ComSIS

Computer Science and Information Systems

Volume 18, Number 3, June 2021

Contents

Editorial

Papers

619 FPGA Implementation of Fuzzy Medical Decision Support System for Disc Hernia Diagnosis
Tijana Šušteršič, Miodrag Peulić, Aleksandar Peulić
641 A New Approximate Method For Mining Frequent Itemsets From Big Data
Timur Valiullin, Joshua Zhexue Huang, Chenghao Wei, Jianfei Yin, Dingming Wu, Iuliia Egorova
657 Reverse Engineering Models of Software Interfaces
Debjyoti Bera, Mathijs Schuts, Jozef Hooman, Ivan Kurtev
687 DrCaptcha: An Interactive Machine Learning Application
Rafael Glikis, Christos Makris, Nikos Tsirakis
703 A Novel Information Diffusion Model Based on Psychosocial Factors with Automatic Parameter Learning
Sabina-Adriana Floria, Florin Leon
729 Arabic Linked Drug Dataset Consolidating and Publishing
Guma Lakshen, Valentina Janev, Sanja Vraneš
749 A Multicriteria Optimization Approach for the Stock Market Feature Selection
Dragana Radojičić, Nina Radojičić, Simeon Kredatus
771 End-to-End Diagnosis of Cloud Systems against Intermittent Faults
Chao Wang, Zhongchuan Fu, Yanyan Huo
791 Distance based Clustering of Class Association Rules to Build a Compact, Accurate and Descriptive Classifier
Jamolbek Mattiev, Branko Kavšek
813 CHEARP: Chord-based Hierarchical Energy-Aware Routing Protocol for Wireless Sensor Networks
Lamia Cheklat, Mourad Amad, Mawloud Omar, Abdellah Boukerram
835 Decision-Making Support for Input Data in Business Processes according to Former Instances
José Miguel Pérez Álvarez , Luisa Parody, Marıa Teresa Gómez-López, Rafael M. Gasca, Paolo Ceravolo
867 Intrusion Prevention with Attack Traceback and Software-defined Control Plane for Campus Networks
Guangfeng Guo, Junxing Zhang, Zhanfei Ma
893 Class Balancing in Customer Segments Classification Using Support Vector Machine Rule Extraction and Ensemble Learning
Sunčica Rogić, Ljiljana Kašćelan
927 Real Time Availability And Consistency Of Health-Related Information Across Multiple Stakeholders: A Blockchain Based Approach
Zlate Dodevski, Sonja Filiposka, Anastas Mishev, Vladimir Trajkovik
957 Predicting Dropout in Online Learning Environments
Sandro Radovanović, Boris Delibašić, Milija Suknović
979 Deep Reinforcement Learning for Resource Allocation with Network Slicing in Cognitive Radio Network
Siyu Yuan, Yong Zhang, Wenbo Qie, Tengteng Ma, Sisi Li
1001 Patient Length of Stay Analysis with Machine Learning Algorithms
Savo Tomović
1023 Multimodal Encoders and Decoders with Gate Attention for Visual Question Answering
Haiyan Li, Dezhi Han
1041 Identifying Key Node in Multi-region Opportunistic Sensor Network based on Improved TOPSIS
Linlan Liu, Wei Wang, Guirong Jiang, Jiang Zhang
1057 Dynamic Fractional Chaotic Biometric Isomorphic Elliptic Curve for Partial Image Encryption
Ahmed Kamal, Esam A. A. Hagras, H. A. El-Kamchochi
1077 Time-aware Collective Spatial Keyword Query
Zijun Chen,Tingting Zhao, Wenyuan Liu
1101 Conflict Resolution Using Relation Classification: High-Level Data Fusion in Data Integration
Zeinab Nakhaei, Ali Ahmadi, Arash Sharifi, Kambiz Badie

ISSN: 1820-0214 (Print) 2406-1018 (Online)

Computer Science and Information Systems

Vol 18, No 3, June 2021

ComSIS

Computer Science and Information Systems

Published by ComSIS Consortium

Volume 18, Number 3
June 2021

ComSIS is an international journal published by the ComSIS Consortium

ComSIS Consortium:

University of Belgrade:
Faculty of Organizational Science, Belgrade, Serbia
Faculty of Mathematics, Belgrade, Serbia
School of Electrical Engineering, Belgrade, Serbia

Serbian Academy of Science and Art:
Mathematical Institute, Belgrade, Serbia

Union University:
School of Computing, Belgrade, Serbia

University of Novi Sad:
Faculty of Sciences, Novi Sad, Serbia
Faculty of Technical Sciences, Novi Sad, Serbia
Technical Faculty "Mihajlo Pupin", Zrenjanin, Serbia

University of Niš:
Faculty of Electronic Engineering, Niš, Serbia

University of Montenegro:
Faculty of Economics, Podgorica, Montenegro

ComSIS Editorial Office:
University of Novi Sad, Faculty of Sciences,
Department of Mathematics and Informatics
Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia
Phone: +381 21 458 888; **Fax:** +381 21 6350 458
www.comsis.org; Email: comsis@uns.ac.rs

Volume 18, Number 3, 2021
Novi Sad

Computer Science and Information Systems

ISSN: 1820-0214 (Print) 2406-1018 (Online)

The ComSIS journal is sponsored by:

Ministry of Education, Science and Technological Development of the Republic of Serbia
http://www.mpn.gov.rs/

Computer Science and Information Systems

AIMS AND SCOPE

Computer Science and Information Systems (ComSIS) is an international refereed journal, published in Serbia. The objective of ComSIS is to communicate important research and development results in the areas of computer science, software engineering, and information systems.

We publish original papers of lasting value covering both theoretical foundations of computer science and commercial, industrial, or educational aspects that provide new insights into design and implementation of software and information systems. In addition to wide-scope regular issues, ComSIS also includes special issues covering specific topics in all areas of computer science and information systems.

ComSIS publishes invited and regular papers in English. Papers that pass a strict reviewing procedure are accepted for publishing. ComSIS is published semiannually.

Indexing Information

ComSIS is covered or selected for coverage in the following:

- Science Citation Index (also known as SciSearch) and Journal Citation Reports / Science Edition by Thomson Reuters, with 2019 two-year impact factor 0.927,
- Computer Science Bibliography, University of Trier (DBLP),
- EMBASE (Elsevier),
- Scopus (Elsevier),
- Summon (Serials Solutions),
- EBSCO bibliographic databases,
- IET bibliographic database Inspec,
- FIZ Karlsruhe bibliographic database io-port,
- Index of Information Systems Journals (Deakin University, Australia),
- Directory of Open Access Journals (DOAJ),
- Google Scholar,
- Journal Bibliometric Report of the Center for Evaluation in Education and Science (CEON/CEES) in cooperation with the National Library of Serbia, for the Serbian Ministry of Education and Science,
- Serbian Citation Index (SCIndeks),
- doiSerbia.

Information for Contributors

The Editors will be pleased to receive contributions from all parts of the world. An electronic version (MS Word or LaTeX), or three hard-copies of the manuscript written in English, intended for publication and prepared as described in "Manuscript Requirements" (which may be downloaded from http://www.comsis.org), along with a cover letter containing the corresponding author's details should be sent to official journal e-mail.

Criteria for Acceptance

Criteria for acceptance will be appropriateness to the field of Journal, as described in the Aims and Scope, taking into account the merit of the content and presentation. The number of pages of submitted articles is limited to 20 (using the appropriate Word or LaTeX template).

Manuscripts will be refereed in the manner customary with scientific journals before being accepted for publication.

Copyright and Use Agreement

All authors are requested to sign the "Transfer of Copyright" agreement before the paper may be published. The copyright transfer covers the exclusive rights to reproduce and distribute the paper, including reprints, photographic reproductions, microform, electronic form, or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder permission to reproduce the paper or any part of it, for which copyright exists.

Computer Science and Information Systems

Volume 18, Number 3, June 2021

CONTENTS

Editorial

Papers

619 **FPGA Implementation of Fuzzy Medical Decision Support System for Disc Hernia Diagnosis**
Tijana Šušteršič, Miodrag Peulić, Aleksandar Peulić

641 **A New Approximate Method For Mining Frequent Itemsets From Big Data**
Timur Valiullin, Joshua Zhexue Huang, Chenghao Wei, Jianfei Yin, Dingming Wu, Iuliia Egorova

657 **Reverse Engineering Models of Software Interfaces**
Debjyoti Bera, Mathijs Schuts, Jozef Hooman, Ivan Kurtev

687 **DrCaptcha: An Interactive Machine Learning Application**
Rafael Glikis, Christos Makris, Nikos Tsirakis

703 **A Novel Information Diffusion Model Based on Psychosocial Factors with Automatic Parameter Learning**
Sabina-Adriana Floria, Florin Leon

729 **Arabic Linked Drug Dataset Consolidating and Publishing**
Guma Lakshen, Valentina Janev, Sanja Vraneš

749 **A Multicriteria Optimization Approach for the Stock Market Feature Selection**
Dragana Radojičić, Nina Radojičić, Simeon Kredatus

771 **End-to-End Diagnosis of Cloud Systems against Intermittent Faults**
Chao Wang, Zhongchuan Fu, Yanyan Huo

791 **Distance based Clustering of Class Association Rules to Build a Compact, Accurate and Descriptive Classifier**
Jamolbek Mattiev, Branko Kavšek

813 **CHEARP: Chord-based Hierarchical Energy-Aware Routing Protocol for Wireless Sensor Networks**
Lamia CHEKLAT, Mourad AMAD, Mawloud OMAR, Abdellah BOUKERRAM

835 **Decision-Making Support for Input Data in Business Processes according to Former Instances**
José Miguel Pérez Álvarez , Luisa Parody, María Teresa Gómez-López, Rafael M. Gasca, Paolo Ceravolo

867 **Intrusion Prevention with Attack Traceback and Software-defined Control Plane for Campus Networks**
Guangfeng Guo, Junxing Zhang, Zhanfei Ma

893 **Class Balancing in Customer Segments Classification Using Support Vector Machine Rule Extraction and Ensemble Learning**
Sunčica Rogić, Ljiljana Kašćelan

927 **Real Time Availability And Consistency Of Health-Related Information Across Multiple Stakeholders: A Blockchain Based Approach**
Zlate Dodevski, Sonja Filiposka, Anastas Mishev, Vladimir Trajkovik

957 **Predicting Dropout in Online Learning Environments**
Sandro Radovanović, Boris Delibašić, Milija Suknović

979 **Deep Reinforcement Learning for Resource Allocation with Network Slicing in Cognitive Radio Network**
Siyu Yuan, Yong Zhang, Wenbo Qie, Tengteng Ma, Sisi Li

1001 **Patient Length of Stay Analysis with Machine Learning Algorithms**
Savo Tomović

1023 **Multimodal Encoders and Decoders with Gate Attention for Visual Question Answering**
Haiyan Li, Dezhi Han

1041 **Identifying Key Node in Multi-region Opportunistic Sensor Network based on Improved TOPSIS**
Linlan Liu, Wei Wang, Guirong Jiang, Jiang Zhang

1057 **Dynamic Fractional Chaotic Biometric Isomorphic Elliptic Curve for Partial Image Encryption**
Ahmed Kamal, Esam A. A. Hagras, H. A. El-Kamchochi

1077 **Time-aware Collective Spatial Keyword Query**
Zijun Chen,Tingting Zhao, Wenyuan Liu

1101 **Conflict Resolution Using Relation Classification: High-Level Data Fusion in Data Integration**
Zeinab Nakhaei, Ali Ahmadi, Arash Sharifi, Kambiz Badie

Computer Science and Information Systems 18(3): i–iv https://doi.org/10.2298/CSIS210300iI

Editorial

Mirjana Ivanović[1], Miloš Radovanović[1], and Vladimir Kurbalija[1]

University of Novi Sad, Faculty of Sciences
Novi Sad, Serbia
{mira,radacha,kurba}@dmi.uns.ac.rs

This third issue of Volume 18 of Computer Science and Information Systems in 2021 contains 22 regular articles. We thank all authors and reviewers for their hard work, without which the current issue, and journal publication in general, would not be possible.

The first regular paper, "FPGA Implementation of Fuzzy Medical Decision Support System for Disc Hernia Diagnosis" by Tijana Šušteršič et al. uses sensor measurements of foot force in combination with fuzzy logic to implement a decision support system for disc hernia diagnostics on a Field Programmable Gate Array (FPGA). Experimental results show that the system performs comparably to its non-FPGA counterpart in both pre-op and post-op scenarios, at the same time representing an inexpensive, portable expert system for real-time data acquisition and processing, as well as disc hernia diagnosis and patient condition tracking.

The second article, "A New Approximate Method for Mining Frequent Itemsets from Big Data" authored by Timur Valiullin et al. proposes a new approach for approximately mining of frequent itemsets in a big transaction database that can miss some true item sets, but on the other hand can be implemented in a distributed environment. The issue of false negatives is tackled by introducing an additional hyperparameter to the algorithm.

"Reverse Engineering Models of Software Interfaces" by Debjyoti Bera et al. presents a passive learning technique for interface models inspired by process mining techniques. The approach is based on representing causal relations between events in an event log and their timing information as a timed-causal graph, which is further processed and transformed into a state machine with a set of timing constraints.

In the article entitled "DrCaptcha: An Interactive Machine Learning Application," Rafael Glikis et al. describe an interactive machine learning system that provides third-party applications with a CAPTCHA service and, at the same time, uses the user's input to train artificial neural networks that can be combined to create a powerful OCR system. This way, two problems with constructing machine learning systems identified in the article are tackled: overfitting the training data and human involvement in the data preparation process.

Sabina-Adriana Floria and Florin Leon, in "A Novel Information Diffusion Model Based on Psychosocial Factors with Automatic Parameter Learning," propose a model for imitating the evolution of information diffusion in a social network. Individuals are modeled as nodes with two factors (psychological and sociological) that control their probabilistic transmission of information, with a genetic algorithm being used to tune the parameters of the model to fit the evolution of information diffusion observed in two real-world datasets.

"Arabic Linked Drug Dataset Consolidating and Publishing," authored by Guma Lakshen et al. examines the process of creating and publishing an Arabic linked drug dataset based on open drug datasets from selected Arabic countries and discusses quality issues considered in the linked data lifecycle when establishing a semantic data lake in the pharmaceutical domain. The article showcases how the pharmaceutical industry can take leverage emerging trends for building competitive advantages, at the same time acknowledging that better understanding of the specifics of the Arabic language is needed in order to extend the usage of linked data technologies in Arabic companies.

Dragana Radojičić et al. in their article "A Multicriteria Optimization Approach for the Stock Market Feature Selection" explore the informativeness of features extracted from limit order book data in order to classify market data vectors into two classes (buy/idle) using a long short-term memory (LSTM) deep neural network. New technical indicators based on support/resistance zones are introduced to enrich the set of features, and multicriteria optimization is employed to perform adequate feature selection among the proposed approaches with respect to precision, recall, and the F-score.

The article "End-to-End Diagnosis of Cloud Systems Against Intermittent Faults," by Chao Wang et al. proposes a fault diagnosis method that can effectively identify and locate intermittent faults originating from processors in the cloud computing environment. The method is end-to-end in that it does not rely on artificial feature extraction for applied scenarios (making it more generalizable than conventional neural network-based methods), it can be implemented with no additional fault detection mechanisms, and is realized by software with almost zero hardware cost.

"Distance based Clustering of Class Association Rules to Build a Compact, Accurate and Descriptive Classifier" by Jamolbek Mattiev and Branko Kavšek introduces new methods that are able to reduce the number of class association rules produced by "classical" class association rule classifiers, while maintaining an accurate classification model that is comparable to the ones generated by state-of-the-art classification algorithms. This is achieved by employing distance-based agglomerative hierarchical clustering as a post-processing step to reduce the number of rules, and different strategies based on database coverage and cluster center in the rule-selection step.

Lamia Cheklat et al., in their article "CHEARP: Chord-based Hierarchical Energy-Aware Routing Protocol for Wireless Sensor Networks" design an energy efficient location-independent routing protocol for data delivery in wireless sensor networks (WSNs). Contrary to existing protocols that connect nodes independently of their physical proximity, this article proposes an approximate logical structure to the physical one, where the aim is to minimize the average paths' length.

In "Decision-Making Support for Input Data in Business Processes according to Former Instances," José Miguel Pérez Álvarez et al. propose learning the evolution patterns of the temporal variation of the data values in a process model extracted from previous process instances by applying constraint programming techniques. The knowledge obtained is applied in a decision support system (DSS) which helps in the maintenance of the alignment of the process execution with the organizational strategic plans, through a framework and a methodology.

The article "Intrusion Prevention with Attack Traceback and Software-defined Control Plane for Campus Networks" by Guangfeng Guo et al. proposes an intrusion prevention system (IPS) based on coordinated control between the detection engine, the attack traceback agent, and the software-defined control plane. The solution includes a novel algorithm to infer the best switch port for defending different attacks of varied scales based on the inverse header space analysis (HSA) and the global view of the software-defined controller.

“Class Balancing in Customer Segments Classification Using Support Vector Machine Rule Extraction and Ensemble Learning” by Suncica Rogic and Ljiljana Kascelan proposes a class balancing approach based on support vector machine-rule extraction (SVM-RE) and ensemble learning in order to improve predictive models of customer segments for effective marketing. The approach also allows for rule extraction, which can help in describing and explaining different customer segments.

In “Real Time Availability and Consistency of Health-Related Information Across Multiple Stakeholders: A Blockchain Based Approach,” Zlate Dodevski et al. examine different approaches and application of blockchain technology and identify which implementations of components are more suitable and beneficial for a specific electronic health record (EHR) eco-system. The article presents alternative way of dealing with information exchange across multiple stakeholders by justifying the use of the decentralized approach, distributed access and solution to comprehensively track and assemble health related data.

“Predicting Dropout in Online Learning Environments” by Sandro Radovanović et al. employs the lasso and ridge logistic regression techniques to create a prediction model for student dropout on the Open University database. Two interesting questions are investigated: how early dropout can be predicted, and why dropouts occur.

The next article, “Deep Reinforcement Learning for Resource Allocation with Network Slicing in Cognitive Radio Network” by Siyu Yuan et al. establishes a cognitive radio network model based on the underlay model and proposes a cognitive network resource allocation algorithm based on the double deep Q network (DDQN) reinforcement learning technique. The algorithm jointly optimizes the spectrum efficiency of the cognitive network and quality of experience of cognitive users through channel selection and power control.

“Patient Length of Stay Analysis with Machine Learning Algorithms,” authored by Savo Tomović, tackles the problem of measuring factor importance on patient length of stay in a medical emergency department. Based on a historical dataset containing average patient length of stay per day, and factors agreed with domain expert, the article solves the task of providing factors’ impact measure on specific days that do not belong to the historical dataset (new observations) for which the average length of stay is higher than the specified threshold.

Haiyan Li and Dezhi Han, in “Multimodal Encoders and Decoders with Gate Attention for Visual Question Answering” present a visual question answering (VQA) model based on multimodal encoders and decoders with gate attention (MEDGA). Each encoder and decoder block in the MEDGA applies not only self-attention and cross-modal attention but also gate attention, so that the new model can better focus on inter-modal and intra-modal interactions simultaneously within visual and language modality.

“Identifying Key Node in Multi-region Opportunistic Sensor Network based on Improved TOPSIS” by Linlan Liu et al. proposes a novel approach based on the improved TOPSIS method to distinguish the key node from the ferry node in a multi-region opportunistic sensor network. Dynamic topology information is represented by a temporal reachable graph, based on which three attributes are constructed to identify the key node. Game theory with a combination weighting method is employed to combine the subjective weight and objective weight, which is then used to improve the TOPSIS method.

The article “Dynamic Fractional Chaotic Biometric Isomorphic Elliptic Curve for Partial Image Encryption” by Ahmed Kamal et al introduced a modular fractional chaotic sine map (MFC-SM) to achieve high Lyapunov exponent values and completely chaotic behavior of the bifurcation diagram for high level security in image encryption. MFC-SM is combined with various other approaches in the image encryption pipeline in order to obtain an algorithm that is robust against common signal processing attacks and provides a high security level and high speed for image encryption applications.

In “Time-aware Collective Spatial Keyword Query,” Zijun Chen et al. define the time-aware collective spatial keyword query (TCoSKQ), which considers the positional relevance, textual relevance, and temporal relevance between objects and query at the same time. Two evaluation functions are defined to meet different needs of users, for each of which an algorithm is proposed, with effective pruning strategies introduced to improve query efficiency based on the two algorithms.

Concluding the issue, “Conflict Resolution Using Relation Classification: High-Level Data Fusion in Data Integration” by Zeinab Nakhaei et al. tackle the problem of conflict resolution in data integration systems by bridging the gap between relation estimation and truth discovery, demonstrating that there is a natural synergistic relationship between machine learning and data fusion. Relational machine learning methods are used to estimate the relations between entities, and then these relations are employed to estimate the true value using some fusion functions.

We hope that this issue brings forth interesting and diverse articles that cover a wide range of contemporary research topics. Besides being an informative read, we believe that the presented research could be attractive and represent a good starting point and/or motivation for other authors to extend the presented scientific achievements and continue with similar research efforts.

Computer Science and Information Systems 18(3):619–640 https://doi.org/ 10.2298/CSIS200224015S

FPGA Implementation of Fuzzy Medical Decision Support System for Disc Hernia Diagnosis

Tijana Šušteršič[1], Miodrag Peulić[2], and Aleksandar Peulić[3,*]

[1] Faculty of Engineering, University of Kragujevac, Sestre Janjić 6,
34000 Kragujevac, Serbia
tijanas@kg.ac.rs
[2] Clinical Centre Kragujevac, Zmaj Jovina 30, 34000 Kragujevac, Serbia
miodrag.peulic@gmail.com
[3] Faculty of Geography, University of Belgrade, Studentski trg 3,
11000 Belgrade, Serbia
aleksandar.peulic@gef.bg.ac.rs

Abstract. The aim of this study was to create a decision support system for disc hernia diagnostics based on real measurements of foot force values from sensors and fuzzy logic, as well as to implement the system on Field Programmable Gate Array (FPGA). The results show that the created fuzzy logic system had the 92.8% accuracy for pre-operational diagnosis and very high match between the Matlab and FPGA output (94.2% match for pre-operational condition, and 100% match for the post-operational and after physical therapy conditions). Interestingly enough, our system is also able to detect improvements in patient condition after the surgery and physical therapy. The main benefit of using FPGAs in this study is to create an inexpensive, portable expert system for real time acquisition, processing and providing the objective recommendation for disc hernia diagnosis and tracking the condition improvement.

Keywords: disc hernia diagnosis, fuzzy inference systems, FPGA, medical decision support system, hardware.

1. Introduction

In diagnosing discus hernia, magnetic resonance imaging (MRI) represents the gold standard. However, MR scans alone are not as accurate as initially believed and should not be used as the only diagnostic tool [1, 2]. While MRI and CT scanning are more or less invasive and not appealing to the examined subject, both of these procedures are time-consuming screening methods, and the queue time is substantial. Therefore, in addition to surgical screening, using one of the appropriate imaging modalities to diagnose lumbar discus hernia, the patient's medical history and physical examination are necessary [3]. Such tests involve a set of measures (neurological evaluation of motor, sensory, deep tendon reflex functions etc.) that will help assess the likely cause of pain and the potential existence of a spinal nerve deficit [4].

[*] Corresponding author

Neurological examination takes into account dermatomes. Dermatomes are described as areas of skin innervated by specific nerves that originate from the spine [5, 6]. The innervation of the muscles and the skin region on toes originate from the nerves between the discs L5 and S1 in the spine, while the innervation of the heels arises from the nerves that arise between the discs L4 and L5 in the spine [7, 8]. This means that when disc herniation is present at either L4-L5 or L5-S1 level, the nerves experience pressure at that level on either left or right side, causing muscle weakness on the corresponding left or right foot. The main medical clausula, used in this paper is that innervation of the muscles and the area of the skin on the forefeet originates from the pressured nerve located between the L5 and S1 vertebrae, while the roots of the nerves innervating the heels are located between the spine disks of L4 and L5 [7, 8]. As a result, if disk herniation is located at the L4-L5 level, the subject would suffer muscle weakness on heels, whilst subjects with disc hernia on level L5-S1, would suffer muscle weakness on forefeet. Therefore, the clinical evaluation of the level of discus hernia includes pushing the patient's forefeet and heels against doctors' hand and attempting to determine whether there is a muscle weakness on the forefoot/heel. Despite the fact that the neurological examination has been identified as a reliable procedure for diagnosing disk herniation [9, 10], the reliability depends on standardization [10] and the tests for muscle weakness are subjective and could be affected by many factors, including disease severity, doctor's experience, etc. It can be concluded that there is a need for an accurate evaluation of the motor function / weakness of the toes, which is non-invasive and can be used as part of the decision support system in the diagnosis of disk hernia. We present the novel methodology for measuring feet forces using the platform with sensors. Such sensors, along with the custom design chip (i.e. programed FPGA chip), would create an independent system for disc hernia diagnostics that can replace the subjective neurological test.

1.1. Related work

Although fuzzy logic is a long-studied topic, and there is a tremendous number of papers that deal with FPGA-based fuzzy logic controllers [11-15], there are not so many studies that deal with implementation of fuzzy systems on FPGAs that are related to medicals diagnostics. Several attempts have been made to resolve fuzzy inference issues using parallel processing and reconfigurable architecture available through Field Programmable Gate Array (FPGA) [16]. Chowdhury et al. have published several papers on the topics of FPGA-based fuzzy architectures for the purposes of medical diagnosis [16-18]. It could be said that their designs focus on defining a processor architecture that would exploit the parallelism fully used in fuzzy inferences and then further use the implemented design in processors like FPGAs, in order to predict possible critical condition of a patient at an early stage [18]. For that purposes, they mainly investigate the coupling between the neuro-fuzzy systems. Their papers theoretically investigate fuzzy neural networks for the purposes of medical diagnosis systems and then apply it on the problem of early detection of the upcoming critical renal condition [17]. In that sense, only rules that have a positive degree of validation are exploited. Developed system is tested with the data from real patients to compare the results from the implemented fuzzy neural network and the "gold standard" which was the actual pathophysiological state of the patient [17, 18]. Accuracy of the predicted critical state was 95.2% using the implemented FPGA based medical diagnostic decision-making system [16]. Similar to their previous work, another research focuses on the development of FPGA based smart processing system that can predict the physiological state of a patient, given the past physiological data of the patient. Developed system can trigger an alarm in advance, before the critical state of a patient and notify the doctor remotely [16]. The main conclusion from their systems is that physiological measures from patients are subjected to noise and uncertainty, and single patient data cannot be enough to derive conclusions due to the sensors' quality and accuracy issues. This is even more true for the cases when decision about the future physiological state of the patient has to be made. Therefore, it is necessary to collect a sequence of patient pathophysiological state data in different time steps [17]. Cintra et al. [19] investigated the method that is capable of detecting cardiopathies in electrocardiograms and implemented that system in FPGA. In order to reduce the amount of data sampling, they adopted the logic of fuzzy clustering. Through a correlation method on the samples, they proved that it is possible to set an initial diagnosis of indication of a cardiopathy and their main contribution is clustering process that improves the hardware implementation without the loss in accuracy (achieved accuracy for correct diagnosis was 91%) [19]. Other authors have also investigated the implementation of VLSI fuzzy classifiers in biomedical applications. For example, Jothi et al. investigated diabetic epilepsy risk level classification and compared it with FPGA output [20]. The results are reported to be a good match between the FPGA and Matlab, however the authors do not specify how the implementation is achieved, nor report any details of implementation. With similar purpose to the previous research, Balamurugan et al. have built a simple and robust SIRM fuzzy processor to classify the epilepsy risk level of diabetic patients and report similar conclusions [21].

Fuzzy logic is often combined with neural networks in order to improve the accuracy. Nilosey et al. employed FPGA based fuzzy interface system cascaded with a feed-forward neural network in order to obtain an optimum decision regarding the future pathology physiological state of a patient. With this methodology, they claim that the chance of predicting the critical diabetic condition of a patient can be achieved accurately, more precisely 30 days ahead of actually attaining the critical condition. The method itself represents an expert system of its kind [22]. Another approach to modelling fuzzy systems is taken by Bariga et al. who adopted a fuzzy logic modeling style using two strategies: behavioral modeling using VHDL and a structural VHDL based on a specific architecture of fuzzy processor. In order to take advantage of the FPGA resources of the devices, they discuss different approaches and employ environment Xfuzzy for the fuzzy system development. This environment is based on the XFL3 specification language, which is a high-level description language, outlining the advantage of focusing on the structure of the system and the behavioral specifications in such way and not on the programming itself, therefore reducing the time for implementing the system [23].

1.2. Advances in FPGA-based processing systems in medical diagnosis

On one side, the main advantage of FPGA-based processing systems is the achieved shorter time span in comparison to the other current processors. High speed simulation of neuron models is becoming the necessity [24]. Intelligent system, which is embedded in one single chip and contains all necessary functionality, gives the FPGAs the advantage over some microcontrollers as it combines high-speed processing and hardware performance with the possibility of software-based changes in the description of the circuits [25]. It has been shown in previous work by Orsila et al. that multiprocessor system-on-chip architecture has the potential to improve the performance of the whole system and reduce the costs [26]. Other researchers like Raychev et al. have outlined the advantages of implementing a processor in hardware [27], such as improved implementation flexibility and design [18]. Chowdhury et al. obtained their results explained in the aforementioned text within an interval of 1.92ms, which guarantees the real-time behavior [18]. Chowdhury et al. also justify the necessity of using FPGAs by explaining that in third world countries like India, doctors are scarcely available in rural areas (only 2% of doctors reside in rural areas). Therefore, these problems necessitate the use of an inexpensive, portable, low power battery operated high-speed equipment that can have the intelligence to predict an imminent health hazard and red alert the patients in rural sectors to contact the doctors for necessary care [17].

On the other side, real time response and controlled accuracy are the biggest outlined issues when it comes to hardware implementation. A delay of a few milliseconds in delivery of results is not something that can be considered a big difference in medical diagnosis. It could be said that software approach could give satisfactory results, especially in more complex systems, as it would be very hard to implement such systems in hardware, without much gain in speed up [16]. The main motivation for a hardware-based implementation in medical diagnostics is the necessity for an inexpensive and non-invasive portable system. Here, it could be said that the main disadvantage one ASIC based hardware are high development costs and low reconfigurability it offers [16]. Contrary, FPGA solution allows new changes in the proposed diagnostic algorithm to be mapped onto the hardware without costly adjustments [16].

It should be also said that medical diagnosis, as a complex and judgmental process, is based not only on the literature data and data obtained from medical tests, but also on the experience of the doctor. However, as already mentioned, subjective decisions made by doctors are affected by many factors including environment and emotional state of a person (being tired or not, influence of private life etc.) [17, 28]. Additionally, there is also inter-physician variability in the decision process [17]. Because of that, the authors of this paper in their previous work [29] have already established the development of a smart system that employs fuzzy logic to predict the level of disc hernia based on objective force values from four sensors placed on two panels designed for feet. This paper aims at implementing that system on FPGA chip to satisfy real time analysis and development of inexpensive portable on-site platform. Main contributions of the proposed system are:

- implementation of the fuzzy logic for the purposes of objective disc hernia diagnostics on FPGA

- fuzzy inference system was realized in the form of look up table (LUT), to meet the demand for effective resource utilization
- the system is able to connect to the portable platform for foot force measurements, achieving the demand for inexpensive portable on-site platform for objective diagnostic, independent of any computer/laptop.
- the fuzzy system was adapted and is using Sugeno method in comparison to the fuzzy system that used Mamdani method described in [29] to be more suitable for hardware implementation
- the portable device is user-friendly with fast analysis in real time, thus reducing the time and queue for patient diagnostics
- the system is able to detect the improvement in muscle strength after the surgery and physical therapy in comparison to the pre-operational condition, which has not been investigated in any other papers before.

2. Materials and Methods

Complex processes that are usually controlled by human experts should be described using uncertainty models [30]. Although an adequate level of precision can be achieved by quantification of uncertainty, for the most of the processes, a better solution is accepting a certain level of imprecision by using fuzzy logic. This logic describes the imprecision and uncertainty using expert knowledge base as the fundament in controlling a complex process control system [30]. Fuzzy inference system (FIS), if organized in the form of addition and subtraction operations, would benefit from implementation on FPGA, in terms of resource utilization and speed-up. To implement division using FPGA is a challenging task and leads to losing some benefits of using hardware, as division occupies a lot of FPGA resources. Therefore, Bhole et al. proposed a novel algorithm that uses fixed to floating point conversion and used it in implementing floating point division in FIS. Their main conclusion was that proposed dignified methodology for fixed to floating point conversion reduces time requirements and improves resource utilization [30].

2.1. FPGA Implementation of Fuzzy Logic for Disc Hernia Diagnostics

As already mentioned, the main challenges in implementation of fuzzy logic are the dimensionality, real time response and accuracy. These objections can be overcome by using reconfigurable architecture, parallel processing and floating-point operations that are available through Field Programmable Gate Array (FPGA). Proposed block diagram for the FPGA implementation of signal processing for Disc Hernia Diagnostics, coupled with the foot force platform measurement system is shown in Fig 1.

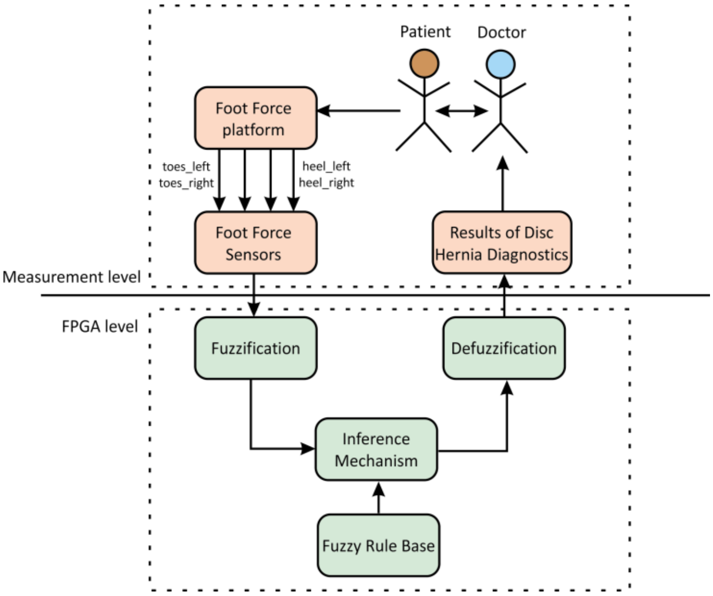


Fig. 1. Block diagram of the proposed coupled FPGA signal processing and foot force platform measurement system

In this figure, on the measurement level, the patient is subjected to the measuring system. The measurement hardware includes two identical platforms with designated surface area for patient's feet. These areas have four sensors per foot placed on specific points of each foot: L1-3 sensors for the left foot placed on the forefeet, L4 sensor placed on the heel of the left foot, R1-3 sensors for the right foot placed on the forefeet, R4 sensor placed on the heel of the right foot. Position of sensors could be adjusted to the patient foot size; therefore, the position is patient specific. Standard Flexi Force A201 [31] sensors are used, with force range from 0 to 440N. The placement choice of four characteristic points of the foot are based on the neurological exam of the doctor. The exam is the standard procedure done by a neurologist and includes pressing each foot of admitted patient against the doctor's hand and examining the pressure that heel and toes are making on the hand. In such a way, the doctor is looking into any differences in pressures between heels and toes of the left and right foot (called muscle weakness in toes/heels). The system is tested with a larger number of sensors (up to 8 sensors per foot - 16 sensors overall), in order to cover greater surface on the platform with sensors. Obtained results were the same, which indicates that at least 4 sensors per foot were enough to capture the phenomena. As a result, the smallest number of sensors that would achieve the highest accuracy was adopted. If another set of sensors (from another producer) would be used, as long as the typical performance of the sensors is not changed (primarily sensor sensitivity and repeatability), the system is able to correctly classify the output diagnosis. Depending on the producer of the sensor, if the operating range is different than the used FlexiForce A201 sensors, the fuzzy system has to be recalibrated to cover the operating range of the sensors. A more detailed explanation of the constructed platform, as well as the place of the sensors L1-L4 and R1-R4 with respect to the platform could be found in [29, 32].

The measurement procedure included three segments which are performed one after another in one continuous recording:

1. patient is standing on both feet normally
2. patient is standing on both feet, but only on the forefeet/toes (term forefoot/toes is used here for standing on the metatarsal heads)
3. patient is standing only on heels

Based on these measurements, average values are calculated directly on the microprocessor:

1. average measurement value from the L1-L3 sensors during the left forefoot standing (further denoted as *toes_left*, also in Fig 1)
2. average measurement value from the R1-R3 sensors during the right forefoot standing (further denoted as *toes_right*, also in Fig 1)
3. average measurement value from the L4 sensor during the left heel standing (further denoted as *heel_left*, also in Fig 1)
4. average measurement value from the R4 sensor during the right heel standing (further denoted as *heel_right*, also in Fig 1)

These values are further sent to FPGA for the fuzzification. The process consists of the usual steps in Fuzzy Logic System (FLS) – fuzzification, inference mechanism that includes knowledge base and defuzzification. Defuzified value is returned to the doctor in the form of indicator whether disc herniation is diagnosed at all, and if it is, at which level it is detected:

1. disc hernia on the left side at the L4/L5 disc level
2. disc hernia on the right side at the L4/L5 disc level
3. disc hernia on the left side at the L5/S1 disc level
4. disc hernia on the right side at the L5/S1 disc level.

2.2. Fuzzy Theory - Fuzzy Inference System Design Steps

Generally, FLS represents a nonlinear mapping of input features into the scalar output based on several steps. The attractiveness of fuzzy logic is the number of possibilities that can be achieved with different mappings [30]. We will further only briefly present the FIS system and describe how it is used to suit the purposes of automatic disc herniation diagnostics. Every FIS should have the following steps:

1. Definition of objectives: Our FIS system should be able to determine the level of disc hernia, based on foot force measurements obtained using the hardware described previously by the same authors [29].
2. Determination of the antecedents, consequents, and fuzzy rules: Fuzzy logic means reasoning using fuzzy sets, and description of linguistic variables, whereas triangular or trapezoidal membership functions are most commonly used. Disc Hernia is a nonlinear process, which can be mapped using trapezoidal membership functions as it was shown in [29]. Normally, since the accuracy of definition is directly dependent on fuzzy patches, it is logical that if the number of fuzzy patches increases, resolution and accuracy increase. However, the complexity of FIS also increases. In order to obtain the balance

between complexity and accuracy, three linguistic variables are selected for each input – very low, low and normal (Fig 2).

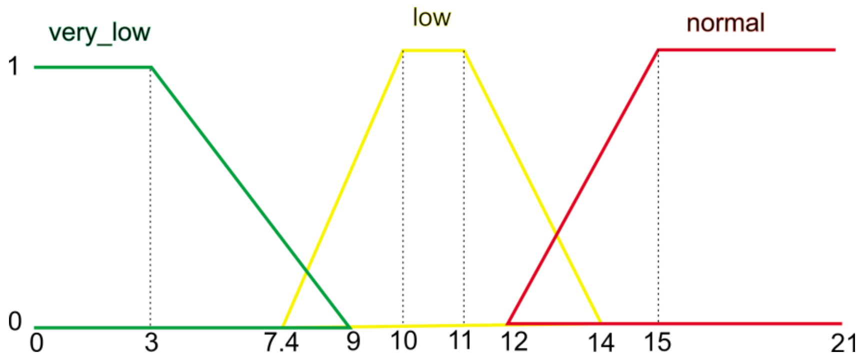


Fig. 2. Linguistic variables used in fuzzification process for disc hernia diagnostics

3. Formulation of the knowledge base: Decision is made based on the expert knowledge, which in our case is defined by the medical doctor. We have defined 42 fuzzy rules implemented as LUT that describe the process of diagnosing the level of disc hernia. Examples of some rules are:
    a. If *toes_left* is VERY_LOW and *toes_righ*t is NORMAL and *heel_left* is NORMAL and *heel_right* is NORMAL, then diagnosis will be L5/S1 ON THE LEFT SIDE
    b. If *toes_left* is NORMAL and *toes_right* is VERY_LOW and *heel_left* is NORMAL and *heel_right* is NORMAL, then diagnosis will be L5/S1 ON THE RIGHT SIDE
    c. If *toes_left* is NORMAL and *toes_right* is NORMAL and *heel_left* is LOW and *heel_right* is NORMAL, then diagnosis will be L4/L5 ON THE LEFT SIDE
    d. If *toes_left* is NORMAL and *toes_right* is NORMAL and *heel_left* is NORMAL and *heel_right* is LOW, then diagnosis will be L4/L5 ON THE RIGHT SIDE
4. Determination of conjunction and disjunction operators: conjunction and disjunction operators are defined in consultation with the medical doctor-expert. For the operators, MAX(x1,x2,x3,x4) was used.
5. Defuzzification: In order to determine the crisp output and final decision from the FIS, several methods can be applied. In Mamdany FIS, crisp conversion is most commonly centre of gravity method where the centroid represents the crisp consequence. However, Mamdani style is not suitable for this project. The reason for this is that Mamdani method requires finding the centroid of a two-dimensional shape by integrating across a continuously varying function [33]. This method is not computationally efficient nor the output in the form of triangle or trapezoidal membership functions is easy to implement. Moreover, defuzzification calculation is very complicated to be achieved in VHDL. Sagaria (2008) proved that with the Mamdani method results are not

necessarily effective or better, and that Sugeno style is much more suitable for hardware implementation [13]. This is due to the fact that the output of this method uses spikes called singletons, meaning there is a unity at single particular point and it is zero elsewhere. This leads to the output of each fuzzy rule being constant [13]. Because of this, Sugeno method has the advantage of being simple as a method, which leads to fast calculations and is relatively easy to implement on hardware. According to [34], fuzzy processor based on the Sugeno method is a good trade-off between the hardware simplicity and efficiency, without the loss in accuracy. Therefore, we have also used Sugeno method to suit the hardware purposes, which is different from the Maamdani method used in [29].

6. Testing and validation: Testing the system and tuning the rules has been achieved through several iterations, until satisfactory results are obtained. The proposed fuzzy inference system has already been proved to be adequate for disc hernia diagnostics in [29].

2.3. FPGA signal analysis

The real time logic (RTL) design of the overall created system is presented in Fig 3. The input to the fuzzy inference system are four values of the foot force measurements as previously mentioned: average measurement value from the L1-L3 sensors during the left forefoot standing (further denoted as *toes_left*), average measurement value from the R1-R3 sensors during the right forefoot standing (further denoted as *toes_right*), average measurement value from the L4 sensor during the left heel standing (further denoted as *heel_left*), and average measurement value from the R4 sensor during the right heel standing (further denoted as *heel_right*). They are all represented as 8 bit *std_logic_vector(7:0).* Of course, inputs to the *top_module* are also clear *(clr*) and master clock (*mclk)* of 50MHz. The reason for using 50MHz is that the master clock on the Nexys 2 board (only available board in our case used for real time implementation) is of that frequency, but the frequency of the *mclk* can be easily changed depending on the available board. Output of the system is the diagnosis in the form of the 4 bit vector (*led(3:0)*), where four 0000 represent the healthy person and each 1 in the four bit vector represents one diagnosis (1000 – L4/L5 on the left side, 0100 – L4/L5 on the right side, 0010 – L5/S1 on the left side; 0001 – L5/S1 on the right side). Another output of the system is the *maximum(7:0),* which represents the certainty value of the calculated diagnosis.

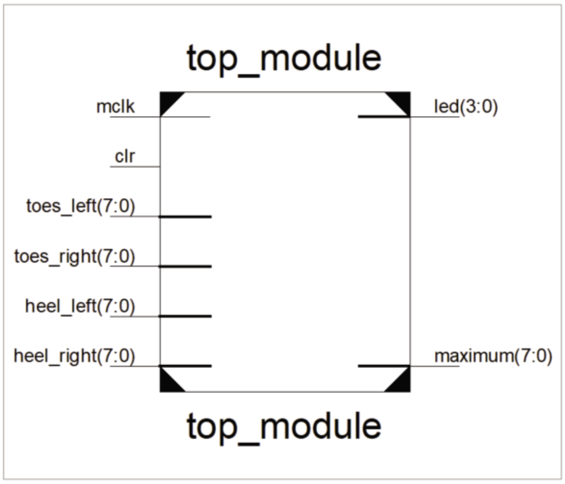


Fig. 3. RTL schematic of the top_module for the fuzzy disc hernia diagnosis system

Inside the *top_module*, the first subsystem is U1 that represents the fuzzification subsystem (Fig 4). The system takes four input values and fuzzifies them according to the defined three linguistic variables - *very low, low and normal*, indicating muscle weakness. After this, based on the LUT table, for each input, an array of three values is created indicating the membership degree to each linguistic variable. This is done in such a way to avoid floating point format, so membership degree is scaled to be in range of 0-100 (instead of 0 - 1). Membership degree is calculated in advance based on the equation for linear function through two points and implemented as LUT table. In the LUT table, the y value with membership degree was calculated for the points on the x axis with the discretization of 0.2. The values are multiplied then with 10, same as for the y axis value, to avoid floating point format and work with integer values. The same logic was applied in [16] and [25]. This means that y value was calculated in advance for 0 (0), 0.2 (20), 0.4 (40) etc. and forwarded to LUT. The input values have to be rounded to the even number, and they are all in the range of 0 – 21 (210), which is covered by 8-bit vector. The system has been tested with both finer and coarser discretization and the adopted described discretization has been shown to have the best tradeoff between the complexity and accuracy. For example, for input *toes_left(7:0)*, an array *fuzzy_t_l* is created, where *fuzzy_t_l(0)(7:0)* indicates the degree of membership to the linguistic variable *very low*, *fuzzy_t_l(1)(7:0)* indicates the degree of membership to the linguistic variable *low* and *fuzzy_t_l(2)(7:0)* indicates the degree of membership to the linguistic variable *normal*. The same output values are further transferred both to U2 and U3.

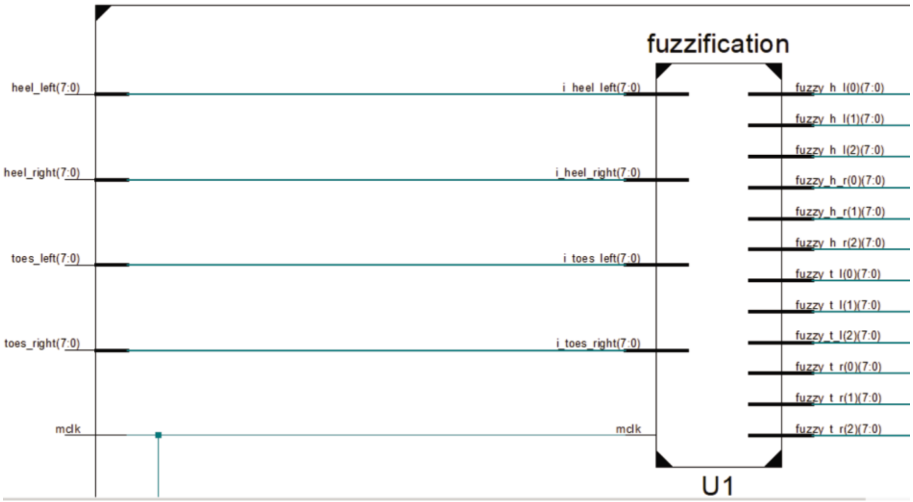


Fig. 4. RTL schematic of U1 subsystem for fuzzification of input data

These variables *fuzzy_t_l*, *fuzzy_t_r*, *fuzzy_h_l*, *fuzzy_h_r* are forwarded to the U2 system (Fig 5) to combine the maximal values for the mentioned variables and determine one overall maximum membership degree and one maximum across three linguistic variables per input. The output of the system is the red colored maximum value (maximal membership value).

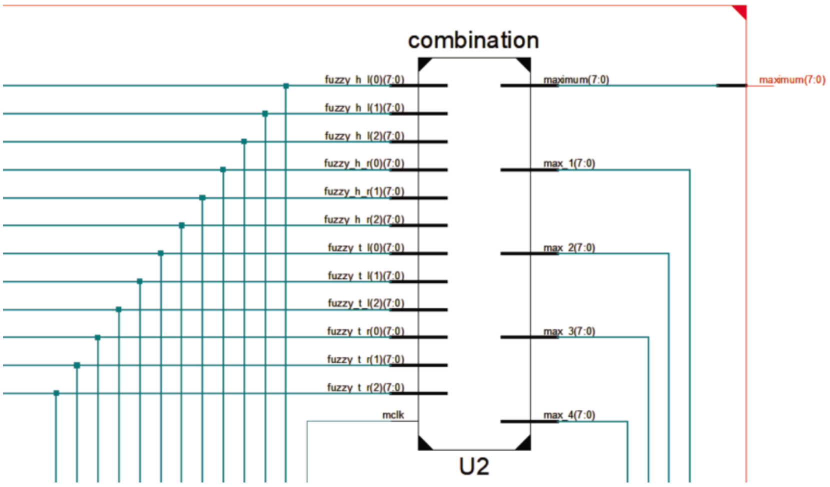


Fig. 5. RTL schematic of U2 subsystem for combination of input fuzzified data

These same variables *fuzzy_t_l*, *fuzzy_t_r*, *fuzzy_h_l*, *fuzzy_h_r* are forwarded to the U3 system (Fig 6), along with the calculated maximum *max_1(7:0)* to *max_4(7:0)* to determine which linguistic variables are active. This is organized in such a way that *max_1(7:0)* to *max_4(7:0)* is compared with the *fuzzy_t_l(7:0)*, *fuzzy_t_r(7:0)*, *fuzzy_h_l(7:0)*, *fuzzy_h_r(7:0)*, and 001 is shifted to the left a number of times that corresponds to the index of the *fuzzy_x_x* that matches the value of *max_x*. This means that only one linguistic variable will be active per fuzzified input bit equals 1 in the place corresponding to the activated linguistic variable. As follows, four outputs r*_toes_left(2:0)*, *r_toes_right(2:0)*, *r_heel_left(2:0)*, *r_heel_right(2:0)* are created, where only one bit will be 1, and the remaining two bits will be 0.

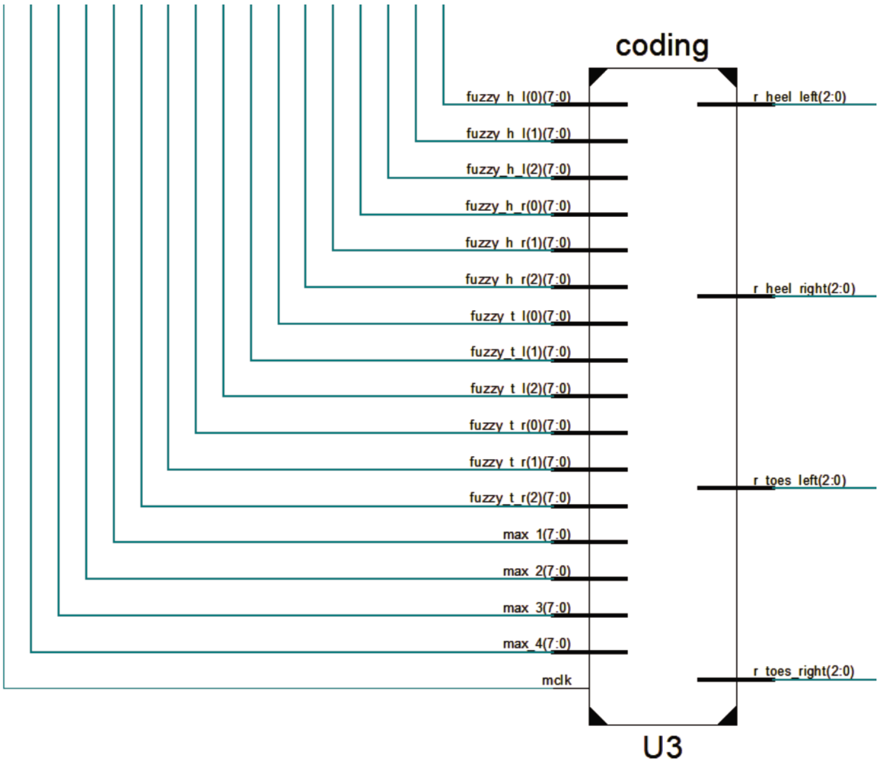


Fig. 6. RTL schematic of U3 subsystem for coding of input data

Calculated variables *r_toes_left(2:0)*, *r_toes_right(2:0)*, *r_heel_left(2:0)*, *r_heel_right(2:0)* serve as the input to the final block U4 shown in Fig 7. This block is responsible for rule activation. Overall, 42 rules were written based on expert knowledge. Final output is a four bit vector output(3:0), where position of the bit with the value 1 indicates the diagnosis (1000 – L4/L5 on the left side, 0100 – L4/L5 on the right side, 0010 – L5/S1 on the left side; 0001 – L5/S1 on the right side). The output diagnosis can be shown as led light on FPGA hardware board, where the position of the led indicates the diagnosis, so no knowledge of the hardware is necessary to understand the final output on the board.

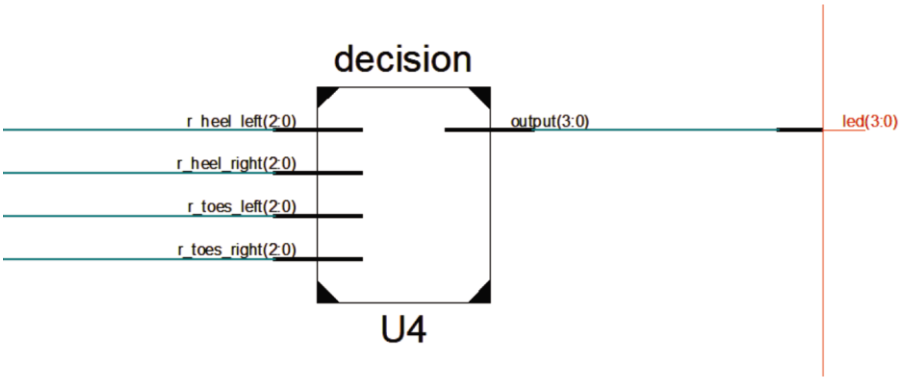


Fig. 7. RTL schematic of U4 subsystem for final decision (defuzzification)

2.4. Dataset

The dataset used in this study included force measurements of 56 adult subjects pre-operationally diagnosed with L4/L5 or L5/S1 discus hernia and 33 adult subjects after surgery and physical therapy. Their medical condition was assessed before operation, after the surgery and after physical therapy using both the designed system and doctor expert. Dataset was collected during the period from 2015 to 2020, in Clinical Centre Kragujevac, Serbia. Demographic details of subjects - gender, age, height and weight were assessed and noted. Information about demographic data is given in Table I in the form of mean ± standard deviation. Inclusion criteria for the study was that patients had only disc herniation at the level of L4/L5 and L5/S1 and were without any other spinal problems. This means that patients with spinal stenosis, spondylolisthesis, cauda equina syndrome, neurogenic claudication or previous spinal surgery, or diseases affecting multiple discs were excluded from the dataset. Patients with pathologies of the lumbar spine, including tumors, infections, inflammatory spondyloarthropathies, fractures, Paget disease, severe osteoporosis, diabetes and pregnancy were excluded from the dataset as well.

Table 1. Demographic details of the tested patients – preoperational, post operational and after physical therapy

Pre-operational	Number	Age (years)	Weight (kg)	Height (cm)
Male with disc hernia	28	43.78±13.41	96.33±14.9	181.33±5.55
Female with disc hernia	28	41.80±9.96	75.5±9.42	172.83±4.96
Post operational and after physical therapy	Number	Age (years)	Weight (kg)	Height (cm)
Male with disc hernia	17	42±13.37	90±13.30	182±5.66
Female with disc hernia	16	42±13	66±9	169±7

Distribution of the patients with disc herniation pre-operationally, belonging to four different categories was as follows:

1. disc hernia on left side at the L4/L5 disc level (12 patients)
2. disc hernia on right side at the L4/L5 disc level (12 patients)
3. disc hernia on left side at the L5/S1 disc level (13 patients)
4. disc hernia on right side at the L5/S1 disc level (19 patients)

Distribution of the patients with disc herniation post-operationally and after physical therapy, belonging to four different categories was as follows:

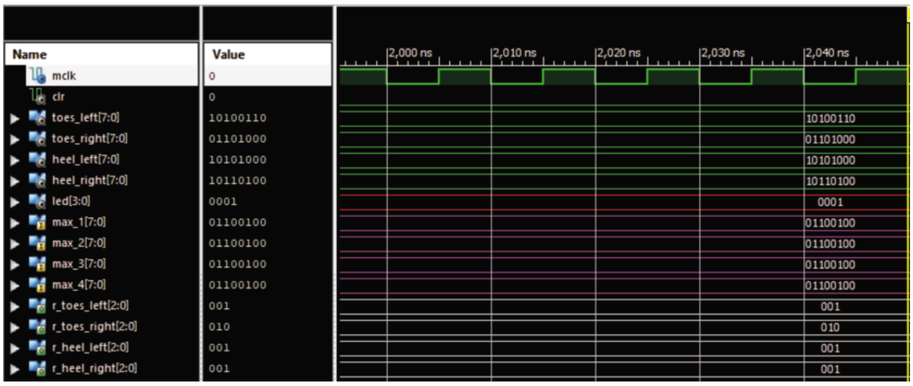
1. disc hernia on left side at the L4/L5 disc level (9 patients)
2. disc hernia on right side at the L4/L5 disc level (5 patients)
3. disc hernia on left side at the L5/S1 disc level (7 patients)
4. disc hernia on right side at the L5/S1 disc level (12 patients)

Patients with disc hernia L3/L4 were not included in this study as there is no proven relationship between the muscle weakness and this diagnosis. Additionally, it is not so common diagnosis, and only one patient was admitted during dataset collection with this diagnosis. This distribution is logical as previous investigation on distribution of disc hernia diagnosis showed that 75% of herniated discs occur at the lumbosacral junction (L5/S1), 20 % at L4/L5 level and the remaining 5% of the upper lumbar levels (L3/L4 etc.) [35].

3. Results and Discussion

The results showed that the designed system is accurate enough to be used as a transportable system with the measurement platforms. Also, the system is able to detect the improvement in muscle strength after the surgery and physical therapy. As it was previously reported, the results show that smaller forefoot force is recorded on the corresponding foot in the case of the patient diagnosed with L5/S1 disc hernia, whilst weaker heel force was recorded on the corresponding foot in the case of the patient diagnosed with L4/L5 disc hernia. For example, in the case of the patient diagnosed with L4/L5 disc hernia on the left side, weaker heel force was detected on the left foot; if the patient is diagnosed with L5/S1 disc hernia on the right side, weaker forefoot force was detected on the right foot etc. The results are in accordance with previously published results by authors and with diagnostic logic used by the doctor. The medical background that supports this logic lies in the fact that innervation of the muscles and skin area that are present on the toes originate from the nerves between L5 and S1 discs in the spine, while innervation of the heels is done via nerves that originate between L4 and L5 discs in the spine [36, 37].

An example of the diagnosed L5/S1 disc hernia on the right side is given in Fig 8. The most important signals were already explained in the section Materials and Methods. Inputs to the system are four recorded values *toes_left(7:0), toes_right(7:0), heel_left(7:0), heel_right(7:0).* Additional variables *r_toes_left(2:0), r_toes_right(2:0), r_heel_left(2:0), r_heel_right(2:0)* show membership functions activated by each input variable after fuzzification. Variables *max_1(7:0), max_2(7:0), max_3(7:0), max_4(7:0)* show the membership degrees of the previously explained activated membership functions. The output is *led(3:0)* where the position of 1 indicates the diagnosis, in this case, number one in the last position indicates the detected diagnosis L5/S1 on the right side.



Name	Value
mclk	0
clr	0
toes_left[7:0]	10100110
toes_right[7:0]	01101000
heel_left[7:0]	10101000
heel_right[7:0]	10110100
led[3:0]	0001
max_1[7:0]	01100100
max_2[7:0]	01100100
max_3[7:0]	01100100
max_4[7:0]	01100100
r_toes_left[2:0]	001
r_toes_right[2:0]	010
r_heel_left[2:0]	001
r_heel_right[2:0]	001

Fig. 8. Simulation results in the case of L5/S1 disc hernia on the right side

The results for the pre-operational state show that out of 56 patients preoperationally, 52 patients were diagnosed with the correct diagnosis by Matlab that matches the gold standard (Table 2). Out of the three wrongly classified diagnosis, two were L4/L5 on the left side and two were L5/S1 on the right side. This means that the overall accuracy was 92.85%.

Table 2. Comparison of the gold standard, Matlab and FPGA results for pre-operational diagnosis

Diagnosis	Gold Standard (medical doctor)	Number of patients correctly diagnosed	
		Matlab	FPGA
L4/L5 left	12	10	9
L4/L5 right	12	12	12
L5/S1 left	13	13	13
L5/S1 right	19	17	15

For the mentioned four patients with wrong diagnosis in comparison to the gold standard, three patients were classified as healthy, meaning the system was not able to detect disc herniation on any level, due to small differences in recorded forces, that were not big enough to diagnose herniated disc. One patient was diagnosed with L4/L5 on the right side (instead of L5/S1 on the right side), because the recorded value of the force on the right heel was close to zero. It is understandable that the system gave such output, both using the Matlab and FPGA, however, the recorded value may not represent the real case, as it could have happened that the sensors were not adjusted well to match the patient’s foot size adequately, and the force was not well recorded.

Additionally, we compared the results of the fuzzy system implemented in Matlab, which uses floating point format in order to compare it with the results obtained with the implemented logic in FPGA (Table 2 second and third column). Three patients showed mismatch between the Matlab and FPGA output. The reason for this was the discretization logic that was not detailed enough for these three cases to capture the phenomena. However, if a more detailed discretization is adopted, these cases could have been included. As a result, the patients were misclassified by FPGA, in comparison to the Matlab output (one diagnosis was L4/L5 on the left side and two were diagnoses L5/S1 on the right side). This means that sensitivity, specificity and accuracy respectively, using FPGA in comparison to the Matlab, were:

- L4/L5 on the left side – 0.818, 0.977, 0.945
- L4/L5 on the right side – 1, 0.977, 0.982
- L5/S1 on the left side – 1, 0.976, 0.982
- L5/S1 on the right side – 0.882, 0.975, 0.947

The promising results when comparing the pre-operational diagnosis between the FPGA and Matlab were confirmed also for the post-operational and after physical therapy results. For the case of post-operational state, Matlab and FPGA results matched 100%, where the improvement was detected in 18 cases (54.5% of patients) (Table 3).

Table 3. Comparison of the gold standard, Matlab and FPGA results for improved post-operational condition

Pre-operational diagnosis	Number of patients with improved condition		
	Gold Standard (medical doctor)	Matlab	FPGA
L4/L5 left	4	4	4
L4/L5 right	3	3	3
L5/S1 left	4	4	4
L5/S1 right	7	7	7

Out of these, 4 of the 9 patients with L4/L5 on the left side showed improvement, 3 of the 5 patients with L4/L5 on the right side showed improvement, 4 of the 7 patients with L5/S1 on the left side showed improvement and 7 of 12 patients with L5/S1 on the right side showed improvement. The doctor's diagnosis confirmed these improvements, meaning that gold standard matched the results obtained by either Matlab/FPGA. One patient was wrongly diagnosed with L5/S1 disc hernia on the left side after operation, while before operation the diagnosis was L5/S1 disc hernia on the right side. There was a problem with decision regarding two patients, which had very low values on some sensors – one patient was the same patient as described above, with the force on the right heel that was close to zero. Since the same situation was noticed again after physical therapy, we can conclude that this patient (female) could have had very small feet size that were not placed well on the platform and the sensors were not adjusted well to fit the feet of this patient. Because we had only 33 patients, we did not want to exclude this patient from the dataset, but instead we give this very plausible explanation for the result, which will serve as a reminder for a doctor/physician that will use this decision support system to pay attention when adjusting the placement of sensors. A recommendation for the future upgrades of the system would be to create several fixed positions according to the size of the feet of the tested subjects, which could be changed similarly to the moving rack, so only some positions are available, and not the infinite number of positions.

For the after physical therapy condition, even more patients showed muscle strength improvements (23 patients) - 7 of the 9 patients with L4/L5 on the left side, 5 of 5 patients with L4/L5 on the right side, 5 of the 7 patients with L5/S1 on the left side and 6 of the 12 patients with L5/S1 on the right side (Table 4).

Table 4. Comparison of the gold standard, Matlab and FPGA results for improved after physical therapy condition

Pre-operational diagnosis	Number of patients with improved condition		
	Gold Standard (medical doctor)	Matlab	FPGA
L4/L5 left	7	7	7
L4/L5 right	5	5	5
L5/S1 left	5	5	5
L5/S1 right	6	6	6

As already mentioned, one female patient had a recording of values close to zero for more than one sensor for pre-operational, post-operational and after physical therapy, leading to false alarms as a result of a systematic error in measurement. Other 5 wrongly diagnosed L5/S1 on the right side, 2 wrongly diagnosed L5/S1 on the left side and 2 wrongly diagnosed L4/L5 on the left side matched the logic described by the doctor when diagnosing the level of herniated discs - it was the measurements that mislead to such conclusions. When it comes to the comparison of the Matlab and FPGA results, the match was 100%, meaning that the same outputs were given by Matlab and FPGA in all cases, as well as the same match was achieved when compared with the gold standard.

The simulation time to obtain results was of ns order of magnitudes, while the device utilization summary is given in Table 5. The available used hardware was Nexys 2 circuit board, which is a complete development platform based on a Xilinx Spartan 3E FPGA [38]. Beside the 500K-gate Spartan 3E-500 FG320 chip, the platform has 50MHz oscillator plus socket for second oscillator, 16MB of Micron PSDRAM &16MB of Intel Strata Flash ROM external memory, and several I/O devices and ports that allow user to perform complex implementation of different algorithms.

Table 5. Device utilization summary

	Used	Available	Utilization
Number of Slices	522	4656	11%
Number of Flip Flops	12	9312	0%
Number of 4 input LUTs	933	9312	10%
Number of IOBs	44	232	18%

Some authors used the idea of processing only the active rules (meaning the rules that give a non-null contribution to the final result), instead of all of them [34]. This is done in order to reduce the utilization of the resources on the board. We have achieved this by using the LUT, and as it can be seen from the Table, even with a simple development board, the results are satisfying.

We wanted to prove that discretization logic proposed in this paper, as well as implementation of the fuzzy logic as LUT table does not lead to the loss in the accuracy and in return gives the benefits of using FPGAs in signal processing like parallel processing, speed up etc. Additionally, the main benefit here is that the board platform could be used to be connected to the measurement platform, achieving the real time processing, without the use of different applications, laptops/computers etc. User-friendly interface is achieved in this study via output result that will be in the form of led light, where the position of the led light indicates the diagnosis. However, it could be also easily achieved that the diagnosis is written on the led display or similar, whatever option the doctor (user) finds more appealing. Standard microcontroller provides flexibility in the definition of the knowledge base and choosing the inference algorithms. Nonetheless, the same microcontrollers become inadequate when solving the problems that demand high inference speeds, small size, and low power consumption. For this reason, more specific hardware solutions must be chosen, such as FPGAs, which are more than adequate when the needs for applications related to portable embedded systems or strong real-time requirements have to be met [22,23].

There are couple of papers such as [6], [37] and [38] that examine the neurological relationship between the nerve roots and the dermatomes. However, these studies are conducted mainly from a medical point of view in the form of case studies and describe the justifications for the use of muscle weakness in the treatment of disk herniation and related spine problems. Except some papers from our research group [29, 32, 39], there are no studies that develop the platform for the purposes of disc hernia diagnostics, nor there are papers concerned with the application of any artificial intelligence algorithms on foot force signals to diagnose disk hernia at the levels of L4/L5 and/or L5/S1 levels. Our research group has developed a novel platform with sensor presented in [29], that would be used to record and detect muscle weakness on toes. Another paper by the same research group investigated the statistical significance of the sensor values in comparison to the clinical manual muscle test [39], while an early disk herniation identification system as a supportive tool for physicians is presented in [32], which will serve as a supportive tool to send the tested patients for further examination. No other work has developed such a framework for accurately calculating muscle weakness and implemented any machine learning algorithms to objectively determine disc hernia level (L4/L5 or L5/S1) and side (left or right).

Fuzzy logic may not be the best approach in solving this kind of a problem in comparison to the more advanced artificial intelligence algorithms. This represents the main limitation of this study. However, fuzzy logic is easily implemented in FPGAs in comparison to the neural networks, SVM etc. and the presented results show the accuracy is high enough to be used even without the application of complex algorithms. The logic behind the disc hernia detection is based on IF-THEN-ELSE rules, and therefore the fuzzy logic stands as a logical choice in solving such problems. The main advantage of the system implemented on FPGA is that real time signal acquisition, processing and decision support system in disc hernia diagnosis and post-surgical recovery can be implemented. In that sense, FPGA have the precedence over GPU when it comes to the real time signal acquisition and analysis.

4. Conclusion

Fuzzy Logic provides a different approach to solving a classification problem, which in this study is the level of disc hernia diagnosis. This method is based on the expert knowledge for the formulation of the rule base, which is a powerful tool in solving the problem. It is very convenient to use when there is no mathematical model to describe the phenomena, which was adequate in this study, as the process of disc hernia diagnosis is described with if-then-else rules. We have already used the explained fuzzy logic on the problem of disc hernia diagnostics as presented in [29]. However, system presented in [29] had to be connected to the laptop or desktop computer, with adequate application that the doctor should get familiar with etc. These main drawbacks were addressed in this paper, which lead to the application of the FPGA in processing the obtained signals. The results show that the adapted fuzzy logic system achieves satisfying results both for pre-operational diagnosis, but also detects the improvements after the surgery and physical therapy. Generally, the system showed 92.8% accuracy and very high match between the Matlab and FPGA output (94.2% match for pre-operational condition, and 100% match for the post-operational and after physical therapy conditions). Some mis-classification results were the problem of measurement, possibly due to the bad adjustments of the sensors to the feet of the patient.

Future research would be focused on employing the described system in real conditions as a portable expert system for acquisition, processing and giving the objective recommendation for a final decision of disc hernia diagnosis.

References

1. Jensen, M., Brant-Zawadzki, M., Obuchowski, N., Modic, M., Malkasian D., Ross, J.: Magnetic resonance imaging of the lumbar spine in people without back pain. New England Journal of Medicine, vol. 331, 69–73 (1994)
2. Trafimow, D. and Trafimow, J.: The shocking implications of Bayes' theorem for diagnosing herniated nucleus pulposus based on MRI scans. Cogent Medicine, vol. 3, no. 1, 1133270-1-7 (2016)
3. Al Nezari, N. H., Schneiders, A. G., Hendrick, P. A.: Neurological examination of the peripheral nervous system to diagnose lumbar spinal disc herniation with suspected radiculopathy: a systematic review and meta-analysis. The Spine Journal, vol. 13, no. 6, 657-674 (2013)
4. Shahbandar, L., Press, J.: Diagnosis and nonoperative management of lumbar disk herniation: Operative Techniques in Sports Medicine, vol. 13, no. 2, 114-121, (2005)
5. McGee, S.: Examination of the Sensory System, in Evidence-based physical diagnosis, Elsevier Health Sciences (2012)
6. Hancock, M. J., Koes, B., Ostelo R., Peul, W.: Diagnostic accuracy of the clinical examination in identifying the level of herniation in patients with sciatica. Spine, vol. 36, no. 11, E712-E719 (2011)
7. Winn, H. R.: Youmans Neurological Surgery, Elsevier Health Sciences (2011)
8. Greenberg, M. S., Arredondo, N.: Handbook of neurosurgery (Vol. 1013). New York: Thieme. (2001)
9. McCarthy, C. J., Gittins, M., Roberts C., Oldham, J. A.: The reliability of the clinical tests and questions recommended in international guidelines for low back pain. Spine, vol. 32, no. 8, 921-926 (2007)
10. Vroomen, P. C., de Krom M. C., Knottnerus, J. A.: Consistency of history taking and physical examination in patients with suspected lumbar nerve root involvement. Spine, vol. 25, no. 1, 91 (2000)

11. Sulaiman, N., Obaid, Z. A., Marhaban, M. H., Hamidon, M. N.: FPGA-based fuzzy logic: design and applications-a review. International Journal of Engineering and Technology, vol. 1, no. 5, 491-503 (2009)
12. Sakthivel, G., Anandhi T., Natarajan, S.: Real time implementation of a fuzzy logic controller on FPGA using VHDL for DC motor speed control. International Journal of Engineering Science and Technology, vol. 2, no. 9, 4511-4519 (2010)
13. Sagaria, S. A. M.: Fuzzy logic design using VHDL on FPGA. School of Computer and Communication Engineering, University Maleysia Perlis (2008)
14. Kumbhar, T. R., Nirmale S. S., Mudholkar, R. R.: FPGA implementation of fuzzy logic controller for temperature control. International Journal of Computer Applications, vol. 62, no. 20, 4511-4519 (2013)
15. Akkaya, Ş., Üzgün H. D., Akbati, O.: Fuzzy Logic Controller Implementation with FPGA in the Loop Simulation. In Proceedings of the 2017 International Conference on Mechatronics Systems and Control Engineering, Kayseri, Turkey (February 2017)
16. Chowdhury, S. R., Chakrabarti D., Saha, H.: FPGA realization of a smart processing system for clinical diagnostic applications using pipelined datapath architectures. Microprocessors and Microsystems, vol. 32, no. 2, 107-120 (2008)
17. Chowdhury, S., Saha, H.: Development of a FPGA based fuzzy neural network system for early diagnosis of critical health condition of a patient. Computers in biology and medicine, vol. 40, no. 2, 190-200 (2010)
18. Chowdhury, S. R., Saha, H.: A high-performance FPGA-based fuzzy processor architecture for medical diagnosis. IEEE Micro, vol. 28, no. 5, 38-52 (2008)
19. Cintra, E. R. F., Pimenta T. C., Moreno, R. L.: The use of fuzzy clustering and correlation to implement a heart disease diagnosing system in FPGA. Journal of Software Engineering and Applications, vol. 4, no. 8, 491-496 (2011)
20. Jothi, M., Balamurugan N. B., Harikumar, R.: Design and Implementation of VLSI Fuzzy Classifier for Biomedical Application. International Journal of Innovative Research in Science, Engineering and Technology, 2641-2648 (2014)
21. Balamurugan, N. B., Jothi, M., Harikumar, R.: FPGA Synthesis of SIRM Fuzzy System-Classification of Diabetic Epilepsy Risk Level. Procedia engineering, vol. 38, 391-404 (2012)
22. Nilosey, A.: FPGA Based Diabetic Patient Health Monitoring Using Fuzzy Neural Network. International Journal of Science and Research (IJSR), 394-396 (2014)
23. Barriga, A., Sánchez-Solano, S., Brox, P., Cabrera, A., Baturone, I.: Modelling and implementation of fuzzy systems based on VHDL. International Journal of Approximate Reasoning, vol. 41, no. 2, 164-178 (2006)
24. Graas, E. L., Brown, E. A., Lee, R. H.: An FPGA-based approach to high-speed simulation of conductance-based neuron models. Neuroinformatics, vol. 2, no. 4, 417-435 (2004)
25. Oliveira, D. N., de Souza Braga, A. P., da Mota Almeida, O.: Fuzzy logic controller implementation on a FPGA using VHDL. In Annual Meeting of the North American Fuzzy Information Processing Society, Toronto, ON, Canada (July 2010)
26. Orsila, H., Kangas, T., Salminen, E., Hämäläinen, T. D., Hännikäinen, M.: Automated Memory-Aware Application Distribution for Multi-Processor System-on-Chips. Journal of Systems Architecture, vol. 53, no. 11, 795-815 (2007)
27. Raychev, R., Mtibaa A., Abid, M.: VHDL modeling of a fuzzy coprocessor architecture. In Proceedings of International Conference on Computer Systems and Technologies, Varna, Bulgaria (2005)
28. Lilford, R. J., Pauker, S. G., Braunholtz D. A., Chard, J.: Decision analysis and the implementation of research findings. BMJ, Vol. 317, no. 7155, 405-409 (1998)
29. Peulić, A., Šušteršič T., Peulić, M.: Non-invasive improved technique for lumbar discus hernia classification based on fuzzy logic. Biomedical Engineering/Biomedizinische Technik, vol. 64, no. 4, 421–428 (2018)

30. Bhole, K., Agashe, S., Deshpande, A.: FPGA implementation of type 1 fuzzy inference system for intravenous anesthesia. In IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (July 2013)
31. Tekscan: Pressure Mapping, Force Measurement & Tactile Sensors (2018) [Online]. Available: http://www.tekscan.com/flexible-force-sensors (current May 2018).
32. Sustersic, T., Rankovic, V., Peulic M., Peulic, A. S.: An Early Disc Herniation Identification System for Advancement in the Standard Medical Screening Procedure based on Bayes Theorem. IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 1, 151-159 (2020)
33. Kusiak, A.: Fuzzy Logic. The University of Iowa, Iowa City (2004)
34. Deliparaschos, K. M., Tzafestas, S. G.: A parameterized TS digital fuzzy logic processor: soft core VLSI design and FPGA implementation. International Journal of Factory Automation, Robotics and Soft Computing, vol. 3, 7-15 (2006)
35. Ahn, U., Ahn, N., Buchowski, J., Garrett, E., Sieber, A., Kostuik, J.: Cauda equina syndrome secondary to lumbar disc herniation: a meta-analysis of surgical outcomes. Spine, vol. 25, no. 12, 1515-1522 (2000)
36. Nexys D.: Reference Manual Nexys2 (2008) [Online]. Available: http://www.ece.umd.edu/class/enee245.F2016/nexys2_reference_manual.pdf (current May 2019)
37. Suri, P., Rainville, J., Katz, J. N., Jouve, C., Hartigan, C. Limke, J.: The accuracy of the physical examination for the diagnosis of midlumbar and low lumbar nerve root impingement. Spine, vol. 36, no. 1, 63 (2011)
38. McCarthy, C., Gittins, M., Roberts C., Oldham, J.: The reliability of the clinical tests and questions recommended in international guidelines for low back pain. Spine, vol. 32, 921-926 (2007)
39. Peulić, M., Joković, M., Šušteršič, T., Peulić, A.: A Noninvasive Assistant System in Diagnosis of Lumbar Disc Herniation: Computational and Mathematical Methods in Medicine. Article ID 6320126 (2020)

Tijana Šušteršič is a Teaching Assistant and Ph.D student at the Faculty of Engineering, University of Kragujevac, Kragujevac, Serbia. She finished M.Sc. and B.Sc. studies at the same Faculty with GPA 10.00/10.00 and as the best student in generation. She participated in the National project by Ministry of Education, Science and Technological Development of the Republic of Serbia [grant number OI174028], three COST projects, several Horizon 2020 projects and was part of bilateral project between Austria-Serbia and Slovenia-Serbia. She is the author and co-author of 14 papers presented at the international conferences and 11 papers published in international journals. She possesses Certificate of Proficiency in English (CPE) level C2 and Goethe Zertifikat in German, level B2 according to the Common European Framework of Reference for Languages. The focus of her current research is on biomedical signal processing using machine learning, as well as implementation of algorithms on Field Programmable Gate Array (FPGA).

Miodrag Peulić is a Teaching Assistant at Faculty of Medical Science - University of Kragujevac and he is a director of Center of Neurosurgery in Clinical Center of Kragujevac. He finished Faculty of Medical Science at University of Kragujevac in 1999. He became specialist of neurosurgery in 2007. at Faculty of Medicine – University of Belgrade. He finished Specialist-academic studies in 2010. and subspecialization of oncology in 2019. at University of Belgrade. He is author and co-author of 8 published papers in international journals. He deals with the operative treatment of patients from various fields of neurosurgery.

Aleksandar Peulić is a Full Professor at the Faculty of Geography, University of Belgrade, Belgrade, Serbia, and part time professor at Faculty of Science, University of Kragujevac, Kragujevac, Serbia. He received the Diploma degree in electronic engineering and the Master of Science degree in electrical engineering both from the Faculty of Electronic Engineering, University of Nis, Nis, Serbia, in 1994 and 2004, respectively and PD degree at University of Kragujevac in computer science 2007. He was a Postdoctoral Research Fellow at the University of Alabama, Huntsville, in 2008 and University of Applied Science, Augsburg, Germany, in 2010. He is the author and co-author more than 80 papers presented at the international conferences and published in peer review journals. His research interests include embedded systems, digital signal processing and image processing.

Received: February 24, 2020; Accepted: January 15, 2021.

Computer Science and Information Systems 18(3):641–656 https://doi.org/10.2298/CSIS200124015V

A New Approximate Method For Mining Frequent Itemsets From Big Data [⋆]

Timur Valiullin, Joshua Zhexue Huang[⋆⋆], Chenghao Wei, Jianfei Yin, Dingming Wu,
and Iuliia Egorova

Big Data Institute, College of Computer Science and Software Engineering
Shenzhen University
518000 Shenzhen, China
{timur, zx.huang, chenghao.wei, yjf, dingming}@szu.edu.cn, jnegorova@gmail.com

Abstract. Mining frequent itemsets in transaction databases is an important task in many applications. It becomes more challenging when dealing with a large transaction database because traditional algorithms are not scalable due to the limited main memory. In this paper, we propose a new approach for the approximately mining of frequent itemsets in a big transaction database. Our approach is suitable for mining big transaction databases since it uses the frequent itemsets from a subset of the entire database to approximate the result of the whole data, and can be implemented in a distributed environment. Our algorithm is able to efficiently produce high-accurate results, however it misses some true frequent itemsets. To address this problem and reduce the number of false negative frequent itemsets we introduce an additional parameter to the algorithm to discover most of the frequent itemsets contained in the entire data set. In this article, we show an empirical evaluation of the results of the proposed approach.

Keywords: Approximation Method, Frequent Itemsets Mining, Random Sample Partition, Big Transactional Database.

1. Introduction

Frequent itemsets mining is the first and most critical step of finding association rules from a transaction database. Association rules mining is one of the main data mining tasks in many applications, such as basket analysis [3], product recommendation [20], cross-selling [10], etc. Huge research efforts have been devoted to solving frequent itemsets mining problem. Many of these studies had considerable impact and led to a plenty of sophisticated and efficient algorithms for association rules mining, such as Apriori [1,2], FP-Growth (Frequent Pattern Growth) [8,6,13,7], and Eclat [22,21,18]. However, decade of a fast development of e-commerce, online and offline shopping has resulted in the fast growth of transaction data, which presents a tremendous challenge to these existing algorithms, because these algorithms require large memory to run efficiently on large transaction databases.

Parallel and distributed association rules mining algorithms were developed to handle large transaction databases. Parallel association rules mining algorithms use in-memory

[⋆] This is an extended version of the DAMDID 2019 conference paper ”A New Approach for Approximately Mining Frequent Itemsets.”

[⋆⋆] Corresponding author

computing to efficiently mine association rules from a large transaction database. However, their scalability is limited by the size of the memory of the parallel systems. Distributed association rules mining algorithms were developed using MapReduce [5], and run on a Hadoop cluster platform. These algorithms have better scalability, but they are not efficient for mining of a large transaction data sets because of frequent I/O operations and communication overhead between nodes.

In this paper, we propose a new approach to solve the problem of mining frequent itemsets from a big transaction data set. Similar to the distributed algorithms in MapReduce, we partition the data set into same size disjoint subsets. However, *we make the distribution of frequent itemsets in each subset similar to the distribution of frequent itemsets in the entire data set by random assignment of the transactions from the entire data set to the subsets without replacement.* As such, we can randomly select a few subsets and run a frequent itemsets mining algorithm on each subset independently to discover the local frequent itemsets from it. After all frequent itemsets are discovered from the subsets, each frequent itemset is voted by all subsets and the one appearing in the majority of subsets is determined as the frequent itemset, called a *popular frequent itemset*.

In this approach, we define the frequency (relative support) threshold for finding all frequent itemsets in the entire transaction database as the global frequency threshold, and the frequency threshold for mining all frequent itemsets in a subset of the transaction database as the local frequency threshold. Based on these two thresholds, we introduce an algorithm for finding the set of approximate frequent itemsets that estimates the set of frequent itemsets in the entire data set. Initially, the algorithm makes the local frequency threshold equal to the global frequency threshold for subsets to mine local frequent itemsets. This setting results in a high-accurate approximate set of frequent itemsets, but the result also contains insignificant amount of both false positive and false negative frequent itemsets. To address this problem, we introduce an additional parameter to make the local frequency threshold smaller than the global frequency threshold for subsets to produce the set of local frequent itemsets. Using a reduced local frequency threshold helps to drastically reduce the number of false negative frequent itemsets and produce high-accurate approximate frequent itemsets that cover most of the frequent itemsets contained in the entire data set with respect to the global frequency threshold.

We conducted experiments to evaluate the approximate solutions on two real world data sets. To evaluate the performance of our method, all popular frequent itemsets discovered from the selected subsets using the local frequency threshold are compared with the frequent itemsets found directly from the entire database using the global frequency threshold. The recalls and precisions of the popular frequent itemsets obtained from the selected subsets are used as evaluation measures. The empirical results have shown that the proposed method is capable of producing high-accurate approximate frequent itemsets and discovering most of the frequent itemsets contained in the entire database that can be found with the global frequency threshold. The comprehensive analysis also shows that reducing the local frequency threshold in the selected subsets enables obtaining all true frequent itemsets.

The remaining part of this paper is organized as follows. Related works are discussed in Section 2. Section 3 describes the proposed approach. In Section 4, the details of the algorithm are presented. Experiment results are shown in Section 5. Finally, conclusions and future work are drawn in Section 6.

2. Related Work

Frequent itemsets mining is a well-studied problem in computer science. However, the enormous data growth made traditional methods inadequate. Therefore, parallel and distributed algorithms came in use.

Authors of [17] implemented a new algorithm called Partition to mine approximate frequent itemsets which achieved both CPU and I/O improvements over Apriori by partitioning the database into a number of non-overlapping partitions so that the partitions are small enough to be handled in the main memory to generate local candidates. On the next step, the local results are merged and the global frequency of the local frequent itemsets is checked in the entire data set. In [19], H. Toivonen introduced new sampling based algorithms to make association rules mining more efficient in terms of computational cost. The proposed approach uses a sample of a data set with a lowered frequency threshold to generate a bigger collection of frequent itemset candidates, and then verifies the candidates with the entire database. Researchers in [9] introduced the parallel implementation of the FP-growth algorithm on GPU. In [11] and [12], the authors introduced two different approaches for mining frequent itemsets in a large database based on MapReduce. In [11], researchers presented two methods for frequent itemsets mining based on Eclat algorithm. The first one is a distributed version of Eclat that partitions the search space more evenly among different processing units, and the second one is a hybrid approach, where the k-length frequent itemsets are mined by an Apriori variant, and then the found frequent itemsets are distributed to the mappers in which frequent itemsets are mined using Eclat. Authors of [12] presented a novel zone-wise approach for frequent itemsets mining based on sending computations to a multi-node cluster. All mentioned approaches have obtained a speed increase over the traditional algorithms and allowed to increase the size of the data set used for mining. However, all introduced approaches require using the entire data set to get the result, which faces the memory bottleneck when dealing with big data and working in a distributed environment.

Researchers in [4], [15] introduced sampling techniques which theoretically proved the existence of the tight bounds of the sample size that guarantees the approximation with respect to the parameters specified by the user. The sample size for mining is not dependent on the size of the database and the number of items. However, in case of a real big data the proposed method might not be applicable since the big sample data, used to achieve the approximation with respect to the parameters specified by the user, may not fit in the memory of a single machine. The proposed approaches are not suitable for the distributed environment. In [14], M. Riondato introduced PARMA algorithm (Parallel Randomized Algorithm for Approximate association rule mining). The algorithm sends random subsets of the database to various machines in the cluster as an input. Then, each machine mines the received subset, and reducers combine the result. Our work follows the idea described in [16]. The random sample partition (RSP) data model was presented, which showed that the block-level samples from an RSP data model can be efficiently used for data analysis.

3. A New Approach

In our approach, we split a big data set into disjoint subsets such that the distribution of frequent itemsets in each subset is similar to the distribution of frequent itemsets in the entire data set. Mining smaller subsets allows using traditional frequent itemset mining algorithms without experiencing memory limit problems. In this research, we have empirically shown that by combining the results of randomly selected subsets, we are able to produce a highly accurate approximate set of frequent itemsets.

3.1. Definitions

A transactional data set $D = \{t_1, t_2, ..., t_n\}$ is represented by a collection of n transactions, where each transaction t is a subset of the set of items $I = \{I_1, I_2, ..., I_m\}$. An itemset A with k distinct items is referred to *k-itemset*. In this paper, we do not distinguish itemsets with different numbers of unique items. Given an itemset A, define $T_D(A)$ as the set of transactions in D which contain A. The number of transactions in $T_D(A)$ is defined as the support of A by D and denoted as $support(A) = |T_D(A)|$. The frequency of A, i.e. the proportion of transactions containing A in D, is denoted as $freq_D(A) = \frac{|T_D(A)|}{n}$, called a *relative support* or *frequency* of A.

Under the above definitions, the task of finding frequent itemsets from D with respect to a minimal global frequency threshold θ is defined as follows:

Definition 1. Given a minimum global frequency threshold θ for $0 < \theta \leq 1$, the frequent itemsets mining with respect to θ is finding all itemsets $\{A_i\}$ for $1 \leq i \leq M$ with $freq(A_i) \geq \theta$, where M is the total number of frequent itemsets found in D. Formally, we define the whole set of frequent itemsets in D as

$$FI(D, I, \theta) = \{(A_i, freq_D(A_i)) : A_i \subset I, freq_D(A_i) \geq \theta\} \tag{1}$$

Definition 2. Let $FI(D, I, \theta)$ be the set of frequent itemsets in D with respect to θ and $M =\mid FI(D, I, \theta) \mid$ the number of frequent itemsets in FI. The accumulative distribution of frequent itemsets in FI is defined as

$$P(f) = \frac{1}{M} \sum_{\forall A_i \in FI} \mathcal{I}(freq_D(A_i) \leq f) \tag{2}$$

where $\mathcal{I}()$ is an indicator function and f is a frequency value for $\theta \leq f \leq 1$. An example of $P(f)$ is shown in Figure 1.

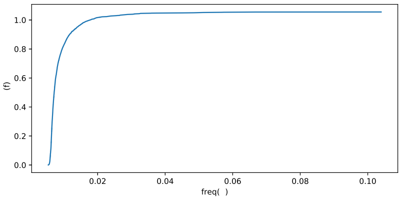


Fig. 1. Example of the accumulative frequent itemsets distribution, where $\theta = 0.005$

Let D be a big transactional data set and $P = \{D_1, D_2, ..., D_k\}$ a partition of D, where $\bigcup_{i=1}^{k} D_i = D$ and $D_i \bigcap D_j = \emptyset$ for $i \neq j$. D_i for $1 \leq i \leq k$ is named as a block of transactions of data set D.

Definition 3. Let $P_D(f)$ be the accumulative distribution of frequent itemsets $FI(D, I, \theta)$ and $P_{D_i}(f)$ the accumulative distribution of frequent itemsets $FI(D_i, I, \theta)$ for $1 \leq i \leq k$. P is a random sample partition of D if

$$P_{D_i}(f) \rightarrow P_D(f) \quad as \quad D_i \rightarrow D \tag{3}$$

where $D_i \rightarrow D$ implies that D_i approaches D as the size of D_i increases. D_i for $1 \leq i \leq k$ is called an RSP data block.

Definition 3 is a redefined definition of random sample partition in [16] with respect to frequent itemsets by replacing the condition of $E[\tilde{F_k}(t)] = F(t)$ with condition (3) where $\tilde{F_k}(t)$ denotes the sample distribution function of t in D_k and $E[\tilde{F_k}(t)]$ denotes its expectation.

3.2. Approximate Frequent Itemsets Mining

When the transaction data set D is big and cannot be held in memory, we cannot run a frequent itemsets mining algorithm on D to find all frequent itemsets $FI_D(D, I, \theta)$. In this situation, we randomly select a set of l RSP data blocks $\{D_1, D_2, ..., D_l\}$ from the partition P and use the frequent itemsets found from the selected RSP data blocks to estimate the set of global frequent itemsets $FI_D(D, I, \theta)$. This approach is called approximate frequent itemsets mining.

Definition 4. Let itemset A be a frequent itemset in $FI_{D_i}(D_i, I, \theta)$ for $1 \leq i \leq l$. A is called a *popular frequent itemset* if

$$\sum_{i=1}^{l} \mathcal{I}(A \in FI_{D_i}(D_i, I, \theta - \epsilon)) > a \tag{4}$$

where $\mathcal{I}()$ is an indicator function, ϵ for $0 \leq \epsilon < \theta$ is a parameter to reduce the local frequency threshold from the global frequency threshold value θ, and a is a given integer greater or equal to $l/2$, so Equation 4 is a simple majority voting.

The set of all popular frequent itemsets PFI from $FI_{D_i}(D_i, I, \theta)$ for $1 \leq i \leq l$ is the estimation of the set of global frequent itemsets $FI_D(D, I, \theta)$. Given PFI and assuming $FI_D(D, I, \theta)$ is known, an itemset A has one of the following statuses:

- true positive if $A \in PFI$ and $A \in FI_D(D, I, \theta)$.
- false positive if $A \in PFI$ but $A \notin FI_D(D, I, \theta)$.
- false negative if $A \notin PFI$ but $A \in FI_D(D, I, \theta)$.

We intentionally omit the true negative frequent itemsets from the above definition because we are not interested in mining infrequent itemsets (itemsets with frequency $< \theta$).

4. An Approximate Frequent Itemsets Finding Algorithm

In this section, we propose a new algorithm for finding the set of approximate frequent itemsets from a set of l RSP data blocks $\{D_1, D_2, ..., D_l\}$ randomly selected from the partition of a big transaction data set D, and using the local frequent itemsets to estimate the set of frequent itemsets in D with respect to a global frequency threshold θ. The algorithm contains four steps: RSP data blocks generation; RSP data blocks selection; local frequent itemsets mining; finding the approximate set of frequent itemsets from the sets of local frequent itemsets by voting.

The pseudo code is presented in Algorithm 1. The inputs are: a transaction database D, the number of transactions in each RSP data block m, the number of RSP data blocks selected for finding frequent itemsets l, two parameters θ and ϵ, and the parameter of the popular frequent itemset voting condition α. The output of this algorithm is the set of the popular frequent itemsets PFI.

The first step in lines 1-6 is to convert D to a partition of k RSP transaction blocks. Each record in D represents one purchase transaction and the transactions with one purchased item are removed. Given the size of the RSP data block m, the number of RSP data blocks in the partition k is the number of transactions in D divided by m. To generate k RSP data blocks, m transactions are randomly selected from D without replacement and assigned to each RSP data block. In the second step in lines 7-9, l RSP data blocks are randomly selected from the k RSP data blocks of the partition without replacement. In the third step in lines 10-15, Apriori algorithm is called with parameter value of $(\theta - \epsilon)$ as the local frequency threshold to find the local frequent itemsets in each of l RSP transaction blocks $\{D_1, D_2, ..., D_l\}$. In the fourth step in lines 16-23, all local frequent itemsets are voted by all sets of local frequent itemsets. The result of a frequent itemsets mining algorithm is a set of string objects, therefore in this article, when comparing the local results and counting the number of appearances of a frequent itemset in all RSP data blocks, we mean an exact match of string objects. Thus, if a local frequent itemset occurs in RSP data blocks more than the given number as shown in Equation 4, the frequent itemset is added to the set of popular frequent itemset; otherwise, it is discarded.

In this article, we provide a comprehensive empirical analysis of the influence of parameter ϵ on the approximate result. We separately consider two cases. The first one is when we set $\epsilon = 0$ which indicates that the local frequency threshold is equal to the global frequency threshold θ. The second one is $0 < \epsilon < \theta$ which indicates that the local frequency threshold is smaller than the global frequency threshold θ. In Section 5, we will show that even the small size of the data block with sufficient number of selected RSP data blocks enables obtaining high quality estimation of the set of global frequent itemsets, however the set of popular frequent itemsets will contain insignificant amount of false positive frequent itemsets, moreover setting $\epsilon = 0$ does not allow discovering all true frequent itemsets from the selected RSP data blocks. To address this problem, we reduce the local frequency threshold for mining local frequent itemsets. Decreasing of the local frequency threshold by increasing the value of ϵ will result in increasing of the number of the local frequent itemsets for each RSP data block. Thus, more true positive frequent itemsets can be discovered from the selected RSP data blocks, i.e. the number of false negative frequent itemsets will be reduced, however it will also increase the content of false positive frequent itemsets in the approxiamte solution. Empirical analysis for choosing ϵ and experiment results for both cases are shown in Section 5.

Algorithm 1 Algorithm for approximate discovery of frequent itemsets from a big transaction database using random sample partition

Input:
-D: transaction database;
-m: number of transactions in each RSP data block;
-l: number of RSP data blocks selected for finding frequent itemsets;
-θ: the global minimum frequency threshold;
-ϵ: local minimum frequency threshold deduction parameter;
-α: popular frequent itemset voting condition parameter.

```
procedure RSP_GENERATION($D, m$)
    $k = \frac{|D|}{m}$ ▷ $|D|$ is the number of transactions in $D$ and $k$ is the number of RSP blocks to be generated.
    for each $D_i$, $1 <= i <= k$ do
        randomly assign $m$ transactions from $D$ to the $i$-th RSP data block without replacement
    end for
end procedure
procedure RSP_BLOCK_SELECTION($\{D_k\}, l$)
    $\{D_l\}$ = random.sample($\{D_k\}$, $l$) ▷ randomly select $l$ RSP data blocks from the set $\{D_k\}$ and put them in set $\{D_l\}$.
end procedure
procedure LOCAL_FIs($\{D_l\}, l, \theta, \epsilon$)
    for each $D_j$, $1 <= j <= l$ do
        $FI_j = Apriori(D_j, \theta - \epsilon)$
        $\{FI_l\}$.append($FI_j$)                ▷ $\{FI_l\}$ is the set of $l$ sets of local frequent itemsets.
    end for
end procedure
procedure POPULAR_FIs($\{FI_l\}$)
    $\{FI\} = dictionary(\cup_{i=1}^{l} FI_i)$ ▷ for all frequent itemsets found, create $<key, value>$ pair, where itemset is a key and the number of repeats in all RSP data blocks is a value.
    for each $frequent\_itemset \in \{FI\}$ do
        if value $>\alpha$ then
            $PFI$.append($frequent\_itemset$) ▷ append a popular frequent itemset to the set of popular frequent itemsets.
        end if
    end for
end procedure
Output: set of the popular frequent itemsets $PFI$
```

5. Experiments

We conducted 30 experiments on two real world transaction data sets. To evaluate the quality of the approximate results, we compared popular frequent itemsets with the frequent itemsets found directly from the entire database with respect to the global frequency threshold θ. The comparison has shown that our approach produced good approximate results and is efficient in mining approximate frequent itemsets from a big transaction database. In this section, we present the real world data sets, experiment settings, evaluation methods and the experiment results.

5.1. Data Sets and Experiment Settings

The two data sets used in these experiments were downloaded from Kaggle.com and Open-Source Data Mining Library, respectively. The characteristics of the data sets are listed in Table 1.

Table 1. The two data sets used in experiments

	Kaggle data set	Online Retail data set
Number of transactions	729148	541908
Number of items	791	2603
Average transaction length	8	4

Thirty experiments were conducted on each data set with different settings on the number of RSP data blocks and the block sizes. The setting values of these experiments are given in Table 2. The global frequency threshold was set as $\theta = 0.005$ so a big set of frequent itemsets was discovered from the entire data set.

Table 2. Settings on number of RSP data blocks and block sizes

Number of RSP data blocks	Block sizes
50	10000, 5000, 3500, 2000, 1000
30	10000, 5000, 3500, 2000, 1000
15	10000, 5000, 3500, 2000, 1000
10	10000, 5000, 3500, 2000, 1000
5	10000, 5000, 3500, 2000, 1000

5.2. Evaluation Measures

In these experiments, we used the following three measures to evaluate the approximate results of the proposed approach. In these evaluations, the set of popular frequent itemsets PFI was compared with the set of global frequent itemsets. The popular frequent itemsets in PFI were divided into two classes: true positive frequent itemsets TP and false positive frequent itemsets FP. There is another class of false negative frequent itemsets FN, which can only be found in the global frequent itemsets. Based on the three sets of frequent itemsets, we define the evaluation measures as follows:

$$Recall = \frac{|TP|}{|TP \cup FN|} \tag{5}$$

where $|TP|$ is the number of frequent itemsets in TP and $|TP \cup FN|$ is the number of the global frequent itemsets because $TP \cup FN = FI_D(D, I, \theta)$. This measure shows how good the algorithm is in discovering true frequent itemsets. Since this measure is very important in frequent itemsets mining, we use the parameter ϵ to reduce the local frequency threshold while mining local frequent itemsets for increasing the recall value. As ϵ approaches to θ, the recall value will get close to 1.

However increasing ϵ to θ will affect precision value defined as:

$$Precision = \frac{|TP|}{|TP \cup FP|} \tag{6}$$

where $TP \cup FP = PFI$. Precision measures the fraction of the true frequent itemsets in the set of popular frequent itemsets. This measure shows how good the algorithm is in avoiding discovering of the false positive frequent itemsets. Using $(\theta - \epsilon)$, $\epsilon > 0$ as the local frequency threshold to mine local frequent itemsets tends to discover more local frequent itemsets, therefore potentially increasing the number of false positive frequent itemsets in the approximate result, which negatively affects the precision measure.

To consider the global frequent itemsets, the accuracy measure is defined as:

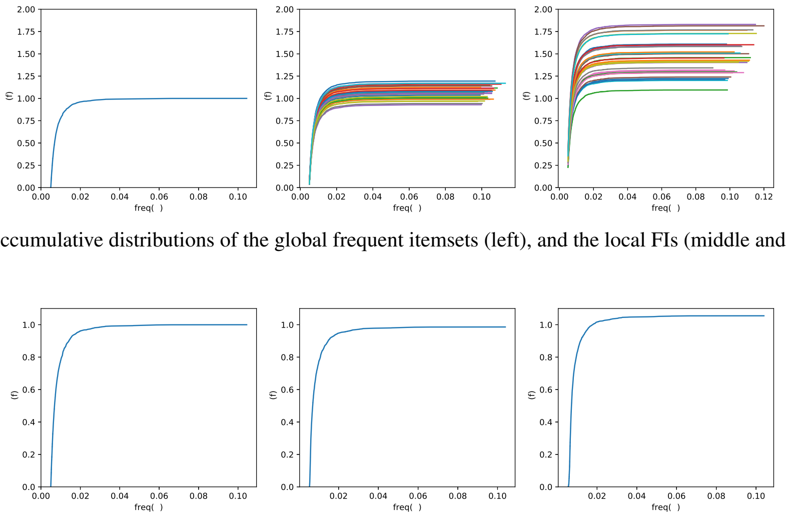
$$Accuracy = \frac{|TP|}{|TP \cup FN \cup FP|} \tag{7}$$

This measure evaluates the approximate result in terms of both precision and recall. Since $TP \cup FN \cup FP = PFI \cup FI_D(D, I, \theta)$, this measure evaluates the quality of the approximate solution in terms of its ability to discover true frequent itemsets, avoiding discovering infrequent itemsets (itemsets with the global frequency $< \theta$).

5.3. Experiment Results With $\epsilon = 0$

In this series of experiments, we set $\theta = 0.005$ and $\epsilon = 0$, i.e. the local frequency threshold is equal to the global frequency threshold. With these settings, we ran Algorithm 1 on the entire data set on different numbers of data blocks and block sizes as specified in Table 2. Figure 2(a) shows the accumulative distributions of frequent itemsets in the whole Kaggle data set and RSP data blocks of different sizes. Subplot on the left shows the distribution in the entire data set. The plots in the middle are the distributions of the frequent itemsets in the RSP data blocks with 10000 transactions. We can see that the distributions of the frequent itemsets in the RSP data blocks are similar to each other and also similar to the distribution in the entire data set. The plots on the right show the distributions of the frequent itemsets in the RSP data blocks with 2000 transactions. Because the block size is much smaller than the blocks for distributions in the middle, a big difference displays among the blocks although the shapes of the distributions are similar to the distribution of the entire data set. We can say that the RSP data blocks in the middle are better samples of the entire data set than the RSP data blocks on the right.

Figure 2(b) shows the accumulative distributions of frequent itemsets in the entire data set and the accumulative distributions of popular frequent itemsets in a number of RSP data blocks of different sizes. Figure 2(b) shows that the three distributions are very similar to each other. The distribution in the middle subplot is very close to the distribution of frequent itemsets in the entire data set. The distribution on the right is also close to the one in the middle. This indicates that the popular frequent itemsets improve the results of frequent itemsets from individual data blocks even with a small block size, and are better estimates of the frequent itemsets in the entire data set with respect to the same frequency



(a) Accumulative distributions of the global frequent itemsets (left), and the local FIs (middle and right)

(b) Accumulative distribution of the global frequent itemsets (left), and accumulative distributions of the popular frequent itemsets (middle and right)

Fig. 2. Accumulative frequent itemsets distributions. Number of RSP data blocks = 30, block size (middle) = 10000, block size (right) = 2000. (Kaggle data set)

threshold θ. Generally speaking, these figures show that the RSP data blocks are good random samples for estimating the frequent itemsets contained the entire data set.

The quality of the approximate results from randomly selected RSP data blocks is evaluated with the measures of Accuracy, Precision and Recall defined in the previous section. Figure 3 shows the changes of accuracy against the data block size in different numbers of selected RSP data blocks. The left figure shows the results of Kaggle data set and the right figure is the results of Online Retail data set. We can see that the accuracy increases as the block size increases. Also, the more the RSP data blocks are selected for voting the popular frequent itemsets, the higher the accuracy of the approximate result is.

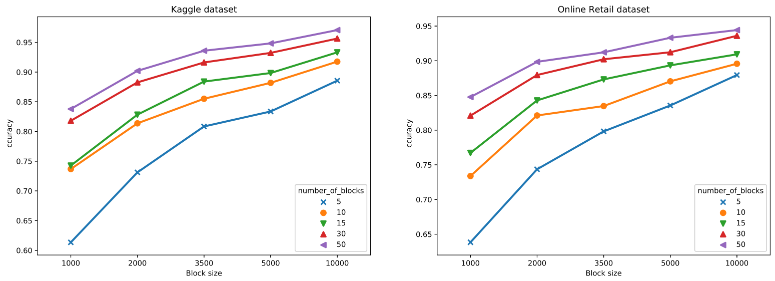


Fig. 3. Accuracy changes with different numbers of RSP data blocks and block size

Figure 4 shows the change of precision against the data block size in different numbers of RSP data blocks. The trends are same as the trends of accuracy because the two measures are essentially same in evaluating the findings of the true positive frequent itemsets in the entire data set from a set of selected RSP data blocks. In all cases, we can approach above 90% of precision with a small portion of the entire data. Based on these figures, we can assume that the bigger number of RSP data blocks is more important than the block size because of the popular frequent itemsets voting. More investigations are needed to find the reasons behind this phenomenon.

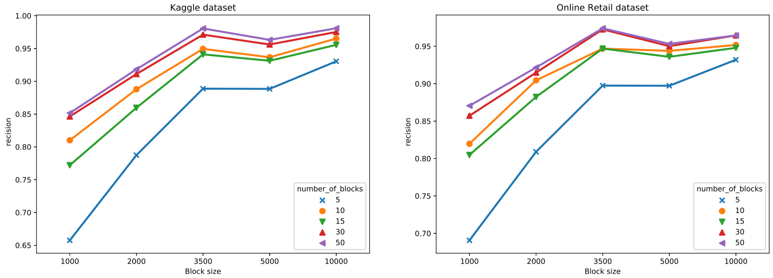


Fig. 4. Precision changes with different numbers of RSP data blocks and block size

In frequent itemsets mining with this approximate approach, Recall is an important measure. Figure 5 shows the change of recalls against the block sizes and different numbers of RSP data blocks. We can see that the trends of recall are different from the trends of accuracy and precision. The general trend is still that the recall increases as the block size and the number of RSP data blocks increase. However, the number of RSP data blocks has a bigger impact on recall. Generally speaking, recall is over 90% even with a few RSP data blocks of a small size, but it rarely arrived 100% in case when the local frequency threshold is equal to the global frequency threshold.

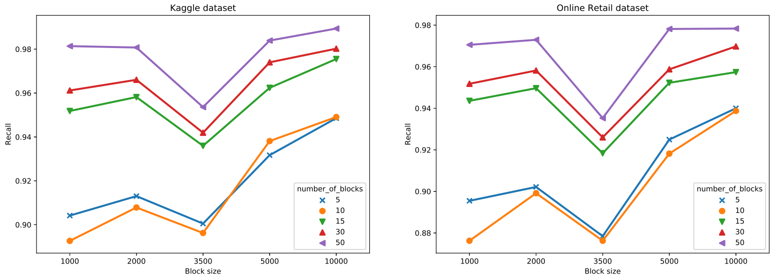


Fig. 5. Recall changes with different numbers of RSP data blocks and block size

5.4. Experiment Results With $0 < \epsilon < \theta$

To increase recall of popular frequent itemsets from the selected RSP data blocks, we set the local frequency threshold to $(\theta - \epsilon)$ and make $\epsilon > 0$. In this series of experiments, we investigate the change of recalls as ϵ increases from 0 to θ. Since the local frequency threshold is reduced, more frequent itemsets will be discovered for the same RSP data blocks. The number of the local frequent itemsets increases as ϵ increases, so the recall of the popular frequent itemsets will increase as well. However, as the number of the local frequent itemsets increases, the number of the false positive frequent itemsets also increases which decreases the accuracy and precision. Therefore, ϵ should be adjusted so that the popular frequent itemsets should not contain too many false positive frequent itemsets. In these experiments, we empirically investigated the value of ϵ which can make a better trade-off between recall, accuracy and precision.

In these experiments, we used the same parameter settings from Table 2. For each setting, we ran Algorithm 1 with a reduced local frequency threshold $(\theta - \epsilon)$, $\epsilon > 0$ to find the popular frequent itemsets from the selected RSP data blocks. Then, we compared the popular frequent itemsets with the frequent itemsets found from the entire data set with the global frequency threshold θ to compute the recall. For each set of parameters from Table 2 we tested several ϵ values. We computed the recall distributions of the popular frequent itemsets discovered with different values of ϵ for corresponding numbers of RSP data blocks and block sizes. Figure 6 shows the recall distributions for all parameter settings from Table 2 for Online Retail Data Set. The rows index is the numbers of RSP data blocks in the descending order as (50, 30, 15, 10, 5), and the columns indicate the sizes of RSP data blocks in transactions as (10000, 5000, 3500, 2000, 1000). Different ϵ values were used in each setting as shown in each display block. From the first row in Figure 6, we can see that for the settings of larger block sizes and more RSP data blocks, a small increase of ϵ, e.g., from 0.0005 to 0.001, results in a big increase of recall which can reach to 1. As the number of RSP data blocks is reduced, the larger ϵ is required to make the recall approach to 1, as shown in the left column of Figure 6. Therefore, as the block size and the number of RSP data blocks are reduced, a big increase of ϵ is required to increase the recall to 1. For example, in the right and bottom display block of Figure 6, ϵ value is 0.002 to make the recall approach 1. Since the global frequency threshold $\theta = 0.005$, the local frequency threshold is reduced with 40% from the global frequency threshold in order to obtain a 100% recall.

We calculated precision with empirically found ϵ value that guarantees the absence of the false negative frequent itemsets in the approximate solution, i.e. recall = 1 for most settings for both data sets. The precisions of the approximate results with different settings are shown in Figure 7. We can see the trends that precision increases sharply as the block size and the number of RSP data blocks increase. In the case of few small RSP data blocks, although the recall is high, the precision is low because a large number of false positive frequent itemsets exist due to a significant reduction of the local frequency threshold. As the block size and the number of RSP data blocks increase, ϵ decreases. Small ϵ allows to maintain low number of false positive frequent itemsets in the approximate solution, which makes the precision stay high. We can see that the precisions in both data sets are over 80% in the settings of large block size and number of RSP data blocks.

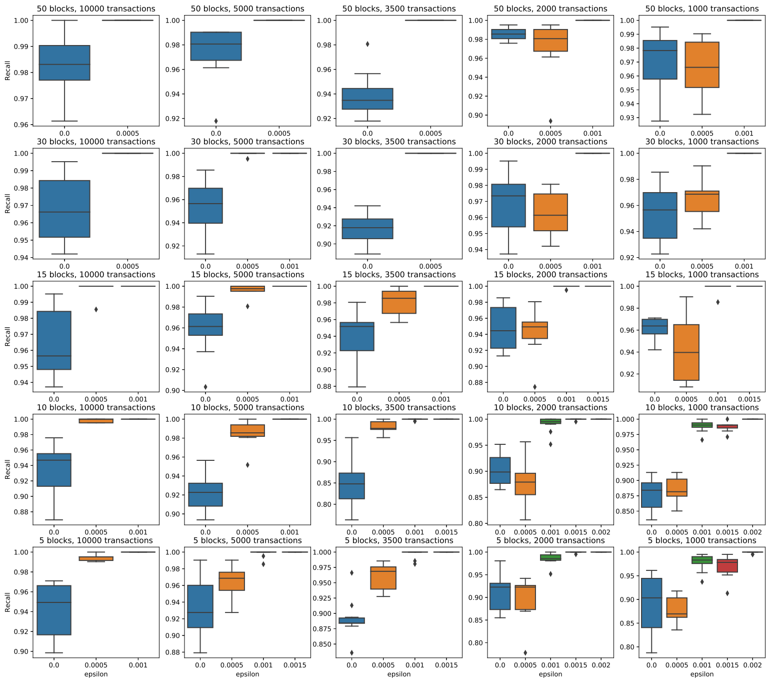


Fig. 6. Distributions of recalls on the different settings of block sizes, numbers of RSP data blocks and ϵ values

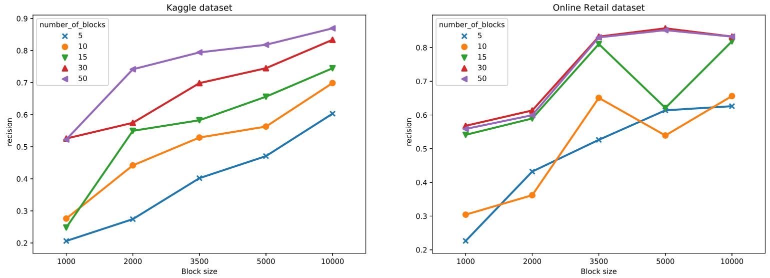


Fig. 7. Precision of the approximate solution with reduced frequency threshold changes with different numbers of RSP data blocks and block sizes

6. Conclusions and future work

In this paper, we have presented a new approach for mining approximate frequent itemsets based on a random sample partition of a big transaction database. We have shown that using the RSP data model for mining frequent itemsets can be very beneficial when the amount of data is very large and traditional single machine or distributed and parallel approaches are no longer able to process it. In this work, we introduced an algorithm for approximate frequent itemsets mining and experimentally showed that the algorithm is able to produce high-accurate frequent itemsets with random sample data blocks, and capable of discovering of all frequent itemsets from the entire data set which is achieved by parameter ϵ. The proposed approach is highly suitable for a distributed architecture and can be effectively run on a computing cluster.

For the further work, we will carry out theoretically statistical analysis on the quality of the outputs of the proposed algorithm and investigate an automatic way of estimating the parameter ϵ. Besides, we are going to implement a parallel version of the algorithm on a cluster and conduct experiments on big data sets in terabyte scale.

Acknowledgments. This research was supported by the National Natural Science Foundation of China under Grant 61972261.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of SIGMOD (1993)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of VLDB (1994)
3. Annie, L., Kumar, A.: Market basket analysis for a supermarket based on frequent itemset mining. IJCSI International Journal of Computer Science Issues 9(3), 257–263 (2012)
4. Chakaravarthy, T., Pandit, V., Sabharwal, Y.: Analysis of sampling techniques for association rule mining. In: ICDT '09 Proceedings of the 12th International Conference on Database Theory. pp. 276–283 (2009)
5. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. In: Proceedings of the CACM. pp. 107–113 (2004)
6. Grahne, G., Zhu, J.: Efficiently using prefix-trees in mining frequent itemsets. In: Proceedings of the CEUR Workshop Proceedings. Melbourne, FL (2003)
7. Grahne, G., Zhu, J.: Reducing the main memory consumptions of fpmax* and fpclose. In: Proceedings of the CEUR Workshop Proceedings. Brighton, UK (2004)
8. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proceedings of the 19th ACM International Conference on Management of Data (SIGMOD). Dallas, TX, USA (2000)
9. Jiang, H., Meng, H.: A parallel fp-growth algorithm based on gpu. In: 2017 IEEE 14th Int. Conf. E-bus. Eng. pp. 97–102 (2017)
10. Kazienko, P., Pilarczyk, M.: Data mining for inventory item selection with cross-selling considerations. New Generation Computing 26, 227–244 (2008)
11. Moens, S., Aksehirli, E., Goethals, B.: Frequent itemset mining for big data. In: 2013 IEEE International Conference on Big Data (2013)
12. Prajapati, D., Garg, S., Chauhan, N.: Interesting association rule mining with consistent and inconsistent rule detection from big sales data in distributed environment. Future Computing and Informatics Journal 2(1), 19–30 (2017)

13. Racz, B.: An fp-growth variation without rebuilding the fp-tree. In: Proceedings of the CEUR Workshop Proceedings. Brighton, UK (2003)
14. Riondato, M., DeBrabant, J., Fonseca, R., Upfal, E.: Parma: a parallel randomized algorithm for approximate association rules mining in mapreduce. In: Proceedings of the ACM International Conference on Information and Knowledge Management (2012)
15. Riondato, M., Upfal, E.: Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. In: ECML PKDD: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 25–41 (2012)
16. Salloum, S., Huang, J., He, Y.: Random sample partition: A distributed data model for big data analysis. IEEE Transactions on Industrial Informatics 15(11), 5846 – 5854 (2019)
17. Savasere, A., Omiecinski, E., Navathe, S.: An efficient algorithm for mining association rules in large databases. In: VLDB ’95: Proceedings of the 21th International Conference on Very Large Data Bases. p. 432–444 (1995)
18. Schmidt-Thieme, L.: Algorithmic features of eclat. In: Proceedings of the Workshop Frequent Item Set Mining Implementations. Brighton, UK (2004)
19. Toivonen, H.: Sampling large databases for association rules. In: VLDB ’96: Proceedings of the 22th International Conference on Very Large Data Bases. p. 134–145 (1996)
20. Wong, R., , Fu, A., Wang, K.: Mining evolving association rules for e-business recommendation. Data Mining and Knowledge Discovery 11, 81–112 (2005)
21. Zaki, M., Gouda, K.: Fast vertical mining using diffsets. In: Proceedings of the 9th ACM International Conference on Knowledge Discovery and DataMining. pp. 326–335. Washington, DC, USA (2003)
22. Zaki, M., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. Newport Beach, CA, USA (1997)

Timur Valiullin received the master’s degree from the Institute of Computational Mathematics and Information Technologies of Kazan Federal University, Kazan, Russia, in 2018. Currently He is a Ph.D. candidate in Computer Science at Shenzhen University, Shenzhen, China.

Joshua Zhexue Huang received the Ph.D. degree in Computer Science from The Royal Institute of Technology, Stockholm, Sweden, in 1993. He is currently a Distinguished Professor of College of Computer Science & Software Engineering, Shenzhen University, Shenzhen, China. He is also the Director of Big Data Institute in Shenzhen University, and the Deputy Director of National Engineering Laboratory for Big Data System Computing Technology, Shenzhen, China.

Chenghao Wei obtained his B.Eng. degree in electronic engineering from the Wuhan University of Science and Technology, Wuhan, China, 2009. He received his M.Eng. degree with distinction from the Department of Electrical Engineering and Electronics in the University of Liverpool, U.K., 2010. Thereafter, he continued his study as a Ph.D. candidate in the same department for oil-immersed power transformer fault diagnosis. Since then, he joined the Big Data Institute of Shenzhen University as a research assistant. His current research focuses on machine learning, big data application, pattern recognition, data analysis.

Jianfei Yin received the Ph.D. degree in Computer Science from the South China University of Technology, Guangzhou, China, in 2005. He is currently an Associate Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

Dingming Wu is Associate Professor of College of Computer Science & Software Engineering at Shenzhen University, China. She received her Bachelor degree in Computer Science at Huazhong University of Science and Technology, Wuhan, China in 2005, and a Master degree in Computer Science at Peking University, Beijing, China in 2008. She received a Ph.D. degree in Computer Science at Aalborg University, Denmark, in 2011. Her research concerns data management, query processing, information retrieval, and data mining.

Iuliia Egorova received the master's degree from the Institute of Computational Mathematics and Information Technologies of Kazan Federal University, Kazan, Russia, in 2020.

Received: January 24, 2020; Accepted: July 15, 2020.

Computer Science and Information Systems 18(3):657–686 https://doi.org/10.2298/CSIS200131013B

Reverse Engineering Models of Software Interfaces *

Debjyoti Bera[1], Mathijs Schuts[2], Jozef Hooman[3], and Ivan Kurtev[4]

[1] ESI (TNO), The Netherlands
debjyoti.bera@tno.nl
[2] Philips, Best, The Netherlands
mathijs.schuts@philips.com
[3] ESI (TNO) and Radboud University, The Netherlands
jozef.hooman@tno.nl
[4] Altran and Eindhoven University of Technology (TU/e), The Netherlands
i.kurtev@tue.nl

Abstract. Cyber-physical systems consist of many hardware and software components. Over the lifetime of these systems their components are often replaced or updated. To avoid integration problems, formal specifications of component interface behavior are crucial. Such a formal specification captures not only the set of provided operations but also the order of using them and the constraints on their timing behavior. Usually the order of operations are expressed in terms of a state machine. For new components such a formal specification can be derived from requirements. However, for legacy components such interface descriptions are usually not available. So they have to be reverse engineered from existing event logs and source code. This costs a lot of time and does not scale very well. To improve the efficiency of this process, we present a passive learning technique for interface models inspired by process mining techniques. The approach is based on representing causal relations between events present in an event log and their timing information as a timed-causal graph. The graph is further processed and eventually transformed into a state machine and a set of timing constraints. Compared to other approaches in literature which focus on the general problem of inferring state-based behavior, we exploit patterns of client-server interactions in event logs.

Keywords: passive learning, process mining, interfaces, model-driven engineering.

1. Introduction

The high-tech industry creates complex cyber-physical systems. These systems consist of many hardware and software components. These components are either developed inhouse or made by third party suppliers. Components interact with each other over software interfaces. Good interface descriptions are crucial for component-based development of cyber-physical systems. Usually software interfaces are only described in terms of their signature, i.e., the set of operations. Sometimes also the allowed sequence of operations is specified, for instance in terms of a state machine or a few example scenarios. The timing behavior of an interface are rarely described. For instance, the expected frequency of notifications and the allowed time between the call of an operation and the corresponding

* This is a revised and extended version of the conference paper [39].

response. Violations of assumptions about timing behavior, however, are an important source of errors over the complete life cycle of these systems.

To overcome the drawbacks of current interface modeling tools, we have developed an Eclipse-based DSL (Domain Specific Language) called ComMA (Component Modeling and Analysis). ComMA [26] is currently used at business unit Image Guided Therapy (IGT) of Philips for specifying interfaces of software components. A ComMA interface specification describes the signature and behavior of a server component. The signature of a server captures the operations it offers to clients and the notifications that it can send to clients. The behavior of a server is specified as a state machine describing the allowed sequence of interactions available to a client. Next to such a state machine, timing constraints on interactions, and data constraints on parameters exchanged during interactions can also be specified. Based on a ComMA specification, a large number of artifacts are generated automatically, for example visualization of the modeled state machine, interface design documentation, simulator, stubs and run-time monitors. The monitor is very useful to check interface compliance after software updates or hardware upgrades. For instance, this is useful when an updated hardware component is obtained from a supplier, or during integration of software developed by different teams. Often during software development, teams are working on different sub-systems with shared interfaces. To facilitate teams to work independently and reduce the chances of costly integration issues later, stubs are particularly useful.

Given the benefits of the ComMA approach, all new system interfaces of Philips IGT are modeled and checked using ComMA. However, there are many existing interfaces and they could benefit from a ComMA specification. A manual transformation based on inspecting event logs and software artifacts (source code, documentation etc.) would require a large reverse engineering effort, is hard to scale up and extremely error prone. On the other hand, the idea of automatically inferring state-based behavior from event logs (also known as passive learning) is not new. This topic has been extensively studied by the two communities of finite state machine inference [11] and process mining [1, 29].

Within both these communities there are popular tools that provide a large variety of techniques to infer automata or Petri net models from event logs. However it is difficult for a non-expert to use these tools directly on event logs to infer interface protocols. For instance, the relation between inferred models and the choice of techniques (and their many configuration parameters and heuristics) is not always clear and may require a deep understanding of these techniques. There is also the additional effort required to translate the output models into a more meaningful domain specific representation. Furthermore, most passive learning approaches focus on ordering of events (actions) but neglect data and time aspects. Moreover, the special case of inferring interfaces of component-based systems can benefit from exploiting patterns of client-server interactions present in event logs. To address these challenges, we present a method to infer component interface models (state machines) and their timing constraints from event logs.

Our approach is based on process mining techniques that exploit the ordering relation between events of an event log (such as α-algorithm [2] and inductive miner [30]). Such techniques usually start by computing causal dependencies between events and represent them as a causal graph (vertices correspond to events). We extend a causal graph to capture both data and time information present in an event log. From a causal graph, we derive a state machine graph and timing constraints over events. Finally, we reduce the state machine graph by merging equivalent states and transform it into a ComMA state machine. At various stages, we exploit patterns of client-server interactions in event logs such as recurring events and compound events to achieve better generalizations in the resulting model.

In contrast, the approach in our previous paper [39], first transforms an event log into a type of Moore automata [24] and then into a ComMA state machine and a set of time constraints. The idea in that paper is to identify *event groups* that start with a client-initiated event (command or signal) followed by zero or more server-initiated events (notifications). Each event group is mapped to a ComMA triggered transition. Such grouping of events often leads to a large number of states since variations in the number and type of notifications will result in different event groups. The approach is also not able to deal with event logs starting with a notification, nor is it correctly able to discover compound events.

Concerning other model learning techniques, we have experimented earlier with active learning which stimulates the system under learning actively and infers an hypothesis based on the responses of the system [41]. Active learning [6] requires the implementation of an adapter to connect the System Under Learning (SUL) with the learner. This adapter has to deal with behavior of the SUL that does not match the assumptions of the learning techniques, such as a SUL which is not input enabled or a SUL which sends no output or multiple outputs after a stimulus. This technique also requires frequent resets of the SUL which may be time consuming. Furthermore, non-determinism of the SUL is a problem for active learning.

A disadvantage of passive learning is that only the behavior that is represented in the used traces will be in the resulting state machine. Hence, compared to the active learning approach, the model might be less complete. An interesting approach presented in [49] exploits the complementary nature of the results produced by passive learning and active learning to improve the final outcome. In our case, however, a passive learning approach is acceptable since the learned model is intended as a starting point for subsequent manual modeling and analysis.

Structure of this paper

The paper is organized as follows. Section 2 provides a brief overview of interface specifications using ComMA. In Section 3 we present our automated workflow to reverse engineer interface models from event logs. Section 4 presents our learning method to infer state and timing behavior. Section 5 describes a few extensions to the learning algorithm. In Section 6, we apply our learning method to two real-life cases at Philips and evaluate the generated models by comparing them to the original models. In Section 7 related work is discussed and we compare our approach to a popular state merging approach for inferring finite state machines. Section 8 concludes the paper.

2. Model-Based Definition of Interfaces in ComMA

In this section, we introduce ComMA [26] as far as needed to understand the remainder of this paper. The ComMA framework consists of the following three main languages: signature, interface and traces. We illustrate the languages using a rather simple example of a vending machine interface called IVendingMachine.

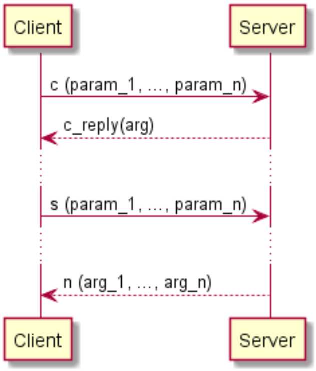


Fig. 1. UML Sequence Diagram of possible event types between a client and server

```
signature IVendingMachine {
    types
    enum Result { OK, NOK }

    commands
    void loadProduct
    Result switchOn
    Result insertCoin
    Result orderCola

    signals
    switchOff

    notifications
    inventoryInfo(int items)
}
```

Listing 1.1. Example of a signature

ComMA Signature. A ComMA signature specification lists a set of events offered by a server to its clients. The events of a signature are distinguished into three types:

- *Commands* are synchronous events from client to server. The client is blocked until it receives a *reply* from the server.
- *Signals* are asynchronous events from client to server.
- *Notifications* are asynchronous events from server to client.

All event types may have data associated with them. Their type definitions are also specified in a signature. To describe data aspects, ComMA provides a set of primitive data types (such as integer, string, boolean, real etc.) and allows the definition of more complex types such as enumerated types, vectors and records.

We refer to data associated with a command or a signal as *input parameters* and data associated with a reply or a notification as *output arguments*.

In the Fig. 1, we give one example of each type of event in the execution context of a client and server. Observe command c and its corresponding reply c_reply. The command c has n input parameters associated with it denoted by *param_1*$, \ldots,$ *param_n*. The reply of command c denoted by *c_reply* has one output argument *arg* associated with it.

```
interface IVendingMachine{
  variables
    int items, coins

  init
    items := 0
    coins := 0

  in all states {
        transition do: inventoryInfo(items)
  }

  initial state Off {
        transition trigger: loadProduct
        do: items := items + 1
        reply
        next state: Off

        transition trigger: switchOn
        guard: items > 0
        do: reply(OK)
        next state: On

        transition trigger: switchOn
        guard: items <= 0
        do: reply(NOK)
        next state: Off
  }

  state On {
        transition trigger: insertCoin
        do: coins := coins + 1
        reply(OK)
        next state: On
        OR
        do: reply(NOK)
        next state: On

        transition trigger: orderCola
        guard: coins > 0 AND items > 0
        do: coins := coins - 1
        items := items - 1
        reply(OK)
        next state: On

        transition trigger: orderCola
        guard: coins <= 0 OR items <= 0
        do: reply(NOK)
        next state: On

        transition trigger: switchOff
        next state: Off
  }
}
```

Listing 1.2. Example of a ComMA state machine

Note that signals and notifications do not have a corresponding reply. In listing 1.1 we present the signature of IVendingMachine.

ComMA Interface. The behavior of an interface in terms of allowed sequence of events is expressed in terms of state machines. A state machine has one or more states (with ex-

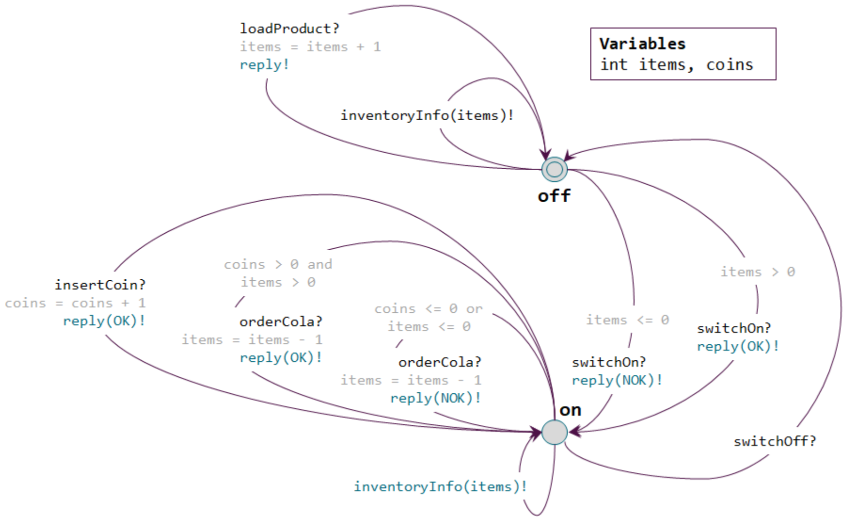


Fig. 2. Server side state machine corresponding Listing 1.2

actly one initial state) and a set of declared and initialized variables. Each state must have one or more transitions. We distinguish between triggered and non-triggered transitions. Both kinds may have an associated guard over the set of defined variables. Triggered transitions (denoted by *transition trigger*) represent an invocation by a client, i.e. either a command or a signal. Non-triggered transitions (denoted by *transition*) represent an event emitted by the server, i.e. notification or reply to a command. The body of a transition consists of one or more *clauses* separated by an OR construct. A *clause* is a sequences of *actions* corresponding variables assignments (using standard mathematical expressions) or events (notifications or replies to commands). Non-determinism between choice of possible transitions in a state is supported by simply defining multiple transitions. Within a transition body we express non-determinism using the OR construct.

We present the interface state machine of our IVendingMachine example in Listing 1.2. A brief explanation is provided below:

- Two variables $items$ and $coins$ are defined and initialized.
- The initial state is Off.
- Notification *inventoryInfo* with parameter $items$ is possible in all states.
- In state Off there is a choice between accepting commands *loadProduct* and *switchOn*. Observe that there are two instances of transition *switchOn* with different guards.
- In state On there is a non-deterministic choice between accepting commands *insertCoin* and *orderCola* and a signal *switchOff*. Observe that *insertCoin* has two possible replies, expressed by the OR construct and different reply values.

In the Fig. 2, we present the communicating state machine (server side) corresponding Listing 1.2 by borrowing the syntax of the popular modeling tool UPPAAL [28] for communicating automata. We extend the notation on arcs between states to represent sequence of communicating events with expressions on variables.

```
timing constraints

TimingRule1
command orderProduct and reply(OK) -> [ 2.4 ms ..  3.8 ms ] between events

TimingRule2
notification inventoryInfo and notification inventoryInfo
    -> [ 400.0 ms .. 550.0 ms ] between events
```

Listing 1.3. Example of a few timing constraints

```
Timestamp: 0.000081 Notification: inventoryInfo Parameter: integer : 0
Timestamp: 2.002300 Notification: inventoryInfo Parameter: integer : 0

Timestamp: 3.030400 Command: switchOn
Timestamp: 3.567788 Reply Parameter: Result::NOK

Timestamp: 4.005600 Command: loadProduct
Timestamp: 4.206180 Reply

Timestamp: 5.640320 Notification: inventoryInfo Parameter: integer : 1
Timestamp: 6.940301 Notification: inventoryInfo Parameter: integer : 1

Timestamp: 7.046780 Command: switchOn
Timestamp: 7.666180 Reply Parameter: Result::OK

Timestamp: 13.100550 Command: orderCola
Timestamp: 13.215671 Reply Parameter: Result::NOK

Timestamp: 17.705500 Notification: inventoryInfo Parameter: integer : 1
Timestamp: 18.905500 Notification: inventoryInfo Parameter: integer : 1

Timestamp: 19.055012 Command: insertCoin
Timestamp: 20.000020 Reply Parameter: Result::NOK

Timestamp: 23.100550 Command: orderCola
Timestamp: 23.215671 Reply Parameter: Result::NOK

Timestamp: 23.908800 Notification: inventoryInfo Parameter: integer : 1
Timestamp: 24.608703 Notification: inventoryInfo Parameter: integer : 1
Timestamp: 25.888101 Notification: inventoryInfo Parameter: integer : 1

Timestamp: 26.000300 Command: insertCoin
Timestamp: 26.006180 Reply Parameter: Result::OK

Timestamp: 27.100550 Command: orderCola
Timestamp: 27.215671 Reply Parameter: Result::OK

Timestamp: 28.030440 Notification: inventoryInfo Parameter: integer : 0
Timestamp: 29.960241 Notification: inventoryInfo Parameter: integer : 0

Timestamp: 30.000300 Signal: switchOff

Timestamp: 36.000330 Command: switchOn
Timestamp: 36.006180 Reply Parameter: Result::NOK
```

Listing 1.4. Fragment of a ComMA trace

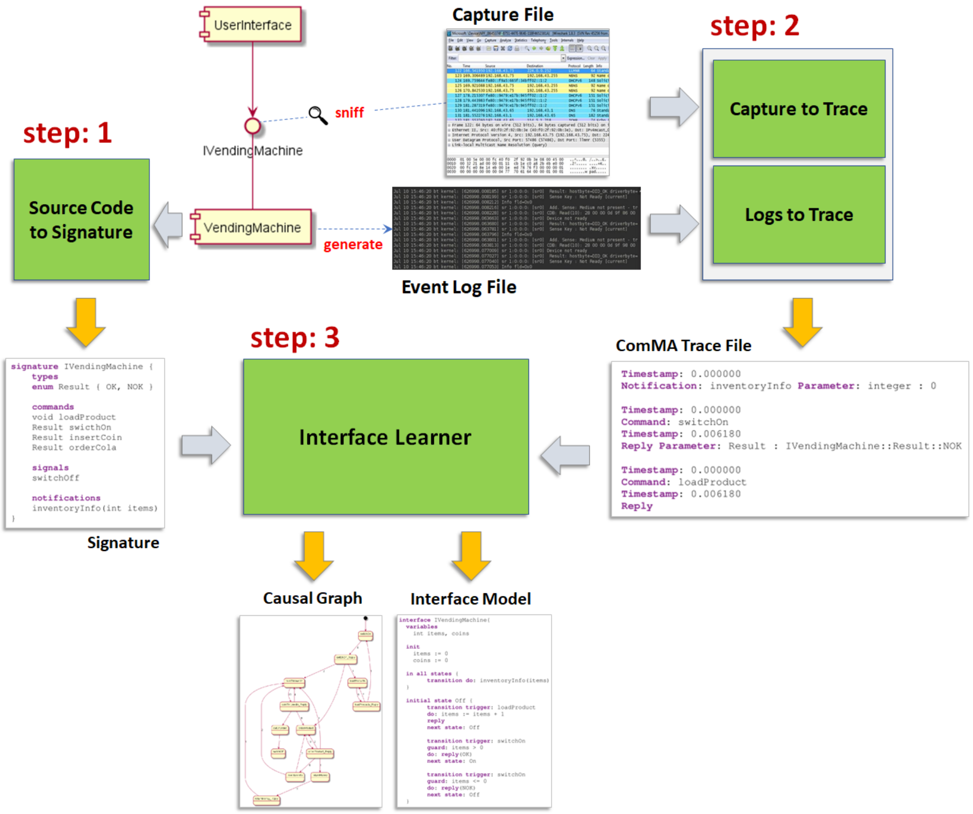
Timing Constraints Next to a state machine specification, a ComMA interface also allows the specification of timing behavior as a set of timing constraints. In ComMA there are four types of timing constraints [25], but we will only consider three of them.

Interval rules specify the time interval between events. *Conditional interval rules* are a further restriction on them. *Rules for periodic events* specify repetitive occurrence of an event within a specified time period and jitter.

Listing 1.3 shows two examples of timing constraints: *TimingRule1* describes the allowed time between an occurrence of command *orderProduct* and its *reply*. The lower bound (LB) is 2.4 ms and the upper bound (UB) is 3.8 ms. *TimingRule2* gives another example of how time intervals between periodic notifications are similarly specified.

ComMA Trace. The trace language in ComMA is used to represent observed interactions between a client and a server in a language independent manner. The idea is to be able to write custom converters from domain specific traces (e.g. sniffing network traffic or from a generated log file) to the ComMA trace language. An example ComMA trace conforming to the IVendingMachine interface is shown in Listing 1.4.

3. Reverse Engineering Interfaces of Legacy Systems



```
signature IVendingMachine {
    types
    enum Result { OK, NOK }

    commands
    void loadProduct
    Result swicthOn
    Result insertCoin
    Result orderCola

    signals
    switchOff

    notifications
    inventoryInfo(int items)
}
```

```
Timestamp: 0.000000
Notification: inventoryInfo Parameter: integer : 0

Timestamp: 0.000000
Command: switchOn
Timestamp: 0.006180
Reply Parameter: Result : IVendingMachine::Result::NOK

Timestamp: 0.000000
Command: loadProduct
Timestamp: 0.006180
Reply
```

```
interface IVendingMachine{
  variables
    int items, coins

  init
    items := 0
    coins := 0

  in all states {
        transition do: inventoryInfo(items)
  }

  initial state Off {
        transition trigger: loadProduct
        do: items := items + 1
        reply
        next state: Off

        transition trigger: switchOn
        guard: items > 0
        do: reply(OK)
        next state: On

        transition trigger: switchOn
        guard: items <= 0
        do: reply(NOK)
        next state: Off
  }
```

Fig. 3. Typical Usage of our Learning Framework

Often the behavior of legacy components is poorly documented and understood. This makes it hard to create an interface model having the right level of abstraction. In the previous section, we have discussed the two ingredients of a ComMA interface model, namely signature and interface. In Fig. 3, we present the different steps to derive a ComMA interface model.

Generating a signature model from available source code is rather easy (step 1). On the other hand, inferring interface behavior in terms of a state machine requires run-time information such as event logs. Often event logs are abundantly available but the information in them may not be very useful due to unclear semantics and incompleteness. In such cases sniffing network channels during system execution and extracting information from them turns out to be a very useful technique to obtain consistent and complete execution data. In all cases, we must create custom transformations to the ComMA trace format (step 2). A prerequisite for performing step 2 is the ability to map the custom messages and their data to the available ComMA event types (as used in signatures) and ComMA data types. ComMA provides the commonly used primitive types such as integer, real, string, enumerations, records, vectors and map types. Our experience shows that these data types are generally sufficient to support non-trivial cases. A limitation of ComMA is the inability to use references to interface instances or services as data types (a feature that can be found in some protocols for distributed computing). The actual conversion from the captured communication or logs to ComMA traces usually requires developing a custom translator. Depending on the complexity of the protocol and the data format, this task may lead to significant efforts. In our practice we have created custom translators that deal with proprietary company specific protocols, and also faced mixed formats that integrate textual, binary and sometimes encrypted data. Clearly, availability of documentation about the protocol is a key enabling factor. Usage of standard protocols and existing tools can greatly reduce the effort in step 2.

The *Interface Learner* is an implementation of the reverse engineering method presented in Sec. 4. Once we have the set of ComMA traces and the signature model, the interface learner generates a timed-causal graph and a ComMA state machine containing time constraints (step 3).

Causal graphs are widely used by many commercial process mining tools [31] [5] as a simple and intuitive means to visualize which activities in a trace can follow one another directly [6]. Most of these tools nicely capture time and frequency based information such as execution times and activity counts.

4. Inferring State Behavior: Interface Learner

We present a step-wise method to transform an event log into a ComMA state machine. First we represent the information in an event log as a *causal graph* (dependency graph) of events extended with time. A *causal graph* is then transformed into a *state machine graph* where edges are events and states are outputs of events (data), i.e. similar to Moore automata [24]. Next all *equivalent* states (i.e. states having same set of possible events) of a state machine graph are discovered and merged. Finally the resulting state machine graph is transformed into ComMA state machine syntax using a simple algorithm.

[5] https://www.celonis.com/, https://processgold.com/en/, https://www. my-invenio.com/

[6] See https://www.gartner.com/doc/3870291/market-guide-process-mining

First we introduce a few notations in Sec. 4.1. In Sec. 4.2 we describe our learning method in steps: logs to causal graphs, causal graphs to state machines, merging equivalent states and finally generating a ComMA interface model and a set of timing constraints.

4.1. Notations

General definitions A finite *sequence* σ over some set S of length $n \in \mathbb{N}$ is a function $\sigma : \{1, \ldots, n\} \to S$. The set of all finite sequences over S is denoted by S^*. We denote a sequence of length n by $\sigma = \langle s_1, \ldots, s_n \rangle$, where $s_1, \ldots, s_n \in S$ and $\sigma(i) = s_i$ for $1 \leq i \leq n$. A sequence of length 0 is called an empty sequence denoted by ϵ.

A *graph* is a pair $G = (V, E)$, where V is the set of vertices and relation $E \subseteq V \times V$ denotes edges. In a directed graph, edges have directions represented by a head and tail. In a directed graph, a sequence $\sigma \in V^*$ of length $n > 0$ is called a *directed path*, if $(\sigma(i), \sigma(i+1)) \in E$ for all $1 \leq i < n$.

We assume a set of *interface actions IA*, consisting of commands, replies, signals and notification, with a function $type$ which yields the type of each action, that is, *COMMAND*, *SIGNAL*, *REPLY*, or *NOTIFICATION*. For a command c we denote its reply by *c_reply*. Henceforth we use $a, a_1, a_2, \ldots$ to denote interface actions, $c, c_1, c_2, \ldots$ to denote commands, $s, s_1, s_2, \ldots$ to denote signals, and $n, n_1, n_2, \ldots$ to denote notifications.

An *event* is a tuple (a, str), where a is an interface action and str is an *output string* which is a string representation of the value of one or more *output arguments* associated with a reply or notification. For commands and signals the output string is empty (see Sec. 2). For now we abstract away from *input arguments* of commands and signals but revisit it later in this section.

Given an event e, we denote its interface action by $action(e)$ and its output string by $output(e)$. The occurrence of an event e at time $t \in \mathbb{R}^+$ is called a *timed event*, denoted as the pair (e, t). We define the projection functions $event(e, t) = e$ and $time(e, t) = t$. Let Σ be the set of all timed events. A *trace* σ is a possibly empty sequence over Σ, $\sigma \in \Sigma^*$. We denote the empty trace by ϵ.
As an example,

$$\sigma = \langle ((c, \text{-}), 0.0); ((c\_reply, \text{OK}), 0.215); ((s, -), 3.51); ((n, 5), 4.11) \rangle$$

is a trace containing first command c at time 0.0 followed by its reply with value OK at 0.215, third signal s at 3.51 and fourth notification n with output 5 at 4.11.

A *log* L is a finite non-empty set of traces $L \subseteq \Sigma^*$. For a given log L, we define $events(L)$ as the set of all *events* occurring in traces of L and $actions(L)$ as the set of all interface actions occurring in $events(L)$.

Restriction [R1] In this section we require for every trace that every occurrence of a command is immediately followed by a reply of this command, that is, there are no intermediate notifications. In Section 5, we will discuss how this restriction can be relaxed.

4.2. The Learning Algorithm

The learning algorithm takes a log and a signature as input and produces an interface model as output. First we convert a log to a causal graph which is later transformed to a state machine.

Logs to Causal Graphs Each trace of a given log L captures a possible order of the occurrences of events in time. A *causal graph* is a directed graph which describes when two events follow each other in a trace. For a log L we define its causal graph $G(L)$ as the graph (V, E) where

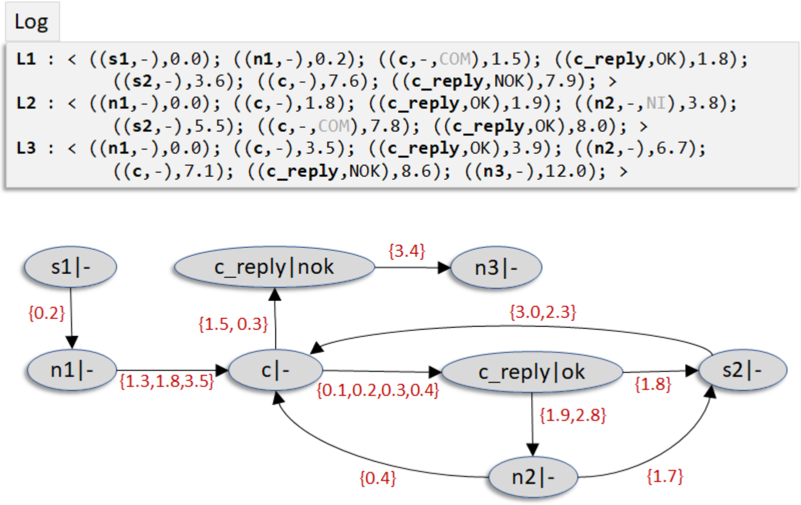
- $V = events(L)$
- $(e, e') \in E$ if and only if there exists a trace $\langle te_1, te_2, \ldots te_n \rangle \in L$ and an $i \in \{1, \ldots, n-1\}$ such that $e = event(te_i)$ and $e' = event(te_{i+1})$.

We denote the set of *initial vertices* as $initial(V) = \{\{event(\sigma(0))\} \mid \sigma \in L\}$.

We associate a set of time durations to each pair of causally related events by the function $\delta : E \to \mathcal{P}(\mathbb{R}^+)$ which is defined as follows:

$$\begin{aligned}\delta(e, e') = \{t \mid & \text{ there exists a trace } \langle te_1, te_2, \ldots te_n \rangle \in L \text{ and an } i \in \{1, \ldots, n-1\} \\ & \text{such that } e = event(te_i),\ e' = event(te_{i+1}) \text{ and} \\ & t = time(te_{i+1}) - time(te_i)\}\end{aligned}$$

We refer to the pair $(G(L), \delta)$ as a *timed causal graph*. An example of a log containing three traces and its corresponding timed causal graph is shown in Fig. 4. The edges are annotated with the set of time durations, as defined by function δ.



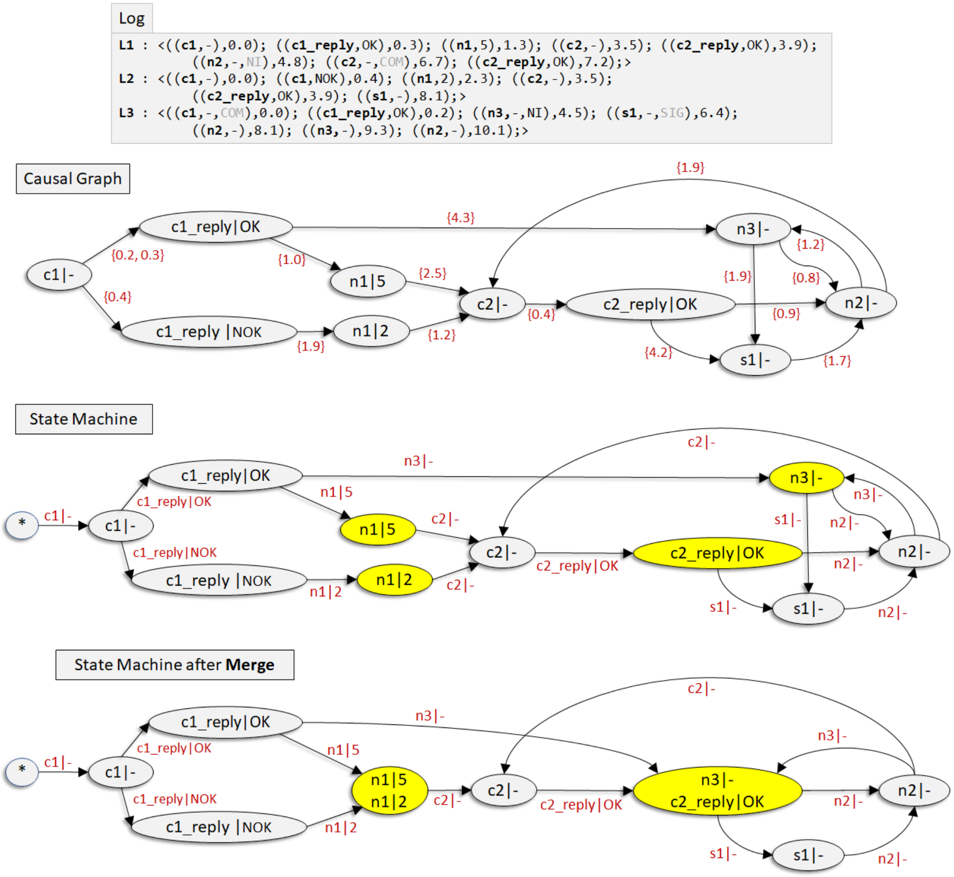
Log

```
L1 : < ((s1,-),0.0); ((n1,-),0.2); ((c,-,COM),1.5); ((c_reply,OK),1.8);
          ((s2,-),3.6); ((c,-),7.6); ((c_reply,NOK),7.9); >
L2 : < ((n1,-),0.0); ((c,-),1.8); ((c_reply,OK),1.9); ((n2,-,NI),3.8);
          ((s2,-),5.5); ((c,-,COM),7.8); ((c_reply,OK),8.0); >
L3 : < ((n1,-),0.0); ((c,-),3.5); ((c_reply,OK),3.9); ((n2,-),6.7);
          ((c,-),7.1); ((c_reply,NOK),8.6); ((n3,-),12.0); >
```

Fig. 4. Three Traces and their Timed Causal Graph

Causal Graphs to State Machines Next we transform a causal graph into a state machine graph, where each state (vertex) represents a set of events and each transition (edge) represents an event. An event of an incoming transition to a state belongs to the set of events associated with that state.

A *state machine* is a tuple (S, A, T, s_0), where S is the set of states, A is the set of actions, $T \subseteq S \times A \times S$ is the set of transitions, and s_0 is the initial state.



Log

```
L1 : <((c1,-),0.0); ((c1_reply,OK),0.3); ((n1,5),1.3); ((c2,-),3.5); ((c2_reply,OK),3.9);
        ((n2,-,NI),4.8); ((c2,-,COM),6.7); ((c2_reply,OK),7.2);>
L2 : <((c1,-),0.0); ((c1,NOK),0.4); ((n1,2),2.3); ((c2,-),3.5);
        ((c2_reply,OK),3.9); ((s1,-),8.1);>
L3 : <((c1,-,COM),0.0); ((c1_reply,OK),0.2); ((n3,-,NI),4.5); ((s1,-,SIG),6.4);
        ((n2,-),8.1); ((n3,-),9.3); ((n2,-),10.1);>
```

Fig. 5. Causal Graph to State Machine

We do not include time durations in a state machine because a timed causal graph is sufficient to derive timing constraints. This is discussed later in the section.

Given a log L and its causal graph $G(L) = (V, E)$, we define a *state machine* $(S, A, T, init)$ where

- $init \notin V$
- $S = \{\{e\} | e \in events(L)\} \cup \{init\}$, i.e. states different from $init$ are singleton sets, each corresponding a distinct event from log L
- $A = events(L)$;
- $T = \{(s_1, e, s_2) \mid e \in s_2$ and $((s_1, s_2) \in E$ or $s_2 \in initial(V) \wedge s_1 = \{init\})\}$ Note that transitions with source $init$ are added to all *initial vertices* of V. The event of a transition is obtained from the target state of the relation in E.

An example of a causal graph and its corresponding state machine is shown in the Fig. 5. Note that we denote the state $init$ by the node containing the symbol $*$.

Algorithm 1: Generate ComMA StateMachine

```
input  : state machine $SM = (S, A, T, s_0)$, name
output: ComMA state machine
print machine name {
for $s \in S$ do
    if $s = s_0$ then
        print initial state StateName($s$) {
    else
        print state StateName($s$) {
    end
    for $(s, e, s_1) \in T$ do
        if $e$ has-type NOTIFICATION then
            print transition do: EventName($e$)
            print next state: StateName($s_2$)
        end
        if $e$ has-type SIGNAL then
            print transition trigger: EventName($e$)
            print next state: StateName($s_2$)
        end
        if $e$ has-typeCOMMAND then
            print transition trigger: EventName($e$)
            print do:
            for $path \in$ getAllPathsToReply($SM, s$) do
                for $(s_1, e, s_2) \in path$ do
                    if $e$ has-type REPLY then
                        print EventName($e$)
                        print next state: StateName($s_2$)
                        if not last for-iteration over paths then
                            print OR
                        end
                    else
                        print EventName($e$)
                    end
                end
            end
        end
    end
    print}
end
print }
```

Merging Equivalent States Once we have a state machine, the goal is to *reduce* it by iteratively discovering all pairs of equivalent states and merging them.

Given a state machine (S, A, T, s_0), two states $s_1, s_2 \in S$ are said to be *equivalent* if and only if $\{a \mid \exists s : (s_1, a, s) \in T\} = \{a \mid \exists s : (s_2, a, s) \in T\}$, i.e., the same set of events are possible.

In the Fig. 5, consider the two pairs of states $\{(n1, 5)\}$ and $\{(n1, 2)\}$, $\{(n3, -)\}$ and $\{(c2\_reply, \text{OK})\}$. It is easy to check that both pairs are *equivalent*. Each state of the first

```
interface ISample{

  initial state init {
        transition trigger: c1
        do: reply(OK)
        next state: executed_c1_OK
        OR
        do: reply(NOK)
        next state: executed_c1_NOK
     }

    state executed_c1_OK {
        transition do: n1(5)
        next state: executed_n1_5_n1_2

        transition do: n3
        next state: executed_n3_c2_OK
    }

    state executed_c1_NOK {
        transition do: n1(2)
        next state: executed_n1_5_n_2
    }

    state executed_n1_5_n1_2 {
        transition trigger: c2
        do: reply(OK)
        next state: executed_n3_c2_OK
    }

    state  executed_n3_c2_OK {
        transition do: n2
        next state: executed_n2

        transition trigger: s1
        next state: executed_s1

    }

    state executed_s1 {
        transition do: n2
        next state: executed_n2
    }

    state executed_n2 {
        transition trigger: c2
        do: reply(OK)
        next state: executed_n3_c2_OK

        transition do: n3
        next state: executed_n3_c2_OK
    }
}
```

Listing 1.5. Generated ComMA State Machine

pair allows a single event $(c2, -)$ (with destination state: $\{(c2, -)\}$), while each state of the second pair allows two events $(s1, -)$ (with destination state: $\{(s1, -)\}$) and $(n2, -)$ (with destination state: $\{(n2, -)\}$).

Given a state machine (S, A, T, s_0) and two equivalent states $s_1, s_2 \in S$, where $s_2 \neq init$, we define a *merge* operation *Merge*$((S, A, T, s_0), s_1, s_2) = (S', A, T', s'_0)$ where

- $S' = S \setminus \{s_1, s_2\} \cup \{s\}$ where s is the union of s_1 and s_2
- T' is obtained from T by replacing all occurrences of s_1 and s_2 by s
- $s'_0 = s$, if $s_1 = init$, $s'_0 = s_0$, otherwise.

It is easy to check that the *merge* operation does not disturb the set of possible paths in the state machine (i.e. action sequences). Also note that the order of applying the *merge* operation does not have an effect on the resulting state machine.

Generating ComMA Interface Model A state machine $SM = (S, A, T, s_0)$ can be transformed into a ComMA interface model, where we assume the following methods:

- *getAllPathsToReply*(SM, s) returns the set of all paths of SM starting at state s and ending with an event of type *REPLY* and no other event in this path is of type *REPLY*. Due to restriction $R1$, a command event is immediately followed by a reply event.
- *StateName*(s) which yields a meaningful string representation (label) of state s, for instance, as a disjunction of event output strings over all incoming edges (see state labels in Fig. 5).
- *EventName*(e) which yields a string representation of event e.

Given a state machine, we present an algorithm (see Alg. 1) to generate its corresponding ComMA interface model. For the last state machine (i.e. after merging) in Fig. 5, the algorithm produces the output shown in Listing.1.5.

Remarks. Often data sets associated with a notification or reply belong to large domains such as integers, real etc. They may also have a complex type such as records and vectors. In such cases the number of resulting states may become very large since the output strings are not equal (for e.g. see notification $n1$ with output 5 and 2 in the Fig. 5). To remedy this we may abstract away from such data sets. This should ideally be indicated by the user as part of configuration parameters of the learner. If we abstract away from all data in notifications and reply (i.e. all output strings are empty), then the size of the resulting state machine is bounded by the number of unique events in an event log. This is easy to check since the number of vertices in a causal graph and its corresponding state machine (excluding *init*) are bounded by that number.

Recall that we did not consider input arguments of commands and signals. It is straightforward to capture them in a similar manner as we did for output data of replies and notifications. To achieve this we only need to extend the tuple representing an *event* to be the tuple $(a, ostr, istr)$, where a is an interface action (as before), $ostr$ is an output string (as before) and $istr$ is an input string which is a string representation of one or more argument values associated with a command or signal. Similar to the previous case, abstraction techniques are needed to avoid an explosion of states.

It is easy to check that the resulting state machine conforms to the input event log since (a) the causal graphs represent all possible ordering of events present in an event log, and (b) the corresponding merged state machine is not disturbing the order of events present in the causal graph. To validate the implementation of our learning algorithm, the monitoring feature in ComMA is used to check for *conformance* between the generated interface model and the set of input event logs [26].

Generating timing constraints. Consider a log L and its timed causal graph $((V, E), \delta)$. From a timed causal graph we could generate all possible timing constraints between pairs of events but not all of them will be useful from a functional requirements point of view, for e.g. between two status reporting notifications or between an error and a status notification etc. Rather timing constraints over response time of an operation to execute a movement on a mechanical device (observable as a specific command and its eventual reply) or between notifications reporting positions of the device are more interesting to check for compliance with safety regulations.

Recall from Sec. 3 that one of the inputs to the interface learner is the signature file containing the syntactic definitions of each unique event occurring in event logs. So as part of the configuration parameters of the interface learner, the user has the possibility to indicate which set of events in the signature are relevant and what *types* of timing constraints over them would be useful to generate.

As pre-requisite for generating timing constraints from a timed causal graph, we assume a few generic methods are present to compute minimum and maximum time durations between events (user indicates which events are useful) present in a log. The notations for these methods are described below.

- Given a command event $c \in V$, we denote the minimum time duration to observe any of its reply events as $c_{min} = \min(\{\min(\delta(c, r)) \mid (c, r) \in E\})$ and maximum time duration to observe any of its reply events as $c_{max} = \max(\{\max(\delta(c, r)) \mid (c, r) \in E\})$.
- Given two events $e1, e2 \in V$ such that $e2$ is reachable from every path starting at $e1$ and there are no cycles in between, let Γ be the set of all paths starting at $e1$ and ending at $e2$. We denote the cumulative minimum and maximum time durations along all paths of Γ by the interval $[e1e2_{min}, e1e2_{max}]$. For the special case of cyclic paths where $e = e1 = e2$, we denote its period by $e_{period} = (e_{max} - e_{min})/2$ and jitter by $e_{jitter} = \max(e_{max} - e_{period}, e_{period} - e_{min})$.

In Listing. 1.6, we provide templates to generate three types of timing constraints. A brief description of these is given below (for detailed semantics, see [25]):

- *TimeForReply* states that if command c is observed then its reply is observed within the specified interval.
- *TimeBetweenEvents* states that if two events $e1$ and $e2$ are observed without observing $e1$ in between then the interval between them is [$e1e2_{min}$ ms ... $e1e2_{max}$ ms].
- *TimeBetweenPeriodicEvents* states that if $e1$ is observed then $e2$ will occur periodically with period $e2_{period}$ and jitter $e2_{jitter}$ until $e3$ observed.

As mentioned earlier, the user indicates the events $c, e1, e2, e3$ as part of configuration parameters of the learner. The Listing 1.7 shows four examples of timing constraints generated from the timed causal graph presented in Fig. 4.

5. Extensions

The learning algorithm presented in the previous section is very general in that it does not exploit patterns of interaction between a client and its server. We present two extensions to the learning algorithm to deal with some commonly occurring patterns.

```
timing constraints

TimeForReply
command $action(c)$
    -[ $c_{min}$ ms ... $c_{max}$ ms ] -> reply

TimeBetweenEvents
$type(action(e1))$ $action(e1)$ and $type(action(e2))$ $action(e2)$
    -> [ $e1e2_{min}$ ms ... $e1e2_{max}$ ms ] between events

TimeBetweenPeriodicEvents
$action(e1)$ then $action(e2)$ with period $e_{period}$ ms jitter $e_{jitter}$ ms until $action(e3)$
```

Listing 1.6. Templates to generate timing constraints

```
timing constraints

TimingConstraint1
command $c$ -[ 0.1 ms ... 1.5 ms ] -> reply

TimingConstraint2
signal $s1$ and command $c$ -> [ 1.5 ms ... 3.7 ms ] between events

%TimingConstraint3
%notification $n1$ and signal $s2$ -> [ 3.2 ms ... 5.7 ms ] between events
```

Listing 1.7. Few timing constraints of the timed causal graph in Fig. 4

The first extension detects recurring events in event logs. In practice these events are usually periodic notifications containing status reporting information but they could also be signals or commands. The second extension detects a generalization of a command-reply pattern by allowing notifications in between. Such patterns are atomic from the point of view of a client because it is blocked until the reply is received. We end the section with a discussion about exploiting domain knowledge to detect hidden dependencies between events present in a log. For e.g. an event in the initialization phase may have an impact on the possible events available in the operational phase.

Using Client-Server Context to Distinguish Recurring Events. Control systems often generate periodic events, for e.g. notifications about status information generated by a server. If we simply transform a log containing recurring events into a causal graph, we may end up with a model where almost every event is possible after a recurring event.

For instance, consider the example in Fig. 6. Here we have a trace containing a periodic notification $n1$. If we ignore $n1$, then we observe that there is a causal relation between client-initiated events, i.e. command c is followed by signal $s1$ and then signal $s2$. When we transform such a trace into a causal graph, the causal relations are lost since $n1$ can be followed by either c, $s1$, $s2$. As a result, we end up allowing too much behavior.

One way to address this problem is by exploiting the fact that client-initiated events (commands and signals) and server-initiated events (notifications) occur in *context* of each other, i.e. a sequence of client-initiated events occur in the context of the last server-initiated event and vice versa. The additional *context* information is easily captured by

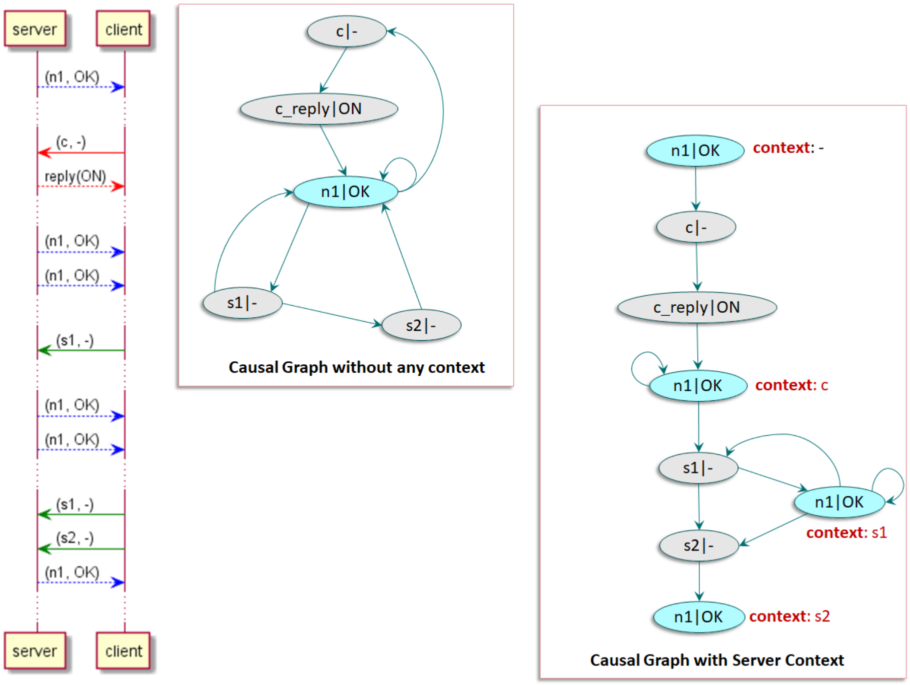


Fig. 6. Handling Periodic Events

extending the vertices of a causal graph with a context attribute. If a client or server initiated event is the first event of a trace then its context attribute is empty.

In the Fig. 6, we show how vertices representing periodic notification $n1$ are extended with context of client-initiated events. The occurrences of $n1$ are now distinguished based on the last occurring event from a client. As a result, causal relations between events $c, s1, s2$ are preserved.

Applying contexts to non-periodic events can have a negative effect on the final result. In practice, the user should be able to decide which events can benefit from a context attribute. This choice can be provided to the learner as part of its configuration parameters.

Inferring Compound Commands Many client-server based control systems support the possibility for a server to raise notifications during the execution of a command, i.e. a sequence of notifications before sending the corresponding reply. We refer to such a pattern as a *compound command*. Note that the client is blocked until it receives a reply from the server. In ComMA, we model such a pattern by adding one or more *notifications* to the body of a *transition trigger* referencing a *command*.

In the Listing 1.8, we give an example where a server can receive a command *switchOn* and as a response, it produces two notification *inventoryInfo* and *powerLevel* before returning a *reply* with value OK.

The trace induced by a *compound command* is a sequence of events starting with a *command*, ending with a *reply* and containing zero or more *notifications* in between, i.e.

```
transition trigger: switchOn
do: inventoryInfo(1)
    powerLevel(85)
    reply(OK)
next state: On
```

Listing 1.8. Compound Command

a sequence of the form $\langle c; n_1; n_2; \ldots n_m; \mathit{c\_reply}\rangle$, where $m \in \mathbb{N}$. In the Listing 1.9, we give an example.

```
Timestamp: 3.030400 Command: switchOn
Timestamp: 13.908800 Notification: inventoryInfo Parameter: integer : 1
Timestamp: 13.908800 Notification: powerLevel Parameter: integer : 85
Timestamp: 3.567788 Reply Parameter: Result::OK
```

Listing 1.9. Trace induced by Compound Command in Listing 1.8

Recall that due to restriction $R1$ in the previous section, we did not consider traces induced by a compound command. To relax this restriction, we make the following modifications to the concepts presented in the previous section.

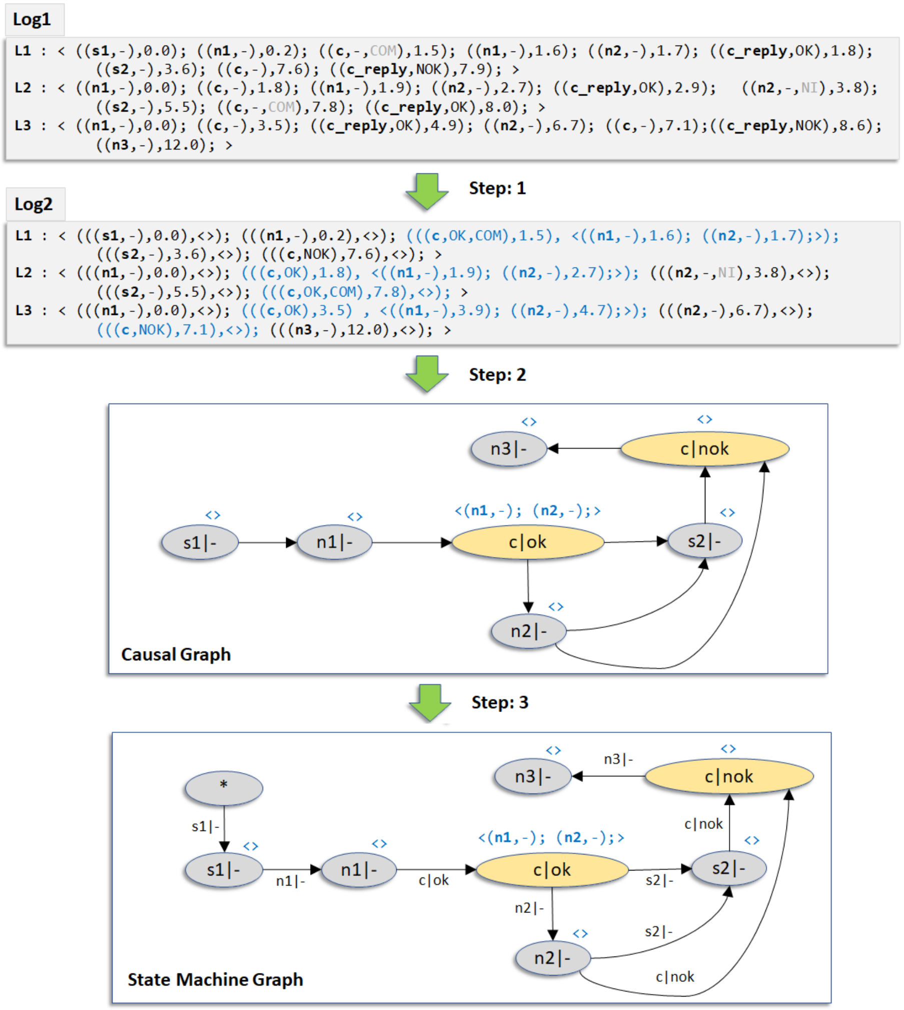
- Drop interface action type *REPLY* and relax the restriction on empty output strings for command events. So instead of using an explicit reply event, we use output string str of a *command* event (c, str) to capture its corresponding *reply* value.
- Lift the definition of an *event* to be the tuple (e, σ), where e is an *event* (as defined before) and σ is a sequence of *notification* events. We require that signal and notification events satisfy $\sigma = \epsilon$.

These modifications imply that event logs must be pre-processed to detect and aggregate subsequences induced by compound commands before constructing the causal graph with the newly extended notion of events. For instance the trace in Listing 1.9 is transformed into event $e = ((switchOn, \text{OK}), \langle \text{inventoryInfo}(5); \text{powerLevel}(85)\rangle)$. We give an example of pre-processing a log in step 1 of Fig. 7.

The transformation to a causal graph and then to a state machine stays the same (see step 2 and step 3 in Fig. 7). However, the algorithm to generate a ComMA state machine syntax must be modified to handle compound commands. This transformation is rather straightforward. For instance, event e must be transformed into the ComMA syntax presented in Listing 1.8.

Note that we still require each input trace to satisfy that for every *command* event there is a matching *reply* event.

Discussion It is often the case that we have some knowledge about the behavior of a component which would otherwise have required a large number of traces. There are two ways to capture such information.



Log1

```
L1 : < ((s1,-),0.0); ((n1,-),0.2); ((c,-,COM),1.5); ((n1,-),1.6); ((n2,-),1.7); ((c_reply,OK),1.8);
          ((s2,-),3.6); ((c,-),7.6); ((c_reply,NOK),7.9); >
L2 : < ((n1,-),0.0); ((c,-),1.8); ((n1,-),1.9); ((n2,-),2.7); ((c_reply,OK),2.9);  ((n2,-,NI),3.8);
          ((s2,-),5.5); ((c,-,COM),7.8); ((c_reply,OK),8.0); >
L3 : < ((n1,-),0.0); ((c,-),3.5); ((c_reply,OK),4.9); ((n2,-),6.7); ((c,-),7.1);((c_reply,NOK),8.6);
          ((n3,-),12.0); >
```

Step: 1

Log2

```
L1 : < (((s1,-),0.0),<>); (((n1,-),0.2),<>); (((c,OK,COM),1.5), <((n1,-),1.6); ((n2,-),1.7);>);
          (((s2,-),3.6),<>); (((c,NOK),7.6),<>); >
L2 : < (((n1,-),0.0),<>); (((c,OK),1.8), <((n1,-),1.9); ((n2,-),2.7);>); (((n2,-,NI),3.8),<>);
          (((s2,-),5.5),<>); (((c,OK,COM),7.8),<>); >
L3 : < (((n1,-),0.0),<>); (((c,OK),3.5) , <((n1,-),3.9); ((n2,-),4.7);>); (((n2,-),6.7),<>);
          (((c,NOK),7.1),<>); (((n3,-),12.0),<>); >
```

Step: 2

Step: 3

Fig. 7. Inferring Compound Command in Traces

One way is to define a set of negative traces (i.e. forbidden order of events) and adapt the learning algorithm to take these orderings into account while constructing the causal graph. In practice such traces are not readily available and are time-consuming to create.

Another way is to specify domain knowledge in terms of temporal constraints [38] and use them in conjunction with a model checker to check for violations [45, 12]. The model checking process can be further augmented by formulating and asking questions to a user in terms of scenarios [16, 19] on the quality of the generalization. The user may accept or reject the generalization, or can provide a new temporal property that handles a larger class of scenarios. Constraints are usually specified as safety (must never be violated) and liveness (must eventually happen) properties [40].

As our algorithm takes into account only causal relations between pairs of directly follows events, a pattern such as event X is eventually followed by event Y is not exploited (long-term dependencies). For instance consider the interface model of IVendingMachine in List. 1.2. The interface state machine does not allow a vending machine with zero *items* to *switchOn* successfully (i.e. with a reply *OK*). It is hard to infer this behavior from traces only because other events can occur in between, e.g. *loadProduct* and *switchOn* (e.g. see List.1.4). Such a dependency can be expressed as a Linear Temporal Logic (LTL) formula.

6. Case Studies

In order to evaluate the learning algorithm we used two example cases for which we already constructed an interface manually earlier. For both interfaces there is a software implementation and a number of traces collected during testing the implementation. The first case is an interface of the power control unit; the second case is an interface of a third-party operating table.

The goal of the presented case studies is to evaluate the interface state machines generated by the learning algorithm in terms of size and understandability. We also compare the generated output to the original manually constructed state machines that were already present. Furthermore, we do not evaluate the generation of timing constraints because it is a work in progress.

Clearly the model inferred by the interface learner depends on the quality of the event log. Most of the times, event logs only contain only a subset of the possible events, usually determined by the execution context. Furthermore, only a part of the interface behavior may be represented by an event log. In order to characterize the input event log we measure the percentage of the used events and the coverage of the logs measured against the original interface models.

For both cases we first automatically checked for conformance of traces against the interface model and then applied the ComMA interface learner. The results from the power control unit case are shown in Table 1. The unit has 5 sub-interfaces: for inspecting the event log (logging); for inspecting the unit self-test results and software version (service); for updating the unit application software and configuration (utility); for monitoring startup and shutdown state (startup/shutdown); and for performing tests by injecting events (test interface).

Table 1. Results of learning experiment with power control unit

Interface	Nr.traces	Coverage	% used events	Size original interface	Size learned interface
Logging	1	83% transitions, 100% states	100%	1 machine, 2 states	4 states
Service	2	100% transitions and states	100%	1 machine, 1 state	5 states
Utility	2	7% transitions, 17% states	25%	1 machine, 6 states	4 states
Startup/shutdown	14	29% transitions, 71% states	54%	2 machines, 7 states	8 states
Test	1	16% transitions, 36% states	50%	1 machine, 11 states	3 states

For every interface the table columns indicate the number of traces used for monitoring and learning, the coverage of the trace set as a percentage of visited states and transitions in the original interface, the percentage of the used interface signature events, the total number of state machines and states in the original interface, and finally the number of states in the learned state machine.

The original Service interface is stateless (hence its specification has a single state). In the traces, the events were observed always in the same order (4 events in total), which explains why the learned state machine contains 5 states. It is anticipated that if the traces contain more permutations of the 4 events then some of the states can be merged by the learning algorithm.

For traces that cover only a small part of the behavior (demonstrated by low coverage and low percentage of used events) the number of states in the learned machine is lower than the original. This observation confirms the intuition that the learned behavior is a subset of the complete one.

All learned state machines are easy to understand partly due to the rules for forming the state names. Their size in terms of number of states is small reflecting the fact that the behavior of the interfaces in this case study is generally not complex.

The interface in the second case study (that of the operating table) is considerably more complex than the one for the power unit. The table can move along 5 axes indicated here as Axis 1 to 5. Moves on several axes can be executed in parallel. All moves have similar behavior captured in a simple state machine with 3 states. Thus the original interface specification consists of 5 orthogonal machines (one for each axis) plus one machine for the startup sequence and the generic parameter notifications. In addition, the table and its clients exchange keep-alive messages with high frequency which results in long traces. The keep-alive messages and parameter value notifications can happen in any state. The experimental set contains 6 traces: one with a move for each axis in isolation and one that combines moves along all axes. For simplicity, the traces do not include startup sequence.

Table 2. Results of learning experiment with the operating table

Trace	Coverage	Size learned behavior
Axis 1	15% transitions, 41% states	13 states
Axis 2	17% transitions, 47% states	13 states
Axis 3	18% transitions, 47% states	13 states
Axis 4	17% transitions, 41% states	13 states
Axis 5	30% transitions, 59% states	18 states
All axes	67% transitions, 94% states	31 states
All traces	71% transitions, 100% states	31 states

The results from this case study are summarized in Table 2. As in the previous case, for each trace set we indicate the coverage over the original interface. First, we applied the learner to one trace per axis to learn the behavior of each move in isolation (rows Axis 1-5). As can be seen, the trace coverage and the size of the results are comparable except for the trace for Axis 5 which appeared to contain moves along 2 axes. The row ”All axes” shows results for a trace containing moves along all axes. As a final step in the experiment we fed the learner will all the 6 traces together (last row). The total coverage of the input increases but the result is not very different in size and topology from the one obtained from the trace with all axes.

A machine with 31 states (obtained when all traces were used) is not easy to comprehend but we have to say that the original behavior specification is not simple either. It was difficult to identify the state behavior for a single move because the move-related events were interleaved with the keep-alive messages. We see a potential to reduce the size of this model by using domain knowledge to filter out keep-alive messages and some status update notifications that can happen at any time. The assumption is that these events do not have an effect on the other events and can therefore be isolated.

As a final observation we would like to note that the generated output with this new algorithm is generally smaller and more manageable than the one reported in the conference version of the paper.

The learned interfaces were inspected by the engineers who created the original specifications thus they had domain knowledge and were experienced in modeling the inspected behavior. As future work we plan to investigate interfaces without pre-existing specifications.

7. Related Work

Inferring state-based behavior from event logs is a well studied topic within Process Mining and Finite State Machine (FSM) Inference communities.

FSM Inference Approaches for FSM-based inference (grammar induction) are based on the learning framework described in [21] which shows that the class of regular languages cannot be identified in the limit from positive strings only, since this almost always leads to over generalization. For instance a self-loop model is always the simplest explanation for any given positive string but such a model is not very useful for analysis. So in practice, heuristics are used to control the amount of generalization present in the final model.

Most popular techniques for FSM inference are based on the *state merging* approach. They start by representing the set of available traces as a prefix tree acceptor and then in steps make generalizations by merging pairs of nodes based on an equivalence notion derived from the well-known Myhill-Nerode relation [36]. So the problem of inferring a state machine from a set of traces is reduced to identifying and scoring equivalent points in the traces that may be suitable merge candidates. Each merge produces a state machine with more allowed behavior (i.e. the set of all possible event orderings). Most popular strategies for generalizing prefix trees can be characterized by Bierman's K-tails algorithm [11] which works on the idea that two points in an execution trace (nodes representing states) are equivalent and can be merged if their future behaviors (up to k-steps) are identical. The work in [15] relaxes the equivalence notion to include subsets of possible behaviors, carried out in the context of discovering software engineering processes. The GK-tails method [33, 46] extends K-tails by considering the influence of data. The method relies on Daikon invariant detection to produce an extended FSM (EFSM), i.e. FSM with data guards on transitions.

When applying simple state merging algorithms to limited traces it is difficult to determine if a compatible merge is truly valid. A bad merge earlier on can have negative consequences on the end result. To some extent this issue is addressed by Evidence Driven State Merging (EDSM) approaches based on the Red-Blue Fringe framework [27]. In an EDSM based approach, each possible merge is given a score based on the amount of evidence of a good merge. The merge with the highest evidence is merged. An extension to the red-blue fringe state merging algorithm that takes into account timing information in traces and infers a timed automata from it, is presented in [43].

Some approaches rely on the user to determine if a merge is good or bad. The work in [16] presents strategies to formulate questions to the user. The user is also able to provide negative traces and temporal constraints to further improve the merge criteria. A similar approach based on model checking is presented in [45] which is extended in [12] using SAT solving techniques. The resulting Mealy machine is transformed into a non-deterministic Moore machine. The trivial solution for the case of positive traces only is the basis for the work in [39]. Note that this is different from the state merging approach.

A popular passive automata learning tool is Flexfringe [44] [7]. The tool provides an efficient implementation of the well-known evidence-driven blue-fringe state-merging algorithm and its probabilistic variants. There are many options to modify search strategies and the user can choose to extend functionality with custom algorithms or rely on standard algorithms such as RPNI [37], ALERGIA [14], EDSM [27], Overlap [23] etc. Another interesting tool is LearnLib [8]. The tool has a focus on active learning but there are also a few RPNI based passive learning algorithms. However most of these tools are difficult to use by a non-expert and solve only the general problem of inferring state machines. It is also not trivial to map the resulting models from these tools to domain specific concepts.

Comparison to the State Merging Approach To compare our approach to state merging approaches based on K-tails, we borrow a nice example of a text editor application from the work in [45]. The idea is simple: once a file is loaded, it can be edited. Only if a file has been edited then it may be saved. Finally a new document can only be loaded when the current document has been closed.

Recall that K-tails like other state merging approaches start by representing the set of available traces as a prefix tree acceptor. In each step, pairs of nodes of this tree are merged if their future behaviors (up to k-steps) are identical, and the resulting model is made deterministic. In addition to the k-value, the choice of pairs and evaluation of a merge rely on heuristics provided by the user.

In the Fig. 8, we present three traces from a log file of such an application and their corresponding FSM generated using the k-tails algorithm with values of $k = 1$ and $k = 2$ (note that the example log and generated state machines are borrowed from [45]).

In general for a user it is difficult to determine the right value of k for which the resulting model is useful, since the user also has not much knowledge about the model being inferred.

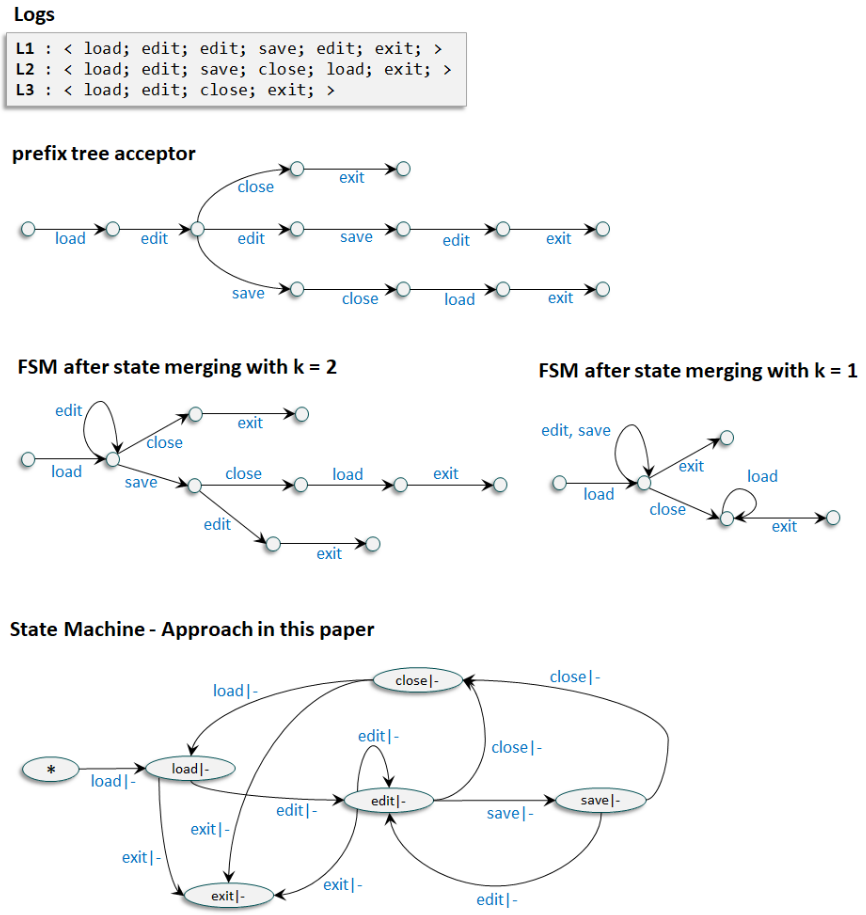
Observe that for higher values of k, we get a FSM with less behavior (i.e. set of all possible event orders). So for $k = 2$ the FSM is almost the prefix tree, whereas for $k = 1$ we get a FSM with more behavior but

- it is possible to save without having edited the file
- it is possible to edit and save the first loaded file but not to the second loaded file.
- it is not possible to load a third file

In the same figure, we also show the state machine generated by our learning method. Since the model preserves the causal relations between events of a given trace, we do not suffer from the problems mentioned above. Furthermore, it is easy for the user to reason about the output state machine, since ordering relations can be checked very easily by either inspecting the set of input traces or its corresponding causal graph.

[7] https://automatonlearning.net/flexfringe/

[8] https://learnlib.de/



```
L1 : < load; edit; edit; save; edit; exit; >
L2 : < load; edit; save; close; load; exit; >
L3 : < load; edit; close; exit; >
```

Fig. 8. Comparing with the K-tails Approach

Process Mining The field of Process Mining aims to discover, monitor and improve processes by extracting knowledge from event logs [1]. Most commercially successful applications of process mining can be seen in the area of organizational business processes (Fluxicon, Celonis, UiPath, Process Gold, ProM, PM4PY)[9] [10]. Petri nets are a widely used formalism to model and analyse business processes [42]. So it is not surprising that most process mining techniques produce their output in terms of a Petri net [1]. In contrast to FSM based inference techniques, process mining techniques take into account the presence of concurrent behavior in event logs.

Early work on process mining can be traced back to three independent papers [5, 17, 15]. The work in [15] developed process discovery techniques in the context of software engineering processes. Among the three methods presented in this paper, the purely algo-

[9] https://www.celonis.com/, https://processgold.com/en/, https://www.fluxicon.com/

[10] https://www. my-invenio.com/, http://www.promtools.org/, https://pm4py.fit.fraunhofer.de/

rithmic approach (based on K-tails [11]) and Markovian approach to deal with noise were considered promising. Around the same time, the work in [5, 17] presented the first applications of process discovery in the context of business processes. The work in [17] adapts the K-tails algorithm with probabilistic elements. However none of the three approaches are able to discover concurrency.

The α-algorithm [2] was one of the first algorithms to mine concurrent behavior along side choices and causal dependencies. It is a simple technique that scans the event log for patterns and distinguishes them in log-based ordering relations, i.e. causal, choice and concurrency. These ordering relations are used by the algorithm to create places to connect transitions of the resulting Petri net.

The learning method presented in this paper uses ordering relations (like in the α-algorithm) between events of a log as its starting point. However for our case of inferring interface models, the presence of concurrent behavior in event logs is not yet a relevant aspect but gives us the nice possibility to extend our method to detect them in the future.

Another interesting approach concerns the theory of regions where the focus is on synthesizing a Petri net from a behavioral specification (for e.g. a transition system), such that the behavior is preserved. There are two main approaches, state-based region [20, 4, 18] and language-based region [10, 48]. Other approaches in process mining include frequency-based techniques such as the heuristics miner [47, 34], abstraction-based techniques such as the fuzzy miner [22] and genetic algorithms [3, 35] which take into account noise and incompleteness of event logs.

The quality of the discovered model with respect to the given log is measured using the four quality dimensions: fitness, simplicity, precision, and generalization [1]. Most approaches guarantee varying levels of fitness and re-discoverability. Among them region-based approaches achieve a good fitness. The problem of guaranteeing sound models with a good fitness is addresed in [30].

Most process mining approaches are not able to handle duplicate tasks since their occurrences are indistinguishable in an event log. As a result, models may become overly connected, negatively affecting the precision and simplicity of the model. Many solutions to detect duplicate tasks have been proposed [35, 32, 13] but the rules to identify them are not sufficiently general for all event logs. Note that we try to address this problem in our learning method by using context information(see Sec. 5).

Freely available tools [11] such as ProM and more recently a Python library (PM4PY) provide access to many mining algorithms. Similar to tools for FSM inference, these tools are also difficult to use by a non-expert and apply them to domain specific concepts. However, the many available professional process mining tools address these problems by providing tailored solutions for the domain of business process management.

8. Conclusions

We have presented a method to infer an interface state machine and a set of timing constraints from an event log. The inferred model is intended to serve as a starting point for subsequent modeling steps. In comparison to other approaches for passive learning, we

[11] http://www.promtools.org/, https://pm4py.fit.fraunhofer.de/

exploit client-server interaction patterns and also take into consideration data and timing information in event logs. Our method can also be configured to deal with recurring events, the choice of generated timing constraints and large data domains of parameters.

Like many process mining techniques [31], we also generate an intermediate causal graph using the directly-follows relation. For a user, such a graph serves as an intuitive way to visualize the information present in an event log and to reason about the resulting interface model. In contrast, the state machines produced by state merging approaches (see Sec. 7 and 4) are difficult for a user to reason about solely based on merge heuristics. Moreover having meaningful state names in interface models greatly improves readability which is also an important aspect for adoption.

Most parts of the method presented in this paper are available in ComMA[12], except support for compound commands and the generation of timing constraints. In future releases, we intend to add these missing features. As future work we see two interesting extensions of our method (1) Extensions to the concept of compound commands to capture more frequently occurring domain specific patterns such as cancelations, etc. [7, 9, 8], and (2) Extensions to infer behavior of components. In typical component-based systems, a component has a set of provided interfaces (to provide services) and a set of required interfaces (to consume services). So the goal is to infer a set of constraints between events of these interfaces based on evidence in event logs. Such an inference must be able to detect concurrent behavior in event logs. In this case exploiting the ordering relation for parallel tasks as presented in the α algorithm [2] could be a nice extension to our method.

Acknowledgments. The research is carried out as part of the VIVACE project under the responsibility of ESI (TNO) with Philips, Best as the carrying industrial partner. The VIVACE research is supported by the Netherlands Organisation for Applied Scientific Research TNO.

References

1. van der Aalst, W.: Process Mining - Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
2. Van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering 16(9), 1128–1142 (2004)
3. Van der Aalst, W.M., De Medeiros, A.A., Weijters, A.: Genetic process mining. In: International conference on application and theory of petri nets. pp. 48–69. Springer (2005)
4. Van der Aalst, W.M., Rubin, V., Verbeek, H., van Dongen, B.F., Kindler, E., Günther, C.W.: Process mining: a two-step approach to balance between underfitting and overfitting. Software & Systems Modeling 9(1), 87 (2010)
5. Agrawal, R., Gunopulos, D., Leymann, F.: Mining process models from workflow logs. In: International Conference on Extending Database Technology. pp. 467–483. Springer (1998)
6. Aslam, K., Cleophas, L., Schiffelers, R., van den Brand, M.: Interface protocol inference to aid understanding legacy software components. Software and Systems Modeling pp. 1–22 (2020)
7. Bera, D.: Petri nets for modeling robots. Ph.D. thesis, Department of Mechanical Engineering (2014)

[12] http://comma.esi.nl/

8. Bera, D., van Hee, K.M., van Osch, M., van der Werf, J.M.E.M.: A component framework where port compatibility implies weak termination. In: Proceedings of the International Workshop on Petri Nets and Software Engineering, Newcastle upon Tyne, UK, June 20-21, 2011. CEUR Workshop Proceedings, vol. 723, pp. 152–166
9. Bera, D., van Hee, K.M., van der Werf, J.M.: Designing weakly terminating ros systems. In: International Conference on Application and Theory of Petri Nets and Concurrency. pp. 328–347. Springer (2012)
10. Bergenthum, R., Desel, J., Lorenz, R., Mauser, S.: Process mining based on regions of languages. In: International Conference on Business Process Management. pp. 375–383. Springer (2007)
11. Biermann, A.W., Feldman, J.A.: On the synthesis of finite-state machines from samples of their behavior. IEEE transactions on Computers 100(6), 592–597 (1972)
12. Buzhinsky, I., Vyatkin, V.: Automatic inference of finite-state plant models from traces and temporal properties. IEEE Transactions on Industrial Informatics 13(4), 1521–1530 (2017)
13. Carmona, J., Cortadella, J., Kishinevsky, M.: A region-based algorithm for discovering petri nets from event logs. In: International Conference on Business Process Management. pp. 358–373. Springer (2008)
14. Carrasco, R.C., Oncina, J.: Learning stochastic regular grammars by means of a state merging method. In: International Colloquium on Grammatical Inference. pp. 139–152. Springer (1994)
15. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. ACM Transactions on Software Engineering and Methodology (TOSEM) 7(3), 215–249 (1998)
16. Damas, C., Lambeau, B., Dupont, P., Van Lamsweerde, A.: Generating annotated behavior models from end-user scenarios. IEEE Transactions on Software Engineering 31(12), 1056–1073 (2005)
17. Datta, A.: Automating the discovery of as-is business process models: Probabilistic and algorithmic approaches. Information Systems Research 9(3), 275–301 (1998)
18. van Dongen, B.F., Busi, N., Pinna, G., van der Aalst, W.: An iterative algorithm for applying the theory of regions in process mining. In: Proceedings of the workshop on formal approaches to business processes and web services (FABPWS'07). pp. 36–55. Publishing House of University of Podlasie, Siedlce, Poland (2007)
19. Dupont, P., Lambeau, B., Damas, C., Lamsweerde, A.v.: The qsm algorithm and its application to software behavior model induction. Applied Artificial Intelligence 22(1-2), 77–115 (2008)
20. Ehrenfeucht, A., Rozenberg, G.: Partial (set) 2-structures. Acta Informatica 27(4), 343–368 (1990)
21. Gold, E.M.: Language identification in the limit. Information and control 10(5), 447–474 (1967)
22. Günther, C.W., Van Der Aalst, W.M.: Fuzzy mining–adaptive process simplification based on multi-perspective metrics. In: International conference on business process management. pp. 328–343. Springer (2007)
23. Heule, M.J., Verwer, S.: Software model synthesis using satisfiability solvers. Empirical Software Engineering 18(4), 825–856 (2013)
24. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to automata theory, languages, and computation. Acm Sigact News 32(1), 60–65 (2001)
25. Kurtev, I., Hooman, J., Schuts, M.: Runtime monitoring based on interface specifications. In: ModelEd, TestEd, TrustEd. pp. 335–356. Springer (2017)
26. Kurtev, I., Schuts, M., Hooman, J., Swagerman, D.J.: Integrating interface modeling and analysis in an industrial setting. In: MODELSWARD. pp. 345–352 (2017)
27. Lang, K.J., Pearlmutter, B.A., Price, R.A.: Results of the abbadingo one dfa learning competition and a new evidence-driven state merging algorithm. In: International Colloquium on Grammatical Inference. pp. 1–12. Springer (1998)
28. Larsen, K.G., Pettersson, P., Yi, W.: Uppaal in a nutshell. International journal on software tools for technology transfer 1(1-2), 134–152 (1997)

29. Leemans, M., van der Aalst, W.M., van den Brand, M.G., Schiffelers, R.R., Lensink, L.: Software process analysis methodology–a methodology based on lessons learned in embracing legacy software. In: 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME). pp. 665–674. IEEE (2018)
30. Leemans, S.J., Fahland, D., van der Aalst, W.M.: Discovering block-structured process models from event logs-a constructive approach. In: International conference on applications and theory of Petri nets and concurrency. pp. 311–329. Springer (2013)
31. Leemans, S.J., Poppe, E., Wynn, M.T.: Directly follows-based process mining: Exploration & a case study. In: 2019 International Conference on Process Mining. pp. 25–32. IEEE (2019)
32. Li, J., Liu, D., Yang, B.: Process mining: Extending α-algorithm to mine duplicate tasks in process logs. In: Advances in Web and Network Technologies, and Information Management, pp. 396–407. Springer (2007)
33. Lorenzoli, D., Mariani, L., Pezzè, M.: Automatic generation of software behavioral models. In: Proceedings of the 30th international conference on Software engineering. pp. 501–510 (2008)
34. de Medeiros, A.K.A., van der Aalst, W.M., Weijters, A.: Workflow mining: Current status and future directions. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 389–406. Springer (2003)
35. de Medeiros, A.K.A., Weijters, A.J., van der Aalst, W.M.: Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery 14(2), 245–304 (2007)
36. Nerode, A.: Linear automaton transformations. Proceedings of the American Mathematical Society 9(4), 541–544 (1958)
37. Oncina, J., Garcia, P.: Identifying regular languages in polynomial time. In: Advances in structural and syntactic pattern recognition, pp. 99–108. World Scientific (1992)
38. Pnueli, A.: The temporal logic of programs. In: 18th Annual Symposium on Foundations of Computer Science (sfcs 1977). pp. 46–57. IEEE (1977)
39. Schuts, M., Hooman, J., Kurtev, I., Swagerman, D.J.: Reverse engineering of legacy software interfaces to a model-based approach. In: 2018 Federated Conference on Computer Science and Information Systems (FedCSIS). pp. 867–876. IEEE (2018)
40. Sistla, A.P.: Safety, liveness and fairness in temporal logic. Formal Aspects of Computing 6(5), 495–511 (1994)
41. Vaandrager, F.: Model learning. Commun. ACM 60(2), 86–95 (2017)
42. Verbeek, H.M., van der Aalst, W.M.: Analyzing bpel processes using petri nets. In: Proceedings of the Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management. pp. 59–78 (2005)
43. Verwer, S., De Weerdt, M., Witteveen, C.: An algorithm for learning real-time automata. In: Benelearn 2007: Proceedings of the Annual Machine Learning Conference of Belgium and the Netherlands, Amsterdam, The Netherlands, 14-15 May 2007 (2007)
44. Verwer, S., Hammerschmidt, C.A.: flexfringe: a passive automaton learning package. In: 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME). pp. 638–642. IEEE (2017)
45. Walkinshaw, N., Bogdanov, K.: Inferring finite-state models with temporal constraints. In: 2008 23rd IEEE/ACM International Conference on Automated Software Engineering. pp. 248–257. IEEE (2008)
46. Walkinshaw, N., Taylor, R., Derrick, J.: Inferring extended finite state machine models from software executions. Empirical Software Engineering 21(3), 811–853 (2016)
47. Weijters, A., van Der Aalst, W.M., De Medeiros, A.A.: Process mining with the heuristics miner-algorithm. Technische Universiteit Eindhoven, Tech. Rep. WP 166, 1–34 (2006)
48. Van der Werf, J.M.E., van Dongen, B.F., Hurkens, C.A., Serebrenik, A.: Process discovery using integer linear programming. In: International conference on applications and theory of petri nets. pp. 368–387. Springer (2008)

49. Yang, N., Aslam, K., Schiffelers, R., Lensink, L., Hendriks, D., Cleophas, L., Serebrenik, A.: Improving model inference in industry by combining active and passive learning. In: 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER). pp. 253–263. IEEE (2019)

Debjyoti Bera received his MSc and PhD degrees from TU Eindhoven. He is currently a researcher at TNO-ESI, an innovation research center with strong partnerships with industry-leading high-tech companies. His research interests include model inference and applications of formal methods in industry.

Mathijs Schuts holds a PhD degree in computer science from the Radboud University Nijmagen. He works at Philips as a software designer. His main research interests include applying formal techniques and domain-specific languages in industry.

Jozef Hooman is a senior research fellow at ESI (TNO), an innovation research center with strong partnerships with industry-leading high-tech companies. In addition, Jozef is a full professor at the Radboud University Nijmegen on model-based development of embedded software.

Ivan Kurtev holds a PhD degree in computer science from University of Twente, the Netherlands. Currently he works as a modeling technologies expert in Altran Netherlands and is a part-time associate professor in the Eindhoven University of Technologies. His main interests are in the area of Model Based Engineering with a focus on applying domain-specific languages for efficient software development.

Received: May 20, 2020; Accepted: December 27, 2020.

Computer Science and Information Systems 18(3):687–702 https://doi.org/10.2298/CSIS200130048G

DrCaptcha: An Interactive Machine Learning Application

Rafael Glikis, Christos Makris, and Nikos Tsirakis

Computer Engineering and Informatics Department,
University of Patras,
26500 Patras, Greece
{rglykys, makri, tsirakis}@ceid.upatras.gr

Abstract. The creation of a Machine Learning system is a typical process that is mostly automated. However, we may address some problems in the during development, such as the over-training on the training set. A technique for eliminating this phenomenon is the assembling of ensembles of models that cooperate to make predictions. Another problem that almost always occurs is the necessity of the human factor in the data preparation process. In this paper, we present DrCaptcha [15], an interactive machine learning system that provides third-party applications with a CAPTCHA service and, at the same time, uses the user's input to train artificial neural networks that can be combined to create a powerful OCR system. A different way to tackle this problem is to use transfer learning, as we did in one of our experiments [33], to retrain models trained on massive datasets and retrain them in a smaller dataset.

Keywords: machine learning, neural networks, transfer learning, interactive machine learning, ensemble techniques.

1. Introduction

A typical process for building a machine learning system starts with the preparation of the data by an expert. This process could be very long and expensive since it must be done manually. Then follows the construction of a model using a learning algorithm on these data. After training, the model is evaluated and if its performance is satisfiable it can be used for predictions in unseen data? But if not this long training process starts again after tweaking the parameters of the learning algorithm. This procedure, many times involves a little trial and error and is repeated until the model performance is sufficient. Depending on the learning algorithm, the size of the training set and the hardware used this process could take from some hours to months! Another problem that occurs very often during the training phase is the over-fitting of the trained model to the training data. That makes the model lose its ability to generalize on unseen data. In this paper, we will try to tackle those problems by using transfer learning [20], ensemble [11] and interactive machine learning [18] techniques as proof of concept.

First, let's take the problems one by one. In order to start training a model, we need to have a sufficient amount of labeled data, and very often the labeling of this dataset must be done by a human. Our approach to this issue is to outsource the labeling to a service we created, that will be used by third-party applications, and use the user's data to prepare our dataset. The next problem we face is the long training time that does not allow the fast formation of a model along with the overtraining of the model in the training set. Our approach to this problem is not overcomplicating things and create simple and neat machine learning models that can be trained in a very short time and give us feedback very fast. But this comes with a price, since the simple models may not be as accurate as the complex models. To overcome this issue we can construct ensembles of models and use those instead. Most of the time we will find out that the ensemble generalizes better than any of the models that make it up. This technique is also useful when the training set is very small and/or the learning algorithm becomes prone to over-fitting.

Having that in mind, we developed “DrCaptcha” a machine learning application that will serve as a Proof of Concept for the solutions we are proposing. “DrCaptcha” is an interactive machine learning (iML) application. The purpose of this application is to take advantage of the feedback provided by users and use it to optimize a machine learning model. The purpose of this model is to perform Optical Character Recognition (OCR) on handwritten characters. The latter has attracted the interest of many researchers [5, 6, 7, 8, 9, 13] with very good results. Our solution to the problem focuses primarily on the minimization of the data preparation phase. We then focus on the speed of the training phase along with the avoidance of overfitting. Our goal was to achieve state-of-the-art results with the above limitations.

“DrCaptcha“ initially provides users with an automated CAPTCHA system with the ability to distinguish whether the user is a human or a machine. The system then compares the values given by the user with the values stored in the database for the CAPTCHA, and if these values match then the system recognizes that the user is human and otherwise not.

The difference of this system from the classic CAPTCHA systems is that the system itself, for some of the images, does not know all the characters. If the system verifies that the user input matches the data in the database, it will take the user values for the non-classified digits, and classify them according to the values given by the user (see Fig. 1.). In case the user login is incorrect, the system will not classify the characters. In this way, the user provides implicit feedback to the system by classifying characters, which will be used later to train machine learning models. This functionality is available through a GUI and an API.

Based on the classified characters the application trains some artificial neural networks. After training and evaluating them, stores their training parameters, structure, and the confusion matrix. The latter will be used to synthesize ensembles of models that operate in the manner described later in this paper.

After the training of the neural networks, an administrator can create an ensemble of neural networks and assign weights to each of them. This ensemble acts as a meta-model that takes the prediction from the base classifiers and then decides which prediction to output. When a user creates or makes a change to this ensemble, the application evaluates it using a test set and saves its parameters again.

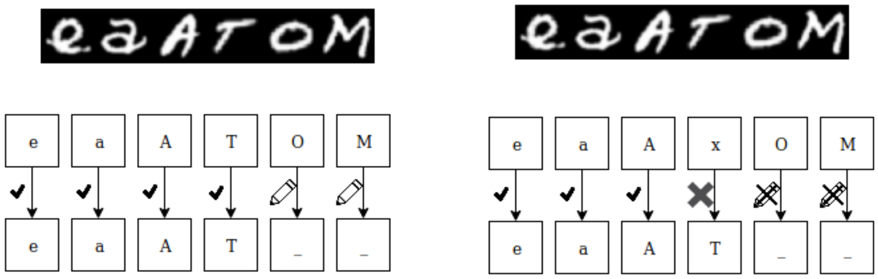


Fig. 1. How our CAPTCHA service works

Finally, the application provides a more interactive evaluation method where the user can "paint" a character and ask the system which character it is. The system will then retrieve the ensemble with the highest accuracy from the database, and give the user its prediction.

Also, if the base classifiers disagree and there are more than one dominant predictions, the system may come up with alternative results for an input. This can be extremely useful for an optical character recognition system. For example with the addition of some extra rules, even word corrections can be made. One possible solution that takes advantage of this feature is, for the system to be expanded to use a dictionary, and look at the different predictions given by the neural network ensemble, and comparing them with the dictionary, making the necessary corrections until the word matches with a word in the dictionary.

2. Related Work

2.1. Interactive Machine Learning

Interactive Machine Learning (iML) is a very promising tool for enhancing both human and computer capabilities. In the past, attempts have been made to create classifiers from humans manually using interactive machine learning systems. Those systems usually follow the same pattern - the training of a machine learning model using training data, and the evaluation of this model by a human. The feedback from this evaluation is considered for the creation of another model. This process repeats until the model performance is sufficient.

One interesting iML application is CueFlik [1]. CueFlik is an image web search application that allows users to quickly create their own rules for re-ranking images based on their visual characteristics. Users can then re-rank any future Web image search results according to their rule. This approach can extend the capabilities of existing computer vision methods to make a web search application more efficient.

ReGroup [2] (Rapid and Explicit Grouping) is a system that applies end-user interactive machine learning to help people create custom, on-demand groups in online social networks. It works by observing a person's normal interaction of adding members to a group while learning a probabilistic model of group membership which it uses to suggest both additional members and group characteristics for filtering a friend list.

Also, some attempts have been made to enable users to explicitly create classifiers. These methods are particularly helpful because an expert should not expect the automatic algorithm to discover something obvious. When such systems used by a domain expert, background knowledge is automatically exploited because the user is involved in almost every decision that leads to the induced models.

In [3], an interactive machine learning system was put in place which enabled its users to create a decision tree with a graphical environment by drawing a line between the points (corresponding to samples) that were graphically represented for a feature using in a two-dimensional visual interface. In [4] transparent boosting tree was proposed which visualizes both the model structure and prediction statistics of each step in the learning process of gradient boosting tree to user, and involves user's feedback operations to trees into the learning process such as add/remove a tree, select feature group for building a new tree, remove a node on the current tree and expand a leaf node. The system also allows the users to go back to any previous model in the learning loop.

2.2. Optical Character Recognition with Neural Networks

One of the most prevalent methods of machine learning is artificial neural networks. EMNIST [8] is the widely used benchmark for the hand-written recognition task. Multiple works [5, 6, 7, 8, 9, 13] have used machine learning models on the EMNIST dataset and have achieved very good results.

In [8], EMNIST's balanced dataset used as input for both a linear classifier and OPIUM-Based classifiers [10] with a different number of hidden neurons each. [6] proposed a deep neural network that is composed of two auto-encoder layers, with 300 and 50 neurons respectively and one softmax layer. [7] proposed a neuro-evolutionary algorithm that evolves simple and successful architectures built from embedding, 1D and 2D convolutional, max pooling and fully connected layers along with their hyper-parameters.

2.3. Ensemble Techniques

Such methods improve the predictive performance of a single model by training multiple models and combining their predictions. In the past, several attempts have been made in creating ensembles, with impressive results [11].

One way to increase the performance of a machine learning model is to learn multiple weak classifiers and boost their performance using a boosting algorithm. One disadvantage of those is that they require re-training based on the misclassified samples, and this may slow down the learning process. [12] proposed an aggregation technique that combines the output of multiple weak classifiers that do not require any re-training.

As a result, the training process is very fast while the model achieves state-of-the-art results. Another advantage of this framework is that it can combine classifiers that were created using different algorithms.

In recent years there have been a few attempts to combine the ensemble techniques and deep neural networks approaches [11]. While most of them center around developing an ensemble of deep individual neural networks, techniques like dropout split the initial neural network into several pieces at training time, to avoid over-fitting. Such techniques aim to reduce complex co-adaptations of neurons since a neuron cannot rely on the presence of particular other neurons in an individual neural network. Thus the neural network is forced to learn more robust features [16].

2.4. Transfer Learning

As stated earlier in this paper, the collection of data is complicated and expensive, making it extremely difficult to build a large-scale, high-quality annotated dataset due to the expense of data acquisition and labeling. Transfer learning [20] is the methodology of transferring knowledge from a source domain to a target domain. Usually, the size of the source domain is much larger. This process enables us to tackle problems with insufficient training data, which has a tremendously positive effect on many areas that are difficult to improve because of inadequate training data [21]. The main idea is that a deep neural network learning process is similar to the processing mechanism of the human brain, and it is an iterative and continuous abstraction process. The network's front-layers can be treated as a feature extractor and identify low-level features of training data. At the same time, subsequent layers can extract more high-level features that provide the network with the information needed to make the final decision.

By assuming that the neural network can learn low-level features from another dataset, we can take a neural network train it on a huge dataset, acquire a lot of low-level features, and retrain a part or all of it with a smaller dataset. This process will enable our model to take advantage of the low-level features learned from the first dataset and combine them with the high-level features learned from the second smaller one to make better predictions. For example in [22] the Inception-v3 [28] was used with weights trained on ImageNet [26] dataset and achieved 70.1% accuracy on CIFAR-10 dataset [23] and 96.5% on Caltech Faces dataset [24]. While transferring knowledge from a model trained on a labeled dataset is ideal, we can also transfer knowledge from models trained on unlabeled datasets. In [25], a new machine learning framework was develop called “self-taught learning” that uses models trained using unlabeled data for classification tasks. Another benefit that emerges is that the first half of the process - the training on the huge dataset - can be done once and use the low-level features learned on various more specialized datasets making the training cycle for these much smaller.

3. Ensemble Methodology

The method described in this section is an ensemble technique that uses artificial neural networks as base classifiers. Any type of neural network could be used. The only limitation is for the classifier to give the normalized probabilities for each category as a result. Although we are using neural networks, any type of classifier could be used as long as the above limitation applies.

This method combines a series of models of artificial neural networks (base classifiers), M_1, M_2, . . . , M_k, aiming at creating an improved model, M*. To understand this concept, let's say we have a patient and we want to make a diagnosis based on his symptoms. Rather than asking for a doctor's opinion, we can ask for an opinion from many. So if we see that the majority of doctors agree on a diagnosis, then we can choose it as the final diagnosis. The final diagnosis can be made by a majority, where the vote of each doctor has the same weight. The majority of voters of a large group of doctors may be more reliable than majority voting by a small group or only one. This is the case in most cases for an ensemble of classifiers.

The structure of artificial neural networks is not particularly relevant, as long as the last layer has a softmax activation so that the output layer gives us what normalized probability that the input sample can belong to. Let us call this probability $p_{i,j}$ where i denote the class, and with j the model from which it emerged. That is, the probability $p_{i,j}$ is the probability for class i by the artificial neural network j. We also can have k training sets, D_1, D_2,. . . , D_k, where D_i ($1 \leq i \leq k$) is used to create the M_i classifier.

Each of the models of artificial neural networks M_1, M_2,. . . , M_k has a weight w_1, w_2,. . . , w_k which is normalized so that the total weights for all artificial neural networks are 1. Continuing with the previous example, we assume that we place weights on the value of diagnosis for each doctor, based on the accuracy of previous diagnoses they have made or based on their specialty. The final diagnosis is then a combination of weighted diagnoses. In the case of artificial neural networks, we can compare education with the training parameters and the structure of the neural network. An examination of previous diagnoses can be likened to the examination of the confusion matrix.

Let us now pass a sample X as an input to all artificial neural networks. At the final layer of each model M_i, we will have the probability of each class for the input. Then we multiply all outputs of the artificial neural network M_i with the weight of the artificial neural network. The same action will be done for all neural networks of the ensemble, and at the end of the process, the probabilities of each model will be added by category, with the corresponding categories of all the models of the ensemble. The category with the largest sum of probabilities will be the output class.

Fig. 2. shows schematically how this method works. On the left side are the base classifiers. On each base classifier, each output neuron is associated with the corresponding neuron that handles the class. The above structure looks intuitive to an artificial neural network in which the output layer is not connected with all the neurons of the previous layer.

This process is mathematically described by Eq. 1. and Eq. 2. In those, with P_i we denote the sum of the probabilities for class i by all neural networks. With $p_{i,j}$ we denote the probability given by the network j the class, for the class i. With w_j we denote the weight of the neural network j, while the number of classes is c.

$$P_i = \sum_{j=1}^{k} p_{i,j} w_j \qquad (1)$$

$$class = argmax(P_{1,} P_{2,..}, P_c) \tag{2}$$

The advantage of this method is that it does not only take into account the only prediction made by neural networks. The result is deduced from the probabilities given by artificial neural networks for each class as output. Thus, the result is not only deducted as a weighted vote but also the confidence of the base classifiers in their prediction. Continuing our diagnosis example, now we take into account the confidence of each doctor along with the previous ones.

As for the choice of weights for each artificial neural network, this can be done automatically by considering the accuracy of the model in a validation set with Eq. 3 where w_i is the weight for neural network i, a_i is the accuracy of the neural network i, and k is the total number of neural networks in the ensemble.

$$w_i = \frac{a_i}{\sum_{j=1}^{k} a_j} \tag{3}$$

This setting can also be done manually by an administrator who can adjust weights by looking at the accuracy of each neural network and other statistics. Those statistics could be different metrics such as precision, sensitivity, specificity, and f-mesure or additional insights we can extract from the confusion matrix of the ensemble and each base classifier. The administrator could also consider the data gathered during each base classifier's training phase, such as the number of epochs, batch size, the increase or decrease of loss, and accuracy on each epoch indicating the level of over-training of each classifier.

Last but not least, the administrator should manually test the ensemble with their data (excluding train and test set), see how the ensemble and each base classifier behave and adjust the weights accordingly. Repeating this process multiple times ensures the classifier's quality and, with each iteration, enhances the ensemble more and more with human expertise.

The whole process can be summarized by the following Algorithm 1.

Algorithm 1. The proposed algorithm

```
1.  Training:
        1. Train all base classifiers (neural networks) with the training
    set(s).
        2. Test all base classifiers with a test set.
        3. Manually add weights to the base classifiers. (Add more base
    classifiers to the ensemble and adjust weights)

2.  Prediction:
        1. Make predictions with all the base classifiers and extract the
    probabilities for each one.
        2. Compute the overall probability for each class.
        3. The ensemble prediction is the class with the highest
    probability.
```

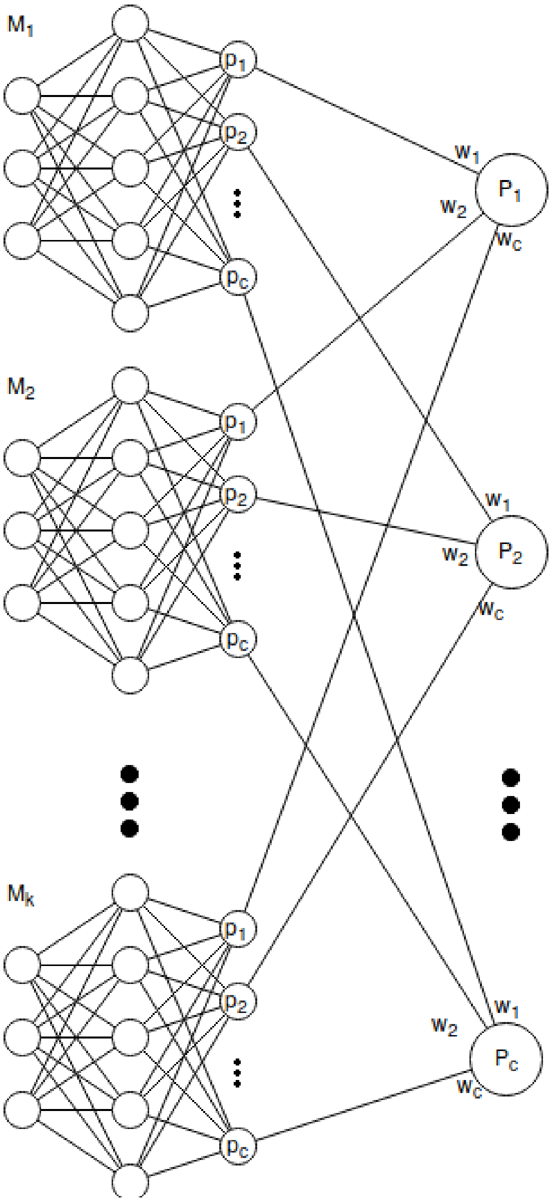


Fig. 2. Schematic representation for the proposed model

4. Context of Experimental Study

To test this methodology, we created two ensembles on two different datasets. The first one was capable of performing optical character recognition tasks, while the second one was a system that can detect pneumonia from chest x-rays. The base classifiers used in the first set of experiments were custom made neural networks (shallow, deep, and convolutional). However, on the second set of our experiments, we used existing state-of-the-art neural network architectures with some modifications to prepare them for the task. This way, we can confirm that our methodology works regardless of the architecture used.

4.1. Optical Character Recognition System

The OCR system predicts handwritten digits, uppercase, and lowercase characters. The dataset used for training was the EMNIST's balanced dataset which consists of 47 classes. For the creation of this system, we designed four neural networks which are described later in this section. Then we used these networks to assemble an ensemble.

The first artificial neural network to be implemented is a simple neural network with 10000 hidden neurons. The network takes an image (matrix) of 28x28 as input and converts it to a vector of size 784. Then it passes this image from a dense layer with 10000 neurons using ReLU [14] function as an activation function and then a 50% dropout layer so that the network is not over-trained. Finally, there is a dense layer with 47 neurons (as many as the categories of EMNIST's balanced). The activation function of the last layer is a softmax function. The categorical cross-entropy was used as a cost function with an adadelta [17] optimizer was used as a cost function. This neural network had about 85.05% accuracy.

The second artificial neural network is a deep neural network consisting of 8 layers. The network takes an image of 28x28 as an input and converts it to a vector of size 784. It then passes the image to a dense hidden layer of 4096 neurons with the ReLU activation function. It then goes through a 10% dropout layer. Then we still have a dense layer with 1024 neurons that also uses ReLU as an activation function. Then we have another 10% dropout layer. Finally, we have a dense layer with 512 neurons and a ReLU activation function followed by a 10% dropout layer. The last dense layer has 47 neurons. The activation function of the last layer is the softmax function. The categorical cross-entropy was used as a cost function with an adadelta optimizer. This neural network achieves an approximate accuracy of 85.07%.

The third artificial neural network is a deep convolutional neural network consisting of 8 layers. The first layer is a convolutional layer with 32 filters, a kernel size of 3x3 and the ReLU activation function. This layer takes as an input a 28x28 image and produces as output 32 images of 32x26, one for each filter. The next layer is also a convolutional layer the same as the previous one but this time with 64 filters that take the previous layer's images as input and produces 64 output images matrices of size 24x24. These "images" go to a max-pooling layer with a 2x2 window size that reduces the size of them to 12x12. The convolutional component of the network ends with a 50% dropout layer. Then we have a layer that undertakes to take the images of the convolutional neural network and converts them to a vector of size 9216. This vector enters a dense layer with 1024 neurons and a ReLU activation function followed by a dropout layer with a percentage of 25%. Finally, we have a dense layer with 47 neurons and a softmax activation function. The categorical cross-entropy was used as a cost function, with an adagrad [18] optimizer. This neural network achieved an average accuracy of 88.24%.

The fourth artificial neural network is a deep convolutional neural network consisting of 8 layers. The first layer is a convolutional layer with 32 filters, kernel size of 3x3 and ReLU activation function. This layer takes as an input a 28x28 image and produces as output 32 images of size 32x26, one for each filter. The next layer is also a convolutional layer identical to the previous one, which takes the previous layer's output as input and produces 32 24x24 "images". These images go to a max-pooling layer with a 2x2 window size that reduces the size of the images to 12x12. The convolutional part of the network ends with a 25% dropout layer. Then we have a layer that takes the output matrices of the convolutional network and converts them to a vector of size 4608. This vector enters as an input to a dense layer with 512 neurons and a ReLU activation function, followed by a dropout layer of 50%. Finally, we have a dense layer with 47 neurons and softmax as the activation function. The categorical cross-entropy was used as the cost function with an adagrad optimizer. This neural network achieved an approximate accuracy of 88.5%.

We choose these four neural networks to ensure the robustness of our ensemble technique by using different styles of neural network architectures. The first one was a primitive type of neural network with only one hidden layer, favoring simplicity over complexity. The second one was a deep neural network that had multiple hidden layers, generally capable to learn more difficult tasks than the first one. The other two were convolutional neural networks that generally train faster and perform better at image classification tasks such as optical character recognition.

Subsequently, we joined the networks designed above to create an ensemble. The weights assigned to those can be seen in table 1. These numbers were assigned after an intensive study of neural network performance on the test set. The overall accuracy of the ensemble in the EMNIST's balanced test set was about 88.89% which was almost 0,4 % more accurate than the most accurate neural network in the ensemble.

Table 1. Accuracy and weights of each neural network in the OCR ensemble.

Neural Network	Accuracy	Weights
1st	85.06 %	10.09 %
2nd	85.07%	10.09 %
3rd	88.24 %	34.78 %
4th	88.5%	43.48%

4.2. Pneumonia Detection System

To ensure our methodology's robustness, we performed another set of experiments on the ChestXRay2017 [27] dataset. Specifically, we created an ensemble of classifiers that is capable of detecting pneumonia from chest x-rays. To assemble that ensemble, we used four state-of-the-art neural networks, InceptionV3 [28], ResNet50 [29], Xception [30], and VGG16[31] with some modifications to prepare these neural networks for the task.

We removed the output layer of each. In its place, we added a global polling layer followed by a dense layer of 256 neurons, followed by a 50% dropout layer and an output layer with two neurons and a softmax activation function. These networks were pre-trained on ImageNet [26] dataset, and we retrained them on ChestXRay2017 using categorical cross-entropy as loss function and Nadam [32] as the optimizer. In table 2, you can see the accuracy and the weights assigned to these networks.

Table 2. Accuracy and weights for each base classifier in the pneumonia detection ensemble.

Neural Network	Accuracy	Weights
Inception-v3	94.39 %	40 %
ResNet50	92.63 %	20 %
Xception	93.43 %	20 %
VGG16	93.59 %	20 %

5. Experiments and Results

The dataset used for our first set of experiments was the balanced version of EMNIST's balanced dataset which consists of 47 classes of digits lowercase and uppercase letters. The official EMNIST's balanced dataset consist of 112,800 training samples, and 18,800 test samples. The samples for the two sets were taken by different groups of people. We trained our models on the official training set and tested it with the official test set.

For the second set of experiments, we used the ChestXRay2017 dataset, which contains 5856 X-Ray images 5232 for the training set and 624 for the test set divided into two classes, pneumonia and normal. 37.5% of the cases were chest x-rays of healthy people, while 62.5% were of patients with pneumonia. The dataset has an equal distribution of the two classes in training and test sets.

5.1. Training on EMNIST

We trained our models for twenty epochs and a batch size of 256 on a Quad-core Intel Core i5-4460. Table 3 shows the difference in training time between us and our main competitors, TextCaps [13]. In our environment, TextCaps training took 1 day, 20 hours and 12 minutes to run for the whole training set while the training time for our methodology was only about two hours and 50 minutes. That's about 15 times faster than our main competitors!

Table 3. Training performance comparisons between TextCaps [13] and DrCaptcha using the whole EMNIST's balanced training set

Method	Training time (minutes)
TextCaps [13]	2652 minutes
DrCaptcha	*172 minutes*

From our experiments, it can be seen that training time was significantly better for our methodology. That's very important for every machine learning methodology since training time is directly related to more expensive equipment and more expenses in general. Subsequently, methodologies with low training time and low hardware requirements are more likely to be used by non-tech users and small businesses on old hardware and mobile devices. Another advantage of our methodology and other ensemble methods is that we can add more models to our ensemble later to make our model even stronger. The latter can be done with only a few clicks in our application.

5.2. EMNIST Test Set Performance

Table 4 illustrates the accuracy achieved by different methods on the EMNIST's balanced dataset. It can be seen that we were able to surpass most of the existing methods including EDEN [7].

Table 4. Performance comparisons between methods for the EMNIST's balanced dataset

Method	Accuracy
OPIUM-Based classifiers [8]	78.02%
CSIM [5]	85.77%
Maximally Compact and Separated Features with Regular Polytope Networks [9]	88.39%
A mixture model for aggregation of multiple pre-trained weak classifiers [12]	88.39%
EDEN [7]	88.3 %
TextCaps [13]	90.46 %
DrCaptcha	*88.89 %*

Even though we were not able to outperform TextCaps in the accuracy performance, we are very close, and this performance gap will not make much difference in a real-life application. In addition we highlight that with our methodology if the prediction a model in the ensemble and another or more predictions are likely to be true then our model returns all of them. This smart heuristic will make a lot of difference in a real-life application, and in some situations could be even better than mathematically defined accuracy.

5.3. Training on ChestXRay17

We trained the four neural networks on an NVIDIA GeForce GTX 1050 (640 Cuda cores, compute capability 6.1). The training took place for a maximum of 40 epochs for all of them, but we stopped training if the loss function stopped getting better for at least five epochs. We also divided each epoch into 50 steps. Because the training set was small for the task, we also performed some random data augmentation for each image during training. Specifically, we performed rotations from 1° to 15°, zooms from 1% to 20%, and shifts within the range of 1-10% both vertically and horizontally. The average training time for each base classifier was 44 minutes.

5.4. ChestXRay17 Test Set Performance

The results of our ensemble were way better than each base classifier individually. Specifically, the ensemble achieved 95.51% accuracy, 97.43% precision, 95.47% sensitivity, 95.57 % specificity, and the f1 score was 96.44%. A summary of each base classifier's performance metrics can be found in table 5.

Table 5. Performance comparison between the ensemble and base classifiers on ChestXRay17.

Classifier	Accuracy	Precision	Sensitivity	Specificity	f-1
Inception V3	94.39 %	95.89 %	95.16 %	93.07 %	95.53 %
ResNet50	92.62 %	93.33 %	94.79 %	89.16 %	94.05 %
Xception	93.58 %	97.94 %	92.27 %	96.19 %	95.02 %
VGG16	93.42 %	93.33 %	96.04 %	89.38 %	94.66 %
Ensemble	*95.51 %*	*97.43 %*	*95.47 %*	*95.57 %*	*96.44 %*

Given the fact that the distribution of training (and test) data was highly imbalanced (62.5% - 37.5%), the best metric for measuring the ensemble's overall performance is the f-1 score which, as we can see, is better than each base classifier individually.

6. Conclusion

In this paper, we dealt with the problem of optical character recognition and its solution by implementing an interactive machine learning system. The accuracy of the system results is quite encouraging, as we managed to achieve up to 88.89% of the EMNIST’s balanced with a total of 47 classes representing lowercase, uppercase and numeric characters in only about two hours and 50 minutes. In addition, our methodology enables us to make proper use of the secondary predictions of our model. The latter could make a significant difference in a real-life application since it enables a user to extend this methodology to fill his needs in order to build a more complete system.

By using user-interactivity, we have solved the problem of labeling additional data as we can insert new images, and let application users label images. This is a great help as labeling many images would require many hours of extra work. With this approach we can provide more data during the training phase of our models, hoping to achieve even better results in the future.

The abundance of data is crucial in the implementation of a machine learning system. Interactivity satisfactorily solves the problem of labeling, but finding data, especially in the case of images, is not a particularly easy process. One way to find more data is to artificially expand the training set with data augmentation techniques, as we did in our second set of experiments. Data augmentation makes the dataset expansion a painless process that can further enhance our models, which we can train with more data to tackle the problem of over-training more effectively. We can also train our models with a large dataset from an entirely different domain and use transfer learning techniques to retrain our models with a smaller dataset and transfer the knowledge acquired from the first dataset to the field that we are interested in.

In addition to expanding the training set, interactivity also serves to synthesize all artificial neural networks. A user can look at the statistics (such as the confusion matrix) of various neural networks, and use them to synthesize a set based on them and assign them the appropriate weights to achieve the desired performance.

Although these ensembles achieve high performance, they add extra complexity to the system. Also, the system gets slower as new models are added to the set. Many researchers today are looking for ways to simplify such systems by exporting the functionality and performance of a set of machine learning models to a simpler model that ideally should have the same behavior and performance as the ensemble.

References

1. Fogarty, James & S. Tan, Desney & Kapoor, Ashish & Winder, Simon. (2008). CueFlik: Interactive concept learning in image search. Conference on Human Factors in Computing Systems - Proceedings. 29-38. 10.1145/1357054.1357061.
2. Amershi, Saleema & Fogarty, James & Weld, Daniel. (2012). ReGroup: Interactive Machine Learning for On-Demand Group Creation. Conference on Human Factors in Computing Systems - Proceedings. 10.1145/2207676.2207680.
3. Ware, Malcolm & Frank, Eibe & Holmes, Geoffrey & Hall, Mark & Witten, Ian. (2001). Interactive Machine Learning: Letting Users Build Classifiers. International Journal of Human-Computer Studies. 55. 281-292. 10.1006/ijhc.2001.0499.
4. Lee, Teng & Johnson, James & Cheng, Steve. (2016). An Interactive Machine Learning Framework. CoRR abs/1610.05463
5. Gao, Yang & Chandra, Swarup & Wang, Zhuoyi & Khan, Latifur. (2018). Adaptive Image Stream Classification via Convolutional Neural Network with Intrinsic Similarity Metrics.
6. Gunawan, Teddy & Noor, A.F.R.M. & Kartiwi, Mira. (2018). Development of english handwritten recognition using deep neural network. Indonesian Journal of Electrical Engineering and Computer Science. 10. 562-568. 10.11591/ijeecs.v10.i2.pp562-568.
7. Dufourq, Emmanuel & Bassett, Bruce. (2017). EDEN: Evolutionary Deep Networks for Efficient Machine Learning. CoRR abs/1709.09161
8. Cohen, Gregory & Afshar, Saeed & Tapson, Jonathan & van Schaik, André. (2017). EMNIST: an extension of MNIST to handwritten letters. CoRR abs/1702.05373
9. Pernici, Federico and Bruni, Matteo and Baecchi, Claudio and Del Bimbo, Alberto (2019), Maximally Compact and Separated Features with Regular Polytope Networks In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop
10. van Schaik, André & Tapson, Jonathan. (2014). Online and Adaptive Pseudoinverse Solutions for ELM Weights. Neurocomputing. 149. 10.1016/j.neucom.2014.01.071.
11. Sagi, Omer & Rokach, Lior. (2018). Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 8. e1249. 10.1002/widm.1249.
12. Chakraborty, Rudrasis & Yang, Chun-Hao & Vemuri, Baba. (2018). A Mixture Model for Aggregation of Multiple Pre-Trained Weak Classifiers. 454-4547. 10.1109/CVPRW.2018.00074.

13. Jayasundara, Vinoj & Jayasekara, Sandaru & Jayasekara, Nipuni Hirunima & Rajasegaran, Jathushan & Seneviratne, Suranga & Rodrigo, Ranga. (2019). TextCaps: Handwritten Character Recognition With Very Small Datasets. 254-262. 10.1109/WACV.2019.00033.
14. Nair, Vinod & E. Hinton, Geoffrey. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair. Proceedings of ICML. 27. 807-814.
15. https://gitlab.com/rafaelglikis/drcaptcha
16. Alex, K & Ilya, S & Hg, E. (2012). Imagenet classification with deep convolutional neural networks. Proceedings of NIPS, IEEE, Neural Information Processing System Foundation. 1097-1105.
17. D. Zeiler, Matthew. (2012). ADADELTA: An adaptive learning rate method CoRR abs/1212.5701.
18. C. Duchi, John & Hazan, Elad & Singer, Yoram. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research. 12. 2121-2159.
19. Amershi, Saleema. (2011). Designing for effective end-user interaction with machine learning. UIST'11 Adjunct - Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. 47-50. 10.1145/2046396.2046416.
20. Torrey, L., & Shavlik, J. (2010). Transfer Learning. In Olivas, E. S., Guerrero, J. D., Martinez-Sober, M., Magdalena-Benedito, J. R., & Serrano López, A. J. (Ed.), Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques (pp. 242-264). IGI Global. http://doi:10.4018/978-1-60566-766-9.ch011
21. Tan, Chuanqi & Sun, Fuchun & Kong, Tao & Zhang, Wenchang & Yang, Chao & Liu, Chunfang. (2018). A Survey on Deep Transfer Learning.
22. Mahbub Hussain and Jordan J. Bird and Diego R. Faria (2018). A Study on CNN Transfer Learning for Image Classification. In Advances in Computational Intelligence Systems - Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK (pp. 191–202). Springer.
23. Krizhevsky, Alex. (2012). Learning Multiple Layers of Features from Tiny Images, Chapter 3 (pp. 32–36). University of Toronto.
24. http://www.vision.caltech.edu/Image_Datasets/Caltech_10K_WebFaces/#Description
25. Raina, Rajat & Battle, Alexis & Lee, Honglak & Packer, Ben & Ng, Andrew. (2007). Self-taught learning: Transfer learning from unlabeled data. Proceedings of the Twenty-fourth International Conference on Machine Learning. 227. 10.1145/1273496.1273592.
26. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255).
27. Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, V2, doi: 10.17632/rscbjbr9sj.2
28. Szegedy, Christian & Vanhoucke, Vincent & Ioffe, Sergey & Shlens, Jon & Wojna, ZB. (2016). Rethinking the Inception Architecture for Computer Vision. 10.1109/CVPR.2016.308.
29. He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90.
30. Chollet, Francois. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. 1800-1807. 10.1109/CVPR.2017.195.
31. Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.
32. Dozat, T. (2016). Incorporating Nesterov Momentum into Adam.
33. https://github.com/rafaelglikis/pneumonia-detector

Rafael Glikis graduated with an Integrated Master from the Department of Computer Engineering and Informatics of the University of Patras specializing in Software Engineering and Artificial Intelligence. During his studies he co-founded Grayhat Infosec Solutions, a startup company that provides software solutions to cybersecurity professionals. Since his graduation in 2019 he is a Software Engineer at Learnworlds, a platform for creating, selling and promoting online courses. His academic interests include Artificial Intelligence, Data Science, Web Development and Cyber Security.

Christos Makris is an Associate Professor in the University of Patras, Department of Computer Engineering and Informatics, from September 2017. Since 2004 he served as an Assistant Professor in CEID, UoP, tenured in that position from 2008. His research interests include Data Structures, Information Retrieval, Data Mining, String Processing Algorithms, Computational Geometry, Internet Technologies, Bioinformatics, and Multimedia Databases. He has published over 100 papers in refereed scientific journals and conferences and has more than 600 citations excluding self-citations (h-index: 18).

Nikos Tsirakis is an Computer and Informatics Engineer and received his Ph.D. degree in Data Mining from the University of Patras, in 2010. His research interests include Information Retrieval, Data Mining, String Algorithmics, Social Network Analysis, Software Quality Assessment & Web Technologies. He has published over 30 papers in refereed scientific journals and conferences and has more than 300 citations.

Received: January 30, 2020; Accepted: December 01, 2020 https://doi.org/10.2298/CSIS200415050F

A Novel Information Diffusion Model Based on Psychosocial Factors with Automatic Parameter Learning

Sabina-Adriana Floria and Florin Leon

Faculty of Automatic Control and Computer Engineering,
“Gheorghe Asachi” TechnicalUniversity of Iași,
Bd. D. Mangeron 27, 700050, Iași, Romania
{sabina.floria, florin.leon}@ tuiasi.ro

Abstract. Online social networks are the main choice of people to maintain their social relationships and share information or opinions. Estimating the actions of a user is not trivial because an individual can act spontaneously or be influenced by external factors. In this paper we propose a novel model for imitating the evolution of the information diffusion in a network as well as possible. Each individual is modeled as a node with two factors (psychological and sociological) that control its probabilistic transmission of information. The psychological factor refers to the node’s preference for the topic discussed, i.e. the information diffused. The sociological factor takes into account the influence of the neighbors’ activity on the node, i.e. the gregarious behavior. Agenetic algorithm is used to automatically tune the parameters of the model in order to fit the evolution of information diffusion observed in two real-world datasets with three topics. The reproduced diffusions show that the proposed model imitates the real diffusions very well.

Keywords: social networks, information diffusion, psychological factors, sociological factors, genetic algorithm.

1. Introduction

The use of social networking websites is currently the most widely used form of communication. Social networks help us to keep in touch with our friends, create new relationships, develop our social life, but these can also influence our decisions especially when a lot of information is false or we may even become dependent through their excessive use. Socialization on social networks has taken on a great extent and the number of users has increased considerably. When analyzing social networks, we focus on discovering patterns of human interactions. Thus, we can observe the social structures, and the actions and friends of an individual are no longer random, but can be modeled according to well-defined rules. If we were to analyze how people interact, we might find that they do not make random connections with one another (e.g. some talk constantly, others often, and some never do). The use of social networks allows users to send and receive messages (both public and private), share photos, videos and gives them the opportunity to join certain groups. We can say that online communication means have become employed in all aspects of everyday life, from business to social life. Thus, online social networks not only connect many users, but also collect data on their daily interactions. Given these collections, we can analyze how information is transmitted through social networks, which is a topic of great interest.

In this paper we propose a model that imitates the information diffusion as well as possible, referring to the behavior of users from a real point of view, both at the psychological and social level. In modeling the individual from a psychological point of view, his/her decision in the transmission of information is a restricted one. This restriction consists in the fact that the individual has a certain interest in the information, but this information may also be useful or not for him/her. In modeling the individual from a sociological point of view, we consider the gregarious behavior of the individual regarding the transmission of information, namely that he/she is influenced by the activity of his/her neighbors. For both models, we take into account both the information credibility, and the fact that users are bored with certain information, either according to the passage of time or depending on the amount of information received. We also use a genetic algorithm to automatically learn the model parameters based on the real information diffusion. This model is applied on two real datasets [1, 2], and the results are promising.

In the literature there are many factors considered to influence the information diffusion, but it is important how these factors are modeled and combined. In our model we introduce the following influencing factors: the boredom of an individual, the activation degree of an individual that is correlated with his/her interest in information and the usefulness of the information, the login rate depending on a certain time of day (daytime/nighttime) etc. The main contribution of this paper is the combination of these influencing factors in a model in order to reproduce the decision of an individual in an agent-based simulation as well as possible. Another contribution is the automatic learning of certain parameters of the diffusion model. This is a great advantage because the evolutions of the diffusions may be different depending on a certain topic being discussed. Moreover, by combining psychosocial modeling with automatic learning of parameters, our model is able to extract the relevant characteristics from various evolutions of diffusion. To the best of our knowledge, no other models have been proposed so far in the literature that combine these influencing factors and automatically adapt their parameters for different diffusion evolutions.

The paper is organized as follows. In Section 2 we present some contributions related to information transmission models. In Section 3 we describe our model, and the experimental results are shown in Section 4. Finally, in Section 5 the conclusions and some development directions are included.

2. Related Work

An analysis of how information is spread through online social networks (OSNs) and simulation of user behavior based on their posts is presented in [3]. The method proposed in this paper uses a stochastic multi-agent approach, in which each agent is in fact a user of the social network. The analysis is made on Barack Obama’s Twitter network in the 2012 US presidential race. The authors show what happens if the central source of the network is inactive, more precisely the node that represents Barack Obama and also highlights the impact of eliminating the most active users in the process of information diffusion. By impact it is understood that the number of messages sent by users is constantly affected over time. Experimental results show that eliminating the first 100 most active users has a greater impact on the number of messages than removing the central source node.

In [4], the authors model the information diffusion using agents with well-defined states, similar to the epidemic SIR (Susceptible, Infected, Recovered) model and use two datasets from the Twitter social network to compare the efficiency of the proposed model regarding the realistic simulation of the diffusion. The model introduced in this paper is based on the fact that those users who may know that a rumor is false, will not spread messages that deny these rumors. Therefore, recovered users will not influence their neighbors, allowing them to recover as well. They use a synthetic scale-free network of 1000 nodes and the Euclidean distance to evaluate the difference between the actual and the simulated diffusion results. The authors compare the Euclidean distance of their model with a basic model and obtain a smaller distance for both datasets, so a more realistic information diffusion.

A basic model for rumor propagation is proposed in [5] and consists of node-level modeling. Nodes can have well-defined states, each state allowing specific actions such as spreading the information, ignoring it etc. As in our work, node activity is modeled in discrete time events, which is why the authors can model various time constraints, such as: some actions of the nodes are completed after a certain period of time, a node checks its information from friends at least once 24 hours and at most once an hour. The proposed model contains a large number of parameters, this being a general impediment in the complex models, hence our motivation to use a genetic algorithm for automatic parameter learning. The authors also use synthetic networks and conclude that those networks with scale-free topology are more suitable for analyzing the simulation of information diffusion.

In [6], a multi-agent model is proposed to reproduce the real transmission of information in scale-free networks. In addition, the authors also propose a mechanism to combat the spread of false information. Each agent has the opportunity to choose whether or not to transmit the information depending on his preparation level on a particular topic, which is a random threshold assigned to him. The authors propose three different ways in which an agent can spread information: spontaneous visualization, collective influence and communication persuasion. An analysis of the real information diffusion is made on a Twitter dataset with the announcement of the discovery of the Higgs Boson [1] in which the authors track the activity of the active users in the network to highlight the evolution of information diffusion. We also follow this aspect in our paper on the same dataset. Running the model on networks of different sizes, the authors observe the same form of diffusion and assume that their model does not depend on the size of the network, but only on the simulation parameters. Also, to study the spread of false information, the authors use the real dataset where fake news was spread during the Occupy Wall Street protest. In order to model the spread of fake news, the authors introduce in their model a new type of agent that is able to recognize fake news and alert its neighbors. In this experiment, based on the number of posts of users over time, they obtain a good dynamic of the event on networks of different sizes.

A spatio-temporal characterization of the information diffusion process and a model that describes the dynamics of information spreading on the Higgs dataset [1] are presented in [7]. Regarding the spatial and temporal characteristics of the observed data, the authors studied the behavior of the user both at the global (macroscopic) and individual (microscopic) levels. The users' activities are: posting message (tweet), sharing post (re-tweet) or replying to existing tweets. Starting from the observed characteristics, the proposed diffusion model takes into account the fact that a user no longer posts a certain period of time after having a recent post. Also, the authors introduce two different rates of activation or deactivation of nodes that are time-varying and can be independently modified. The probability that a node will post a message is also influenced by the number of its neighbors who repeatedly post over time. Their model has a good accuracy in reproducing the information diffusion and could be applied in other processes of diffusion of social networks.

A protocol in which the network becomes more immune to the spread of false information based on the evidence theory (Dempster-Shafer theory and Yager's rule) is presented in [8]. Their model is based on the choice of two source nodes, one that transmits true information and one that transmits false information. The effects of the collision of the two pieces of information through the network are shown, but also the effect of using the evidence after establishing the ground truth. This approach based on the evidence theory plays an important role in the individual's decision to transmit or not the received information. The authors also consider the confidence degree of the neighbors regarding the character of the information spread by a certain source. Once the ground truth is established, the authors show how the spread of false information is blocked. Also, the work [9] is an extended version of the work [8] in which the following case studies are considered: different positioning of the source nodes, a source node might not always transmit the same information during a simulation, use of a larger network, adding new connections to the original networks and analyzing the number of messages during information diffusion.

There are many other approaches that analyze the process of information diffusion through the network. For example, a dynamic model is proposed in [10] to investigate the influence of node activity on the information spread process. Through an active node, the authors refer to the fact that it can contact all its neighbors, while an inactive node can only communicate with its active neighbors. The behavior of the model is studied on both homogeneous and heterogeneous networks. In [11], the authors study the dynamics of the information diffusion on homogeneous social networks in which they consider a mechanism to combat false information. A stochastic model for information diffusion is proposed in [12] and the authors mention the limited attention property of the users, in which they may lose some of the received messages if they have many connections. In [13], an extension of the Susceptible-Infected diffusion model is proposed, in which the authors include elements of human dynamics, such as bursty and limited attention, with a significant impact on the diffusion process. In [14] a competitive model of information diffusion is presented, which consists in the simultaneous spread of two different pieces of information. A diffusion model called GT is presented in [15], in which the nodes are considered intelligent and rational agents and have two types of payoff: a social and an individual one. Also, the proposed model can be used to predict what behavior the users will have in a certain time frame.

Other models of information diffusion are also presented in surveys on this topic: [16-18].

3. Model Description

In this paper we propose a protocol of information diffusion in social networks in which we take into account as many realistic factors as possible in order to model the individual's decision to transmit information or not. We consider both the personality of the individual from a psychological point of view, as well as his degree of sociability. We model user behavior using two categories of influences: internal and external. These categories are presented by [19] in a detailed analysis of consumer behavior. Some examples of internal influences of a consumer's behavior presented by the authors are perception, motivation, learning, memory, attitude, and the main external influences are those of groups or different factors of a society (e.g. demographic or cultural factors). We refer to these internal factors as psychological factors of an individual, while external factors are correlated with sociological factors. These terms can be jointly referred to as "psychosocial" factors. In psychological modeling we chose the perception and motivation of an individual as internal factors: perception is modeled as the usefulness of information, while motivation is modeled as a combination of the individual's interest and the usefulness of information. In sociological modeling, we chose the external factors related to the influence of the neighbors on an individual.

In addition, we propose that an individual may be influenced by the information credibility when making a decision in its transmission. The credibility of the information or the credibility of the source of information is difficult to assess. In an online social environment, a user usually assesses credibility based on certain indicators provided by the social platform. For example, in [20] the authors analyze the relevance of certain indicators on Twitter based on which users try to assess the credibility of posts. The most important indicators are those that refer to an official source, or posts that contain links, facts, informative or professional messages. To analyze these indicators, the authors chose different evaluators to judge the credibility of the posts and used the majority vote for the final evaluations. The aforementioned credibility indicators cannot be used in our model because we do not make an analysis based on the content of the messages. However, in [20] the authors show that the number of posts is also an indicator of credibility. Therefore, in our model we propose to use the number of the neighbors' messages as an indicator of information credibility. We propose that the information held by a node has an initial credibility, which is a value in the range (0, 1]. In our model there are two ways in which a node can have information: it is assigned to it by simulation (source node or informant node) or it can receive it from neighbors (special node). The mechanism for determining credibility applies only to special nodes. Majority voting is an intuitive strategy to model an individual's decision when multiple options are available. For example, [20] and [21] use majority voting to assess the final credibility of posts, and [22] uses majority voting as a mechanism for modeling the gregarious behavior of a node. The majority vote cannot be applied in our model because the credibility of the information is not a categorical variable, but a real one. Therefore, as an alternative, we propose that a node weigh the credibility of information received from neighbors. Moreover, in this model we assume that the credibility of the information increases as it is discussed more. To achieve this growth, we increase the weight of that credibility received from the neighbor with the highest number of messages.

Users cannot always send information, but only during certain periods. We start from the premise that they log on to a social platform with a certain average login rate.

Most of the parameters of the proposed model are learned using a genetic algorithm. Therefore, having a diffusion model and the automatic learning of the parameters, our objective is to obtain an accurate evolution of the information diffusion, comparing it with the real diffusion.

The first dataset we use contains both the structure of a social network on Twitter and the activity of users during the announcement for the discovery of the Higgs Boson. The second dataset does not contain the structure of the network, but only the activity of users in the Twitter social network on certain topics, of which we have chosen two, namely: “lipstick on a pig” and “fundamentals of our economy are strong”. We chose datasets that contain the timestamps of communication between users. Based on these timestamps, we managed to extract the evolution of information diffusion. Regarding the second dataset, the subjects were chosen at random.

3.1. User Login Rate and User Handling

When a person initiates an activity on a social network, we say that by this action he or she logs in. For this purpose, we choose that users have an average login rate (λ_{login}) following a negative exponential distribution law according to equation (1), where u represents a randomly generated number in the range [0, 1) and t_{login} is the time period between two successive logins of a user expressed in minutes:

$$t_{login} = -\frac{\ln(1-u)}{\lambda_{login}} \tag{1}$$

For example, if an individual logs in every two hours, then the average login rate is $\lambda_{login} = \frac{1}{2[h]} = \frac{1}{2 \cdot 60[m]} = 0.0083$ logs per minute. For this login rate, one can see in Fig. 1 on the Y axis that t_{login} is generated in an interval of approximately [0,600] minutes. We can also see that we have a higher chance of generating short duration times and we mark with the dotted line the duration of 200 minutes, i.e. in about 80% of cases we will have small values.

We consider that the nodes are handled in the order of the login duration. Thus, after the login duration of a node has been generated, it is added to a sorted list. We chose to increment the simulation clock in discrete steps, where each step represents a period of one minute. After each increment of the simulation clock, the list is checked to identify which nodes are able to log in, i.e. a minute has passed and some nodes may be able to log in. A node that is able to log in is extracted from the list and then it is checked whether it can transmit its information to its neighbors according to the two models (from a psychological and sociological point of view). Subsequently, a new time period is generated for this node and it is added back to the list, such way that the list remains sorted (i.e. the node that has the smallest login period is placed first in the list).

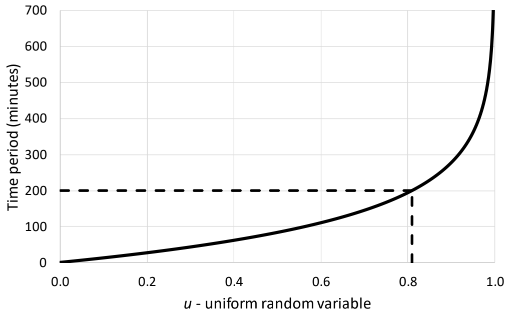


Fig. 1. Generation of time between two successive logins according to the uniformly distributed random variable *u*

Before describing how we handle nodes, we specify that in our model we have three types of nodes, namely: source node, special node and informant node. A source node holds the information and its transmission is based only on the basic probability attached to the node. Also, the transmission of a source node is not influenced by the information credibility or by the psychosocial modeling. A source node will change its type into a special node when at least one piece of information is received from one of its neighbors. If the source node does not receive any information from its neighbors, it automatically becomes a special node after a period of time to avoid the continuous transmission of messages. The second type of node, the special node, has a transmission probability that is determined according to the two models and it is also influenced by the information credibility. The last type of node, the informant node, has 100% probability of transmission and 100% credibility for information. We propose this type of node for the moment when we want to suddenly encourage socialization between the nodes, i.e. several nodes adopt the information when it is transmitted very often. Informant type nodes spread information for a certain period of time ($T_{max\_informant}$), after which they become special nodes.

3.2. Transmission of Information

In the initial phase only the source nodes hold the information along with the credibility attached to it. After a node has logged in, its decision to transmit or not the information is a probability determined according to the type of node:

$$Prob_{send} = \begin{cases} P_b \text{ , } \mathit{source\ node} \\ P_f \text{ , } \mathit{special\ node} \\ 1 \text{ , } \mathit{informant\ node} \end{cases} \qquad (2)$$

where P_b is the basic probability (the same for all nodes), and P_f is the final probability determined according to the psychosocial modeling. Depending on this probability, if the node has the chance to transmit the information, it will spread the information to all its neighbors along with the credibility attached. A node can transmit information multiple times due to repeated logins and also a node counts the information received from each of its neighbors separately. The special node has a particular behavior when it spreads information. For this type of node, we determine a final probabilityP_f, which is based on both the psychological and sociological modeling and also on the information credibility:

$$P_f = (P_P \cdot W_p + P_S \cdot W_S) \cdot InfoCred(n_t) \tag{3}$$

where $P_p$is the probability from the psychological modeling, $P_s$is the probability from the sociological modeling, $W_p$and $W_s$are weights that control the impact of P_p and P_s, and *InfoCred*(n_t) is the credibility that the node n_t has on the information. *InfoCred*(n_t) is computed based on all information received from the neighbors of the n_t node.

3.3. Determining the Credibility that a Node Has on the Information

In our model, a node stores statistics for each neighbor. Thus, when a node receives information from a neighbor, it stores the information credibility and also the number of messages received. The node has these two fields separately for each neighbor. We choose that the information from certain neighbors should be more important or less important depending on the number of messages received. To determine the information credibility of a transmitter node (n_t), we weight each credibility received from neighbors according to equation (4), where v represents the number of neighbors of node n, and *InfoCred*(i) is the credibility received from neighbor i:

$$InfoCred(n_t) = \sum_{i=1}^{v} W_i \cdot InfoCred(i) \tag{4}$$

The weight W_i associated with the neighbor i is computed as the ratio between the number of information received from the neighbor$i$ and the total number of the information pieces received from all the neighbors:

$$W_i = \frac{Info(v_i)}{Info_{total}} \tag{5}$$

In order to implement a mechanism for increasing the credibility of information, we choose that the maximum weight should be encouraged by a percentage increase. In other words, we take into account to a greater extent the credibility of that neighbor who transmitted the highest number of messages:

$$W^*_{\max} = W_{\max} + W_{\max} \cdot W_{cred} \tag{6}$$

$$W_{\max} = \max(W_i), i = 1,...,v \tag{7}$$

where W^*_{max} is the maximum adjusted weight, and W_{cred} represents a parameter that controls the increase in credibility (the same for all nodes). The value of the control parameter W_{cred} is learned using the genetic algorithm.

This process can indeed be manipulated by artificially developing a high number of neighbors or another approach by spamming messages [20]. We do not use a mechanism for detecting and correcting such manipulations, but we propose a simple mechanism for combining the information that a user has from friends.

In Fig. 2 we show an example on a small network in which we determine the credibility that node 3 has for the information.

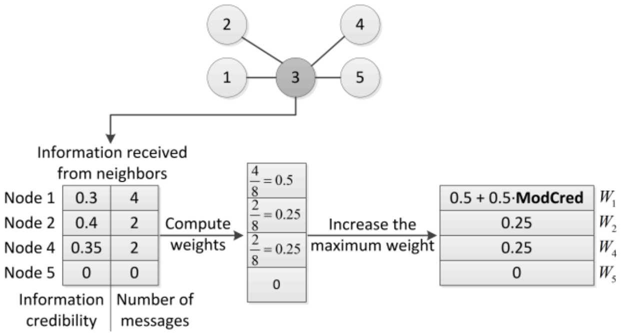


Fig. 2. Example for computing the information credibility of a node

3.4. Modeling the Individual from the Psychological Point of View

Regarding the modeling of the individual from the psychological point of view, we propose that his/her decision to spread or not the information (probability P_p) should be influenced by the individual's activation degree and an attenuation factor. We also consider the basic probability of the node (P_b) and a weight for the activation degree (W_{act}):

$$P_p = Activation \cdot W_{act} \cdot P_b \cdot \text{Attenuation}_{\mathrm{P_p}} \qquad \textbf{(8)}$$

We propose that the node's activation degree should be correlated both with the node's interest for information and with the level of usefulness of the information. Therefore, we define the following measures: $N_{interest}$ is the level of interest that the node has for the information and $N_{usefulness}$is the level of usefulness of the information. $N_{interest}$ and N_{util} have the same definition domain: integers in the range [1, 10]. Before starting the simulation, we initialize $N_{interest}$ from each node with a random value in the range [1, 10], thus each node has its own interest for the information. $N_{usefulness}$ is attached to the information that is spreading and this measure does not differ from one node to another. $N_{usefulness}$ is also initialized with a random value in the range [1,10]. Our model is capable of spreading a single type of information during the simulation, thus the value of $N_{usefulness}$ remains constant. $N_{interest}$ and $N_{usefulness}$ are used to determine the node's activation degree and this step is done before the node transmits the information:

$$Activation = \frac{N_{interest} \cdot N_{usefulness}}{Max_{interest} \cdot Max_{usefulness}} = \frac{N_{interest} \cdot N_{usefulness}}{10 \cdot 10} \qquad \textbf{(9)}$$

where $Max_{interest}$ and $Max_{usefulness}$ are the maximum limits for $N_{interest}$ and $N_{usefulness}$. The node's activation degree is directly proportional to both $N_{interest}$and $N_{usefulness}$. Basically, the activation degree increases in greater proportion as both levels ($N_{interest}$ and $N_{usefulness}$) increase (Fig. 3).

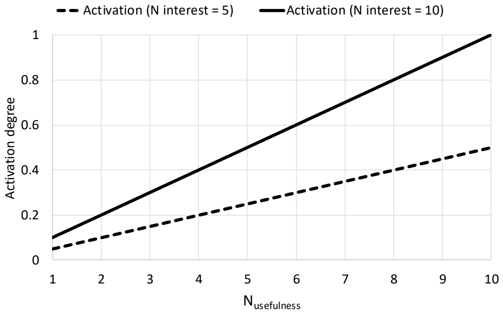


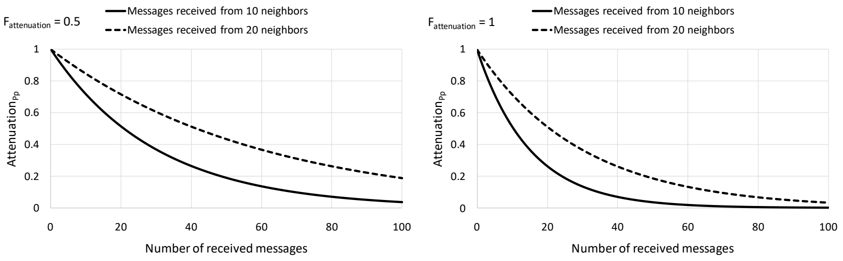
Fig. 3. Example of activation degree for $N_{interest}$=5 and $N_{interest}$ =10

Using this approach, we want to model various boundary cases. For example, when the interest of a node is very high, but the usefulness of the information is low, we can say that the activation degree has a small value. This behavior is due to the fact that although the node is very interested in information, the reduced utility of the information does not satisfy the node. The behavior is the same if $N_{interest}$ is small and the $N_{usefulness}$ is very large: the information is satisfactory, but the individual has no interest in it. We also attach a weight (W_{act}) for the activation degree in order to control the impact it has on the sending probability of the psychological model (P_p). Depending on the diffusion evolution, the value of the W_{act} weight is learned by the genetic algorithm in the range [0.2, 1].

As a topic is discussed more often, the number of messages received by the node increases. We propose that the node should get bored of the topic discussed as the number of messages received becomes larger. In other words, we introduce an attenuation (10) that depends on the total number of messages received by a node (Msg_{total}) from all its neighbors and by an attenuation factor ($F_{attenuation}$) learned by the genetic algorithm.

$$Attenuation_{P_p} = e^{-\frac{F_{attenuation} \cdot Msg_{total}}{v}} \qquad \textbf{(10)}$$

We divide the exponent into the total number of neighbors (v) because we would obtain different results for networks of different sizes. The total number of messages of a node in a small network is smaller than in a large network (where the nodes have more neighbors).



a) b)

Fig. 4. Example of attenuation for a) $F_{attenuation}$ = 0.5; b) $F_{attenuation}$ = 1

In Fig. 4 we show the attenuation evolution according to the number of messages received from a certain number of neighbors for two cases: $F_{attenuation}$ = 0.5 and $F_{attenuation}$ = 1. In the case of a node with 10 neighbors and 40 received messages, one can see that the attenuation $Attenuation_{Pp}$ changes as follows: when $F_{attenuation}$ is 0.5, $Attenuation_{P_p} \cong 0.25$ (Fig. 4.a), and when $F_{attenuation}$ is 1, $Attenuation_{P_p} \cong 0.07$ (Fig. 4.b).

3.5. Modeling the Individual from the Social Point of View

In this type of modeling we focus on the percentage of active neighbors of the node because we want the probability provided by this modeling (P_s) to depend only on the activity of its neighbors and not on the amount of information received by the node. To model this probability, we start from a sigmoid function (P_{social_infl}) to which we include an attenuation ($Attenuation_{Ps}$):

$$P_s = P_{social\_\inf l} \cdot Attenuation_{P_s} \tag{11}$$

$$P_{social\_\inf l} = \frac{1}{1 + e^{-(\theta + Pct_{neighbors} \cdot F_{social\_\inf l})}} \tag{12}$$

We choose the sigmoid function (12) because, considering its shape, we want $P_{social_infl}$to have a lower value in the beginning, when the number of active neighbors is relatively small. Then, as users discuss more, we want $P_{social_infl}$to have a sudden increase at one point, thus modeling the social behavior of a node. In order for the evolution of the sigmoid function to start from the origin on the X axis and not from the negative domain, we introduce a parameter θ with the value –5. Also, we introduce in P_{social_infl}a social influence factor (F_{social_infl}) in order to control the shape of the curve. We show in Fig. 5 the impact of $F_{social_infl}$for two different values, 5 and 10. These values are actually the limits of this parameter.

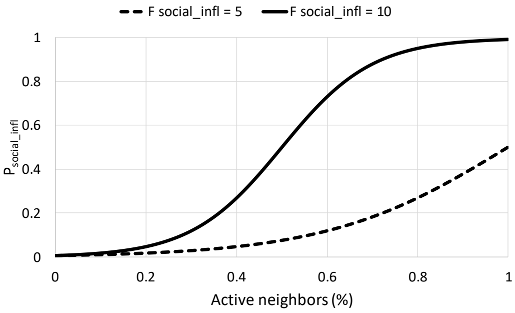


Fig. 5. Evolution of $P_{social_infl}$for two values of F_{social_infl}: 5 and 10

We choose the value 5 for the lower limit of F_{social_infl} because in this case the sigmoid function shape is incomplete and one can see that P_{social_infl}reaches the maximum value of 0.5. This is the situation in which the individual is weakly influenced by the percentage of his active neighbors. We choose the value 10 for the upper limit of F_{social_infl}because we want a complete sigmoid shape. Therefore, P_{social_infl}tends to value 1, modeling the fact that an individual is more influenced by the activity of his neighbors. The F_{social_infl}parameter, which represents the social influence of the node, is not learned by the genetic algorithm, but is randomly generated in the range [5, 10] for each node at the beginning of the simulation. As in the case of psychological modeling, we introduce an attenuation ($Attenuation_{Ps}$) in the probability P_s (11) to simulate the fact that a node gets bored with the activities of its neighbors. In P_s probability, the node does not take into account the amount of information from its neighbors, but the number of active neighbors expressed as a percentage. Therefore, $Attenuation_{Ps}$ (13) does not dependent on the number of messages received by the node, but we choose to be time dependent. However, a node does not have many neighbors active at the beginning of the diffusion, so it is important to choose a start time (T) from which we can consider that the neighbors of the node are quite active. We choose to define the moment T when W_s>W_p, i.e. when probability P_s is more important than probability P_p. We consider that this criterion is suitable because the weight of W_s is dependent on the percentage of active neighbors of the node:

$$Attenuation_{P_s} = \begin{cases} e^{-T_{elapsed} \cdot F_{social\_tolerance}} & ,W_s > W_p \\ 1 & ,W_s < W_p \end{cases} \tag{13}$$

$$T_{elapsed} = \frac{Simulation\_clock - T}{T_{max\_socialization}} \tag{14}$$

In Fig. 6 we show an example where the moment T is defined. The P_s probability is affected by the attenuation $Attenuation_{Ps}$ from the beginning of time T. After defining the moment T, we can also determine the elapsed time ($T_{elapsed}$) to evaluate the degree of attenuation.

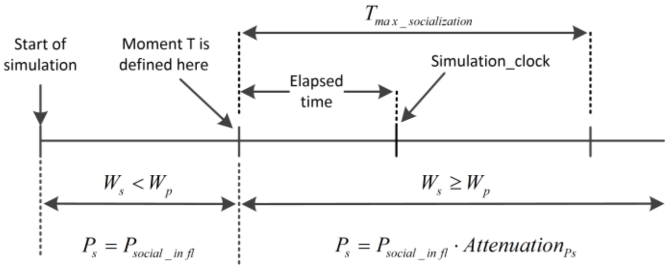


Fig. 6. Example in which the moment *T* is defined

The elapsed time (14) is the difference between the current simulation time (*Simulation_clock*) and the start time *T*. We normalize $T_{elapsed}$ according to a maximum socialization time ($T_{max\_socialization}$) to obtain a period expressed in percentages and which is specific to each node. Thus, $Attenuation_{Ps}$ tends to 0 (maximum attenuation for P_s) as $T_{elapsed}$ tends to 100%. In our model, $T_{max\_socialization}$ is the time required for a node to become completely bored with the activity of its neighbors. Each node has its own value for $T_{max\_socialization}$ and is generated randomly at the beginning of the simulation with a period between 1 and 7 days. In this way, each node has a different period length in which it gets bored with the activity of its neighbors. The genetic algorithm controls $Attenuation_{Ps}$ by learning the parameter $F_{social\_tolerance}$ defined on the range [1, 15].

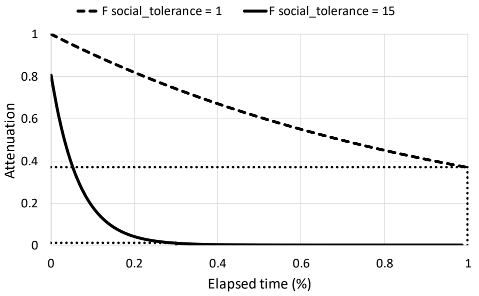


Fig.7. Example of attenuation for a) $F_{social\_tolerance}$ = 1 b) $F_{social\_tolerance}$ = 15

In Fig. 7 we show two examples for the evolution of $Attenuation_{Ps}$. One can see a severe attenuation when $F_{social\_tolerance}$ =15, i.e. probability P_s is completely suppressed when the elapsed time is 30% of the $T_{max\_socialization}$. In the case of $F_{social\_tolerance}$=1, the attenuation is very low and $Attenuation_{P_S} \cong 0.4$ at 100% elapsed time.

3.6. Determining the Weights

The final sending probability (P_f) for a special node (3) depends on the P_p and P_s probabilities. P_p and P_s are weighted by W_p and W_s, which are complementary weights (i.e.$W_p = 1 – W_s$). So, we will only discuss about W_s, which is defined as follows:

$$W_s = 1 - e^{-Pct_{neighbors} \cdot F_{act\_social\_mo\,del}} \qquad \textbf{(15)}$$

where $Pct_{neighbors}$ is the percentage of active neighbors of the node, and $F_{act_social_model}$is a control parameter. We say that a node is active if it has transmitted at least one information, so the weights $W_p$and W_s are not influenced by the number of messages from the neighbors of a node. Instead, W_p and W_s are influenced by the state of the neighbors (i.e. active or inactive node).

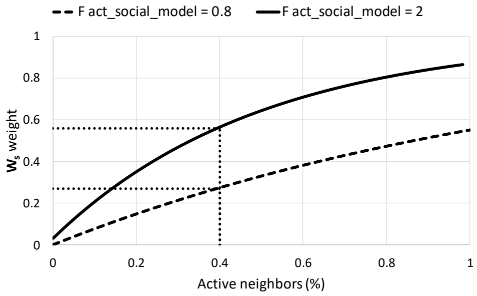


Fig. 8. Evolution of W_s for $F_{act\_social\_model}$= 0.8 and $F_{act\_social\_model}$= 2

We control the evolution of the weight W_s by using the parameter $F_{act\_social\_model}$. In this way we actually control the degree of an individual to be influenced by others, or how quickly the individual adopts a gregarious behavior. In Fig. 8 we show the evolution of W_s for two values of the parameter $F_{act\_social\_model}$: 0.8 and 2. These two values are in fact the limits in which $F_{act_social_model}$is learned by the genetic algorithm.

If a node has, for example, 40% active neighbors, one can see that $W_s \cong 25\%$ when $F_{act\_social\_model}$ has the value 0.8 and $W_s \cong 55\%$ when $F_{act\_social\_model}$ has the value 2. Therefore, when the value of $F_{act\_social\_model}$ is higher, the node is more encouraged to follow its neighbors during information diffusion. Also, the login rate of a node is influenced by W_s during the simulation:

$$\lambda_{lo\,gin} = \begin{cases} \lambda_{normal}, W_s < W_p \\ \lambda_{social}, W_s \geq W_p \end{cases} \qquad \textbf{(16)}$$

We choose a lower login rate (λ_{normal}) for a node when it is not strongly influenced by the activity of its neighbors and a higher login rate (λ_{social}) when the node has a gregarious behavior:

$$\lambda_{final\_lo\,gin} = \begin{cases} \lambda_{lo\,gin}, 11AM \leq simulation\_clock < 11PM \\ \dfrac{\lambda_{lo\,gin}}{2}, 11PM \leq simulation\_clock < 11AM \end{cases} \qquad \textbf{(17)}$$

Moreover, we consider a change in the login rate of the nodes depending on the time of day. Thus, we define two time periods (17): the time interval 11AM – 11PM is considered the daytime period (λ_{login} remains unchanged for users), and the time interval

11PM – 11AM is considered the nighttime (λ_{login} is halved for each user). In this way, we model the fact that users are less active at night.

3.7. Learning the Model Parameters

The proposed information diffusion protocol contains a large number of parameters that influence the evolution of the diffusion. It is difficult to adjust so many parameters in order to obtain an evolution of the diffusion as close as possible to the real one. The automatic learning of the parameters is the solution that helps us in this problem and we have chosen to use a genetic algorithm. In Table 1 we show the parameters of our diffusion model that are learned by the genetic algorithm, and in Table 2 we show the parameters that are not learned.

Table 1. The parameters of the diffusion model that are learned by the genetic algorithm

Parameter	Definition range	Comment
λ_{normal}	[120, 240]	The login rate used by the nodes with a small number of active neighbors (e.g. the minimum value means 1 login every 120 minutes)
λ_{social}	[30, 60]	The login rate used by the nodes with a high number of active neighbors
$T_{max\_informant}$	[1, 90]	The maximum time period (in minutes) in which an informant node spreads the information
$F_{attenuation}$	[0.01, 1]	Attenuation factor for probability P_p
W_{cred}	[0.01, 0.2]	The control weight over increasing the information credibility over time
$F_{act\_social\_model}$	[0.8, 2]	Control factor to adjust the evolution of the W_s weight
$F_{social\_tolerance}$	[1, 15]	Control factor to adjust the attenuation of the P_s probability
W_{act}	[0.2, 1]	The control weight of the activation level from the $P_p$probability

Table 2. The parameters of the diffusion model that are not learned by the genetic algorithm

Parameter	Definition range	Comment
F_{social_infl}	[5,10]	Control factor to adjust the evolution of P_{inf_social}
Source nodes	5%	Percentage of source nodes
Informant nodes	30%	Percentage of informant nodes
P_b	0.8	The basic probability of the nodes
$T_{max\_socialization}$	[1, 7]	The maximum time period (in days) in which a node becomes bored with the activity of its neighbors
$Cred_{Info}$	0.3	Initial credibility of the information (used by the source nodes)
$T_{start\_informant}$	Depending on the real diffusion	The time when the informant type nodes activate and suddenly encourage the spread of information

3.8. The Genetic Algorithm

Background. Genetic algorithms are based on the principles of natural selection [23], which states that the survival of an organism consists in the survival of the most adapted species. For a species to survive in time, the following stages are required: selection, reproduction and mutation. The *selection* process consists in the fact that certain organisms of the species better tolerate the environment in which they live and, consequently, have a greater chance of survival. Thus, these organisms are more adapted (fitted) to the environment, due to specific genes (the set of all genes is called a chromosome). More adapted organisms have a higher chance of reproducing. In the *crossover* process, parents transmit certain genes to the offspring. The new generation that results from the crossover is a new epoch and this process represents a way to simulate the evolution of the species over time. However, selection and reproduction are not sufficient to ensure long-term improvement of an organism's adaptation. Also, there is the possibility that an organism will suffer changes of the genes that did not result from the crossing of the parents and these changes may lead to a better adaptation of the new individual. The process in which these new genes suffer unexpected changes is called *mutation*.

By analogy with the search for solutions to a problem, the genetic algorithm is based on the concept of biological evolution to simulate a finite number of epochs in order to find the most fitted individuals that ultimately represent the desired solutions. [24]. The set of all individuals of an epoch is called population. In our case, the genetic algorithm learns certain parameters (Table 1) to obtain an evolution of the information diffusion as close as possible to the real one. The real evolution of diffusion and the result provided by our diffusion model from a particular individual are both represented as a one-dimensional array. We choose that the stop condition of the genetic algorithm should be given by the iteration of a certain number of epochs. The following operations are performed at each epoch: selection, crossover and mutation.

Selection. There are different methods of selecting parents to create the new generation, and we choose the *tournament selection*. The basic idea for this type of selection is as follows:

- We sample k random individuals from the current population and choose the one with the best fitness as a parent (k=2 in our case);
- The procedure is repeated to select more parents.

We also use the *elitism* operation in which we get the individual with the best fitness from the previous population and add it to the new population. In this way we will never lose the best solution found throughout the epochs.

Crossover. After the selection process is completed, the selected parents (i.e. mating pool) are used for *crossover*. We choose pairs of two parents to create children with new genes for the new population. In Fig. 9 we show how the parents are paired. If the number of created children is not sufficient to create the new population at the expected size, then the pairs of parents are crossed again to obtain other children.

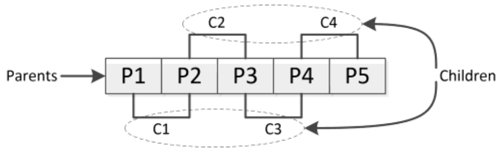


Fig. 9. The way the parents are paired for mating

In our case, the genes from each individual have a real numerical representation, and we use the *arithmetic crossover*:

$$z_i = \alpha \cdot x_i + (1-\alpha) \cdot y_i \qquad \textbf{(18)}$$

where x_i and y_i are the *i*-th gene of the two parents, z_i is the *i*-th gene of the child, and α is a uniformly distributed random number in the interval [0, 1].

Mutation. Each gene of a new child has a small chance of undergoing unexpected changes. This unexpected event represents the *mutation* operation. We choose the *random resetting mutation*, in which the value of a gene is replaced by a random value from its given range (Table 1).

General Pseudocode. Below one can see the general pseudocode of a genetic algorithm (Algorithm 1), as used for the experiments in the present paper.

Algorithm 1. The proposed algorithm

```
For epoch = 1 to MAX_EPOCH do
//Compute the fitness of all individuals in the
  //population. The dissemination of information is
  //simulated for each individual and is compared to the
  //real one
Compute_fitness(Population);
//Parent selection for the mating pool
Mating_pool = Selection(Population);
//Elitism: the best individual is always chosen for the
  //next population
Best_individual = Best(Population);
//Empty the population and keep only the best individual
Clear_population(Population);
Population.Add(Best_individual);
//Create (MAX_INDIVIDUALS - 1) children
Children = Crossover(Mating_pool, MAX_INDIVIDUALS - 1);
//Modify newly obtained children
Final_children = Mutation(Children);
//Add children to the population
Population.Add(Final_children);
End
//The best solution obtained
Get_parameters(Best_individual);
```

The Fitness Function. The fitness of an individual, which is used in the selection process, is computed using a fitness function that is problem specific. In our case, the fitness function (19) computes the difference between the real and the simulated diffusion using the Euclidean distance, where O_i is a sample from the real diffusion, S_i is a sample from the simulated diffusion, and N is the size of the arrays.

$$f = \sqrt{\sum_{i=1}^{N} (O_i - S_i)^2} \tag{\textbf{19}}$$

In our case, a better individual will have a lower value for *f*: the smaller the *f*, the closer the individual is to the optimal solution.

In Fig. 10 we show an example in which the parameters of each individual are used in the information diffusion model to provide an evolution of the diffusion. Then, the obtained evolution can be compared with the real one using the fitness function to obtain the fitness of an individual.

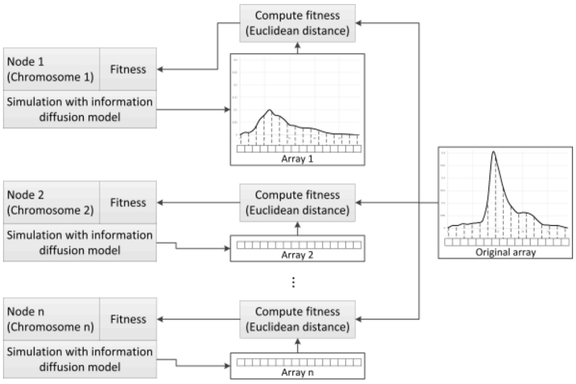

Compute fitness (Euclidean distance)
Node 1 (Chromosome 1)
Fitness
Simulation with information diffusion model
Array 1
Node 2 (Chromosome 2)
Fitness
Simulation with information diffusion model
Compute fitness (Euclidean distance)
Array 2
Original array
⋮
Node n (Chromosome n)
Fitness
Simulation with information diffusion model
Compute fitness (Euclidean distance)
Array n

Fig. 10. Example of fitness computation for each individual

In our case, the genetic algorithm has the following configuration: epochs – 100, population size – 100 individuals, and mutation rate – 10%. This algorithm aims to learn 8 parameters from our diffusion model, shown in Table 1.

4. Experimental Results

4.1. Description of the Two Real-World Datasets

In order to evaluate the efficiency of our information diffusion model, we use two real datasets that contain the activity of users over time. We use the genetic algorithm to learn the parameters of our information diffusion model for each real diffusion of a dataset. The first dataset is a collection of information on the activity of Twitter users during the announcement for the discovery of the Higgs Boson [1], which also contains the network structure. The activity of a user on Twitter represents his/her action to share a piece of information he/she saw on a neighbor's page (retweet). The shared information is visible to all of the user's neighbors. The dataset also contains the timestamps when users share their information. Therefore, we can extract the activity of users over time and correlate it with the evolution of information diffusion, i.e. the target solution used by the genetic algorithm. The second dataset, *memetracker9* [2], is a large collection of data in which the exchange of information between users is described by text messages or links to other web pages. This dataset contains the diffusions of several topics that are collected over several months. The diffusion of each topic discussed by users is easily identified in paper [25], and we choose two of them: "lipstick on a pig" and "fundamentals of our economy are strong". The diffusion evolutions from each dataset are provided to the genetic algorithm to learn the parameters of our diffusion model and to obtain evolutions as close as possible to the real ones.

4.2. Results for the *Higgs* Dataset

For the Higgs dataset, we apply the genetic algorithm on a synthetic network of 1000 nodes with scale-free topology because the original network contains 456,626 nodes and 14,855,842 connections, which results in a very high simulation time and cannot be used in the genetic algorithm. In [6], the authors state that the evolution of information diffusion in scale-free networks is similar even if the networks have different sizes. Thus, we can apply the genetic algorithm without having to simulate a very large network.

The parameters learned by the genetic algorithm are initialized with random values in their specific range (Table 1), and it is expected to obtain weaker solutions in the first epochs. In Fig. 11 we show an evolution example of the best solutions from each epoch on the diffusion of the Higgs dataset and one can see that the solutions are drastically improving in the first 20 epochs. We consider that 100 epochs is an acceptable stop condition for the genetic algorithm because no significant improvements can be observed for a higher number of epochs.

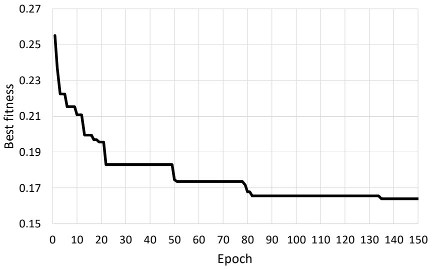


Fig. 11. The best fitness obtained at each epoch

In the case of the Higgs dataset, we show in Fig. 12.a the evolution of the information diffusion obtained by our diffusion model on the 1000-node synthetic network. The continuous line represents the real evolution of the diffusion, while the dotted line represents the diffusion obtained by our model. The simulated diffusion is obtained by counting the active nodes at intervals of one hour. Then, we use the learned values of the diffusion model parameters to run a simulation on the real network provided by this dataset.

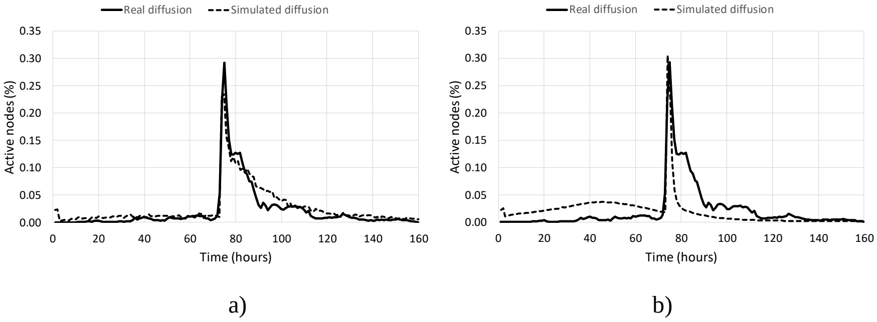
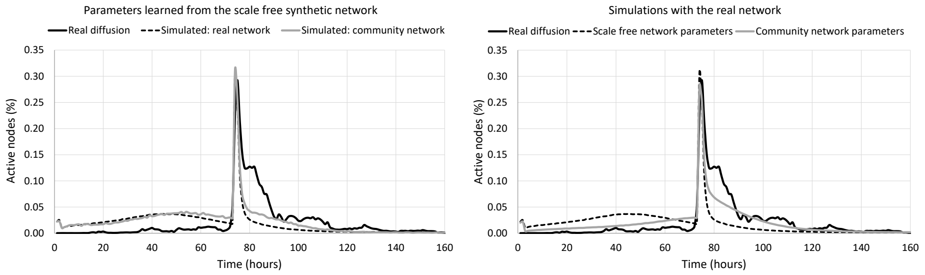


Fig. 12. Simulated diffusion on: a) synthetic 1000-node network, b) real network

We show the result in Fig. 12.b and one can see that the obtained diffusion is close to the real one, except that the simulated diffusion has a greater attenuation. We could not make any further adjustments of the parameter values on this large network because it takes a day to run 2-3 simulation rounds.

We mention that for the Higgs dataset we do not use the modeling for the daytime and nighttime periods because users had different geographical positions on several distant continents, according to [7], and our algorithm does not take into account different times of the day between users.

Using a Community Network. Given that real networks contain many communities, we want to observe the impact of using a synthetic network with communities in our information diffusion model. The motivation of this study is due to the fact that the simulated diffusion has a greater attenuation (Fig. 12b). Regarding the high peak obtained with our model on the real network (Fig. 12b), in our investigations we observe a very similar behavior when we use a synthetic network with communities. This community network has 1000 nodes and was generated using the Gaussian random partition graph [26]. In Fig. 13a, one can see the evolution of simulated diffusion both on the real network and on the network with communities. These evolutions are provided by our model using the parameters learned by the genetic algorithm on the synthetic scale free network. Due to the similarity between these two evolutions, we use the genetic algorithm to learn the parameters of the diffusion model on this network with communities. We want to simulate the diffusion of information on the real network with two different sets of parameters: the first set is learned by the genetic algorithm on the synthetic scale free network and the second set is learned on the synthetic network with communities. We can observe in Fig. 13b that the simulation of the real network with the second set of parameters now has an evolution closer to the real diffusion. Based on these observations we can say that a limitation of our model is the choice of a synthetic network with a certain topology.



a) b)

Fig. 13. Comparative simulations: a) real network and community network, b) real network with different model parameters

4.3. Results for the *memetracker9* Dataset

The second dataset (*memetracker9*) is very large and contains conversations between users for several months, between 2008 and 2009. This dataset does not provide the real network, so we show experimental results using only the 1000-node synthetic network. In paper [25], the authors provide a picture that contains various diffusions from this dataset and highlight the main phrases (i.e. the topic of discussions) for each diffusion. In our work, we use specific keywords to extract the diffusion of the following two topics from September 2008: “lipstick on a pig” and “fundamentals of our economy are strong”. The keywords used to identify the phrases of the first topic are “lipstick” and “pig”, while for the second topic we use the keywords “strong”, “economy” and “fundamentals”. Our diffusion model is capable of providing a single evolution of the diffusion for a single topic discussed by users. In Fig. 14 we stack the diffusion of the two topics on the same plot to easily observe the different time periods in which these diffusions are active and also it is easier to compare the evolution of the diffusions with those of [25]. The continuous lines represent the real diffusions and the dotted lines are the simulated diffusions. We also distinguish the two topics by line width: high width for the “lipstick on a pig” topic, and low width for the second topic. The simulated diffusion is obtained by counting the active nodes at six-hour intervals.

The evolution of diffusion from each topic is obtained by a separate simulation using the genetic algorithm, therefore the parameters learned for the diffusion model have different values for the two topics. Regarding the parameters that are not learned with the genetic algorithm, the main difference between the simulation of the two topics is that we change $T_{start\_informant}$, which is the moment when the information is suddenly spread (i.e. users discuss a lot about the given topic). Although the diffusion model is applied on the synthetic network, one can see in Fig. 14 that the obtained diffusion is very close to the real one.

In our model we propose that the user login rate depends on the time of day (i.e., day or night). In [27] a detailed analysis of user activity is presented on two datasets and a certain sinusoidal periodicity is observed in this activity. The authors notice that these activities are periodic at 24-hour intervals on both datasets. They also illustrate the activity of users during a week, and an interesting aspect is that they identify a consistent drop in activity during weekends on both datasets. Our model does not take into account an attenuation of user activity during weekends, therefore it cannot reproduce these low amplitudes of periodicity as the real data (e.g. Fig. 14, time frame 230-270 or 460-480).

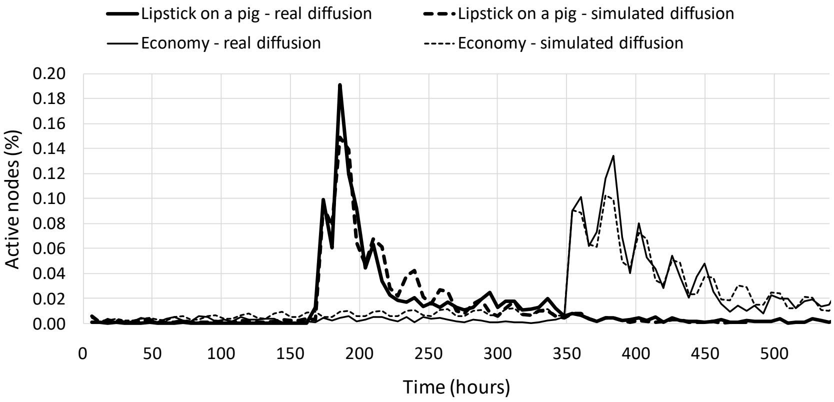
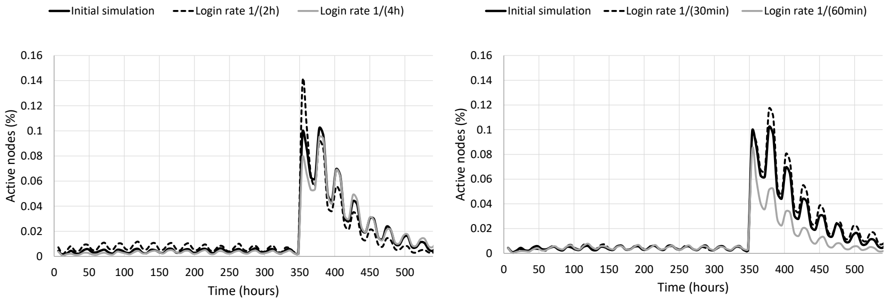


Fig. 14. Information diffusion simulation for two topics in the *memetracker9* dataset

4.4. Influence of Parameters

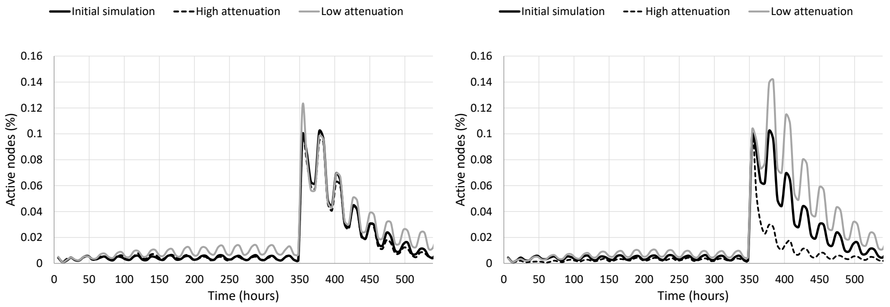
In this section we make an analysis for some of the individual parameters of our diffusion model. The impact of a parameter can be analyzed separately by simulating the diffusion if we keep all the parameters constant and adjust only the parameter of interest. If we refer to the parameters learned by the genetic algorithm, we can show, for example, the impact of the login rate (Fig.15) and that of the boredom of the nodes (Fig. 16). Depending on the two models we have:

- psychological modeling: normal login rate (λ_{normal}), boredom over the amount of information ($F_{attenuation}$)
- sociological modeling: social login rate (λ_{social}), boredom over the activity of neighbors ($F_{social\_tolerance}$)



a) b)

Fig. 15. Login rate modeling: a) normal login rate; b) social login rate



a) b)

Fig. 16. Boredom modeling: a) over the amount of information; b) over the activity of neighbors

We can see that the parameters of sociological modeling have a great impact on diffusion. The continuous black line represents the simulated diffusion using the values of the parameters obtained with the genetic algorithm. The other two evolutions are obtained by varying a single parameter (the one of interest) to observe its impact on the diffusion.

Unlike other works (e.g. [5, 6]) in which the parameters are manually adjusted through repeated simulations, we present a model in which its parameters are automatically learned, and the obtained diffusions are very promising compared to the real ones.

5. Conclusions

Developing models capable of imitating the information diffusion on a social network is a challenging task at the moment. In this paper we propose such a model that imitates the diffusion of information as well as possible. The model is based on stochastic node-level decisions. Each node has its own set of rules by which its actions are defined. The decision of a node, whether or not to transmit information, is modeled both from a psychological point of view and from a sociological point of view. In modeling from the psychological point of view, we propose that the decision of a node should be influenced by its preferences on the content of the information. On the other hand, when modeling from the sociological point of view, we propose that the decision of the node should be influenced by the activity of its "friends", i.e. we model the gregarious behavior of the node. Also, most of the parameters of our diffusion model are learned by means of a genetic algorithm to eliminate the effort of adjusting their values. Then, we can use the learned values in the proposed diffusion model to obtain an evolution of information diffusion as close as possible to the real one. We use two datasets that contain real diffusions, and the results show that our model reproduces them very well.

However, one must take into account the fact that there is no unique model for all situations. One goal of our work was to show that the proposed model with psychosocial factors is capable of approximating real data. But every case will likely need different values of the parameters, which can be found through automatic search using genetic algorithms or other optimization methods, e.g. based on gradients.

As a future direction of investigation, one can investigate the inclusion of additional parameters in the model (such as those accounting for weekend activity or the different geographical distribution of users on different continents), in order to further increase the prediction accuracy. One must also consider the trade-off between increase flexibility and the growth of the search space, while applying the automatic determination of parameter values.

References

1. Stanford Network Analysis Platform (SNAP): Higgs Twitter Dataset. [Online]. Available: https://snap.stanford.edu/data/higgs-twitter.html
2. Stanford Network Analysis Platform (SNAP): 96 million memes from Memetracker. [Online]. Available: https://snap.stanford.edu/data/memetracker9.html
3. de C Gatti, M. A., Appel, A. P., dos Santos, C. N., Pinhanez, C. S., Cavalin, P. R., Neto, S. B.: A Simulation-based Approach to Analyze the Information Diffusion in Microblogging Online Social Network. In Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World. IEEE Press, Washington, DC, USA, 1685-1696. (2013)
4. Serrano, E., Iglesias, C. A., Garijo, M.: A Novel Agent-Based Rumor Spreading Model in Twitter. In Proceedings of the 24th International Conference on World Wide Web. Association for Computing Machinery, Florence, Italy, 811-814. (2015)
5. Chen, J., Song, Q., Zhou, Z.: Agent-Based Simulation of Rumor Propagation on Social Network Based on Active Immune Mechanism. Journal of Systems Science and Information, Vol. 5, No. 6, 571-584. (2017)
6. Mazzoli, M., Re, T., Bertilone, R., Maggiora, M., Pellegrino, J.: Agent Based Rumor Sspreading in a Scale-free Network. (2018). [Online]. Available: https://arxiv.org/abs/1805.05999
7. De Domenico, M., Lima, A., Mougel, P., Musolesi, M.: The Anatomy of a Scientific Rumor. Scientific Reports, Vol. 3, No. 2980. (2013)
8. Floria, S.-A., Leon, F., Logofătu, D.: A Credibility-based Analysis of Information Diffusion in Social Networks. In Proceedings of the 27th International Conference on Artificial Neural Networks (ICANN). Springer International Publishing, Rhodes, Greece, 828–838. (2018)

9. Floria, S.-A., Leon, F., Logofătu, D.: A Model of Information Diffusion in Dynamic Social Networks Based on Evidence Theory. Journal of Intelligent & Fuzzy Systems, Vol. 37, No. 6, 7369-7381. (2019)
10. Huo, L., Ding, F., Liu, C., Cheng, Y.: Dynamical Analysis of Rumor Spreading Model Considering Node Activity in Complex Networks. Complexity, Vol. 2018, 1-10. (2018)
11. Zhao, L., Wang, X., Wang, J., Qiu, X., Xie, W.: Rumor-Propagation Model with Consideration of Refutation Mechanism in Homogeneous Social Networks. Discrete Dynamics in Nature and Society, Vol. 2014. (2014)
12. Liu, L., Qu, B., Chen, B., Hanjalic, A., Wang, H.: Modeling of Information Diffusion on Social Networks with Applications to WeChat. Physica A: Statistical Mechanics and its Applications, Vol. 496, 318-329. (2018)
13. Yan, Q., Wu, L., Liu, C., Li, X.: Information Propagation in Online Social Network Based on Human Dynamics. Abstract and Applied Analysis, Vol. 2013, 1-6. (2013)
14. Sun, Q., Li, Y., Hu, H., Cheng, S.: A Model for Competing Information Diffusion in Social Networks. IEEE Access, Vol. 7, 67916-67922. (2019)
15. Li, D., Zhang, S., Sun, X., Zhou, H., Li, S., Li, X.: Modeling Information Diffusion over Social Networks for Temporal Dynamic Prediction. IEEE Transactions on Knowledge and Data Engineering, Vol. 29, No. 9, 1985-1997. (2017)
16. Guille, A., Hacid, H., Favre, C., Zighed, D. A.: Information Diffusion in Online Social Networks: A Survey. ACM SIGMOD Record, Vol. 42, No. 2, 17-28. (2013)
17. Li, M., Wang, X., Gao, K., Zhang, S.: A Survey on Information Diffusion in Online Social Networks: Models and Methods. Information, Vol. 8, No. 4, 118. (2017)
18. Dey, K., Kaushik, S., Subramaniam, L. V.: Literature Survey on Interplay of Topics, Information Diffusion and Connections on Social Networks, (2017). [Online]. Available: https://arxiv.org/abs/1706.00921
19. Hawkins, D. I., Mothersbaugh, D. L.: Consumer Behavior: Building Marketing Strategy (11th ed.). Boston, MA: McGraw-Hill. (2010)
20. Shariff, S. M., Zhang, X., Sanderson, M.: User Perception of Information Credibility of News on Twitter. In Proceedings of the 36th European Conference on IR Research, ECIR 2014. Springer Cham, Amsterdam, The Netherlands, 513-518. (2014)
21. Castillo C., Mendoza, M., Poblete, B.: Information credibility on twitter. In WWW'11: Proceedings of the 20th international conference on World wide web. Association for Computing Machinery, New York, NY, United States, 675-684. (2011)
22. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, chapter 16. (2010)
23. Darwin, C.: Origin of the Species. (1859).[Online]. Available: http://darwin-online.org.uk/converted/pdf/1861_OriginNY_F382.pdf
24. Baeck, T., Fogel, D. B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. Institute of Physics Publishing Publishing and Oxford University Press. (1997)
25. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the Dynamics of the News Cycle. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Association for Computing Machinery, Paris, France, 497-506. (2009)
26. NetworkX developers: NetworkX Python package. [Online]. Available: https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.generators.community.gaussian_random_partition_graph.html#networkx.generators.community.gaussian_random_partition_graph (current September 2020)
27. Flamino, J., Dai, W., Szymanski, B. K.: Modeling Human Temporal Dynamics in Agent-Based Simulations. In Proceedings of the 2019 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. Association for Computing Machinery, New York, NY, United States, 99-102. (2019)

Sabina-Adriana Floria received the B.Sc. degree in Computer Science (2014), the M.Sc. degree in Embedded Computers (2016) and the Ph.D. degree in Computer Science and Information Technology (2020) from the "Gheorghe Asachi" Technical University of Iași, Romania, Faculty of Automatic Control and Computer Engineering. She became a Teaching Assistant at the Department of Computer Science and Engineering, "Gheorghe Asachi" Technical University of Iași in 2017, where she teaches the following subjects: Discrete Mathematics, Computer Programming, Computational Logic, Logical Design, Modelling and Simulation, Computer System Testing. Her research interests include the following aspects: information diffusion in social networks and performance evaluation in complex networks.

Florin Leon received a Ph.D. degree in computer science from the "Gheorghe Asachi" University of Iași, Romania in 2005, followed by a postdoctoral fellowship completed in 2007. In 2015, he defended his habilitation thesis. He has been a faculty member at the Department of Computers and Information Technology of the same university since 2005. In 2015, he became a Full Professor at the same department. He authored and co-authored more than 160journal articles, book chapters and conference papers, and 14 books. He was a member in the guest editorial boards for three journal special issues, and he participated in 29 national and international research projects, three of which as principal investigator. His research interests include: artificial intelligence, machine learning, multiagent systems and software design. Prof. Leon was a member of the organizing committees or program committees chair of five conferences. He is currently a member of IEEE Systems, Man and Cybernetics Society: Computational Collective Intelligence Technical Community and the Romanian Association for Artificial Intelligence.

Received: April 15, 2020; Accepted: December 10, 2020

Computer Science and Information Systems 18(3):729–748 https://doi.org/10.2298/CSIS200510047L

Arabic Linked Drug Dataset Consolidating and Publishing

Guma Lakshen[1], Valentina Janev[2], and
Sanja Vraneš[2]

[1] School of Electrical Engineering, University of Belgrade,
11000 Belgrade, Serbia
jlackshen65@yahoo.com
[2] Mihajlo Pupin Institute, University of Belgrade,
11060 Belgrade, Serbia
{valentina.janev, sanja.vranes}@pupin.rs

Abstract. The paper examines the process of creating and publishing an Arabic Linked Drug Dataset based on open drug datasets from selected Arabic countries and discusses quality issues considered in the linked data lifecycle when establishing a semantic Data Lake in the pharmaceutical domain. Through representation of the data in an open machine-readable format, the approach provides an optimum solution for information and dissemination of data and for building specialized applications. Authors contribute to opening the drug datasets from Arabic countries, interlinking the data with diverse repositories such as DrugBank, and DBpedia, and publishing it in a standard open manner that allows further integration and building different business services on top of the integrated data. This paper showcases how drug industry can take full advantage of the emerging trends for building competitive advantages. However, as is elaborated in this paper, better understanding of the specifics of the Arabic language is needed in order to extend the usage of linked data technologies in Arabic companies.

Keywords: drug management applications; Linked Data; methodology; open ecosystems; quality assessment.

1. Introduction

Today, data is growing at a tremendous rate on the Web, and is expected to reach 35 Zettabytes (1 ZB= 10^{21} bytes) by the end of 2019, and exceeds 175 zettabytes by 2025 [1]. This amount of data creates new opportunities for modern enterprises, especially in the context of analyzing value chains in a broader sense. The value chain considered in this study is the one presented in Fig. 1 that can be divided into 3 layers:

- Data sources layer, where different data sources and systems generate data. The interconnected systems in this layer are property of the organization or its partners, or the data is freely available on the Web.
- Data management layer, where the data is acquired via customized interfaces or crawled from the Web and transmitted using interconnected networks into storage

data centers. The data management layer in a modern data ecosystem is composed of data lakes and data warehouses.

– Data analytics and business intelligence layer, which refers to the application of artificial intelligence, mining algorithms, machine learning, and deep learning to process the data and extract useful knowledge for better decision making. Additionally, data visualization tools are used for visually examining processed data.

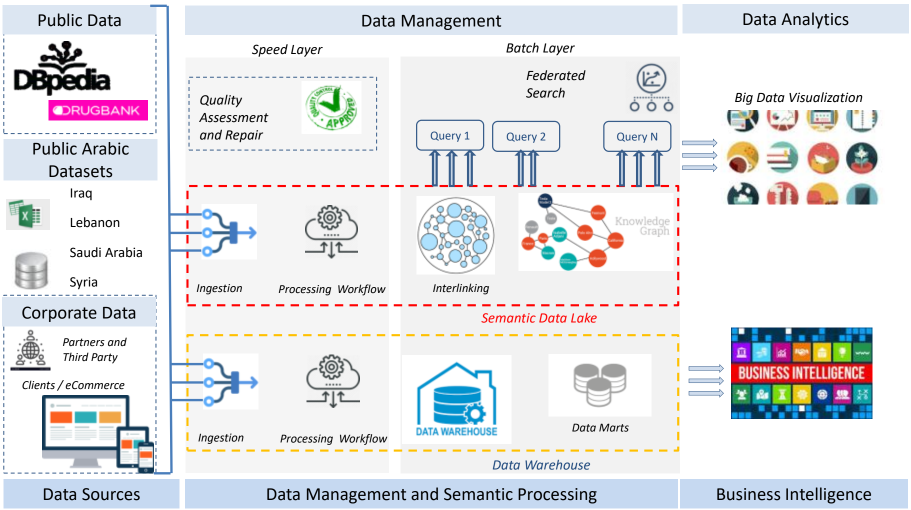


Fig. 1. Modern data ecosystem

The development of business intelligence services is simple when all data sources collect information based on unified file formats and the data is uploaded to a data warehouse. However, the development of a distributed software system requires the interaction of services and the use of resources from diverse organisations throughout the Web [2]. The biggest challenge that enterprises face is the undefined and unpredictable nature of data appearing in multiple formats. Additionally, in order to gain competitive advantage over their business rivals, the companies utilize open data resources that are free from restrictions and can be reused, redistributed, and can provide immediate information and insights. Thus, in a modern data ecosystem, data lakes and data warehouses are both widely used for storing big data. A *data warehouse* [3] is a repository for structured, filtered data that has already been processed for a specific purpose. A *data lake* is a large, raw data repository that stores and manages the company data bearing any format. The concept of data lake was introduced in the last decade in order to address issues related to processing big data [4]. Moreover, recently, the *semantic data lakes* [5] are introduced as an extension of the data lake supplying it with a semantic middleware, which allows the uniform access to original heterogeneous data sources. The data management life cycle (see Fig. 1) is divided into two parts. Data pre-processing activities like data integration, enrichment, transformation, reduction, and cleansing occur in the speed layer, while maintaining the knowledge graphs (part of the *semantic data lake*) and data marts is part of the batch layer. In addition to the speed and batch layers, in big data applications, a third layer is often added named (merged serving layer that makes the combined data available for data analysis and reporting, see also Fig. 3.

In the last decade, more and more corporations have introduced semantic processing technologies also to improve the interoperability, i.e., they use the *Linked data principles and standards* recommended by the W3C consortium [6]. The use of common data models provides a standard way to store and query the data and, furthermore, creates an opportunity to build a virtual middleware under which the heterogeneous formats are homogenized on-the-fly without data transformation or materialization.

Taking the drug industry and drug management as an example, this study was motivated by the following research questions (related to operations in the depicted semantic layer, see red rectangular in Fig. 1):

– What are the benefits from integrating freely available data sources (e.g., DBpedia) into the existing business value chain and what are the drawbacks of this approach?
– What is the quality of open data e.g. the Arabic DBpedia?
– How can business intelligence services (e.g. a search operation) be implemented on top of a semantic drug data lake?

The paper is structured as follows. Section 2 presents the research framework and proposes a methodology for consolidating heterogeneous data sources using the Linked Data principles [6], [7]. Section 3 presents the process of transforming selected Arabic two-star drug datasets published on various websites, into a five-star linked open data (LOD)[1], connected to the DBpedia [8] and DrugBank [9]. The overall count of the distinct data is 31,906 drugs, while 23,971 drugs are interlinked to DBpedia. The proposed methodology advances the state-of-the-art taking into account quality issues and specifics for the Arabic language and provide examples of how the drug data lake (knowledge graph interlinked with DBpedia[2] and DrugBank[3]) can be analyzed with business objectives in mind (retrieval of drug information). Section 4 discusses the results and presents the main conclusions. Section 5 concludes the article.

2. Research Overview: Building Linked Data Application on Top of Arabic Open Data

In the drug industry, the rapidly increasing amount of data on the Web opens new opportunities for integrating and enhancing drug knowledge on a global scale. As far as medical data available in the Arab region, there are only a handful of Arabic drug applications such as Webteb[4], Altibbi[5], 123esaaf[6], Kuwait Pharmacy KP[7] which provides their services in Arabic and English, but unfortunately, the data is not open and most are

[1] https://www.w3.org/2011/gld/wiki/5_Star_Linked_Data
[2] https://wiki.dbpedia.org/
[3] https://www.drugbank.ca/
[4] https://www.webteb.com/aboutusen
[5] https://www.altibbi.com/
[6] https://www.123esaaf.com/
[7] http://www.kuwaitpharmacy.com/default.aspx

not free. Arabic language content on the Web is less than 3%. The situation is even worse regarding Arabic open data, linked data, and open linked data on drugs.

This limitation of Arabic content motivated the authors to propose a solution that will enable Arabic-speaking end users to benefit from private datasets interlinked with public data and local data enriched with information from the Web. The goal of the innovative Arabic drug application proposed in this study is to enable end-users to answer inquiries about drug availability in the open datasets and to enrich the local data store with information from the Web. Examples of key business queries are:

1. For a particular drug, retrieve relative information in the Arabic language (if exists) from other identified datasets, such as DrugBank and DBpedia.
2. For a particular drug, retrieve equivalent drugs, and compare their active ingredients, contradictions, and prices.
3. For a particular drug, retrieve valuable information about equivalent drugs with different commercial names, manufacturers, strengths, forms, prices, etc.
4. For a particular drug, retrieve its reference information to highlight possible contradiction, e.g., in combination with other drugs, allergies, or special cases (e.g., pregnancy).
5. For a particular active ingredient, retrieve advanced clinical information, i.e., pharmacological action, pharmacokinetics, etc.
6. For a particular drug, retrieve its cost, manufacturer, and country.

The authors also propose to split the implementation of a linked data application development into three software development phases as is presented in Table 1.

Table 1. Linked data application phases

Phase	Description
Initialization	Business objectives and requirements: Requirement specification, technical characterization, and setting up of the demo site; Establishing acceptance (success) criteria for pilot applications validation based on performance characteristics, usability, as well as EU and national regulations (e.g., related to data access and security measures); Data categorization and description: Analysis of the datasets to be published in linked data format and selection of vocabularies and development other specifications for metadata description. *Example.* In addition to corporate data, the targeted data is selected from the Arabic drug datasets (Iraq, Saudi Arabia, Syria, and Lebanon) along with the public datasets (DrugBank and DBpedia). Appropriate vocabularies are selected or developed, and mapping rules are defined.
Innovation	Integrating datasets in the form of a knowledge graph: Data access, transformation, and enrichment. For instance, in this phase, the data lake is established, and semantic processing is performed, which includes all the stages of data preparing, modeling, and conversion. At each stage, quality issues are revised, and if the quality is not satisfactory, the appropriate stage is revisited. After the transformation, master data is stored for subsequent use. Generic component selection and tool customization for the pilot applications: Customization of linked data components for use in the targeted domain. *Example.* In this phase, tools for federated search and data are selected. Additionally, big data analytics tools are selected, and custom visualization and user interfaces are created [8].
Specific tools development and Validation	In this phase open-source tools are validated for reuse; feedback is provided for improving the solution components; and new interfaces are built.

3. Validation of Software Development Methodology

3.1. Initialization (Stage I): Data Preparation

Data Selection. As a use case scenario, the authors selected four drug data files from four different Arabic countries, Iraq, Saudi Arabia, Syria, and Lebanon, as shown in Table 2. Most of the open published files in the Arab region are either in PDF or XLS format. The reasons for choosing XLS format were data fidelity, ability to source from a wider range of public sector domains, and to have increased value that comes from many information linkages. The authors believe that for many years to come, more drug data will be published in XLS format in the Arab countries.

The selected datasets are open data published by health ministries or equivalent bodies in the respected governments. They are regularly updated, usually after a two-year period. As it can be noticed from the difference in the number of columns, the structure of the datasets is not unified, which makes the unification and mapping of data necessary.

Table 2. Selected Arabic open drug datasets

Country	DataSet URI	No. of drugs
Iraq	http://www.iraqipharm.com/upfiles/drug/dreg.xls	9090
Lebanon	https://moph.gov.lb/userfiles/files/HealthCareSystem/.../7.../ WebMarketed20170307.xls	5822
Saudi Arabia	https://www.sfda.gov.sa/en/drug/search/pages/default.aspx	6386
Syria	http://www.moh.gov.sy/LinkClick.aspx	9375

Data Analysis. The data quality (DQ) of the selected files is too low, e.g., most XLS documents do not represent the generic name or their ATC code, which makes the data almost unusable for further transformation. However, the data from Lebanon and Saudi Arabia is in a form of generic online drug database, see Table 2.

These two databases contain 13,445 records. In order to gather the data in HTML format, the authors built HTML Crawlers based on JSOUP, which is a Java library for extracting and manipulating data. It iterates through the drug list (link by link), gathering information for each drug separately. Unfortunately, Syria and Iraq do not provide such databases, so the authors have to use their XLS files and implement additional transformations to extract active ingredient information.

Data Cleaning. OpenRefine[8] (version 2.6-rc1) was used to clean the selected data in order to make it coherent and ready for further operations according to the methodology. A well-organized cleaning operation minimizes inconsistencies and ensures data standardization among a verity of data sources.

[8] http://openrefine.org

Quality Assessment. In what follows, we will describe the process for transforming, linking, and publishing the Arabic drug data. When it comes to quality assessment of the DBpedia Arabic Chapter, there are problems specific to the Arabic language that result in:

1. Presentation of characters as symbols via web browsers due to errors during the extraction process.
2. Wrong values in numerical data, due to the use of Hindu numerals in some Arabic sources.
3. Occurrence of different names for the same attribute, for instance, the birth date attribute appears in various info-boxes by different names: one time as "تاريخ الميلاد" [birth date], another time as"تاريخ الولادة" [delivery date], the third time as "الميلاد" [birth].
4. Inconsistency of names between the infobox and its template; for instance, there is a template called "مدينة" [city] while the infobox name is called "معلومات مدينة" [city information].
5. Geo-names templates formatting problems when placed in the infobox.
6. Errors in *owl:sameAs* relations and problems in identifying the *owl:sameAs* relations due to heterogeneity in different data sources.

However, some of the problems present in other DBpedia chapters are also identified in the Arabic chapter, specifically:

1. Wrong Wikipedia Infobox information; for example, the height of minaret of the grand mosque in Mecca (the most valuable mosque for all Muslims) is given as 1.89 m, where the correct height is 89 m.
2. Mapping problems from Wikipedia, such as unavailability of infoboxes for many Arabic articles; for example, “Man-made River in Libya النهر الصناعي في ليبيا,” it is considered as the biggest water pipeline project in the world, or not containing all the desired information.
3. Object values incompletely or incorrectly extracted.
4. Data type incorrectly extracted.
5. Some templates may be more abstract, thus cannot map to a specific class.
6. Some templates are unused or missing inside the articles.

3.2. Innovation (Stage 2 and Stage III): Integrating Datasets in the Form of a Knowledge Graph

Ontology Definition and Data Mapping Schema. The ontology development was based on re-use of classes and properties from existing ontologies and vocabularies including Schema.org vocabulary[9], DBpedia Ontology[10], UMBEL (Upper Mapping and Binding Exchange Layer)[11], DICOM (Digital Imaging and Communications in Medicine)[12], and DrugBank. Each instances of the drug class has properties such as generic drug name, code, active substances, non-proprietary name, strength value, cost

[9] https://schema.org/

[10] https://wiki.dbpedia.org/services-resources/ontology

[11] http://umbel.org/

[12] https://www.dicomstandard.org/

per unit, manufacturer, related drug, description, URL, license, etc. Additionally, in order to align the drug data with generic drugs from DrugBank, properties brandName, genericName, atcCode, and dosageForm from the DrugBank Ontology were used. The relation *rdfs:seeAlso* can be used to annotate the links which the drug product entities will have to generic drug entities from the LOD Cloud dataset. The nodes are linked according to the relations these classes, tables, or groups have between them. There exist a few tools for ontology and vocabulary discovery, which should be used in this operation, such as Linked Open Vocabularies (LOV)[13] and DERI Vocabularies[14].

Data Conversion. *Create RDF dataset:* The previously mapped schema can produce an RDF graph by using RDF-extension of LODRefine tool. This step transforms raw data into RDF dataset based on a serialization format. The transformation process can be executed in many different ways and with various software tools, e.g., OpenRefine (which the authors used), RDF Mapping Language[15], and XLWrap[16] which is a Spreadsheet-to-RDF Wrapper, among others.

*Interlinking***.** LODRefine[17] was used for reconciliation in interlinking the data. In this case, columns *atcCode*, *genericName1*, *activeSubstance1*, *activeSubstance2* and *activeSubstance3* reconciled with DBpedia. This operation enables interoperability between organization data and the Web through establishing semantic links between the source dataset (organization data) with related datasets on the Web. Link discovery can be performed in manual, semi-automated, or fully-automated modes to help discover links between the source and target datasets. Since the manual mode is tedious, error-prone, and time-consuming, and the fully-automated mode is currently unavailable, the semi-automated mode is preferred and reliable. Link generation yields links in RDF format using *rdfs:seeAlso* or *owl:sameAs* predicates. The activities of link discovery and link generation are performed sequentially for each data source. The last activity within the interlinking stage is the generation of overall link statistics, which showcase the total number of links generated between the source and target data sources.

Storage and Publishing. OpenLink Virtuoso server (version 06.01.3127)[18] on Linux (x86_64-pc-Linux-gnu), Single Server Edition have been used to run the SPARQL endpoint queries: http://aldda.b1.finki.ukim.mk/sparql. RDF graph can be accessed on the following link: http://aldda.b1.finki.ukim.mk/. For publishing linked data on the Web, a linked data API is needed, which makes a connection with the database to answer specific queries. The HTTP endpoint is a webpage that forms the interface. A REST API is used to make a web application. It makes it possible to give the linked data back to the user in various formats, depending on the user's requirements. The linked data can be made visible in HTML on a website as HTTP links or as RDF data in a browser or a graphic visualization in a web application, which would be the most user-friendly.

[13] http://lov.okfn.org/
[14] http://datahub.io
[15] https://github.com/RMLio
[16] http://xlwrap.sourceforge.net/
[17] https://sourceforge.net/projects/lodrefine/
[18] https://github.com/openlink/virtuoso-opensource

3.3. Specific Tools Development and Validation (Stage II and Stage III)

Tools for Quality Assessment. In our approach [10], [11], quality assessment is an ongoing operation in all stages as the quality of the content of the document on the Web varies, see also Fig. 2. The authors strongly recommend assessing quality at every stage of the transformation process based on characteristics such as accuracy, consistency, and relevancy. Therefore, we have developed an evaluation scheme that addresses the DQ before starting data analytics. It is carried out by estimating the quality of data attributes or features by applying a dimension metric to measure the quality characterized by its accuracy, completeness, and consistency.

The expected result is DQ assessment suggestions indicating the quality constraints that will increase or decrease the DQ. The authors believe also that DQ must be handled in many other phases of the big data lifecycle. In our approach, we distinguish between quality on data level and quality on metadata level. The data pre-processing improves DQ by executing many tasks and activities such as data transformation, integration, fusion, and normalization.

Example. For every quality dimension, quantification and measurement are needed (see the discussion on dimensions in Section 3.1). Therefore, metrics have been defined and linked to particular dimensions. Usually, most metrics used for measuring DQ are within a range from 0 to 1, with 0 representing incorrect value and 1 representing a correct value. Dimensions such as accuracy, completeness, and consistency, among others, are calculated by the function $M\_D = 1 - (N_{iv}/N_{tv})$, where M_D is the metric for a given dimension, N_{iv} is the count of incorrect values, and N_{tv} is the total number of values for the dimension concerned. Regarding DQ dimensions relevant for quality assessment of Arabic DBpedia, we have identified three dimensions accuracy, consistency, and relevancy, as shown in Table 3.

Table 3. Data Quality dimensions relevant for quality assessment of Arabic DBpedia (*Specific to DBpedia, **Specific to Arabic DBpedia)

Category	Sub-category
Accuracy	• Incorrectly extracted triple • Special template not properly recognized* • Wrong values in numerical data (due to dual numbering used) **
	Incorrectly extracted data type
	Implicit relationship between attributes • One/ Several fact encoded in one/several attributes* • Attribute value computed from another attribute value**
Consistency	• Inconsistency in representation of number values**
Relevancy	Irrelevant information extracted • Extraction of attributes containing layout information** • Redundant attribute values • Image related information* • Other irrelevant information

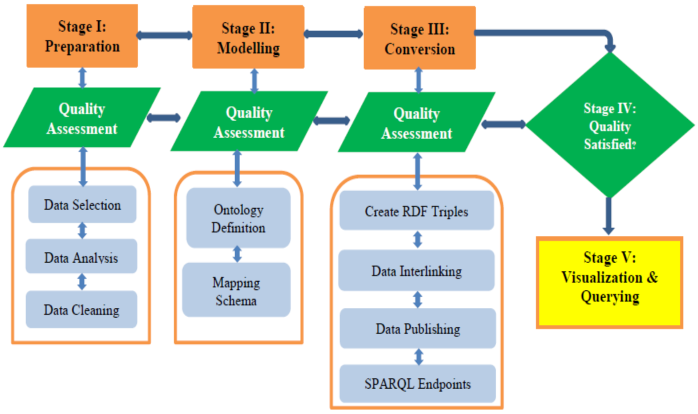


Fig. 2. A novel linked data methodology with a focus on quality assessment

Tool for Workflow Automation. The processing steps discussed so far refer to the initial load of the knowledge graph available online for experimental purposes at this location[19]. Currently, underway is testing of the solution and deployment of the adopted tools (LODRefine, OpenLink Virtuoso, PoolParty UnifiedViews for a client from Lybia. The PoolParty UnifiedViews (relevant for the speed layer in the Big Data architecture presented in Fig. 1) is considered for automation of the Extract-Transform-Load processes. The UnifiedViews's pipeline shall integrate also the custom quality assurance services discussed above.

3.4. Visualization and Querying (Stage IV)

After publishing the data on the Web in a form of a knowledge graph, it becomes available to other web applications for retrieval and visualization [12]. Using standard vocabularies for modeling allows end users to use different visualization approaches, e.g., freely available libraries can be used that offer diverse types of visualization, such as a table or in a diagram formatted in different ways as shown in Fig. 3. Custom visualization and query applications enable the user to interact with the data. In order to visualize the statistics about drug types and/or manufacturers, we use the exploratory spatial-temporal analysis (ESTA-LD) [12] tool[20]. The tool enables us to select the endpoint from where the data should be retrieved. This Section gives few examples of SPARQL queries that answer the business questions introduced in Section 1.

[19] http://aldda.b1.finki.ukim.mk/sparql, http://aldda.b1.finki.ukim.mk.

[20] http://geoknow.imp.bg.ac.rs/ESTA-LD

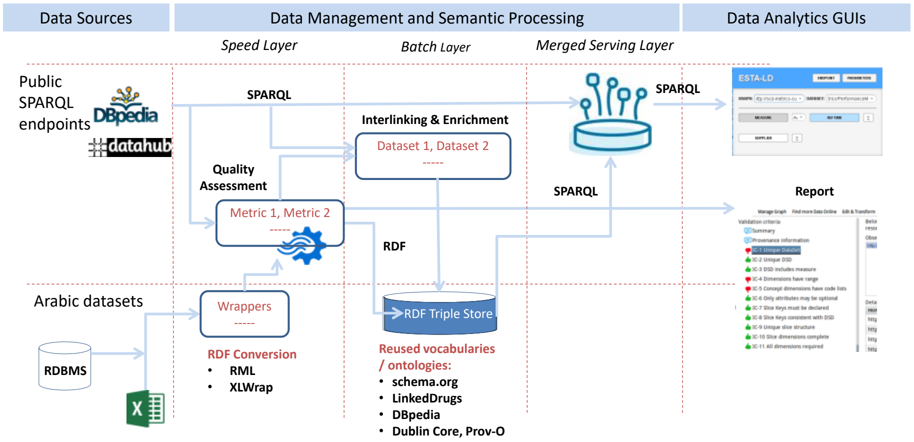


Fig. 3. Knowledge graph visualization and querying

Example. Query: Count all distinct drugs

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT count distinct ?drug
FROM <http://aldda.b1.finki.ukim.mk/lod/data/drugs>
WHERE
    { ?drug a <http://schema.org/Drug>  }
```

Output: 31906 distinct drugs

- Query: Count all interlinked drugs

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT count distinct ?drug
FROM <http://aldda.b1.finki.ukim.mk/lod/data/drugs>
WHERE
    { ?drug a <http://schema.org/Drug> .
     ?drug rdfs:seeAlso ?seeAlso }
```

Output: 23971 interlinked drugs

It is notable that >75% of the merged datasets are interlinked with DBpedia and can obtain additional information regarding drugs from DBpedia.

- Query: Extract abstract info from DBpedia in Arabic language for the ‘taxol’ which is an Organic composite similar to the ‘paclitaxel’ drug

```
prefix dbo: <http://dbpedia.org/ontology/>
prefix drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
SELECT * WHERE {
?drug a <http://schema.org/Drug>   .
?drug drugbank:genericName ?genericName .
?drug rdfs:seeAlso ?seeAlso .
{ SERVICE<http://dbpedia.org/sparql>
{ ?seeAlso dbo:abstract ?abstract  } }
FILTER (?genericName = ‘paclitaxel’)
FILTER (langMatches(lang(?abstract), "ar")) }
```

- Output

```
محضر من لحاء ، وهو مركب taxol في 1988 توصل الباحثون في جامعة جونز هوبكنز إلى أن تاكسول"
بسرطان حاد في المبيض. كما اقترح الباحثون شجر الطقسوس بالمحيط الهادي ، يمكن أن يفيد النساء المصابات
للسرطان في هيوسطن أن مادة تاكسول يمكن أن تفيد السيدات المصابات          سنة 1991 في مركز أندرسون
الثدي أيضاً. في دراسات تمت على 25 سيدة مصابة بسرطان متقدم في الثدي ولمتتمكن من الاستجابة  بسرطان
@ar".بعد تسع شهور من العلاج التجريبي للعلاج الكيميائي ، شعر غالبية السيدات بانكماش الورم
```

- Query: Equivalent drugs comparison

```
    prefix dbo: <http://dbpedia.org/ontology/>
    prefix drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
    prefix schema:<http://schema.org/>
    prefix dbp: <http://dbpedia.org/ontology/>
    SELECT distinct ?drug1, ?drug1GenericName, ?drug1ManufacturerLegalName,
?drug1ActiveIngredient, CONCAT(str(?drug1CostPerUnit),' ',?drug1CostCurrency) as
?drug1CostFull, ?drug1AddressCountry,
    ?drug2,?drug2GenericName, ?drug2ManufacturerLegalName, ?drug2ActiveIngredient,
CONCAT(str(?drug2CostPerUnit),' ',?drug2CostCurrency) as ?drug2CostFull,
?drug2AddressCountry WHERE {
    ?drug a <http://schema.org/Drug> .
    ?drug drugbank:genericName ?drug1GenericName .
    ?drug schema:addressCountry ?drug1AddressCountry .
    ?drug schema:cost ?drug1Cost .
    ?drug schema:manufacturer ?drug1Manufacturer .
    ?drug1Manufacturer schema:legalName ?drug1ManufacturerLegalName .
    ?drug schema:activeIngredient ?drug1ActiveIngredient .
    ?drug1Cost schema:costPerUnit ?drug1CostPerUnit .
    ?drug1Cost schema:costCurrency ?drug1CostCurrency .
    ?drug rdfs:seeAlso ?seeAlso .
    ?drug2 rdfs:seeAlso ?seeAlso .
    ?drug2 drugbank:genericName ?drug2GenericName .
    ?drug2 schema:addressCountry ?drug2AddressCountry .
    ?drug2 schema:cost ?drug2Cost .
    ?drug2 schema:manufacturer ?drug2Manufacturer .
    ?drug2Manufacturer schema:legalName ?drug2ManufacturerLegalName .
    ?drug2 schema:activeIngredient ?drug2ActiveIngredient .
    ?drug2Cost schema:costPerUnit ?drug2CostPerUnit .
    ?drug2Cost schema:costCurrency ?drug2CostCurrency .
    FILTER (?drug != ?drug2)}
```

- Output:

	Drug1	Drug2
Drug Number	aldda.b1.finki.ukim.mk/lod/data/drugs#35704	aldda.b1.finki.ukim.mk/lod/data/drugs#36482
GenericName	glimepiride	metformin and sulfonamides
ManufacturerLegalName	Sadco	Benta Trading Co s.a.l.
ActiveIngredient	Glimepiride	Metformin HCl
CostFull	12415.0 L.L	28800.0 L.L
AddressCountry	LB	LB

- Query: Drugs with different brand name comparison.

```
    prefix dbo: <http://dbpedia.org/ontology/>
    prefix drugbank: <http://www4.wiwiss.fu-berlin.de/drugbank/resource/drugbank/>
    prefix schema:<http://schema.org/>
    prefix dbp: <http://dbpedia.org/ontology/>
    SELECT   ?drug1BrandName,?drug1GenericName,          ?drug1ManufacturerLegalName,
?drug1ActiveIngredient,              ?drug1DosageForm,       CONCAT(str(?drug1CostPerUnit),'
',?drug1CostCurrency) as ?drug1CostFull, ?drug1AddressCountry,
    ?drug2BrandName,?drug2GenericName,                       ?drug2ManufacturerLegalName,
?drug2ActiveIngredient,              ?drug2DosageForm,       CONCAT(str(?drug2CostPerUnit),'
',?drug2CostCurrency) as ?drug2CostFull, ?drug2AddressCountry WHERE {
    ?drug a <http://schema.org/Drug> .
    ?drug drugbank:brandName ?drug1BrandName .
    ?drug drugbank:genericName ?drug1GenericName .
    ?drug schema:addressCountry ?drug1AddressCountry .
    ?drug schema:cost ?drug1Cost .
    ?drug schema:manufacturer ?drug1Manufacturer .
    ?drug1Manufacturer schema:legalName ?drug1ManufacturerLegalName .
    OPTIONAL {
    ?drug drugbank:dosageForm ?drug1DosageForm }
    ?drug schema:activeIngredient ?drug1ActiveIngredient .
    ?drug1Cost schema:costPerUnit ?drug1CostPerUnit .
    ?drug1Cost schema:costCurrency ?drug1CostCurrency .
    ?drug rdfs:seeAlso ?seeAlso .
    ?drug2 rdfs:seeAlso ?seeAlso .
    ?drug2 drugbank:brandName ?drug2BrandName .
    ?drug2 drugbank:genericName ?drug2GenericName .
    ?drug2 schema:addressCountry ?drug2AddressCountry .
    ?drug2 schema:cost ?drug2Cost .
    ?drug2 schema:manufacturer ?drug2Manufacturer .
    ?drug2Manufacturer schema:legalName ?drug2ManufacturerLegalName .
    ?drug2 schema:activeIngredient ?drug2ActiveIngredient .
    OPTIONAL { ?drug2 schema:availableStrength ?drug2Strength .}
    OPTIONAL {?drug2 drugbank:dosageForm ?drug2DosageForm }
    ?drug2Cost schema:costPerUnit ?drug2CostPerUnit .
    ?drug2Cost schema:costCurrency ?drug2CostCurrency .
    FILTER (?drug1BrandName != ?drug2BrandName &&
    ?drug1DosageForm != ?drug2DosageForm &&
    ?drug1ManufacturerLegalName
    !=drug2ManufacturerLegalName)}
```

- Output

	Drug1	Drug2
BrandName	EBETREXAT	METOJECT
GenericName	methotrexate	methotrexate
ManufacturerLegalName	Codipha	Alfamed S.A.L.
ActiveIngredient	methotrexate	methotrexate
DosageForm	7.5mg/0.75ml	15mg/0.3ml
CostFull	32984.0 L.L	51182.0 L.L
AddressCountry	LB	LB

4. Discussion of Results: Analysis of Linked Data Methodologies

In literature, not many papers have dealt with linked data methodologies i.e., the process of generating, linking, publishing, and using linked data; to name a few: *W3C Best Practices for Publishing Linked Data* (W3C-Government Linked Data Working Group, 2014)[21] [13]; *A Cookbook for Publishing Linked Government Data on the Web* [14]; *Linked Data Life Cycles* [15]; *Guidelines for Publishing Government Linked Data* [16]; *Managing the Life-Cycle of Linked Data with the LOD2 Stack* [6]; and *Methodological Guidelines for Consolidating Drug Data* [17]; see Table 4 for a comparison. One of the first linked data methodologies was developed in the European research project LOD2 (Creating Knowledge out of interlinked Data, 2011-2014)[22] that was mainly dedicated to the publishing process, i.e., opening data in a machine-readable format and establishing the tools and technologies for interlinking and integrating heterogeneous data sources in general.

Jovanovik and Trajanov [17] concluded that "the LOD2 methodology which provides software tools for the denoted steps still misses some key elements of the linked data lifecycle, such as the data modeling, the definition of the URI format for the entities and the ways of publishing the generated dataset". They also stated, "The LOD2 tools are general, and cannot be applied in a specific domain without further work and domain knowledge." (page 4). Therefore, they proposed a new linked data methodology with a focus on reuse. It provides guidelines for data publishers defining reusable components in the form of tools, schemas, and services for the given domain (i.e., drug management).

The methodology presented in this paper meets the needs of the case study. Hence we suggest an approach to standardize the quality assessment of Linked Data lifecycle as is presented in Table 5.

[21] W3C Best Practices for Publishing Linked Data. http://www.w3.org/TR/ld-bp/ (2018)

[22] https://linkeddata.rs/project/LOD22010–2014

Table 4. Comparison of previous methodologies

Authors	Title / Steps	
W3C Government Linked Data Working Group (2014)	Best Practices for Publishing Linked Data:	
	(1) Prepare stakeholders, (2) Select a dataset, (3) Model the data, (4) Specify an appropriate license, (5) Good URIs for linked data, (6) Use standard vocabularies,	Initialization
	(7) Convert data, (8) Provide machine access to data,	Innovation
	(9) Announce new data sets, (10) Recognize the social contract	Validation & Maintenance
Hyland et al. (2011)	A Cookbook for Publishing Linked Government Data on the Web:	
	(1) Identify, (2) Model, (3) Name, (4) Describe,	Initialization
	(5) Convert, (6) Publish,	Innovation
	(7) Maintain	Validation & Maintenance
Hausenblas et al. (2016)	Linked Data Life Cycles:	
	(1) Data awareness, (2) Modeling,	Initialization
	(3) Publishing, (4) Discovery, (5) Integration,	Innovation
	(6) Use-cases	Validation & Maintenance
Villazón-Terrazas et al. (2011)	Guidelines for Publishing Government Linked Data:	
	(1) Specify, (2) Model,	Initialization
	(3) Generate, (4) Publish,	Innovation
	(5) Exploit	Validation & Maintenance
Auer, et all. (2012)	Managing the Life-Cycle of Linked Data with the LOD2 Stack:	
	(1) Extraction,	Initialization
	(2) Storage, (3) Authoring, (4) Interlinking, (5) Classification,	Innovation
	(6) Quality, (7) Evolution/Repair, (8) Search/ Browsing/ Exploration	Validation & Maintenance
Jovanovik and Trajanov (2017)	Methodological guidelines for consolidating drug data:	
	(1) Domain and Data Knowledge, (2) Data Modeling and Alignment,	Initialization
	(3) Transformation into 5-star Linked Data, (4) Publishing the Linked Data Dataset on the Web,	Innovation
	(5) Use-cases, Applications and Services	Validation & Maintenance

Table 5. The proposed methodologies

Guma Lakshen	Methodological guidelines for quality assessment of Linked Data:	
	(I) (1)Data Selection, (2) Data Analysis and (3) Data Cleaning,	Initialization
	(II) (4) Ontology Definition, (5) Mapping Scheme taking into consideration Quality metrics, (III) (6) Conversion into 5-star Linked Data taking into consideration the specific requirements of the Arabic language, and (7) Interlinking, (8) Publishing the Linked Data Dataset on the Web,	Innovation
	(IV) (9) Quality Assessment, (V) (10) Use-cases, Applications and Services.	Validation & Maintenance

4.1. Analysis of the Knowledge Graph Model and Transformation Tools

In this study, the authors decided to use the RDF, because it is recommended by W3C, and it has advantages, such as an extensible schema, self-describing data, dereferenceable URIs. Further on, since RDF links are typed, it enables good structure, interoperability, and safely linking different datasets.

Before converting XLS data to RDF, the authors selected the target ontology to describe the drugs contained in the drug availability dataset. Authors selected the LinkedDrugs[23] ontology [17], Schema.org vocabulary, and DBpedia, as they have the needed properties and provide easier interlinking possibilities for further transformation. The Web Ontology Language allows complex logical reasoning and consistency checking for RDF and OWL resources. These reasoning capabilities helped the authors to harmonize the heterogeneous data structures found in the input datasets.

Web drug data availability in some Arabic countries is basically public as a two-star format data, i.e., PDF or XLS format. Most of the available drug data is provided in the English language with a few columns in Arabic, this is because English is widely used among physicians and pharmacists; it is the predominant language in their communications.

Following the authors' proposal described above, the authors transformed the selected drug data into five-star linked open data and established relations in the RDF knowledge graph (31,906 drugs, more than 300 000 triples) toward outside entities, including the DBpedia and DrugBank. The *owl:sameAs* relation allows interlinking related drug descriptions that refer to the same real-world entity. For research purposes, the knowledge graph has been published via the SPARQL endpoint available at http://aldda.b1.finki.ukim.mk/sparql.

4.2. Quality Analysis of Integrated Open Data

Many authors have pointed out issues such as the completeness, conciseness, and consistency of open data. In 2014, Kontostas et al. [18] provided several automatic quality tests on LOD datasets based on patterns modeling various error cases, and they detected 63 million errors among 817 million triples. At the same time, Zaveri et al. [19, 20], conducted a user-driven quality evaluation which stated that DBpedia indeed has quality problems (e.g., around 12% of the evaluated triples had issues). They can be summarized as incorrect or missing values, incorrect data types, and incorrect links. Based on the survey, the authors developed a comprehensive quality assessment framework based on 18 quality dimensions and 69 metrics. Based on the work of Zaveri et al. [120], and the ISO 25012 DQ model, Radulović et al. [21] developed a linked data quality model and tested the model with DBpedia with a special focus on accessibility quality characteristics.

Based on the analysis of quality issues with DBpedia and the problems identified, the authors conclude that most important dimensions to be taken into consideration are the following.

[23] http://linkeddata.finki.ukim.mk/sparql

- Accuracy: triple incorrectly extracted, data type problems, errors in the implicit relationship between attributes.
- Consistency: representation of numerical values.
- Relevancy: irrelevant information extracted.

Different metrics were further defined, and web services were implemented to be used for data curation.

4.3. Proposal for Further Development of Quality Assessment Tools

There were several attempts in the past to design and implement a generic tool for linked data quality assessment. One of the first open-source frameworks for flexibly expressing quality assessment methods, as well as fusion methods, was Sieve (http://sieve.wbsg.de) Mendes et al. [22], released in 2012. As part of the Linked Data Integration Framework (LDIF; http://sieve.wbsg.de/), Sieve supports users in accessing data from the LOD cloud. Taking into consideration that DBpedia is a core element in the LOD cloud, in 2014, Kontokostas et al. enabled the RDFUnit Testing Suite (https://github.com/AKSW/RDFUnit) to run automatically-generated (i.e., based on a schema) and manually-generated test cases against an endpoint, e.g., the DBpedia SPARQL endpoint. Recognizing the large variety of DQ dimensions and measures, Luzzu (https://github.com/EIS-Bonn/Luzzu) [23], was developed at the same time to allow knowledgeable engineers without Java expertise to create quality metrics in a declarative manner. LOD Laundromat (http://lodlaundromat.org) was designed to help crawl the LOD cloud, converting all its contents in a standards-compliant way (i.e., gzipped N-Triples), as well as removing all data stains, such as syntax errors, duplicates, and blank nodes. TripleCheckMate (https://github.com/AKSW/TripleCheckMate) is a tool for crowdsourcing the assessment of Linked Open Data. It was developed for evaluating the correctness of DBpedia. TripleCheckMate provides an easy user interface with multiple resource assignment methods and a ready-to-use error classification scheme. The quality assessment methods implemented in these tools can be grouped into automatic, semi-automatic, manual, or crowd-sourced approaches. Initial results of the analysis and a comparison of the selected tools are provided in Table 6. However, as these tools have not been tested with the Arabic datasets yet, the quality assessment operations needed in our case study were implemented with custom web services.

Table 6. Comparison of open source quality assessment tools according to several attributes

Tool	Extensibility	Last Update	Collaboration	Cleaning
RDFUnit	SPARQL	03/2018	✕	✕
Luzzu	JAVA, LQML	07/2017	✕	✕
TripleCheck Mate	✕	03/017	✔	✕
Laundromat	SPARQL	05/2018	✔	✔
Sieve	XML	2014	✕	✔

5. Concluding Remarks

Most of the available drug datasets nowadays are still provided in 2-star format and in English language due to the fact that the English language is widespread among physicians and pharmacists and also a predominant language in communications between physicians and pharmacists. In order to showcase the possibilities for large-scale integration of drug data, the authors proposed a piloting methodology and tested the approach with datasets from Arabic countries. The authors presented the transformation process of 2-star drug data into a 5-star Linked Open Data with DrugBank and DBpedia. The data is open for research purposes, while the OpenLink Virtuoso server (version 06.01.3127) on Linux (x86_64-pc-linux-gnu), Single Server Edition has been used to run the SPARQL endpoint (see http://aldda.b1.finki.ukim.mk/sparql).

The paper showcases the benefits from the Linked Data approach, in particular the possibility of enriching the private datasets with selected open data such as DBpedia. Main conclusion is that the Linked Data approach (1) contributes to the standardization on the metadata level and the semantic interoperability; (2) opens possibilities for improving the existing business value chain and insights by integration of valuable free information. However the quality issues in the Big Data ecosystems, Linked Drug Data in particular are still wide open for further study and evaluation, especially in the Arab countries.

The main research goal was to identify, collect, analyze, and evaluate the quality of selected drugs data sets, to allow quantifying and improving their value for the benefit of the user's especially with deficiencies in English language. The main contributions can be summarized as follows:

– This work introduced a modified process model based on previous methodologies shown in table 3 above.

– It is recommended to implement custom quality assessment services for transformation and processing in order to ensure that the process is conducted in high quality manner.

– For the first time, the paper discusses the issues with drug data from Arabic countries based on the selected four drug data files from four different Arabic countries, Iraq, Syria, Saudi Arabia, and Lebanon).

Taking into consideration the issues identified with quality of the open data (in particular, the issues with drug data from Arabic countries), the future work will include implementation of additional services for repairing the errors observed in the Arabic Linked Drug dataset.

Acknowledgment. The research leading to these results has received funding from the European Community’s H2020 Programme under grant agreement no 809965 (LAMBDA – Learning, Applying, Multiplying Big Data Analytics) and from the Ministry of Science and Technological Development of Republic of Serbia.

References

1. Patrizio, A.: IDC: Expect 175 zettabytes of data worldwide by 2025, Network World, Network World. https://www.networkworld.com/article/3325397/idc-expect-175-zettabytes-of-data-worldwide-by-2025.html (2018)
2. Aljazzaf, Z. M.: Modelling and measuring the quality of online services, Kuwait J. Sci. 42 (3), 134-157. (2015)
3. Kern, R., Kozierkiewicz, A., Pietranik, M.: The data richness estimation framework for federated data warehouse integration. Information Sciences, Volume 513, 2020, pp. 397-411. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2019.10.046 (2020)
4. Sawadogo, P., Darmont, J.: On data lake architectures and metadata management. Journal of Intelligent Information Systems. DOI: https://doi.org/10.1007/s10844-020-00608-7
5. Mami, M.N., Graux, D., Scerri, S., Jabeen, H., Auer, S., Lehmann, S.: Uniform Access to Multiform Data Lakes using SemanticTechnologies. Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services, December 2019, pp. 313–322. DOI: https://doi.org/10.1145/3366030.3366054 (2019)
6. Auer, S., A., Lorenz, B., Dirschl, C., Erling, O., Hausenblas, M., Isele, R., Lehmann, J., Martin, M., Mendes, P.N., Nuffelen, P.v., Stadler, C., Tramp, S. & Williams, H.: Managing the Life-Cycle of Linked Data with the LOD2 Stack. The Semantic Web-ISWC 2012. Boston: Springer Berlin Heidelberg: 1–16. (2012)
7. Berners-Lee, T.: Design issues: Linked data. http://www.w3.org/DesignIssues/LinkedData.html (2006)
8. Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z. (2007) DBpedia: A Nucleus for a Web of Open Data. In: Aberer K. et al. (eds) The Semantic Web. ISWC 2007, ASWC 2007. Lecture Notes in Computer Science, vol 4825. Springer, Berlin, Heidelberg.
9. Wishart DS, Knox C, Guo AC, Shrivastava S, Hassanali M, Stothard P, Chang Z, Woolsey J. DrugBank: a comprehensive resource for in silico drug discovery and exploration. Nucleic Acids Res. 2006 Jan 1;34(Database issue):D668-72. DrugBank Release Version 5.1.7, https://www.drugbank.ca/releases/latest#open-data
10. Lackshen, G., Janev, V., Vraneš, S.: Quality Assessment of Arabic DBpedia. In R. Akerkar et all (Eds.) Proc. of 8th International Conference on Web Intelligence, Mining and Semantics. June 25 – 27. 2018, Novi Sad, Serbia. ACM New York, NY, USA. DOI: https://doi.org/10.1145/3227609.3227675 (2018)
11. Lackshen, G., Janev, V., Vraneš, S.: Linking Open Drug Data: Lessons Learned. In K. Saeed et al. (Eds.): CISIM 2019, LNCS 11703, pp. 1–12, 2019. Springer International Publishing. https://doi.org/10.1007/978-3-030-28957-7 (2019)
12. Mijović, V. Janev, V., Paunović, D., Vraneš, S.: Exploratory Spatio-Temporal Analysis of Linked Statistical Data, Journal of Web Semantics, Web Semantics: Science, Services and Agents on the World Wide Web 41C (2016), pp. 1-8. ISSN: 15708268. DOI: https://doi.org/10.1016/j.websem.2016.10.002 (2016)

13. Janev, V., Mijović, V., Vraneš, V.: Using the Linked Data Approach in European e-Government Systems. International Journal on Semantic Web and Information Systems 14(2):27-46, April 2018. DOI: https://doi.org/10.4018/IJSWIS.2018040102 (2018).
14. Hyland, B. & Wood, D.: The Joy of Data: A Cookbook for Publishing Linked Government Data on the Web. In: Linking Government Data, New York: Springer New York: 3–26. (2016)
15. Hausenblas, M.: Linked Data Life Cycles. http://www.slideshare.net/mediasemanticweb/linked-data-life-cycles (2016)
16. Villaz´on-Terrazas, B. Vilches-Bl´azquez, L. Corcho, O. & G´omez-P´erez, A.: Methodological Guidelines for Publishing Government Linked Data Linking Government Data. In Linking Government Data. Springer NewYork, New York, NY. chapter 2: 27–49. (2011)
17. Jovanovik, M. & Trajanov, D.: Consolidating drug data on a global scale using linked data. Journal of Biomedical Semantics, 8(3). (2017)
18. Kontokostas, D. Westphal, P. Auer, S. Hellmann, S. Lehmann, J. & Cornelissen, R.: Test driven Evaluation of Linked Data Quality. In Proceeding of the 23rd International Conference on World Wide Web, pp. 747-758. New York, NY, USA. DOI: http://dx.doi.org/10.1145/2566486.2568002 (2014)
19. Zaveri, A. Kontokostas, D. Sherif, M.A. Bühmann, L. Morsey, M. Auer, S. & Lehmann, J. : User-driven Quality Evaluation of DBpedia. In Proceedings of the 9th International Conference on Semantic Systems. New York. USA:97–104. (2013)
20. Zaveri, A. Rula, A. Maurino, R. Pietrobon, R. Lehmann, J. & Auer, S.: Quality assessment for linked data: A survey. Semantic Web– Interoperability, Usability, Applicability. (2016)
21. Radulovic, F. Mihindukulasooriya, N. García-Castro, R. & Gómez-Pérez, A.: A Comprehensive Quality Model for Linked Data. Semantic Web – Interoperability, Usability, Applicability, Vol. 9, No. 1. Special issue on Quality Management of Semantic Web Assets (Data, Services and Systems), pp: 3-24. DOI: https://doi.org/10.3233/SW-170267 (2018)
22. Mendes, P.N. Mühleisen, H. & Bizer, C.: Sieve: Linked Data Quality Assessment and Fusion. In Proceedings of the 2012 Joint EDBT/ICDT Workshops, pp: 116-123. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/2320765.2320803 (2012)
23. Debattista, J. Auer, S & Lange, C.: Luzzu -- A Framework for Linked Data Quality Assessment. In Proceeding of the 2016 IEEE Tenth International Conference on Semantic Computing (ICSC), pp: 124-131. Laguna Hills, CA, 2016. IEEE. DOI: https://doi.org/10.1109/ICSC.2016.48 (2016)

Guma Lackshen is a PhD student at the School of Electrical Engineering. His interest includes NLP techniques, Linked Data, Big Data tools and technologies, and quality assurance in IT systems.

Valentina Janev is a Senior Researcher at the Mihajlo Pupin Institute, University of Belgrade, Serbia. Her interest includes business intelligence, decision support systems, Linked Data and Big Data tools, and applications of semantic technologies and W3C standards in different industrial domains. She has participated in many information systems projects for clients in Serbia and the region, as well as national and EU research projects. She serves as a Coordinator of the EU project LAMBDA, https://project-lambda.org/.

Sanja Vraneš, PhD is jointly appointed as the Director General of the Institute Mihajlo Pupin and as a Full Professor of Computer Science at the University of Belgrade. Her research interests include artificial intelligence, semantic web, linked data web, knowledge management, decision support systems, etc. From 1999 she has been engaged as a United Nations Expert for information technologies, and from 2005 as an expert evaluator and reviewer of EC Framework Programme Projects and H2020 projects.

Received: May 10, 2020; Accepted: November 01, 2020.

Computer Science and Information Systems 18(3):749–769 https://doi.org/10.2298/CSIS200326044R

A Multicriteria Optimization Approach for the Stock Market Feature Selection

Dragana Radojičić[1], Nina Radojičić[2] and Simeon Kredatus[1]

[1] TU Wien, Institute of Statistics and Mathematical Methods in Economics
Wiedner Hauptstr. 8/E105-01-05 FAM
1040 Vienna, Austria
{gagaradojicic, simeon.kredatus}@gmail.com
[2] Faculty of Mathematics, University of Belgrade
Studentski trg 16, 11000 Belgrade, Serbia
nina@matf.bg.ac.rs

Abstract. This paper studies the informativeness of features extracted from a limit order book data, to classify market data vector into the label (buy/idle) by using the Long short-term memory (LSTM) network. New technical indicators based on the support/resistance zones are introduced to enrich the set of features. We evaluate whether the performance of the LSTM network model is improved when we select features with respect to the newly proposed methods. Moreover, we employ multicriteria optimization to perform adequate feature selection among the proposed approaches, with respect to precision, recall, and F_β score. Seven variations of approaches to select features are proposed and the best is selected by incorporation of multicriteria optimization.

Keywords: Limit order book, multicriteria optimization, time-series, feature selection, machine learning.

1. Introduction

The stock market has been the subject of studies of many research projects and publications, as well as of many financial institutions.

It has already been noted in the literature that the financial markets hold memory properties, see [48], [11], [38]. A wide range of research is done analyzing the financial data and exploring adequate mathematical models for modeling financial phenomena. In [32], the optimal strategy and the dynamic volatility derivatives pricing in an incomplete financial market were established using the expected utility maximization formulation. On the other hand, the authors in [21] proposed a smart market model that reduces peak-demand charges and has a positive final profit in energy systems.

Since there are more and more data on stock markets nowadays, there is a potential in developing algorithms to analyze historical data and take advantage of it. Nowadays, most exchanges have been automatized, because more trades are observed via automated electronic markets (see [6]). Therefore, most exchanges have switched from the traditional stock exchange, which had a physical location, to the electronic stock exchanges.

The electronic marketplace mechanism is comprehensively described in [7].

Various Artificial Intelligence approaches have been used for, usually very challenging, financial forecasting, which has been attracting many researchers. The authors in [37] provided a review on various techniques that perform well in the forecasting of stock markets, as well as they proposed an approach through data discretization by fuzzistics and rule generation by rough set theory.

The authors in [2] investigated the role of sample size and class distribution in credit risk assessments and their results indicate that various factors play a role in determining the optimal class distribution (the performance measure, classification algorithm and data set characteristics). Moreover, seven different machine and deep learning algorithms were proposed in [17] to learn the inherent patterns and predicting future movements in the stock market. Also, in [9] authors have used the weighted support vector machine (WSVM) method above the principal components to predict the stock trading signals. In [43] authors analyzed how Window Size influences the performance of future price direction prediction models that use technical indicators as input. The description of the trading strategy evolution and the strategy valuation is interpreted in [39].

In this research, we tend to cross compare the performance of the Long short-term memory (LSTM) network (initially introduced in [23]) when the model is fed with different groups of features extracted from the market limit order book (LOB) data. The research presented in this paper is unique in the sense that we used the customized data transformation, included the resistance support features and multicriteria optimization is used to select the best option among seven different approaches proposed to perform feature selection. Therefore, the presented research is not comparable to any other work available in the literature. But note that in [41] the authors extracted the Fourier transformation based features from the LOB and evaluated the performance of the GRU based model when fed with different groups of extracted features. Further, note that multicriteria optimization approaches for choosing the set of features for LSTM network models have not been considered in the literature up until now.

The LSTM network model is employed within this research in order to estimate if it is a good time to open a trade or it is better to idle. During the data transformation part, a huge amount of features are extracted from the limit order book and dealing with a large amount of various types of data has become an important challenge in computer science in recent years (see [47] and [27]). These extracted features describe market behavior. In this research, seven groups of features are chosen from the set of all features by employing proposed selection criteria. To evaluate the performance of the proposed LSTM network model when fed with the different groups of features several relevant metrics can be used.

In order to be able to better express and make objective decisions about the efficiency of the proposed method for feature selection, we incorporated multicriteria optimization, which has become used with its software support in various applications (see [16]). Besides the fact that choosing the most reliable set of features is multi-layered problems, the trades are occurring very fast and decisions need to be made quickly. Therefore, an intelligent approach for selection is necessary to make quick and precise decisions in order to capture opportunities. Further, multicriteria optimization enables to improve the quality of decisions, and it allows the decision-maker to point out the preference of one alternative over the other ones compared with several arguments. Therefore, to make a reliable financial decision and to evaluate opportunities and investments, financial intelligence needs to be utilized, see [22] and [8].

We start our paper with a brief introduction of the theory that stands behind the LOB concept, which is important for the research presented in this paper. After that, in Section 2 we describe the dataset which is used for our research. Then, new features based on the support/resistance zones are introduced in Section 3. In Section 4 we introduce criterion with the respect which we group features. Section 5 contains a multicriteria approach needed in this research and basics on the used method. Moreover, the results are presented in Section 6. Final conclusions and prospective work are given in Section 7.

1.1. A short review of a mathematical description of the Limit Order Book

The main job of the limit order book is to record all incoming and outgoing orders. The Limit Order Book is defined on a discrete price grid, where every grid's point models a price level, see Fig. 1. The minimum distance between two price levels is called a *tick*. On the ASK side the orders that need to be sold are placed, while on the BID side the orders that need to be bought are placed. The minimum price at the moment t on the ASK side, denoted by P_t^a, is named the best ask price, while the maximum price at the moment t on the BID side, denoted by P_t^b, is called the best bid price.

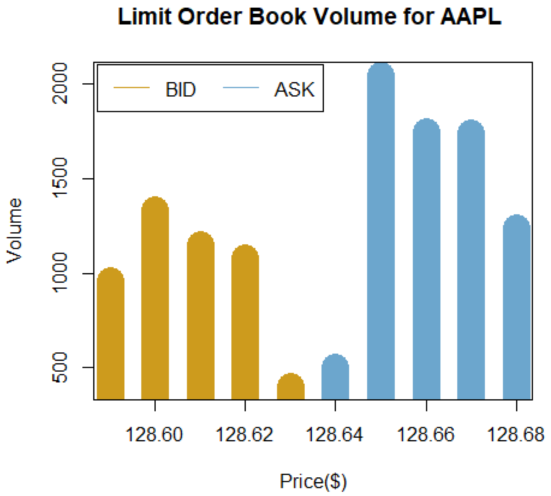


Fig. 1. Snapshot of the NASDAQ limit order book for AAPL stock symbol for 5 levels at 09:28:47am (on the 20th Februar 2015). On the bid/ask side are placed outstanding buy/sell orders (gold/blue), and the best bid price is $\$128.63$ with volume of 500 shares, while the best ask price is $\$128.64$ with volume of 600 shares.

The difference between the best ask and the best bid price is called the quoted spread, i.e.:

$$QuotedSpread = P_t^b - P_t^a.$$

The mid-price is defined as a arithmetic average of the best bid and best ask price, i.e.:

$$MidPrice = \frac{1}{2}(P_t^b + P_t^a).$$

An detailed mathematical explanation of the limit order book (LOB) can be found in Section II in [19].

The study of the limit order book dynamic is crucial for the financial world. The modeling of the next price-flip event in limit order book using the ability of the RNN is presented in [15]. In [12] a continuous-time stochastic model for the limit order book dynamics, which extracts key empirical properties of the LOB is established. Considering the volume of the LOB at different distances to the best ask price authors in [1] derived the mid price diffusion limit. In [24] authors introduce a two-sided limit order book stochastic model and prove the limit theorem when the tick size is converging to zero. A framework for manipulation in a computational model of a limit-order book, which emphasis information asymmetry between buy and sell makes profitable manipulation possible, was presented in [46].

Our research is based on the real market data from the past, more precisely on the high-quality online limit order book data tool Lobster[3], which replicates the entire NASDAQ Stock Market (the second largest exchange in the world) from the 27th of June 2007 up to the two days ago from the current day. Although in [25] the LOBSTER reconstruction approach is presented, in order to extract essential structures from the limit order book for our research we have used our customized data processing reconstructor which is presented in the next section. Moreover, since we are working with a huge data set, to process data we have used a processing engine supported by an advanced technology, which is explained in the [40].

2. The market data summary and data processing reconstructor

For each trading day and for each selected ticker (e.g. AAPL which stands for Apple, MSFT which stands for Microsoft Corporation, etc.) LOBSTER has 'orderbook' and 'message' file. The 'orderbook' file has information on the Price and Volume up to the requested number of levels. The 'message' file contains the following columns: time (represents the time when an event has occurred), type of the event (submission of a new order, cancellations, deletions of an order, execution of a visible order, execution of a hidden order, trading halt), order ID, size of the order, price, direction of the trade (sell/buy).

We define a market data vector x_t at a time point t, which contains market data features, i.e.:

$x_t = (bidLevel_1, bidVolume_1, askLevel_1, askVolume_1, bidLevel_2, bidVolume_2,$ $askLevel_2, askVolume_2, \ldots, bidLevel_n, bidVolume_n, askLevel_n, askVolume_n,$ $Time, EventType, OrderId, Size, Price, Direction)$.

Note that the limit order book is continuously evolving, so the vector x_{t+1} is derived from the type of the event that is compressed in the vector x_t.

Stock markets are producing a huge amount of data and there are over a million events for each stock symbol data in one trading day. Therefore, to avoid working with a vast amount of data, and in order to extract relevant features from the limit order book, we employ the data aggregation with respect to the 3min interval. During the 3min interval aggregation, more features are extracted, such as a number of canceled limit orders, a number of executed limit orders, open price (price at the beginning of the interval), close price, maximum price (during that interval), etc.

[3] Lobster academic research data. https://lobsterdata.com.

Furthermore, we compute the set of standardly known technical indicators in order to get more features describing market behavior. Since, there are various factors that have an influence on the stock market, using adequate technical indicators is important in achieving good forecasting results, see for example [49]. For a comprehensive description of the technical indicators and their use see [34], [10] and Table 2 in [49]. Note that these technical indicators are computed using an open source library ta-lib[4]. At this stage, for each timestamp t the data vector $\tilde{x}_t$ (enhanced vector x_t) contains features extracted from the LOB shape (e.g. Volume at each price level, number of executed trades, Open Price, Maximum Price, etc.) and technical analysis indicators (e.g. Bollinger Bands, moving average). Now for a particular trading day d, we have a dataset $\mathcal{D}_d = \{\tilde{x}_t | 0 \leq t \leq \lfloor \frac{duration\ of\ a\ trading\ day\ d\ in\ s}{180s} \rfloor\}$.

The goal is to classify vector of market data $\tilde{x}_t$ into the one of the labels from the set $S_+ = \{buy, idle\}$, if we consider semi-strategy emitting buy signals, or from the set $S_- = \{sell, idle\}$, if we consider semi-strategy emitting sell signals. Considering a semi strategy emitting buy signals, we examine whether from given vector $\tilde{x}_t$ every subsequent vector $\tilde{x}_{t+1}$, reach certain profit until the end of the day, with only exposing ourselves to a certain risk. Thus, the main idea is to see if each point reaches the desired risk-reward ratio (RRR), and to assign the label to each vector with respect to the Boolean function: 1 if the buy order should be issued, and 0 if it is better to idle. The particular algorithm is presented in the Appendix (Algorithm 1). Note that it is based on the same idea as Algorithm 1 in [40] and Algorithm 2 in [41]. In order to label our training set for this research, we have run the Algorithm 1 with the following values: $REWARD = 0.08$ and $RISK = 0.04$. The aforementioned data transformations extract features of interest and prepare data set as needed of our research. Note that [40] contains plain, but similar, concepts of the presented data transformation reconstructor. More precisely, the data lapsing transformations are employed in [40] and therefore other features are extracted. Also, in [41] a similar data reconstructor is used and further features based on the Fourier transformations are extracted.

3. Support-Resistance zones based indicators

In this section, we propose an algorithm that estimates support-resistance zones.

The support and resistance levels are often applied for short-term financial market forecasting. For example, the predictive power of support and resistance levels for intraday exchange rates are presented in [36].

Our algorithm takes any of the prices as the input. Be it open price, close price, or eventually the price level of a limit order book, such as the best bid price, the value of the third bid level, etc.

Let the window $\mathcal{W}_{index}$ be the window containing all the points found in the daily price series between the indexes *index* - *WINDOW_SIZE* to *index*, where *index* is the current position of the iterator in a daily price series. We define the *upper / lower bound line* as a line which approaches the data contained in the window $\mathcal{W}_{index}$ from the top/bottom, such that the absolute distance between points contained in the $\mathcal{W}_{index}$ and the line itself is minimal and there is no point touching the line. Each of the respective lines can be

[4] Ta-lib: Technical analysis library. https://www.ta-lib.org/

described by two parameters slope and offset, i.e.:

$$y_{upperbound,x} = slope_{upperbound,x} * x + offset_{upperbound,x},$$

$$y_{lowerbound,x} = slope_{lowerbound,x} * x + offset_{lowerbound,x},$$

where x stands for the index within the window $\mathcal{W}_{index}$ and $y_{upperbound,x}$ and $y_{lowerbound,x}$ are the predicted bounding lines at that index. For a given window size, for each level and for amount of points N in a single trading day series, we extract slope and offset in order to describe the behavior of the support/resistance over time. However, we also need to identify the strength of such a zone. We define this to be a percentage of total points contained in a window $\mathcal{W}$ which falls into the distance no more than $d > 0$ from a respective line. Let i be the current index in the interval $[\textit{index - WINDOW\_SIZE}, \textit{index}]$. A point $w_i \in \mathcal{W}_{index}$ is called a challenger if a distance between the point and the bounding line is less than $100 * C$ percents of the bounding line value, where $C \in [0, 1]$. So to define the resistance strength R_s as the relative proportion of challengers when compared to the whole window size, i.e.:

$$R_{s,index} = \frac{|\{w_i \in \mathcal{W}_{index} : w_i \geq ((slope_{upperbound} * i + offset_{upperbound}) \times (1 - C)\}|}{|\mathcal{W}_{index}|},$$

where C is the constant controlling the size of the band which if exceeded shall be considered as a challenger of the resistance zone.

For the support zone, we can proceed similarly:

$$S_{s,index} = \frac{|\{w_i \in \mathcal{W}_{index} : w_i \leq ((slope_{upperbound} * i + offset_{upperbound}) \times (1 + C)\}|}{|\mathcal{W}_{index}|}.$$

A visualization of the bounding lines can be found in the Figure 2. Two green lines represent the currently chosen window. The yellow line represents fitted upper bound and the blue line represents fitted lower bound. Apart from extracting slopes, offsets and strengths for each line we also extract the angle between the upper and lower bound. If the lines are diverging (i.e. they have intercepted one another prior to the currently evaluated index i), then the angle is kept positive. If the lines have not intercepted one another before currently evaluated index i (as seen in the Figure 2, the second green vertical line depicts currently evaluated index i - intersection occurs after it) we set the angle to the negative value. In order to compute the angle we compute the direction vectors $\boldsymbol{d_u}, \boldsymbol{d_l}$ for each of the lines ($y_{upperbound}$ and $y_{lowerbound}$). Furthermore, we apply the well known equation to extract the angle itself:

$$cos\alpha = \frac{\boldsymbol{d_u} \cdot \boldsymbol{d_l}}{|\boldsymbol{d_u}| \cdot |\boldsymbol{d_l}|}.$$

We extracted our support/resistance (SR) based features from all the limit order book level prices (ask / bid) - we work with the data of depth five. In Figure 3 we provide mutual information values computed for the SR features extracted from the Ask 1 level prices. As a reader can note, these features provide significant mutual information with the close price. Since there is a low correlation between support/resistance (SR) based features and features extracted in Section 2, it implies the fact they shall bring different information and thus they could nicely encompass one another.

The best performing feature *intersecting_ys* denotes the price value at which the upper and lower bounding line intersect.

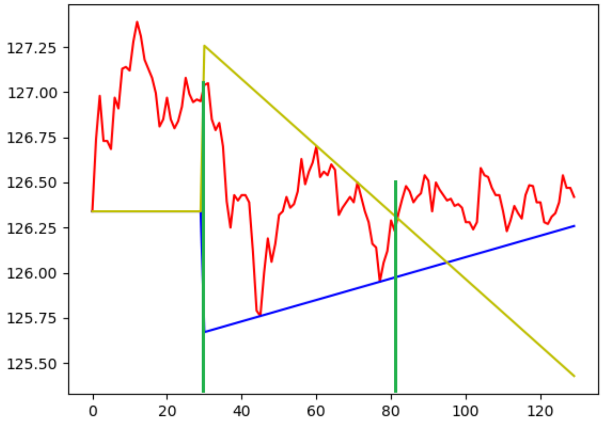


Fig. 2. Visualization of the upper and lower bounding lines estimating the S/R zones. Red line depict the price.

Furthermore, we can study the behavior of the same feature when extracted over different limit order book levels to understand which level holds relevant information.

Surprisingly, in some of the extracted features, we experience an increasing portion of mutual information with increasing depth of limit order book. As can be observed in Figure 4, the angle of intersection tends to have more information when extracted over the deeper level and poses linear growth with respect to the depth.

Another feature exhibiting similar interesting behavior is the $'intersecting\_ys'$ feature, which again shows higher importance with increasing depth level 5.

The observed behavior can be explained by a concept that the deeper level of the market is usually where the price lands if a change starts. This means that if there is a significant market movement theoretically the fifth level might become the first level value in the future and thus there is higher mutual information.

4. Proposed methodology

4.1. Motivation

During the data transformation part, we extracted features from the LOB and we enhanced that set with technical indicators. Furthermore, we included the new features based on the support-resistance zones, and thus there are nearly 3800 features. Since there is a huge number of features for each timestamp, if we plug all of them in the LSTM model, the model tends to over-fit. Therefore, feature selection is important for the model to perform well. But from such a large amount of possible features that could be extracted, how to choose the relevant ones? Considering any feature, the key question here is how to determine its quality.

Different approaches to perform feature selection have been explored in the literature. In [26] four different models, based on different feature selection methods, which use

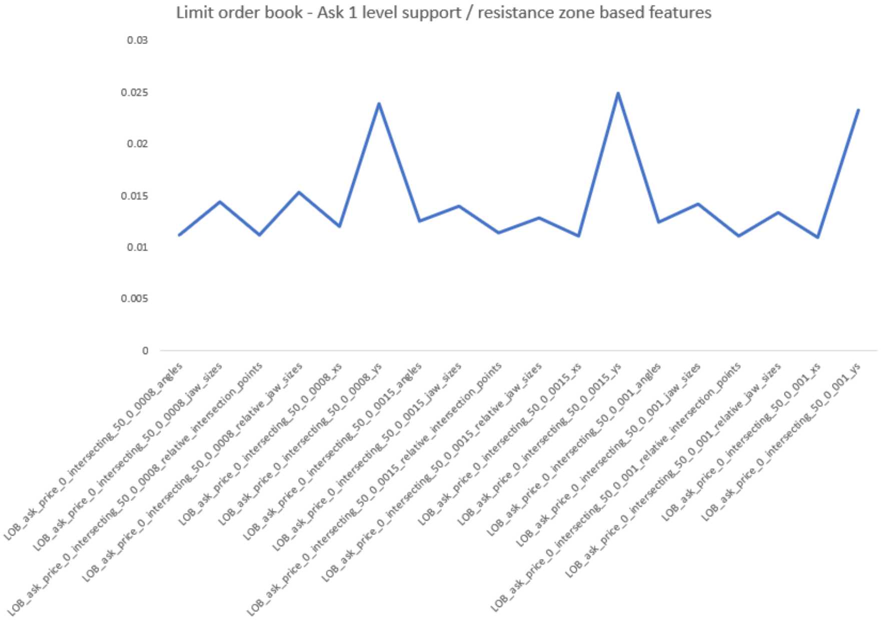


Fig. 3. Mutual information of the support / resistance zone based features for the level 1 prices on the ask side.

fifty-five technical indicators as input variables to predict the direction of the price, were compared. Moreover, hybrid Artificial Neural Network (ANN) models were employed in [18] to choose technical indicators that are relevant to determine stock market price direction. Feature selection processes have been applied in various fields. The authors in [4] aimed of this research is to develop a Majority Vote based Feature Selection algorithm (MVFS) to identify the most valuable software metrics and the thorough experiments showed the ability of the proposed method to find out the most significant software metrics that enhance defect prediction performance. On the other hand, the authors in [30] analyzed the correlations among different commodities sales to identify interesting patterns to increase cross-marketing quality.

4.2. Basic measures used in the paper

Here, we mention two measures used in the following of the paper.

Mutual Information The mutual information (MI), also known as the information gain, of two random variables is a quantity that measures the amount of information about one random variable which is communicated with another one.

Formally, the mutual information of a pair of two random variables X and Y with values in $\mathcal{X}$ and $\mathcal{Y}$ respectively, with the joint distribution $p_{X,Y}$ and the marginal distri-

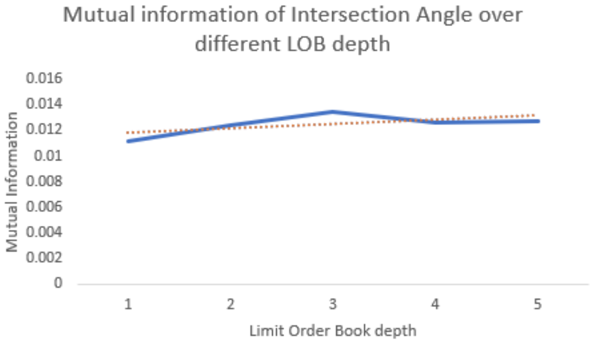


Fig. 4. Mutual information of the angle of 'intersection' feature when computed over different limit order book levels.

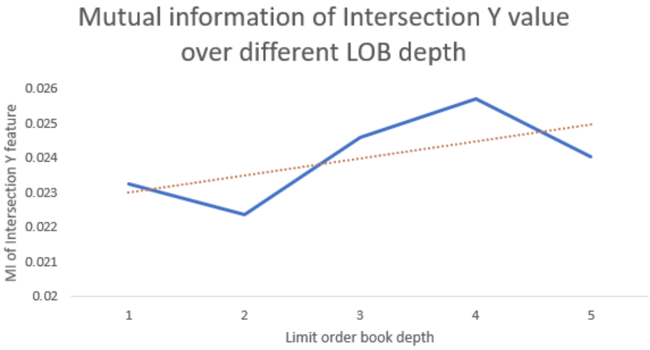


Fig. 5. Mutual information of the 'intersection Y' feature plotted over different limit order book depth levels.

butions p_X and p_Y, is calculated by the formula:

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}.$$

For a deeper theoretical overview on a mutual information we refer the reader to [44]. Furthermore, a comprehensive overview of feature selection methods based on mutual information is stated in [45].

Autocorrelation Autocorrelation measures a correlation of a signal with its lagged version over different successive time periods. Informally, it is a degree of similarity between a signal's present value and its former values.

4.3. Feature groups

In this paper, two assumptions have been made in order to perform feature selection. We assume that a feature with higher autocorrelation is more powerful when it comes to classifying the market data vector into the label than a feature having lower autocorrelation. Another assumption is that a feature that has higher mutual information with close price contains more information than a feature with lower mutual information.

Therefore, we construct two lists with features, which are sorted descended, one with respect to feature's autocorrelation and one with respect to the feature's mutual information with the close price. Moreover, we use also the initial list of all features. These three different lists correspond to the column named list of features Table 1, while in the last column the selection method was presented. The first group of features was selected completely randomly from the set of all features. For other groups, we employ more interesting selection criteria: if two features A and B have the absolute value of cross-correlation higher than $\alpha = 0.7$, we choose one which is higher in the list. More precisely, from the set of all features in the lists, sorted descending, we choose features with the highest value (autocorrelation or mutual information) such that each pair of selected features have the absolute value of cross-correlation less than 0.7. We measure cross-correlation with respect to Pearson's and Spearman's correlation coefficient, which measures the degree of linear and monotonic correlation between two signals respectively. The list of features for the construction of group 7 was randomly sorted before the selection method was applied.

To summarize, we constructed seven groups of 200 features each and the construction method of each group is presented in Table 1. We compare the performance of the model fed with the features from the aforementioned groups.

For this research, seven different datasets (one per each group of features) were prepared. In order to perform LSTM, each dataset is split into three separate parts: for the training, for the validation and for the testing phase. The first one needs 60% of the data, while validation and testing use 20% each. The data for the testing phase is not visible during the training phase and then the confusion matrix is calculated.

Table 1. The lists of features and selection methods for all seven considered groups of features

	list of features	selection method
Group 1	ALL	Random
Group 2	mutal information	cross corr Pearson
Group 3	autocorrelation	cross corr Pearson
Group 4	ALL	cross corr Pearson
Group 5	autocorrelation	cross spearman
Group 6	mutal information	cross spearman
Group 7	ALL	cross spearman

5. Multi-criteria optimization for choosing optimal group of features

Please note that none of these groups leads to high classification accuracy. That is completely expected due to the specific behavior of the stock market, i.e. a lot of noise present in the data.

However, we aim to compare the performance of the model when fed with different groups of features. The confusion matrix (see [34]) is one of the most used metrics to measure the performance of classification based supervised learning models. Two widely used metrics (e.g. see Section 4 in [49]) are precision and recall, defined as presented with equations (1) and (2).

$$precision = \frac{truepositive}{truepositive + falsepositive} \tag{1}$$

$$recall = \frac{truepositive}{truepositive + falsenegative} \tag{2}$$

These two metrics are different in a way that precision calculates how accurate the model is in a sense how many of the predicted positive are actual positive, while recall measures how many of the actual positives are predicted to be positive. Thus, precision is an adequate measure to use when the cost of false positive is high, while recall shall be used when the cost of false negative is high. In this research we also use an interesting measure, a function of precision and recall, called F_β score (Eq. (3)). In addition to the widely used F_1 score, which is employed when both precision and recall are equally important, commonly used ones are $F_{0.5}$ and $F_{1.5}$. F_1 score is a very convenient measure if we want a balance between precision and recall (see [34]). Note that the more we decrease β, the more we prefer precision over recall. Conversely, the more we increase x, the more we favor recall over precision. Note that as more we decrease β we prefer more precision over recall, and vice versa. The F_β-measure with a smaller β value, such as the $F_{0.5}$-measure, is utilized when more weight is put on precision, and less weight is put on recall, i.e. when minimizing false positives is more important than minimizing false negatives. Therefore, the $F_{0.5}$-measure is used in order to increase the importance of precision and decrease the importance of recall. On the other hand, F_β-measure with β value higher than 1, such as the $F_{1.5}$-measure, is utilized when slightly more attention is put on recall, and less is put on precision. Thus, the $F_{1.5}$-measure is used when minimizing false negatives is more relevant than minimizing false positives, i.e. when more attention is put on raising the importance of false positives rather than false negatives. An example, usually found in the literature, is the F_2-measure. However, in this research, the focus was to just slightly increase the importance of recall over precision and thus, the $F_{1.5}$-measure is used.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \tag{3}$$

There is no alternative optimizing all the criteria (precision, recall, F_1, etc.) at the same time, but there are many approaches within multicriteria optimization that can be used for problems having multicriteria nature. More on multicriteria decision analysis can be found in [20]. The enhanced concept of a confusion matrix, which evaluates the performance of a trading strategy was established in [14].

In order to choose the optimal group of features, a multi-criteria decision-making methodology, namely PROMETHEE (Preference Ranking Organization Method for Enrichment Evaluations) method, was used. The basic version of the PROMETHEE was presented in [5], while a comprehensive literature review on methodologies and applications of the PROMETHEE family can be found in [3]. Various versions, numerous modifications and additions of the PROMETHEE method have been applied to many problems. In [33] authors used PROMETHEE II to employ a multicriteria evaluation of the statistical and machine learning classifiers for financial decision making.

In this paper, calculations regarding the PROMETHEE method are performed by using the academic version of the Visual PROMETHEE software[5]. We formulate a new multicriteria problem with 7 alternatives/actions (which corresponds to the aforementioned groups from Table 4.3) and 5 criteria (precision, recall, F_1, F_β for $\beta = 0.5$, F_β for $\beta = 1.5$). Six different preference functions were proposed in [5]. In this research, the V-shaped preference function was used for all the criteria due to its simplicity and compatibility with the used criteria.

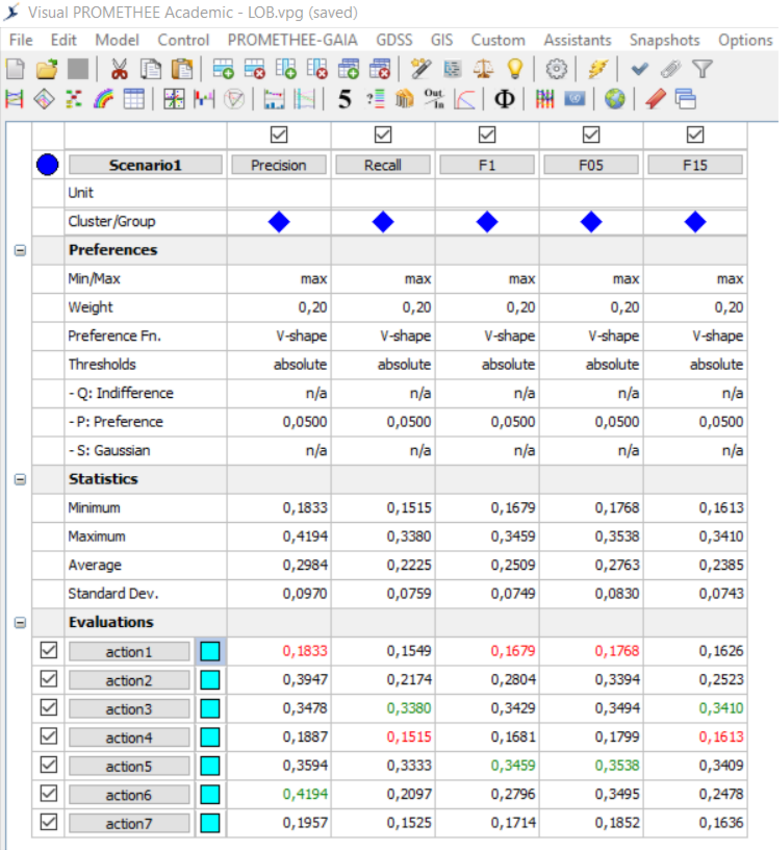
For determinations of weights (relative significance) of each criterion in practical applications of multicriteria optimization various methods can be used depending on the type of the problem, its structuredness as well as the knowledge and experience of the decision maker. A systematic review of the possible methods is given in [35], which can be used as an instruction for an adequate selection of methods for setting the criteria weights in practical applications of multicriteria optimization. One of the presented methods is the Delphi method, which is used in this paper in order to set up the weights of each criterion according to the assessment of the experts in order to obtain the realistic and objective model of the analyzed problem. The Delphi method is based on systematic and organized acquiring and processing of data obtained by the individual predictions given by a small group of experts in order to obtain the mean value from a series of questionnaires (see [13] and [31]). The basic concepts can be found in [42]. Today, this method has many modifications depending on the application, see [28], [29], etc. The simplicity and wide range of application of the Delphi method are based also on the fact that it is suitable to perform this method to tackle the problems that are not structured in a precise way, as well as due to the availability of software support for this method [28]. It can be used for the problems where there is no available data that is relevant and precise enough to perform further research without the usage of experts' assessments, e.g. for setting the weights for multicriteria optimization problems. In this paper, according to the results obtained by using the Delphi method, the system with five different criteria is formulated and the problem is solved for three different variants of weights.

6. Results

To perform the PROMETHEE method, the presented seven different groups (actions) and five presented criteria were entered into the Visual PROMETHEE software, as presented in Fig. 6. The values for all criteria for each action were calculated and entered into the table (see Fig. 6), and the software colored the best values for each criterion in green, while the smallest value for each criterion is in red. Moreover, the software calculates basic statistics for each criterion, e.g. minimum, maximum, average and standard deviation. In order for the PROMETHEE method to be performed, the preference functions had to be defined. The V-shaped preference functions are used to represent linear preference between compared alternatives until the threshold value after which preference is set to one.

[5] http://www.promethee-gaia.net/academic-edition.html

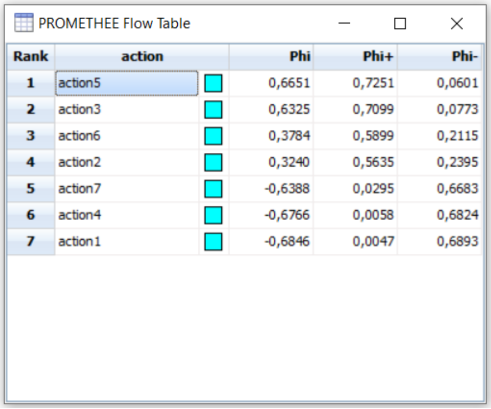
The threshold value is set to 0.05 according to standard deviations of the values for each criterion.


Visual PROMETHEE Academic - LOB.vpg (saved)
File Edit Model Control PROMETHEE-GAIA GDSS GIS Custom Assistants Snapshots Options

			Precision	Recall	F1	F05	F15
			☑	☑	☑	☑	☑
●	Scenario1		Precision	Recall	F1	F05	F15
	Unit						
	Cluster/Group		◆	◆	◆	◆	◆
⊟	**Preferences**						
	Min/Max		max	max	max	max	max
	Weight		0,20	0,20	0,20	0,20	0,20
	Preference Fn.		V-shape	V-shape	V-shape	V-shape	V-shape
	Thresholds		absolute	absolute	absolute	absolute	absolute
	- Q: Indifference		n/a	n/a	n/a	n/a	n/a
	- P: Preference		0,0500	0,0500	0,0500	0,0500	0,0500
	- S: Gaussian		n/a	n/a	n/a	n/a	n/a
⊟	**Statistics**						
	Minimum		0,1833	0,1515	0,1679	0,1768	0,1613
	Maximum		0,4194	0,3380	0,3459	0,3538	0,3410
	Average		0,2984	0,2225	0,2509	0,2763	0,2385
	Standard Dev.		0,0970	0,0759	0,0749	0,0830	0,0743
⊟	**Evaluations**						
☑	action1	🟦	0,1833	0,1549	0,1679	0,1768	0,1626
☑	action2	🟦	0,3947	0,2174	0,2804	0,3394	0,2523
☑	action3	🟦	0,3478	0,3380	0,3429	0,3494	0,3410
☑	action4	🟦	0,1887	0,1515	0,1681	0,1799	0,1613
☑	action5	🟦	0,3594	0,3333	0,3459	0,3538	0,3409
☑	action6	🟦	0,4194	0,2097	0,2796	0,3495	0,2478
☑	action7	🟦	0,1957	0,1525	0,1714	0,1852	0,1636

Fig. 6. Inserting data into the Visual PROMETHEE software.

An important aspect of multicriteria optimization is stating the importance/weights of each criterion. When all criteria are equally important, weights for all five criteria are set to the same value, i.e. 0.2, the result flow table is presented in Fig. 7. This table also contains the positive and negative outranking flow (denoted as Phi+ and Phi-), as well as calculated net outranking flow (denoted as Phi), which represents the balance between the positive and the negative outranking flows (see [20]). It can be seen that the best option is action number 5, while the worst one is action number 1. Therefore, an illustrative representation of the resulting order of groups can be seen in Figure 8.



PROMETHEE Flow Table

Rank	action		Phi	Phi+	Phi-
1	action5	🟦	0,6651	0,7251	0,0601
2	action3	🟦	0,6325	0,7099	0,0773
3	action6	🟦	0,3784	0,5899	0,2115
4	action2	🟦	0,3240	0,5635	0,2395
5	action7	🟦	-0,6388	0,0295	0,6683
6	action4	🟦	-0,6766	0,0058	0,6824
7	action1	🟦	-0,6846	0,0047	0,6893

Fig. 7. The result flow table for weights (0.2, 0.2, 0.2, 0.2, 0.2)

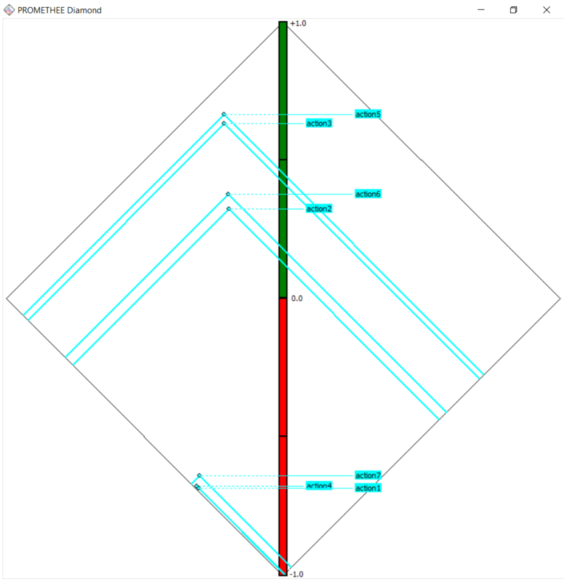
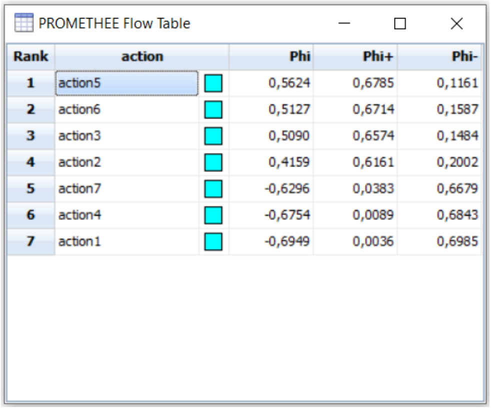


Fig. 8. The diamond representation of results for weights (0.2, 0.2, 0.2, 0.2, 0.2)

Moreover, it is interesting to investigate the significance of sensitivity analysis in assessing how different weights (relative significance) of each criterion affect the results. In this research, we were specifically interested in changing weights of two criteria: precision and recall. In order to perform the comparison, the results are found for the situation when the precision has weight 0.4, while others have 0.15. The result flow table for this situation can be seen in Figure 9, while the diamond representation is represented in Figure 10. The best alternative is still the action number 5. Thus, we can say it stays stable when we increase the significance of the precision criteria from 0.2 to 0.4. The second best option is now number 6.

Similarly, the results are found for the situation when the recall has weight 0.4, while others have 0.15. The result flow table for this situation can be seen in Figure 11, while the diamond representation is represented in Figure 12. The action number 5 is still slightly better than the second best option. However, there is a change at the end of the table, and the action number 4 is considered the worst according to these weights of criteria, by being slightly worse than the action 1.



PROMETHEE Flow Table

Rank	action		Phi	Phi+	Phi-
1	action5	🟦	0,5624	0,6785	0,1161
2	action6	🟦	0,5127	0,6714	0,1587
3	action3	🟦	0,5090	0,6574	0,1484
4	action2	🟦	0,4159	0,6161	0,2002
5	action7	🟦	-0,6296	0,0383	0,6679
6	action4	🟦	-0,6754	0,0089	0,6843
7	action1	🟦	-0,6949	0,0036	0,6985

Fig. 9. The result flow table for weights (0.4, 0.15, 0.15, 0.15, 0.15)

Also, we suggest that not only the price as a source of the signal shall be investigated, however respective limit order book levels born significant informativeness too. Furthermore, some of the features extracted out of simple support/resistance zones turned out to have an interesting property - the deeper the limit order book level is, the higher the informativeness (mutual-information) when predicting the target variable.

7. Conclusion and future work

In this paper, the performance of the model based on LSTM to predict profitable trades was investigated when fed with different features extracted from the LOB data. To perform that, a good feature selection approach had to be employed. Among presented seven variants of feature selection methodology, in order to choose the most suitable option with respect to the five proposed criteria a multicriteria optimization technique (i.e. PROMETHEE method) was performed. Moreover, it was investigated what happens when different values for the relative significance of the proposed criteria are used. The best feature selection strategy was shown to be the strategy that produced the group that consists of the features having high autocorrelation, which are pairwise not monotonically correlated in the sense that each pair of features has Spearman's correlation coefficient less

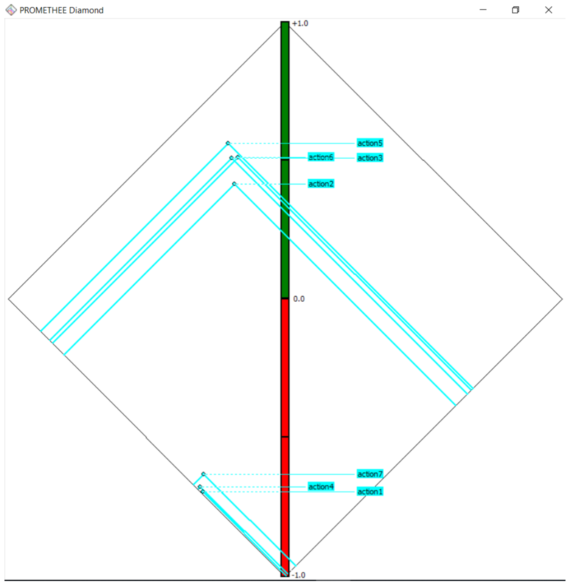
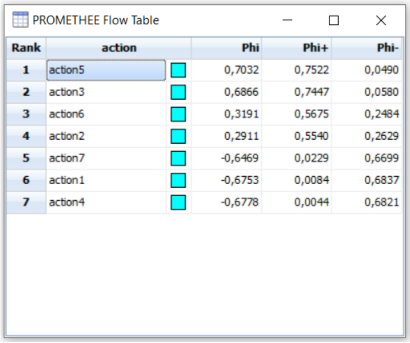


Fig. 10. The diamond representation of results for weights (0.4, 0.15, 0.15, 0.15, 0.15)

than $\alpha = 0.7$. This confirms our initial assumption that features that have higher autocorrelation hold more information. Note that group number 5 stayed the best even when we changed the significance of the precision and recall, meaning that this is the best option even when the cost of false positive high and when the cost of false negative high.

Future work will be continued in several directions. We will try to introduce some new features in order to explore how that would affect the proposed model and decision-making method's results. On the other hand, we want to use other multicriteria optimization methods.

Acknowledgments. The authors are thankful to Thorsten Rheinländer for providing insightful comments and for his help in doing the LOBSTER data analysis and valuable, constructive and insightful suggestions. This research was supported by TU Wien in Austria, the Faculty of Mathematics, University of Belgrade and the Serbian Ministry of Education, Science and Technological Development through Mathematical Institute of the Serbian Academy of Sciences and Arts.



PROMETHEE Flow Table

Rank	action		Phi	Phi+	Phi-
1	action5	🟦	0,7032	0,7522	0,0490
2	action3	🟦	0,6866	0,7447	0,0580
3	action6	🟦	0,3191	0,5675	0,2484
4	action2	🟦	0,2911	0,5540	0,2629
5	action7	🟦	-0,6469	0,0229	0,6699
6	action1	🟦	-0,6753	0,0084	0,6837
7	action4	🟦	-0,6778	0,0044	0,6821

Fig. 11. The diamond representation of results for weights (0.15, 0.4, 0.15, 0.15, 0.15)

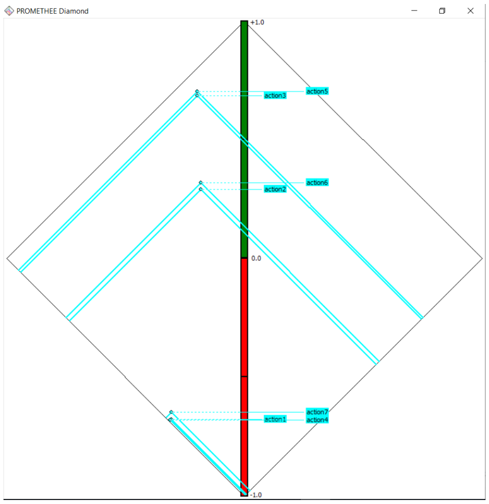


Fig. 12. The diamond representation of results for weights (0.15, 0.4, 0.15, 0.15, 0.15)

8. Appendix

Algorithm 1 Sweep forward algorithm labelling the desired trades

```
Input: $\mathcal{D}_d$,$RISK$, $REWARD$
Output: $\mathcal{D}_d^*$ - the labelled data set
$\mathcal{D}_d^* = None$
for $t = 0$ to $|\mathcal{D}_{day}|$ do
    $tPrice = getPrice(\mathcal{D}_d[t])$
    $stopLoss = tPrice \cdot (1 - RISK)$
    $targetReward = tPrice \cdot (1 + REWARD)$
    $x_t = \mathcal{D}_d[t]$
    $labelled = False$
    for $subsequent\_t = t$ to $|\mathcal{D}_d|$ do
        $currentPrice = getPrice(\mathcal{D}_d[subsequent\_t])$
        $tempStopLoss = currentPrice \cdot (1 - RISK)$
        if $(tempStopLoss > stopLoss)$ then
            $stopLoss = tempStopLoss$
        end if
        if $(currentPrice < stopLoss)$ then
            $\mathcal{D}_d^*.append(labelAsIdlePoint(x_t))$
            $labelled = True$
            break
        end if
        if $(currentPrice > targetReward)$ then
            $\mathcal{D}_d^*.append(labelAsBuyPoint(x_t))$
            $labelled = True$
            break
        end if
    end for
    if $(labelled == False)$ then
        $\mathcal{D}_d^*.append(labelAsIdlePoint(x_t))$
    end if
end for
return $\mathcal{D}_d^*$
```

References

1. Abergel, F., Jedidi, A.: A mathematical approach to order book modeling. International Journal of Theoretical and Applied Finance 16(05), 1350025 (2013)
2. Andrić, K., Kalpić, D., Bohaček, Z.: An insight into the effects of class imbalance and sampling on classification accuracy in credit risk assessment. Computer Science and Information Systems 16(1), 155–178 (2019)
3. Behzadian, M., Kazemzadeh, R., Albadvi, A., Aghdasi, M.: Promethee: A comprehensive literature review on methodologies and applications. European Journal of Operational Research 200(1), 198 – 215 (2010), http://www.sciencedirect.com/science/article/pii/S0377221709000071

4. Borandag, E., Ozcift, A., Kilinc, D., Yucalar, F.: Majority vote feature selection algorithm in software fault prediction. Computer Science and Information Systems 16(2), 515–539 (2019)
5. Brans, J.P., Vincke, P.: Note—a preference ranking organisation method: (the promethee method for multiple criteria decision-making). Management science 31(6), 647–656 (1985)
6. le Calvez, A., Cliff, D.: Deep learning can replicate adaptive traders in a limit-order-book financial market. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1876–1883. IEEE (2018)
7. Cartea, Á., Jaimungal, S., Penalva, J.: Algorithmic and high-frequency trading. Cambridge University Press (2015)
8. Chen, S.H., Wang, P.P.: Computational intelligence in economics and finance. In: Computational Intelligence in Economics and Finance, pp. 3–55. Springer (2004)
9. Chen, Y., Hao, Y.: Integrating principle component analysis and weighted support vector machine for stock trading signals prediction. Neurocomputing 321, 381–402 (2018)
10. Colby, R.W., Meyers, T.A.: The encyclopedia of technical market indicators. Dow Jones-Irwin Homewood, IL (1988)
11. Cont, R., Kukanov, A., Stoikov, S.: The price impact of order book events. Journal of financial econometrics 12(1), 47–88 (2014)
12. Cont, R., Stoikov, S., Talreja, R.: A stochastic model for order book dynamics. Operations research 58(3), 549–563 (2010)
13. Dalkey, N., Helmer, O.: An experimental application of the delphi method to the use of experts. Management science 9(3), 458–467 (1963)
14. Dixon, M.: A high-frequency trade execution model for supervised learning. High Frequency 1(1), 32–52 (2018)
15. Dixon, M.: Sequence classification of the limit order book using recurrent neural networks. Journal of computational science 24, 277–286 (2018)
16. Eschenauer, H., Koski, J., Osyczka, A.: Multicriteria design optimization: procedures and applications. Springer Science & Business Media (2012)
17. Ghosh, I., Jana, R.K., Sanyal, M.K.: Analysis of temporal pattern, causal interaction and predictive modeling of financial markets using nonlinear dynamics, econometric models and machine learning algorithms. Applied Soft Computing 82, 105553 (2019), http://www.sciencedirect.com/science/article/pii/S1568494619303333
18. Göçken, M., Özçalıcı, M., Boru, A., Dosdoğru, A.T.: Integrating metaheuristics and artificial neural networks for improved stock price prediction. Expert Systems with Applications 44, 320–331 (2016)
19. Gould, M.D., Porter, M.A., Williams, S., McDonald, M., Fenn, D.J., Howison, S.D.: Limit order books. Quantitative Finance 13(11), 1709–1742 (2013)
20. Greco, S., Figueira, J., Ehrgott, M.: Multiple criteria decision analysis. Springer (2016)
21. Grgic, D., Vdovic, H., Babic, J., Podobnik, V.: Crocodileagent 2018: Robust agent-based mechanisms for power trading in competitive environments. Computer Science and Information Systems 16(1), 105–129 (2019)
22. Gupta, S.: Financial intelligence. Journal of Global Economy 13(3) (2017)
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
24. Horst, U., Paulsen, M.: A law of large numbers for limit order books. Mathematics of Operations Research 42(4), 1280–1312 (2017)
25. Huang, R., Polak, T.: Lobster: Limit order book reconstruction system. Available at SSRN 1977207 (2011)
26. Kumar, D., Meghwani, S.S., Thakur, M.: Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets. Journal of Computational Science 17, 1–13 (2016)
27. Kune, R., Konugurthi, P.K., Agarwal, A., Chillarige, R.R., Buyya, R.: The anatomy of big data computing. Software: Practice and Experience 46(1), 79–105 (2016)

28. Lawnik, M., Banasik, A.: Delphi method supported by forecasting software. Information 11(2), 65 (2020)
29. Lee, S., Cho, C., Hong, E.k., Yoon, B.: Forecasting mobile broadband traffic: Application of scenario analysis and delphi method. Expert Systems with Applications 44, 126–137 (2016)
30. Li, H., Wu, Y.J., Chen, Y.: Time is money: Dynamic-model-based time series data-mining for correlation analysis of commodity sales. Journal of Computational and Applied Mathematics 370, 112659 (2020), http://www.sciencedirect.com/science/article/pii/S0377042719306648
31. Linstone, H.A., Turoff, M., et al.: The delphi method. Addison-Wesley Reading, MA (1975)
32. Mi, H., Xu, L.: Optimal investment with derivatives and pricing in an incomplete market. Journal of Computational and Applied Mathematics 368, 112522 (2020), http://www.sciencedirect.com/science/article/pii/S0377042719305278
33. Mousavi, M.M., Lin, J.: The application of promethee multi-criteria decision aid in financial decision making: Case of distress prediction models evaluation. Expert Systems with Applications p. 113438 (2020)
34. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
35. Odu, G.: Weighting methods for multi-criteria decision making technique. Journal of Applied Sciences and Environmental Management 23(8), 1449–1457 (2019)
36. Osler, C.L.: Support for resistance: technical analysis and intraday exchange rates. Economic Policy Review 6(2) (2000)
37. Pal, S.S., Kar, S.: Time series forecasting for stock market prediction through data discretization by fuzzistics and rule generation by rough set theory. Mathematics and Computers in Simulation 162, 18 – 30 (2019), http://www.sciencedirect.com/science/article/pii/S0378475419300011
38. Palguna, D., Pollak, I.: Mid-price prediction in a limit order book. IEEE Journal of Selected Topics in Signal Processing 10(6), 1083–1092 (2016)
39. Pardo, R.: The evaluation and optimization of trading strategies, vol. 314. John Wiley & Sons (2011)
40. Radojičić, D., Kredatus, S., Rheinländer, T.: An approach to reconstruction of data set via supervised and unsupervised learning. In: 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI). pp. 000053–000058. IEEE (2018)
41. Radojičić, D., Kredatus, S.: The impact of stock market price fourier transform analysis on the gated recurrent unit classifier model. Expert Systems with Applications 159, 113565 (2020), http://www.sciencedirect.com/science/article/pii/S0957417420303894
42. Rowe, G., Wright, G.: Expert opinions in forecasting: the role of the delphi technique. In: Principles of forecasting, pp. 125–144. Springer (2001)
43. Shynkevich, Y., McGinnity, T.M., Coleman, S.A., Belatreche, A., Li, Y.: Forecasting price movements using technical indicators: Investigating the impact of varying input window length. Neurocomputing 264, 71–88 (2017)
44. Thomas, J.A., Cover, T.: Elements of information theory. John Wiley & Sons, Inc., New York. Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, MPH (2009),“Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems,” Journal of the Royal Society Interface 6, 187–202 (1991)
45. Vergara, J.R., Estévez, P.A.: A review of feature selection methods based on mutual information. Neural computing and applications 24(1), 175–186 (2014)
46. Withanawasam, R., Whigham, P., Crack, T.: Characterising trader manipulation in a limit-order driven market. Mathematics and Computers in Simulation 93, 43 – 52 (2013), http://www.sciencedirect.com/science/article/pii/S0378475412002212, selected Papers of the MSSANZ 19th Biennial Conference on Modelling and Simulation, Perth, Australia, 2011

47. Zhang, J., Huang, M.L., Meng, Z.P.: Visual analytics for bigdata variety and its behaviours. Computer Science and Information Systems 12(4), 1171–1191 (2015)
48. Zheng, B., Moulines, E., Abergel, F.: Price jump prediction in limit order book. arXiv preprint arXiv:1204.1381 (2012)
49. Zhou, F., Zhang, Q., Sornette, D., Jiang, L.: Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices. Applied Soft Computing 84, 105747 (2019), http://www.sciencedirect.com/science/article/pii/S1568494619305289

Dragana Radojičić received her PhD in Mathematics in 2020 from the Department for Financial and Actuarial Mathematics, the Vienna University of Technology, where she has been working as a Teaching assistant. She obtained her MSc degree from the Faculty of Mathematics, Technical University of Berlin in 2016 and her BSc degree from the Faculty of Mathematics, University of Belgrade in 2014. Her research is mainly focused on stochastic analysis models, financial mathematics and machine learning in finance.

Nina Radojičić is an Assitant Professor at the Department of Computer Science, Faculty of Mathematics, University of Belgrade. She received her PhD degree in Computer Science from the University of Belgrade, Serbia, in 2018. Her main research interests include Artificial Intelligence, Computational Intelligence, Metaheuristic Optimization algorithms, Mathematical Optimization, etc.

Simeon Kredatus holds a Master's degree in Computational Intelligence from Technical University in Vienna and has over 10 years of experience in the machine learning and data processing industry. Data and Artificial Intelligence applications are among his passions as well as a career choice.

Received: June 23, 2020; Accepted: November 10, 2020.

Computer Science and Information Systems 18(3):771–790 https://doi.org/10.2298/CSIS200620040W

End-to-End Diagnosis of Cloud Systems against Intermittent Faults

Chao Wang[1,3], Zhongchuan Fu[2,*], and Yanyan Huo[1]

[1] Computer School, Beijing Information Science and Technology University,
North 4th Ring Mid Road 35,
100101 Beijing, China
wangchao@bistu.edu.cn

[2] Computer Science & Technology Department, Harbin Institute of Technology,
Xidazhi Street 92,
150001 Heilongjiang, China
fuzhongchuan@hit.edu.cn

[3] Beijing Advanced Innovation Center for Materials Genome Engineering,
North 4th Ring Mid Road 35,
100101 Beijing, China

Abstract. The diagnosis of intermittent faults is challenging because of their random manifestation due to intricate mechanisms. Conventional diagnosis methods are no longer effective for these faults, especially for hierachical environment, such as cloud computing. This paper proposes a fault diagnosis method that can effectively identify and locate intermittent faults originating from (but not limited to) processors in the cloud computing environment. The method is end-to-end in that it does not rely on artificial feature extraction for applied scenarios, making it more generalizable than conventional neural network-based methods. It can be implemented with no additional fault detection mechanisms, and is realized by software with almost zero hardware cost. The proposed method shows a higher fault diagnosis accuracy than BP network, reaching 97.98% with low latency.

Keywords: cloud system, intermittent fault, fault diagnosis, end-to-end, LSTM, PNN.

1. Introduction

The diagnosis of intermittent faults has drawn increasing attention in recent years. This problem is challenging because of the random manifestation of such faults due to intricate mechanisms. This can be mainly attributed to two reasons: a) The long time operation, high-load operation, and large cluster scale could more easily lead to phenomena such as PVT variation, cross talk, and interference, as the computing density increases (along with energy throughput) in cloud systems; b) On the other hand, aggressive chip feature sizes increase the hardware fault susceptibility of the single device itself [1].

[*] Corresponding author

Hardware faults can be classified into 3 categories: transient, intermittent, and permanent faults, depending on the duration. Transient faults often manifest as bit-flips and were first detected through a radioactive material contained in the chip package. High-energy radioactive particles, such as thorium and uranium, in the package emit α particles with energy > 8 MeV. When the accumulated electric quantity exceeds the charge threshold, the behavior of the PN junction changes, resulting in a bit-flip [2]. The duration of this type of fault is in the picosecond scale; it can be recovered by writing or refreshing. By contrast, a permanent fault is due to the aging of components or irreversible physical damage such as open or short failures in the circuits [3]. Permanent faults need to be replaced or repaired. Existing state-of-the-art diagnosis technologies are mainly designed for transient and permanent faults.

The mechanism of intermittent faults is complex, and was put forward as early as the 1970s [4]. The causes of device failures include time-dependent dielectric breakdown (TDDB), negative bias temperature instability (NBTI), electromigration (EM), stress migration (SM), and thermal cycling (TC) [5]. An intermittent fault is non-periodic, i.e., the time, frequency, probability, and amplitude of fault occurrence are random [6]. As reported, faults that occur in electronic devices are typically intermittent. Intermittent faults in integrated circuits are 10~30 times more frequent than permanent faults [7]. An error report [8] from Microsoft Windows on 950,000 personal computers showed that approximately 39% of the hardware errors reported in microprocessors are intermittent faults.

Most existing fault diagnosis technologies are based on replay, i.e., after detecting the fault, the process instance is executed again on the standby core, and the former and latter are then compared for the diagnosis. This type of method is only applicable to distinct transient and permanent faults, because intermittent faults occur non-periodically and are not necessarily reproduced in the process of replay. This let us to conclude that the diagnosis of intermittent faults is challenging, from the embed devices to the cloud computing systems (with redundant threads or cores, but not avail against the uncertainty and propagation thereafter). Raghavan [9] compared the outputs of a tested circuit and reference circuit, and distinguished permanent and intermittent faults based on whether the number of faults exceeded a certain threshold. The number and threshold of faults are typically determined with respect to conditions such as the fault rate. To diagnose intermittent faults, Lafortune adopted the monitoring theory to study the diagnosability of discrete event systems (DESs), to check whether the fault can be diagnosed within a limited time [10, 11]. Due to the assumption that the fault type is known (assuming that the known fault is intermittent), the purpose is to identify how the system works in the recovery state against an intermittent fault, but not to diagnose an unknown fault, based on observable events, to be an intermittent fault or not. Therefore, the diagnosability of intermittent fault cannot be analyzed.

A novel intermittent fault diagnosis algorithm for cloud systems is presented to overcome the above limitations. Our contributions are as follows: (1) End-to-End. To avoid the heavy reliance on environment feature extraction, this method is intuitively designed as an end-to-end diagnosis method, which, although requires information selection, does not use any potential function; it explores the best characteristic representation to solve problems from the perspective of "intuition”; (2) Covers all the hardware types. Unlike most conventional methods, this method covers all the hardware types and the fault locating responsibility, meaning that this method needs to identify each instance to be a golden run or an isolating faulty category (transient/ intermittent/ permanent faults), and also to locate where the fault originates from; (3) No additional fault detection mechanism. This method does not rely on additional fault detection mechanisms. It uses only hardware interrupt handlers as the basis for inspection; these are commonly used in central processor units, and it is realized via software, so the hardware cost is almost zero. The experimental results show that the diagnosis accuracy reaches 97.98% for all the three types of faults (transient, intermittent, and permanent faults).

The rest of this paper is organized as follows. Section 2 presents the related work in literature. Section 3 describes a novel end-to-end diagnosis framework, including corresponding algorithms. In Section 4, the method is validated with experimental work. Section 5 concludes the paper.

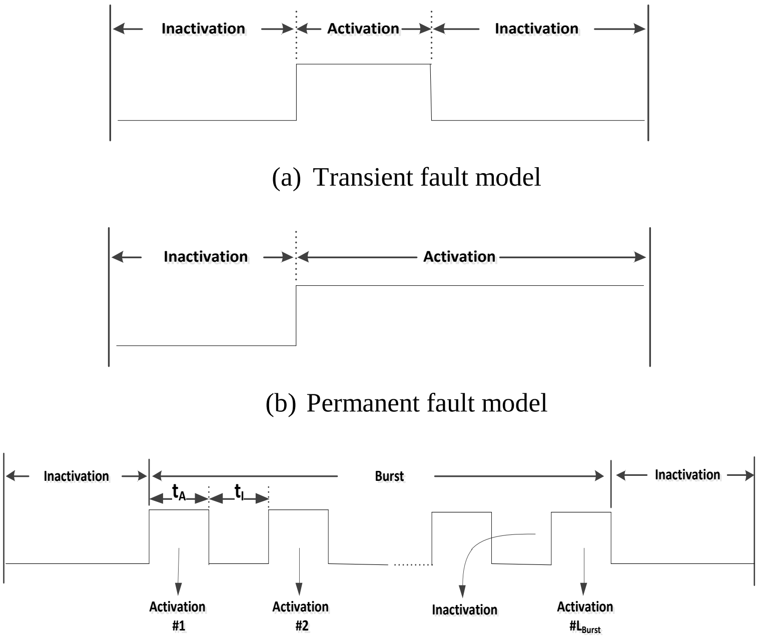
2. Related Work

We would like to present the research work in literature of the fault diagnosis area, especially in the hardwired faults those originated in the computer devices. In this part of work, the description of transient, intermittent, and permanent fault models, the fault diagnosis method and fault injection technologies are introduced.

2.1. Fault Models

The pulse description method can uniformly describe the hardware fault models, by using the activation time and the inactivity time as the parameters during the fault occurs. In the case of irradiation, when the illuminated high-energy particles reach the fault threshold, the transient fault will be triggered, causing a bit flip. As the fault duration increases, the energy is released and the transient fault disappears(see Fig. 1(a)). A permanent fault is an irreversible physical defects in the circuit, and a fault phenomenon will always exist (see Fig. 1(b)).

The intermittent faults are different from these two. The occurrence and disappearance of intermittent faults happens mutually (see Fig.1(c)). The fault location is fixed (same as transient and permanent faults). In fact, the intermittent fault model with 101 order duration of the clock cycle has now been accepted by the academic community, and thus, is adopted in this paper and is applied at different levels such as processor structure, virtual machine monitor, operating system and even application level.



(a) Transient fault model

(b) Permanent fault model

(c) Intermittent fault model

Fig. 1. Pulse-based description method for hardware faults [12, 33].

2.2. Fault diagnosis methods

Research that designs scheme for the post-silicon debugging mechanism records the footprint of every instruction as it is executed in the processor [13-15]. Some of them (e.g., IFRA[16]) requires the presence of hardware-based fault detectors to limit the error propagation, while others are implemented in a hybrid hardware-software manner, and with no additional detectors [17, 18]. Carratero et al. [19] propose their method to diagnose faults in the load-store unit (LSU) which is performed during post-silicon validation, and it only covers design faults. In contrast, SCRIBE [20] is proposed to diagnose intermittent faults during regular operation. After the fault is detected, the program is replayed on the standby core, and a data dependence graph (DDG) is constructed by extracting the runtime information (microstructure-level devices). By comparing the data flow graphs of two runs [21], the diagnosis and location of the intermittent fault are realized. Our work is similar to theirs in some aspects. However, as SCRIBE's potential assumption that the fault type is known (assuming that the known fault is intermittent or permanent), the purpose is to diagnose how the system is currently in a recovery or intermittent fault state based on observable events. Therefore, in fact, the diagnosability of intermittent fault are remained unsolved, and additional detection mechanism is still needed by this method.

Hari et al. designed a trace-based fault diagnosis (TBFD) mechanism to diagnose permanent faults. Although the diagnosis accuracy reached 95%, heavy-weight overheads, such as hardware buffers and re-executions, were required [22]. Furthermore, TBFD is only effective for permanent faults. Considering the burst and non-periodic characteristics of intermittent faults, TBFD is not an alternative solution for intermittent fault diagnosis. Deng et al. proposed a stochastic automata-based method that can diagnose both of the permanent fault and the intermittent fault. They set up a finite automaton model by introducing the fault identification mechanism, wherein the state transformation of the system is invested, and the probability of the fault event is made out [23].

The above methods depend on the scale of sample space: few samples cannot guarantee the accuracy of the diagnosis, which in turn can easily cause false alarms. As the existing samples are often limited in the real-world [24], fault injection is an effective method to accumulate the fault instances.

2.3. Fault injectors

Fault injectors are developed and realized toward upper levels in view of systematization. VFIT [25], INJECT [26], and VERIFY [27] are fault injection platforms developed on very high-speed hardware description language (VHDL), supporting fault models on the switch-level, gate level, and register transfer level (RTL). Wang et al. extended their fault injection simulator to multi-core architecture. They selected the UltraSPARC processor (8 cores, 64 threads) as Device Under Test (DUT) to characterize the effects of intermittent faults at the RTL level, and showed that some systematic events can be used as detection symptoms [28-29]. Rashid set up a pure software-based fault injector that is designed on SimpleScalar, and investigated the characteristic of intermittent faults at the application program level (Spec CPU2006) [30]. Hu et al. set up a system-level fault injection platform based on the Simics simulator, and studied the impact of hardware fault on a multi-core system through software simulation, including operating system and application program [31]. Le and Tamir proposed fault injection tools based on cloud environments, taking advantage of virtualization environment (virtual machine monitor) to implement a fault injection interface toward the upper layers [32]. As fault injection modules are (and can only be) implemented in a virtual machine monitor, only misbehaviors of the guest operating system fall into the observation scope and can be tracked.

In this study, the cloud platform is selected as the injection target. Unlike the above fault injector, this work is not implemented merely “on” the cloud (the fault behavior propagation path only covers the operating system level and above); in fact, this work is different in that the virtualization firmware can be tracked even at the CPU structural level, which is beyond the operating system level. Thus, the fault propagation behavior can be tracked with more accuracy than injectors set up on the cloud.

3. Approach

This paper presents an end-to-end fault diagnosis method. The fault log is recorded in the fault injection camp in the cloud environment. Based on the system level run-time information, features are automatically extracted and inputted to the neural network. This method covers all the hardware types as the target fault set, including transient, intermittent, and permanent faults.

We first perform fault diagnosis based on a BP neural network through the statistical analysis of the log. Although artificial feature extraction is less computationally complex than the end-to-end method, there are drawbacks in the way it relies on manual feature extraction, which has two disadvantages: First, the selection of the features needs to be conducive to the classification. Therefore, features are combined through statistical or potential function methods for processing. This method strongly depends on the quality of the feature extraction, even more important than the learning algorithm used. For example, if the color of hair is extracted as a feature, the classification effect for gender will be poor regardless of the classification algorithm used. Therefore, features need enough training for design, which is increasingly difficult in the case of large amounts of data and complex systems. In addition, useful information may be potentially lost in the calculation of the original features. Second, the data element in the feature set may change (information or attributes need to be updated) depending on the operating environment in order to avoid the lack of generalization ability, and the repeated tuning and optimization processes for evaluating how the extracted features may influence the back-end performance, which may increase the time cost of model development. Therefore, an end-to-end diagnosis framework for system-level symptoms is proposed in this paper, providing an efficient solution to the implementation of intermittent fault diagnosis.

3.1. Challenges and solutions of end-to-end model

In the non-end-to-end algorithm, a significant amount of preparatory work is required. For example, in speech recognition, "phoneme" has been invented by linguists. Although it improves the efficiency in the processing step, it will undoubtedly lead to other information loss in the speech. The algorithm requires less data. However, the feature extraction depends on humans, and the feature needs to be redefined for application scenario migration (such as changing language), so the generalization ability is not high.

Hence, the end-to-end method has been proposed, in which the original data are pre-processed and selected as features that are learned without any potential functions. Hence, it can be integrated into the algorithm without human intervention, in order to explore the best characteristic representation to solve problems from the perspective of "intuition". As a result, the input (original data or feature sequence) and the output (fault categories or locations) have been directly connected to both ends of a neural network. However, the end-to-end learning algorithm does not require much human intervention, but it needs a lot of labeled data.

Based on a fault behavior tracking (FBT) system [33], we have applied a two-month period fault injection campus to obtain the systematic-level fault propagation behavior in the cloud computing environment. We obtained statistics from 42,000 experiments on fault injection under SPEC2006 workloads, including eon, gcc, parser, perlbmk, and twolf. For each instance, one of the three types of faults is chosen and injected into the target fault location. We set a time window (within the time of 1,000,000 instructions starting from fault injection) and collected the system-level fault propagation behavior sequence generated in this window. For intermittent faults, an total of 24,000 runs (300 injections * 4 units * 5 benchmarks * 4 Lburst) were conducted; for transient and permanent faults, we conducted 12,000 and 6,000 runs, respectively, since there are two types of permanent faults, namely permanent stuck@0 and permanent stuck@1, compared with the transient faults, which are only of one type. Based on this behavior, the input neural network extracts the features and carries out fault diagnosis. Currently, the simulator covers all the hardware types as the target fault set, including transient, intermittent, and permanent faults, and supports fault injections into four targets, namely the Address generator, Decoder, ALU_FPU, and Register Files in the processor, and monitors the run-time log trace from the instruction buffer and state registers. We developed FBT modules to monitor the software stack.

Given that millions of experimental instances are required to produce numerical labeled data for training the end-to-end framework, we implemented fault injection automatically in the FBT, wherein blue screen recognition and dead loop detection were developed in the controller module, to recognize system crashes due to illegal memory address access, trap stack overflow, and/or other severe perturbations.

3.2. Overall architecture

The reliability modules include the fault injector, fault tracer, and analyzer modules. In **Step 1**, we developed the fault injector module in the FBT to inject the three types of faults (transient/intermittent/permanent) into the specified location in the target unit. The target system is a multi-layer cloud system simulator, wherein the CPU/memory/hard disk is located beyond the VMM and guest operating systems. We adopted the prototype of UltraSPARC T2 processor as the target CPU. UltraSPARC T2 is a commercial chip multi-threading (CMT) processor, which has eight 64-bit cores and 8/16 threads in each core. Instead of exploiting instruction-level parallelism (ILP) and deep pipelining, this processor model achieves a good performance by taking advantage of thread-level parallelism (TLP), which is an optimized CPU model for cloud computing environment, instead of using the ILP architecture.

The cloud software stack, comprising a VMM layer and the operating system for control domain and other virtual domains, is overlaid on top of the simulated hardware. Inside these domains, user applications (in our fault injection campaigns, the benchmark) are processed. The execution environment includes the computer hardware and host operating system. The latter is responsible for the simulator and other fault injection relevant modules. Below the host operating system is a (real world) hardware computing device that is responsible for executing all the software layers in **Step 2**, and the logs are then recorded in the host operating system in **Step 3**. The system-level symptoms are collected so that the fault propagation can be logged at all levels.

Step 4: Feature selection

When using the machine learning technique, feature selection is the most important part. Based on the statistical distribution we just proved, we can take the number of times of a system call shown up in the trace as the major feature and other features, such as the trap level and high OS, as the complement features (unlike feature extraction, feature selection does not require a calculation process for the potential function, which belongs to the original data, because we cannot and do not need to input all the original data into the neural network). The exceptions and interrupts in the cloud environment are collectively referred to as trap. In SPARC architecture, the related attribute values of the trap are stored in specific registers (as listed in Table 1). TL is the trap-level register, which specifies the trap nesting level of the current program state. Under normal circumstances, the value of TL is 0, which means no trap. When the processor enters a trap, the value of TL is increased by 1. When the nesting level of the trap is greater than 1, nest failure occurs. The SPARC architecture requires that at least five layers of nesting are supported. A nest fault is determined by the value of TL. When TL is greater than or equal to 2, nest fault occurs. TT is the trap-type register, indicating the trap-type number. The values of CCR, ASI, pstate, and CWP are also saved in the TSTATE register. The HP and P states represent the privilege level of the processor, indicating hypervisor authorization and operating system administrator, respectively. When a trap occurs, the hardware will automatically save PC/NPC to TPC/TNPC, and save CCR/ASI/pstate/CWP to TSTATE. Otherwise, the trap state program counter (TPC), trap state next program counter (TNPC), and TSTATE are saved in the hardware register stack. The CPU then enters the privilege execution mode and jumps to the trap vector entry to execute the relevant trap service program.

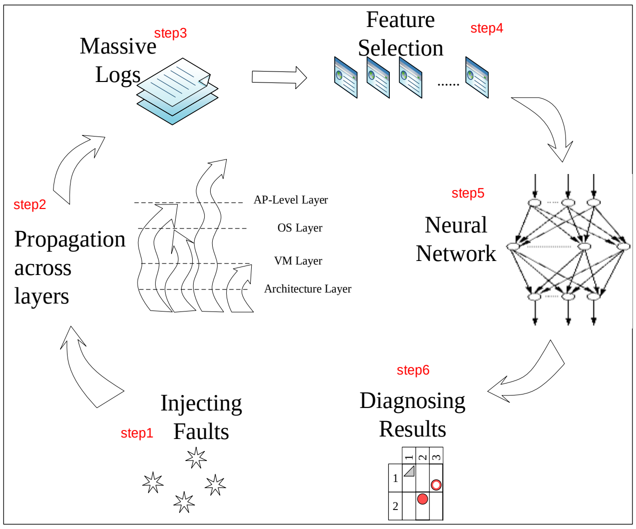


Fig. 2. Block diagram of the proposed end-to-end diagnosis algorithm

High OS: the trap handler only takes a small piece of the coding fragment, except in two cases: 1) to allocate time slices to the application, the operating system may take a longer time to execute. And we record that the maximum continuous instructions is 10000, by tracking the instructions running in the priviledged mode (operating system); 2) to execute the system call procedures, the operating system executes 105 or 106 continuous instructions before returning to the unpriviledged mode (application program). Therefore, under normal states, the number of continuous instructions executed in the priviledged mode will not exceed 106. When this threshold is exceeded, the behavior is considered abnormal.

Table 1. Functional trap registers.

Register	Description
TL	Register to record Trap Level
TT	Register to record Trap Type
TSTATE	Register to record Trap State

Steps 5 & 6: Diagnose algorithms

In the process of fault diagnosis, both of the two learning strategies have been investigated--offline and online. By analyzing the fault behavior (based on the log files), it is not difficult to find that the sample can be regarded as a sequence. For each fault injection simulation instance, several trap events are generated and then logged. Based on this, a sequence can be simply setup as sample towards a learning strategy. In this paper, the method based on the long and short term neural network is adopted, that is, the trap sequence is constructed as the input vector to input to the long short term memory (LSTM). Before the diagnosis framework starts to works, it requires to collect the entire trap event as the input sequence (from the beginning of the simulation to the finish), so it is called the offline learning strategy; on the other hand, each fault can be treated as an event that needs to be diagnosed immediately, and hence the serialized data can be expanded into vector data and submitted one by one. This are often called the online learning strategy. We implement the online mode based on Back Propagation Neural Network (BP) and Probabilistic Neural Network (PNN), respectively. The performance of the learning strategies will be discussed in section 4.

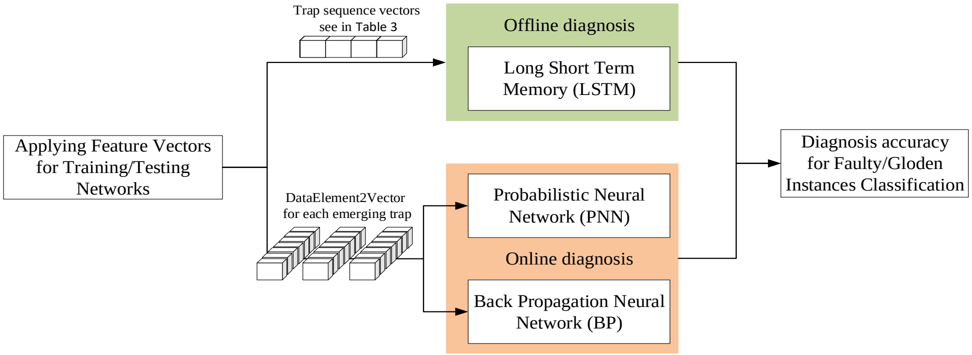

Trap sequence vectors see in Table 3
Offline diagnosis
Long Short Term Memory (LSTM)
Applying Feature Vectors for Training/Testing Networks
Diagnosis accuracy for Faulty/Gloden Instances Classification
DataElement2Vector for each emerging trap
Probabilistic Neural Network (PNN)
Online diagnosis
Back Propagation Neural Network (BP)

Fig. 3. The diagnosis framework consists of offline and online learning strategies.

Offline learning strategy

LSTM. In the course of training, RNN neural network often has gradient disappearing or exploding, so Hochreiter et al. [34] put forward long short term memory neural network. This problem is well overcome in LSTM by adding three gate structures: forget gate, input gate and output gate, to keep and update the status information of each unit module. The input gate receives the current information of the system; the forgetting gate filters the information and discards the useless memory; the output gate filters the value of the next hidden state. In this scheme, the output result is defined as the fault categories, in which we can select the maximum value as the diagnosis result. Cross entropy loss is chosen as the loss function, which is suitable for multiple classifiers. There are two parameters for cross entropy: input value and label, representing the specific gravity of classification of the samples and the category index [0, n-1]. In Equation 1, where is the true value and is the predicted value.

$$\text{loss} = -\sum_{k=1}^{N}(m_k * \log n_k) \qquad (1)$$

Online learning strategy

BP neural network. This is an artificial neural network based on the learning mechanism of back propagation. In the BP neural network, linear transformation is used to map nodes in the input layer to nodes in the hidden layer. The activation function of hidden layer and the linear transformation are co-operated to map nodes from the hidden layer to the output layer. The hidden layer can be one or more layers. We adopt softmax to be the activation function, which converts each vector value to the [0, 1]. See the calculation formula in Equation 2:

$$f_i(x) = \frac{e^{(x_i-\alpha)}}{\sum^j e^{(x_j-\alpha)}}, \alpha = \max(x_i) \tag{2}$$

Wherein x_i is the i_{th} element in the input vector， α is the maximum element among x_i.

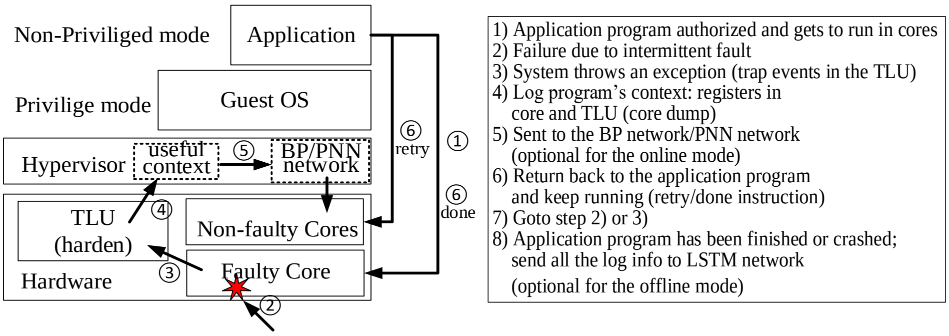
PNN. Unlike BP network, probabilistic neural network is a forward propagation classifier that uses Bayesian decision theory to classify samples. Bayesian decision-making refers to taking the test sample as the classification with the highest probability. The PNN consists of four layers: one input layer, one output layer, and two hidden layers. The two hidden layers are the sample and competition layers. The neuron activation function of the sample layer is used to calculate the distance between the input value and the category center. If the distance is close to a center, the probability of this value in the corresponding area is set high. Theoretically, the output function of the PNN adopts the Bayesian classification method, wherein using Gauss function (equation 3) to compute the distance between input vector and center point in order to classify the data with the maximum probability.

$$\mathrm{y}_g(x;\sigma) = \frac{1}{l_g (2\pi)^{n/2} \sigma^n} \sum_{i=1}^{l_g} \exp\left(-\sum_{j=1}^{n} \frac{\left(x_{ij}{}^{(g)} - x_j\right)^2}{2\sigma^2} \right) \tag{3}$$

wherein n represents number of feature dimension, l_g represents the number of samples in the g_{th} category, x_{ij} represents the j_{th} data of the i_{th} neuron, and σ is a hyper parameter.

3.3. Implementation

The following assumptions about the system are illustrated before we introduce the working flow: a) we assume a commodity multi-core system in which all cores are homogeneous, and are able to communicate with each other through a shared address space. b) We assume the availability of a fault-free core to perform the diagnosis. This is similar to the assumption made by and Li et al. [34]. The fault-free core is only needed during diagnosis. c) Trap logic unit (TLU) in processor is hardened in need to assure the correct exception information is logged. Note that UltraSparc T2 processor provides two trap return instructions, retry and done. Retry makes trap return to the instruction where trap is raised, and re-executes the instruction again when the done instruction returns to continue with the program. When the system detects a fault, it may use the retry instruction to return to the abnormal instruction for re-execution, or use the done instruction to transfer the trap to the operating system when the hypervisor may not be able to process the trap.



1) Application program authorized and gets to run in cores
2) Failure due to intermittent fault
3) System throws an exception (trap events in the TLU)
4) Log program's context: registers in
core and TLU (core dump)
5) Sent to the BP network/PNN network
(optional for the online mode)
6) Return back to the application program
and keep running (retry/done instruction)
7) Goto step 2) or 3)
8) Application program has been finished or crashed;
send all the log info to LSTM network
(optional for the offline mode)

Fig. 4. Working flow of online and offline diagnose methods. The steps in the figure are explained in the box.

Overheads. Compared to other diagnosis schemes, our technique incurs low performance and power overheads with reasons as follows: a) as it initiates diagnosis only when error detection occurs, the diagnosis overhead is not incurred during fault-free execution; b) our scheme do not need to log the context information in the processor continuously (only when an error detection occurs, and not like SCRIBE [19] which needs to do this continuously); c) the complex task of figuring out the fault type and faulty component is done in software. Hence, the power overhead is low.

4. Experimental result

In this section, we evaluated the performance of the proposed end-to-end diagnosis framework against hardware faults for cloud computing systems. We used the FBT simulator based on the software asset management (SAM) to emulate the considered case studies.

Figure 5 shows the coverage of systematic-level fault behavior in the cloud system environment in our FBT simulator. high OS is large. In the transient fault, the coverage of high OS is the highest, in the permanent fault model, the coverage of high OS is the lowest, and almost 0 in the ALU and the decoder. The coverage of high OS decreases with the increase of the burst length. In the ALU, the coverage rate of high OS is significantly higher than that of other components. The overall coverage of nest is also high, and with the increase of the burst length, the coverage is significantly increased. In the transient fault model, the coverage is basically 0, which can be used as the diagnostic feature of the model.

In these traps, the coverage of 0x10 and 0x34 is high. In the ALU, the trap is mainly 0x34, which is caused by the address reading error of the ALU. In the decoder, the coverage of 0x10 and 0x34 is about 50%, which may be caused by the illegal instruction caused by bit flipping, or by the error of the target address or register number caused by the fault, resulting in the wrong instruction address, etc. In the program counter (PC) register, the coverage of 0x10 and 0x34 is high, which may be caused by the change of PC value. The coverage of 0x10 is almost 0 in ALU, but higher in other components, which can be used as the diagnosis feature of ALU. 0x30 only appears in the faulty ALU, and 0xd only appears in the faulty PC register, which can also be used as feature of faulty location diagnosis.

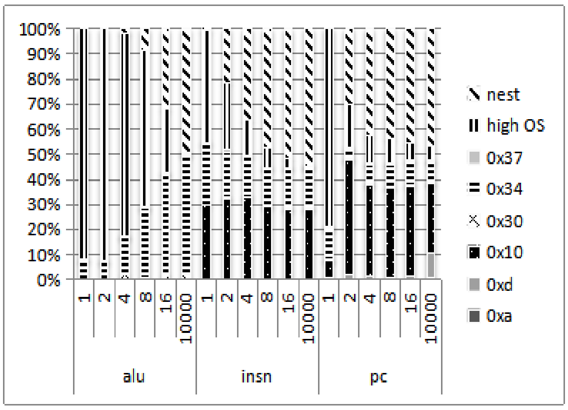


Fig. 5. Systematic-level fault behavior occupation

Offline diagnosis scheme(LSTM):

Cross entropy loss adopts "one hot" mode. As shown in table 2, when hidden reaches 249, the accuracy is the highest; when hidden is determined as 249, it is found that when batch is 50, the accuracy is the highest. With the Adam optimizer, the learning rate is 1e-4, the regularization parameter is 1e-5, the number of neurons in the hidden layer is 249, the batch is set to 50, and the accuracy is 59.8%.

Table 2. Hidden nodes and batch size tuning of LSTM.

Hidden nodes tuning				Batch size tuning	
Hidden	Accuracy	Hidden	Accuracy	Batch	Accuracy
25	53.40%	248	58.70%	16	55.50%
50	59.10%	**249**	**59.80%**	**50**	**59.80%**
100	53.40%	250	59.10%	100	55.50%
150	57.70%	255	58.40%	150	59.10%
200	56.20%	260	56.60%	200	54.50%
245	57.30%	300	58.00%	250	50%

Online diagnosis scheme(BP/PNN):

For BP network, we adopt Adam optimizer, the learning rate is 1e-2, the number of neurons in hidden layer is 150. The sample proportion of fault type is 334:1205:294, and the accuracy rate is 77.83%. After further learning with smote data enhancement strategy, the accuracy of the fault type (T/I/P) is 89.71%, and the loss is stable at about 0.4. Figure 6 (d, e, f) shows the difference between the real value of the fault duration and the diagnosis value after the data enhancement. The sample proportion of fault parts is 284:884:666; the accuracy of fault location is 82.30%, and the loss is stable at about 0.4. See Table 3 column "fault location" for the accuracy of each fault location. See Figure 6 (a, b, c) for the true value and diagnosis value of fault diagnosis.

For PNN network, the transmission factor is set to 0.007 after tuning, and the best accuracy is 97.98%. Figure 7 shows the difference between the test result and the real value (the yellow line represents the real value and the blue line represents the predicted value). It can be seen that compared with the BP network, the diagnose performance of PNN is much more stable.

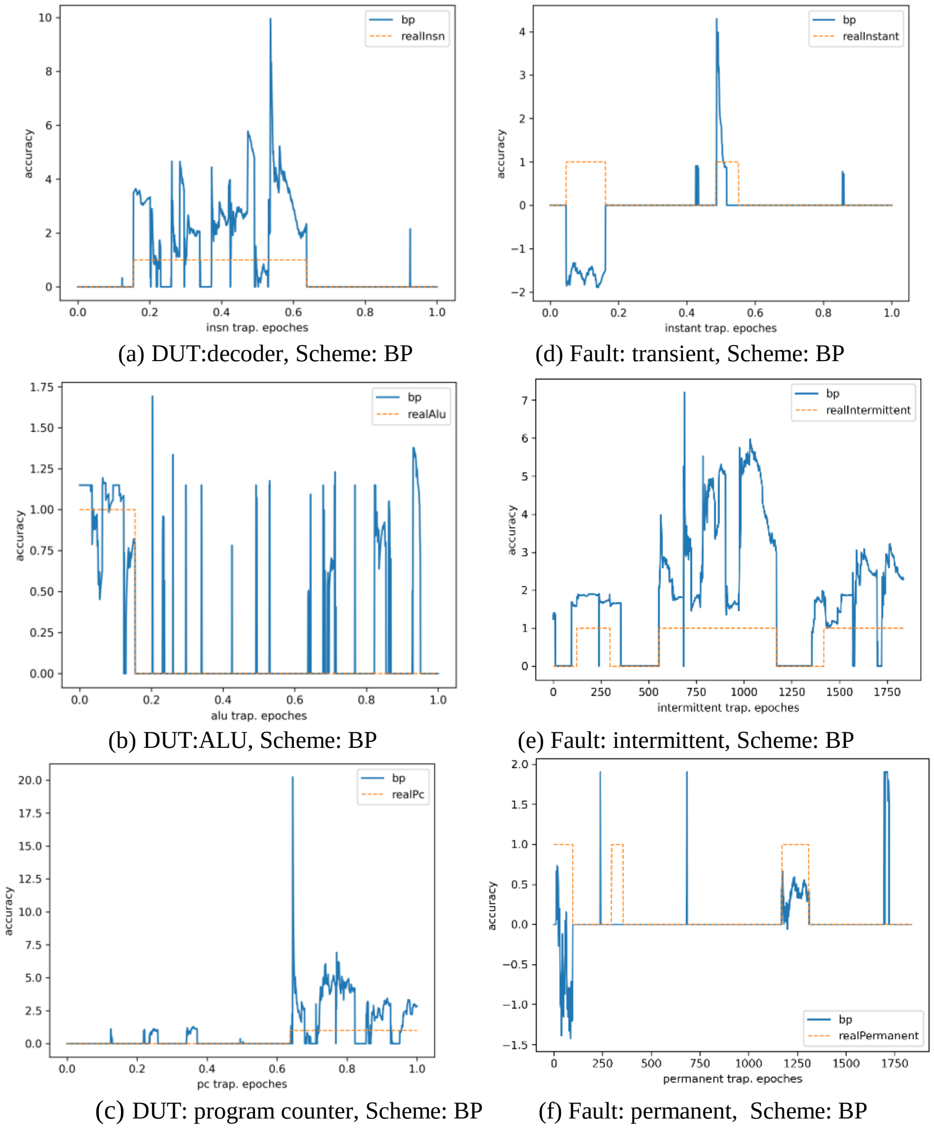
Figure 8 shows the training process of the LSTM network, in which the upper coordinate system represents the change of accuracy during training, the red line represents the change of test data loss value, and the blue line represents the change of training data loss value. The accuracy and loss changes of the BP network are shown in Figure 9. From the comparison, we can see that the BP network training process is stable, but the LSTM network training is more tough, in which the loss value and accuracy are changing unsteadily; then it can be concluded that the LSTM network is relatively poor in the ability to acquire knowledge from fault data compared with BP network.

Table 3. Diagnose accuracy of BP network.

Fault type (after SMOTE)			Fault location		
Fault type	Accuracy	Sample	DUT	Accuracy	Sample
transient	72.84%	334	decoder	75.09%	284
intermittent	97.68%	1205	ALU	84.18%	884
permanent	76.27%	294	PC	82.88%	666
total	**89.71%**		total	**82.30%**	

Table 4. Tuning of transmission factor in PNN network.

Factor	Accuracy
0.1	57.79%
0.05	72.06%
0.01	94.13%
0.008	95.64%
0.007	**97.98%**
0.006	**97.98%**



(a) DUT:decoder, Scheme: BP

(d) Fault: transient, Scheme: BP

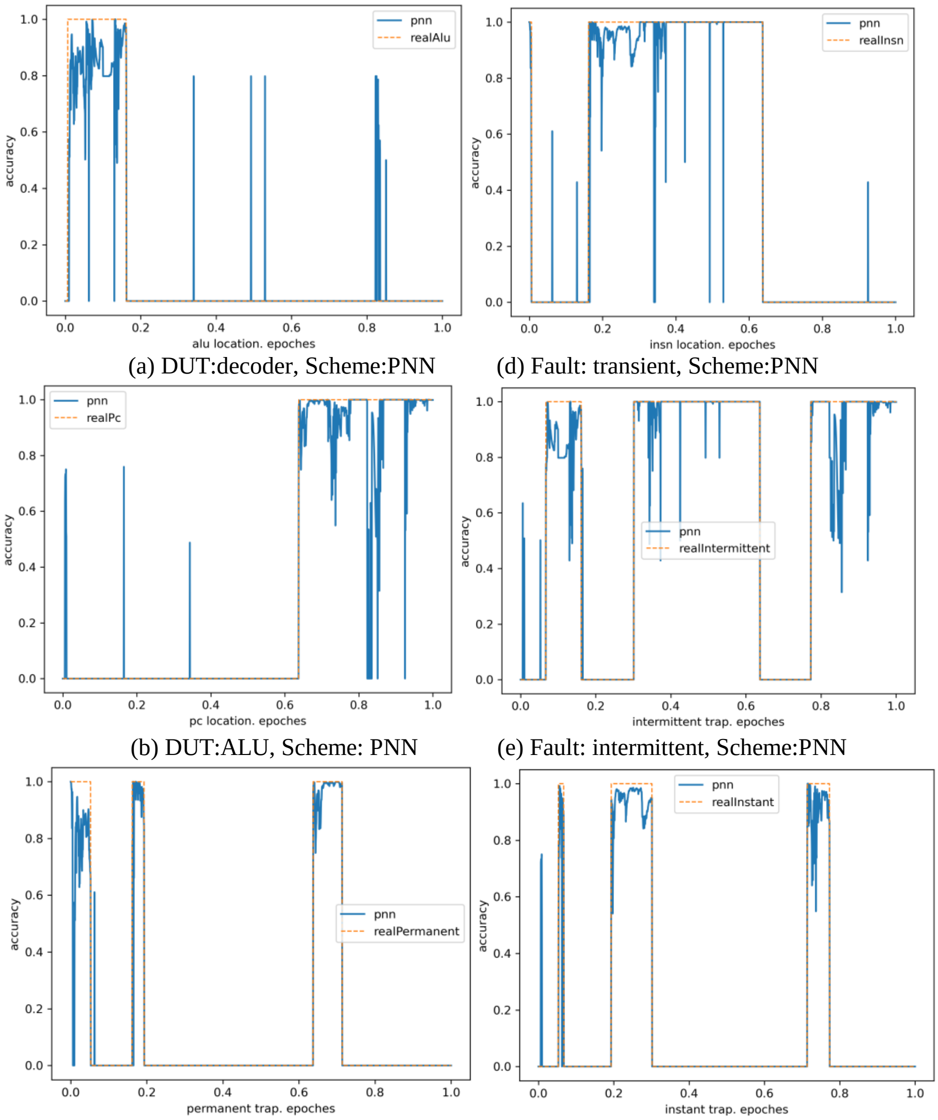
(b) DUT:ALU, Scheme: BP

(e) Fault: intermittent, Scheme: BP

(c) DUT: program counter, Scheme: BP

(f) Fault: permanent, Scheme: BP

Fig. 6. Diagnose result diagrams of BP network.



(a) DUT:decoder, Scheme:PNN (d) Fault: transient, Scheme:PNN

(b) DUT:ALU, Scheme: PNN (e) Fault: intermittent, Scheme:PNN

(c) DUT: program counter, Scheme: PNN (f) Fault: permanent, Scheme:PNN

Fig. 7. Diagnose result diagrams of PNN network.

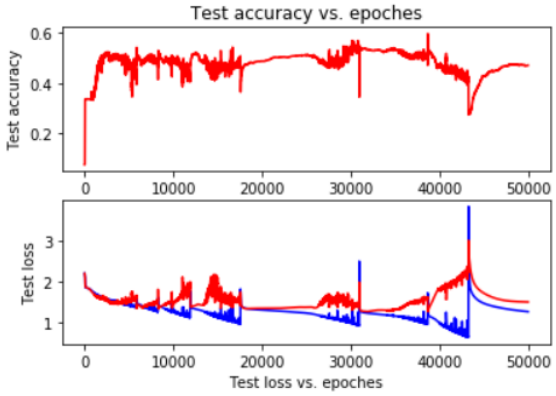


Fig.8. Test accuracy and loss of LSTM.

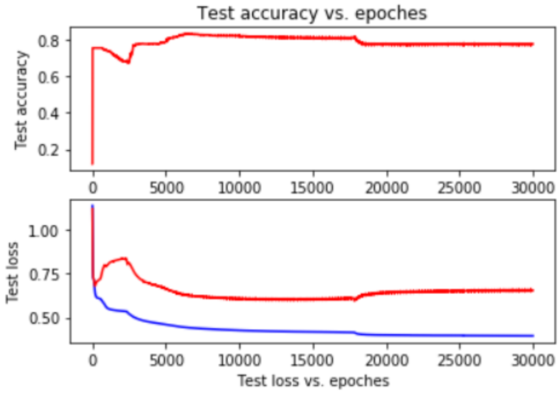


Fig.9. Test accuracy and loss of BP network.

The training process of the BP network is error back-propagation learning, and the basic requirement is that the error function has continuity (because it needs to ensure that the error function can be biased). Thus, the final fitting result of the BP network is a continuous function in multi-dimensional space; however, the general result of fault diagnosis is discontinuous: it is neither 0 nor 1, so the BP network has a larger error than the PNN network. In contrast, to classify data with the lowest risk, the PNN directly uses the Bayesian classification method based on the Gaussian density function. Hence, its output is either 0 or 1, so it has a higher fault diagnosis rate than the BP network.

The latency is 0.0952 seconds (BP) and 0.0286 seconds (PNN) alternatively for the online mode and 0.421 seconds (LSTM) for the offline mode. These algorithms run on Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, with 2 x 32 KB L1 cache and 2 x 256 KB L2 cache. The statistics do not include time for core dump. However, we observe that a considerable proportion of the system call traces of faulty instances is repeated in most of the cases, so there should be observable reduction in the latency. Accordingly, a software recovery mechanism is favorable. In addition, training of the network is done offline. Hence, there is no need to recompute the weights of each neural connection when performing the diagnosis, thus saving significant time.

5. Conclusion

In this paper, we propose an offline/online diagnosis mechanism for cloud system against intermittent faults. We take systematic-level behavior as a high-level representation of fault behavior. We implement an end-to-end neural network-based method that takes advantage of the log information to perform feature selection. Then, we set up a unified diagnosis framework based on LSTM/BP/PNN classifiers. Among the three classifiers, the PNN performs best in diagnosis accuracy. It employs the Bayesian probability analysis method to make fault category and fault location close to the actual label. The offline training/online diagnosis ensures that this method can be implemented in firmware, with zero hardware costs.

References

1. Gil, P., Arlat, J., Madeira, H. et al, Fault Representativeness. Deliverable ETIE2. DBench European Project (IST-2000-25425).
2. Nishant J. George, Carl R. Elks, Barry W. Johnson and John Lach. Transient fault models and AVF estimation revisited//In International Conference on Dependable Systems and Networks (DSN), Chicago, IL, 2010:477-486.
3. Process Integration, Devices and Structures, The International Technology Roadmap for Semiconductors, 2012.
4. C. Constantinescu. Impact of deep submicron technology on dependability of VLSI circuits. Proceedings of Dependable Systems and Networks Conference, 2002:205-209.
5. Gracia-Moran J, Gil-Tomas D, Saiz-Adalid L J, et al. Experimental validation of a fault tolerant microcomputer system against intermittent faults[C]// IEEE/IFIP International Conference on Dependable Systems & Networks. 2010.
6. Tharf M S H. Computer modeling of electromagnetic interference, radiation, and crosstalk in electronic systems[M]. 1993.
7. C. Constantinescu. Intermittent Faults in VLSI Circuits[C]. Proceedings of the IEEE Workshop on Silicon Errors in Logic - System Effects, 2007.
8. Rashid L, Pattabiraman K and Gopalakrishnan S. Intermittent hardware errors recovery: modeling and evaluation. In: 2012 ninth international conference on quantitative evaluation of systems (QEST), London, 17– 20 September 2012. New York: IEEE.
9. Raghavan V. On Asymmetric Invalidation with Partial Test[J]. IEEE Transactions on Computers, 1993, 42(6): 764-768.
10. Lafortune S, Sengupta R, Sampath M, et al. Failure Diagnosis of Dynamic Systems an Approach Based on Discrete Event Systems[C]. Proc of the American Control Conference, Arlington, USA, 2001: 2058-2071.
11. Deng Guanqian, Qiu Jing, Li Zhi, Yan Ning. A Survey on Intermittent Fault Diagnosis Technology[J].Ordnance Industry Automation, 2015-01, 34(1): 15-20.
12. Daniel Gil-Tomás, Joaquín Gracia-Morán, J.-Carlos Baraza-Calvo, Luis-J. Saiz-Adalid, and Pedro-J. Gil-Vicente. Studying the effects of intermittent faults on a microcontroller[J]. Elsevier Microelectronics Reliability, 2012, 11(52):2837-2846

13. Rohan Garg, Tirthak Patel, Gene Cooperman, Devesh Tiwari. Shiraz: Exploiting System Reliability and Application Resilience Characteristics to Improve Large Scale System Throughput [C]// IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE, 2018, pp. 83-94.
14. Guanpeng Li, Karthik Pattabiraman. Modeling Input-Dependent Error Propagation in Programs[C]// IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE, 2018, pp. 279-290.
15. Sam Ainsworth, Timothy M. Jones. Parallel Error Detection Using Heterogeneous Cores[C]// IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE, 2018, pp. 338-349.
16. S.-B. Park and S. Mitra. IFRA: Instruction footprint recording and analysis for post-silicon bug localization in processors[C]// DAC, 2008, pp. 373-378.
17. A. DeOrio, Q. Li, M. Burgess, and V. Bertacco. Machine learning-based anomaly detection for post-silicon bug diagnosis[C]// DATE, 2013, pp. 491-496.
18. J. Carretero, X. Vera, J. Abella, T. Ramirez, M. Monchiero, and A. Gonzalez. Hardware/software-based diagnosis of load-store queues using expandable activity logs[C]// HPCA, 2011, pp. 321-331.
19. Dadashi M , Rashid L , Pattabiraman K , et al. Hardware-Software Integrated Diagnosis for Intermittent Hardware Faults[C]// IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE, 2014.
20. Jiaqi Yan, Guanhua Yan, Dong Jin. Classifying Malware Represented as Control Flow Graphs using Deep Graph Convolutional Neural Network[C]// IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE, 2019, pp. 52-63.
21. Deng Guanqian, Jing Qiu, Liu Guanjun, et al. A Discrete Event Systems Approach to Discriminating Intermittent from Permanent Faults[J]. Chinese Journal of Aeronautics, 2014, 27(2): 390-396.
22. Deng Guanqian, Jing Qiu, Liu Guanjun, et al. A Stochastic Automaton Approach to Discriminate Intermittent from Permanent Faults[J]. Journal of Aerospace Engineering, 2014, 228(6): 880-888.
23. Maurice G, Diaz F, Coti C, et al. Downtime statistics of current cloud solutions, 2014, http://iwgcr.org/wp-content/uploads/2014/03/downtime-statistics-current-1.3.pdf
24. Baraza J C, Gracia J, Gil D, et al. A Prototype of a VHDL based Fault Injection Tool: Description and Application [J]. Journal of Systems Architecture, 2002, 47 (10) : 847-867.
25. Zarandi H R, Miremadi G, Ejlali A R. Fault Injection into Verilog Models for Dependability Evaluation of Digital Systems[C]// Proceedings of the International Symposium on Parallel and Distributed Computing, IEEE Press, 2003: 281-287.
26. Sieh V, Tschche O, Balbach F. VERIFY: Evaluation of Reliability Using VHDL Models with Embedded Fault Descriptions[C]// Proceedings of the 27th International Symposium on Fault-tolerant Computing. Seattle, USA: IEEE Press, 1997: 32-36.
27. Chao Wang, Zhongchuan Fu, Hong-Song Chen, Gang Cui. Characterizing the Effects of Intermittent Faults on a Processor for Dependability Enhancement Strategy[J]. The Scientific World Journal, 2014:1-12.
28. Chao Wang, Wei Zhang. Intermittent fault injection platform implemented in register transfer level[J]. Journal of Beijing Information Science & Technology University, 2015 (30): 46-50.
29. Rashid L., Pattabiraman K., Gopalakrishnan S.. Characterizing the Impact of Intermittent Hardware Faults on Programs[J]. IEEE Transactions on Reliability, 2015, 64(1):297-310.
30. Radha Venkatagiri, Khalique Ahmed, Abdulrahman Mahmoud, Sasa Misailovic, Darko Marinov, Christopher W. Fletcher, Sarita V. Adve. gem5-Approxilyzer: An Open-Source Tool for Application-Level Soft Error Analysis[C]// IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE, 2019, pp. 214-221.

31. Qian Hu, etc al. Simics-based System Level Fault Injection Platform[J]. Computer Engineering, 2015(41):57-63.
32. Le M and Tamir Y. Fault injection in virtualized systems-challenges and applications. IEEE T Depend Secure 2015, 12(3): 284-297.
33. Chao Wang, Zhongchuan Fu. Quantitative evaluation of fault propagation in a commercial cloud system[J]. International Journal of Distributed Sensor Networks, 2020,16(3):1-11.
34. Hochreiter S , Schmidhuber J . Long Short-Term Memory[J]. Neural computation, 1997, 9(8):1735-1780.
35. M. L. Li, P. Ramachandran, S. Sahoo, S. Adve, V. Adve, and Y. Zhou. Trace based microarchitecture-level diagnosis of permanent hardware faults[C]//IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE, 2008, pp. 22-31.

Wang Chao is an Assistant Professor at the School of Computer Science in Beijing Information Science and Technology University of China. His main research interests are directed to reliability qualification of cloud computing using simulation and fault injection, and high performance computing in deep learning model accelerator development. He is also interested in unmanned ground vehicle and system, including objection detection, navigation and decision making.

Fu Zhongchuan is a vice professor in Computer Science department in Harbin Institute of Technology of China. His main research interests involve fault injection simulator development, and quantitative analysis of reliability technology in cloud computing environment. He is also interested in high performance computing in CPU model development.

Huo Yanyan is an algorithm engineer graduated from the School of Computer Science in Beijing Information Science and Technology University of China. She is interested in deep learning and AI area.

Received: June 20, 2020; Accepted: November 15, 2020.

Computer Science and Information Systems 18 (3):791–811 https://doi.org/10.2298/CSIS200430037M

Distance based Clustering of Class Association Rules to Build a Compact, Accurate and Descriptive Classifier

Jamolbek Mattiev[1,3] and Branko Kavšek[1,2]

[1] University of Primorska,
Glagoljaška 8, 6000 Koper, Slovenia
jamolbek.mattiev@famnit.upr.si, branko.kavsek@upr.si
[2] Jožef Stefan Institute,
Jamovacesta 39, 1000 Ljubljana, Slovenia
branko.kavsek@ijs.si
[3] Urgench State University,
Khamid Alimdjan 14, 220100 Urgench, Uzbekistan
jamolbek_1992@mail.ru

Abstract. Huge amounts of data are being collected and analyzed nowadays. By using the popular rule-learning algorithms, the number of rule discovered on those "big" datasets can easily exceed thousands. To produce compact, understandable and accurate classifiers, such rules have to be grouped and pruned, so that only a reasonable number of them are presented to the end user for inspection and further analysis.

In this paper, we propose new methods that are able to reduce the number of class association rules produced by "classical" class association rule classifiers, while maintaining an accurate classification model that is comparable to the ones generated by state-of-the-art classification algorithms. More precisely, we propose new associative classifiers, called DC, DDC and CDC, that use distance-based agglomerative hierarchical clustering as a post-processing step to reduce the number of its rules, and in the rule-selection step, we use different strategies (based on database coverage and cluster center) for each algorithm. Experimental results performed on selected datasets from the UCI ML repository show that our classifiers are able to learn classifiers containing significantly fewer rules than state-of-the-art rule learning algorithms on datasets with a larger number of examples. On the other hand, the classification accuracy of the proposed classifiers is not significantly different from state-of-the-art rule-learners on most of the datasets.

Keywords: Frequent Itemset, Class Association Rules (CAR), Associative Classification, Agglomerative Clustering.

1. Introduction

Huge amounts of data are being collected and stored nowadays in many world applications. Mining association rules from those datasets and reducing is becoming a popular and important knowledge discovery technique [1]. A huge number of rules are being discovered in "real-life" datasets that will lead to combinatorial complexity. To overcome this problem, rules have to be pruned and clustered while the compact, accurate and understandable classifier (model) is being built to reduce the number of rules.

Association rule (AR) mining [2] aims to generate all existing rules in the database that satisfy some user-defined minimum support and confidence thresholds, while classification rule mining tries to extract a small subset of rules to form accurate and efficient models to predict the class label of unknown objects. Associative Classification (AC) is a combination of these two important data mining techniques, namely, classification and association rule mining [3]. Recently, researchers have proposed several associative classification methods [4-11] that aim to build accurate and efficient classifiers based on association rules. Research studies prove that AC methods could achieve higher accuracy than some of the traditional classification methods, although the efficiency of AC methods depends on the user-defined parameters such as minimum support and confidence. Other important approaches are clustering methods (unsupervised learning) studied in [12-14]. These clustering techniques are split into two main parts: partitional and hierarchical clustering. In partitional clustering [15,16], objects are grouped into disjoint clusters such that objects in the same cluster are more similar to each other than objects in another cluster. Hierarchical clustering [17], on the other hand, is a nested sequence of partitions. In the bottom-up method, larger clusters are built by merging smaller clusters, while the top-down method starts with the one cluster containing all objects and divides into smaller clusters.

In this research work, we propose new associative classification methods based on hierarchical agglomerative clustering (complete linkage). We define the new normalized distance metrics based on direct and indirect measures to measure the similarities between CARs, which we later use to cluster CARs in a bottom-up hierarchical agglomerative fashion (firstly, we group the class association rules based on their class label and then rules that are in the same group are clustered together). Once we cluster the rules, the natural number of clusters is identified for each group of CARs by cutting the dendrogram from the point that achieves the maximum difference between two consecutive cluster heights.

Once CARs are clustered, we define a “representative” CAR within each cluster. We propose two methods of extracting the “representative” CAR for each cluster, (1) we choose the CAR based on database coverage and (2) based on cluster center.

We have performed experiments on 14 selected datasets from the UCI Machine Learning Database Repository [18] and compared the performance of our proposed methods with the 8 most popular associative and classical classification algorithms (Decision Table and Naïve Bayes (DTNB) [19], Decision Table (DT) [20], FURIA (FR) [21], PART (PT) [22], C4.5 [23], CBA [3], Ripple Down Rules (RDR) [24], Simple Associative Classifier (SA) [25]).

The rest of the paper is organized as follows. Section 2 highlights the related work to our research work. The problem statement and our goals are provided in section 3. Our proposed method is described in section 4. Section 5 highlights the experimental evaluation. Conclusions and future plans are given in Section 6. The Acknowledgement and References close the paper.

2. Related Work

The novelty in our proposed approach is in the way we select “strong” class association rules, how we cluster them and how we choose the “representative” class association rule for each cluster. Other related research also deals with the notion of clustering CARs, but all of them use different approaches. This section presents these related approaches to clustering CARs and emphasizes the similarities and differences related to our proposed approach.

In [26], researchers have proposed a new method to cluster the association rules by K-means (partitional) clustering algorithm. The main goal of this research is the clustering of discovered association rules to make it easy for users to choose the best rules. The algorithm is divided into 4 steps: (1) ARs generated from the frequent pattern by the “APRIORI” algorithm is extracted; (2) interestingness measures such as Lift, Cosinus, Conviction and Information Gain are computed for all rules generated in step 1; (3) a set of association rules is partitioned into disjoint clusters by using K-means algorithm; they try to cluster the rules which have the smallest similarities degree between them, and Euclidian and Degree of similarity distances are used to apply the K-means algorithm; (4) finally, they classify the group of rules from the best to the worst by using a centroid of each cluster. Our proposed method uses the hierarchical agglomerative approach for clustering CARs instead of k-means and employs a different way of selecting “strong” CARs in the first place. The distance metric used by our approach during clustering is also different.

The FURIA (Fuzzy Unordered Rule Induction Algorithm) [21] is a rule based classification method which is a modified and extended version of RIPPER [32] algorithm. FURIA learns fuzzy rules instead of conventional rules and unordered rule sets (namely a set of rules for each class in a one-vs-rest scheme) instead of rule lists. Moreover, to deal with uncovered examples, it makes use of an efficient rule stretching method. The idea is to generalize the existing rules until they cover the example.

In [33], we produced a relatively simple, descriptive and accurate classifier (J&B) by exhaustively searching the entire example space. More precisely, we select the strong class association rules according to their contribution for improving the overall coverage of the learning set. J&B has a stopping criterion in the rule-selection process based on the training dataset’s coverage. In our current research, we just applied J&B method without stopping criterion in the representative CAR-selection process. Since the number of clusters (the size of the classifier) is identified with different strategy, we don’t need to apply stopping criterion in this approach.

Another approach is conditional market-basket difference (CMBP) and conditional market-basket log-likelihood (CMBL) methods proposed in [27]. This approach uses a new normalized distance metric to group association rules. Based on the distances, agglomerative clustering is applied to cluster the rules. The rules are further embedded in a vector space with the use of multi-dimensional scaling and clustered using self-organizing maps. This method is very similar to ours, but we propose a new normalized distance metric based on “direct” and “combined” distances between class association rules, whereas “indirect” measures are used based on CARs support and coverage.

Mining clusters with ARs [28] is another related approach. Here the rules are first generated using the “APRIORI” algorithm, but an “indirect” distance metric (based on coverage probabilities) is later used to find the similarities between rules. Rules are then clustered using a top-down hierarchical clustering method for finding clusters in a population of customers, where the list of products bought by the individual clients is given. Once the rules are clustered, a specific distance metric is introduced to measure the quality of the clustering.

Another interesting clustering-based approach [29] is "Tightness" which quantifies the strength of binding between the items of an association rule. The idea is that certain items in the application domain might get bound together because they are so strongly correlated that they often occur together in transactions. This tightness of binding is not covered by traditional measures like support or confidence. They build their distance function based on indirect measures, that is, the items in AR that obtain the maximum and minimum support. Our proposed methods are all utilized different distance metrics based on "direct" and "combined" measures.

The absolute market-basket difference (AMBD) approach discussed in [30] also aims to cluster the ARs. The CAR-generation part of this method is similar to ours. The procedure is similar except for the clustering part. They focus on clustering the sorted (support and confidence based) association rules with the same consequent. The key differences are in the clustering part: indirect distance metric is used to cluster the rules (the distance gives the percentage of examples in the dataset that are not covered by both rules), whereas our methods use different distance metrics.

3. Problem Definition and Contributions

We assume that a normal relational table is given with *N* examples (transactions). Each example is described by *A* distinct attributes and is classified into one of the *M* known classes. Since our algorithm supports just attributes of a nominal type (like the vast majority of association rule miners), we had to perform discretization on numeric attributes in some cases (the details about discretization are provided in Section 5). The goals and contributions of our proposed approach are the following:

1. Generate "strong" CARs that satisfy some user-defined minimum support and minimum confidence constraints;
2. Propose new normalized similarity measures based on the "direct" and "indirect" distances between two class association rules;
3. Cluster class association rules by using this normalized similarity measure and automatically determine the optimal number of clusters for each class value;
4. Define two methods of extracting a representative CAR for each cluster to produce the final, compact and meaningful classifier;
5. Experimentally, show the usefulness of our methods in reducing the number of CARs, while retaining the classifier's accuracy.

4. Proposed Method

Our approach (Compact, Accurate and Descriptive Associative Classifier) can be divided into 4 steps mentioned in the previous section. Each of these 4 steps is presented in detail in the following 4 subsections.

4.1. Generating the Strong Class Association Rules

We discuss how to discover the strong CARs from frequent itemsets in this subsection. Generation of ARs is usually split up into two main steps: first, minimum support is applied to find all frequent itemsets from the training dataset; second, we use these frequent itemsets and minimum confidence to generate strong association rules. Discovering of CARs is also followed to the same procedure as in AR-generation. The only difference is that in the rule-generation part, the consequence of the rule contains only the class label in CAR-generation while the consequence of rule in AR-generation can include any frequent itemset.

In the first step, the “APRIORI” algorithm is used to find the frequent itemsets. The ‘downward-closure’ technique is used in the “APRIORI” algorithm to accelerate the searching procedure by reducing the number of candidate itemsets at any level. The “APRIORI” finds the 1-frequent itemset, then, the 1-frequent itemset is used to generate the 2-frequent itemset and so on. If it finds any infrequent itemsets at any level, it is removed in place, because infrequent itemsets cannot generate frequent itemset. The “APRIORI” performs this process before computing their support at any level to reduce the time complexity of the algorithm.

After all frequent itemsets are generated from the training datasets, it is straightforward to generate the strong CARs that satisfy both minimum support and minimum confidence constraints from frequent itemsets found in the first step. Confidence of the rule can be computed by the following formula:

$$confidence(A \rightarrow B) = \frac{support\ \_count\ (A \cup B)}{support\ \_count\ (A)}. \quad (\mathbf{1})$$

The equation (1) is expressed by support count of itemset, where *A* is a premise (itemset in the left-hand side of the rule), *B* is a consequence (class label in the right-hand side of the rule), *support_count*$(A \cup B)$ is the number of transactions that matches the itemsets $A \cup B$, and *support_count*(*A*) is the number of transactions that matches the itemsets*A*. Strong class association rules that satisfy the minimum confidence threshold can be generated based on the above equation, as follows:

- All nonempty subsets *S* are generated for each frequent itemset *L* and a class label *C*;
- For every nonempty subset *S* of *L*, output the strong rule R in the form of “*S*→*C*” if, $\frac{support\ \_count\ (R)}{support\ \_count\ (S)}$ ≥*min_conf*, where *min_conf* is the minimum confidence threshold.

4.2. Distance Metrics

Once we generate strong class association rules in 4.1, our next goal is to cluster them. Since we intend to apply hierarchical agglomerative clustering, we must define a way of measuring the similarity between CARs, that is, how far two rules are apart. Unfortunately, there is not any distance metric for CARs. However, researchers have proposed some indirect distance metrics for association rules, namely, Absolute Market

Basket Difference (AMBD), Conditional Market-basket Probability (CMBP), and Conditional Market-basket Log-likelihood (CMBL) [27] and "Tightness" [29].

Indirect Distance Metrics

We highlight the indirect distance metric for association rules in this section. We call rule distances that are obtained from the data Indirect Distance Metrics. An indirect distance is defined as a function of the market-basket sets that support the two considered rules.

To begin with a simple distance measure (AMBD) introduced to compute the similarity between association rules. Let $rule1{:}\, A \Rightarrow C$ and $rule2{:}\, B \Rightarrow C$ be two association rules, the distance is defined between rules in terms of the number of market-baskets that they differ in (meaning one rule is supported, but not the other). Based on the number of non-overlapping market-baskets, a distance metric d^{AMBD} between *rule1* and *rule2* can be defined by the following equation:

$$d_{rule\,1,rule\,2}{}^{AMBD} = |m(BS_{rule\,1})| + |m(BS_{rule\,2})| - 2 * |m(BS_{rule\,1}, BS_{rule\,2})| \qquad \textbf{(2)}$$

Where, *BS* is the both side of the rule, that is, the itemset for the association rule. *m(BS)* denotes the set of transactions (baskets) matched by BS and |m(BS)| is the number of such transactions.

Equation (2) illustrates that rules valid for exactly the same baskets have a distance of zero. Rules applying to disjoint sets of baskets have a distance equal to the sum of the numbers of transactions for which each rule is valid. There are several problems with this measure. One such problem is that it grows as the number of market-baskets in the database increases. This can be corrected by normalizing (dividing the measure by the size of the database) and it is appropriate for rules only with the same consequent while this approach is intuitive. However, the measure is still strongly correlated with support. High support rules will on average tend to have higher distances to everybody else. This is an undesired property. For example, two pairs of rules, both pairs consisting of non-overlapping rules, may have different distances. High support pairs have a higher distance than low support pairs.

Based on the previous approach, another indirect distance measure in the CMBP method is proposed as an improvement of AMBD using the support values of two association rules. Researchers tried to solve two problems that occurred in AMBD: (1) it grows as the database grows, and (2) due to the focus on support values, rules with high support will on average tend to have higher distances to everybody else. To solve the above-mentioned problems, they proposed the new indirect distance metric based on conditional probabilities. Using a probability estimate for distance computation has many advantages. Probabilities are well understood, are intuitive, and a good measure for further processing. The distance d^{CMBP} between two rules *rule1* and *rule2* is the (estimated) probability that one rule does not hold for a basket, given at least one rule holds for the same baskets. This distance is defined as follows:

$$d^{CMBP}_{rule\,1,rule\,2} = 1 - \frac{|m(BS_{rule\,1}, BS_{rule\,2})|}{|m(BS_{rule\,1})| + |m(BS_{rule\,2})| - |m(BS_{rule\,1}, BS_{rule\,2})|} \qquad \textbf{(3)}$$

With this metric, rules having no common market baskets are at a distance of 1, and rules valid for an identical set of baskets are at a distance of 0. The CMBP does not suffer from the support correlation problem of AMBD. Let us call a distance interesting if it is neither 0 nor 1. Rule pairs with an interesting distance are called good neighbors. In most real databases, the majority of all rule pairs are not good neighbors. Manual exploration of a rule's good neighbors showed that intuitive relatedness was captured very well by this metric. For example, rules involving different items but serving equal purposes were found to be close good neighbors. Super-set relationships of the itemsets associated with the rules often lead to very small distances.

The new "Direct" Distance Metric

We compute the distance between two rules by ignoring the class label because we are clustering the rules belonging to the same class label. When we apply indirect distance measures to our proposed method, we get a larger natural number of clusters, that is, the classifier includes a larger number of rules. Therefore, we propose a new normalized Item Based Distance Metric (IBDM) in this research work by considering the differences in rule items.

Let $R = \{r_1, r_2, \ldots.., r_n\}$ be a set of class association rules found from relational dataset D that are defined by $A = \{a_1, a_2, \ldots.., a_m\}$ distinct items (attribute's value) classified into $C = \{c_1, c_2, \ldots.., c_l\}$ known classes. Each rule is denoted as follows:
$r = \{x_1, x_2, \ldots.., x_k\} \rightarrow \{c\}$ where, $\{x_1, x_2, \ldots.., x_k\} \subseteq A$ and $c \in C$ for $\forall\, r \in R$.
Given two rules $rule1, rule2 \in R$:

$$rule1 = \{y_1, y_2, \ldots.., y_s\} \rightarrow \{c\}$$
$$rule2 = \{z_1, z_2, \ldots.., z_t\} \rightarrow \{c\}$$

Where$\{y_1, y_2, \ldots.., y_s\} \subseteq. A$, $\{z_1, z_2, \ldots.., z_t\} \subseteq. A$, and $c \in C$. We compute the similarity between $rule1$ and $rule2$ as follows:

$$distance_q(rule1, rule2) = \begin{cases} if\ y_q = z_q \mid y_q = \emptyset \,\&\, z_q = \emptyset\,, 0; \\ else\ if\ y_q = \emptyset \,\&\, z_q \neq \emptyset \mid y_q \neq \emptyset \,\&\, z_q = \emptyset\,, 1; \\ else\ \ 2\ (y_q \neq z_q). \end{cases} \qquad \textbf{(4)}$$

Where q is the index of rule items that cannot exceed from *border* value (defined below).

Equation (4) expresses how close two rules are one from another. If rules have similar items, then the distance function has a low value. An empty rule item is considered closer than a different rule item.

$$border = Max(s, t); \qquad \textbf{(5)}$$

border is the length of the longest rule, (5) is used to normalize the distance metric. The distance between two rules is denoted as follows:

$$d^{IBDM}_{rule\,1, rule\,2} = \sum_{i=1}^{border} distance_i / 2 \times border \qquad \textbf{(6)}$$

Distance (6) ranges between 0 and 1. CARs having the same items and the same size are at a distance of 0, CARs containing the different items are at a distance of 1.

The new "Combined" Distance Metric

Since conditional market-basket distance is appropriate for the rules having the same consequent, we decided to propose a new Weighted and Combined Distance Metric (WCDM) by combining direct (IBDM) and indirect distance (CMBP) measures. When we apply CMBP distance to our proposed method, we got a larger number of clusters on some datasets. WCDM combines direct measure (rule items) and indirect measure (rule coverage). Both distance metrics (IBDM and WCDM) have their advantage: on some datasets IBDM produces lower number of rules with higher accuracy while WCDM achieves better results on some other datasets. The weighted distance d^{WCDM} between two rules *rule1* and *rule2* is defined as follows:

$$d_{rule\,1,rule\,2}^{WCDM} = \alpha \times d_{rule\,1,rule\,2}^{IBDM} + (1-\alpha) \times d_{rule\,1,rule\,2}^{CMBP} \qquad \textbf{(7)}$$

where, α is a weight parameter. We set $\alpha = 0.5$ parameter, the final weighted and combined distance measure is described as follows:

$$d_{rule\,1,rule\,2}^{WCDM} = 0.5 \times d_{rule\,1,rule\,2}^{IBDM} + 0.5 \times d_{rule\,1,rule\,2}^{CMBP}. \qquad \textbf{(8)}$$

4.3. Clustering

Clustering algorithms aim to group similar examples; the examples in the same cluster should be similar and dissimilar to the examples in other clusters. There are two types of hierarchical clustering algorithms: top-down (*divisivehierarchical clustering*) and bottom-up (*hierarchical agglomerative clustering*). Bottom-up algorithms initially assume each example as a single cluster and then merge the two closest clusters in every iteration until all clusters have been merged into a unique cluster that contains all examples. The resulting hierarchy of clusters is represented as a tree (or dendrogram). The root of the tree is the unique cluster that gathers all the examples; the leaves are considered as clusters with only one sample. The top-down approach is the opposite of the hierarchical agglomerative clustering method. It considers all examples in a single cluster, and then it splits the clusters into smaller parts until each example forms a cluster or until it satisfies the stopping condition.

We apply the complete linkage method ofhierarchical agglomerative clustering. In the complete linkage (farthest neighbor) method, the similarity of two clusters is the similarity of their most dissimilar examples, therefore, the distance between the farthest groups are taken as an intra-cluster distance. We assume that we have given $N \times N$ distance matrix *D*, where *N* is the total number of rules (that is, total number of clusters). The clusters are numbered 0,1,..,(*N*-1) and *m* is the sequence number of clusters. *L[k]* is the level of the *k*-th clustering and the distance between two clusters *cl1* and *cl2* is defined as *D[cl1,cl2]*. Complete linkage of the hierarchical agglomerative clustering algorithm is outlined in algorithm 1.

We need to apply hierarchical clustering algorithm twice: first, we apply *AHCCLH* algorithm to find the cluster heights that we will use later to identify the optimal number of clusters. In this case, number of cluster *S=1* and distance matrix are defined as input parameters. Because if *S=1*, then, *AHCCLH* iterates *N-1* times to find the heights of all the clusters. Second, *AHCCLC* is utilized to identify the cluster of class association rules. In *AHCCLC,* we provide the number of cluster *S* (found by using the cluster heights) and distance matrixto identify the clusters of CARs.

Algorithm 1: Agglomerative Hierarchical Clustering with Complete Linkage (AHCCLH: Heights || AHCCLC: Clusters)

Input: a distance matrix *D* and number of clusters *S*
Output: Cluster heights (AHCCLH), Cluster of CARs (AHCCLC)

1. **Initialization:** Each rule is a unique cluster *C* at level 0 *(L[0]=0)*, sequence number *m*=0 and the optimal number of cluster *S* is identified, so, to get the intended number of clusters (*S*), the algorithm should iterate *K* times *K*=*N*–*S*;
2. **Compute:** Find the most similar pair of clusters, *cl1* and *cl2*and merge them into a single cluster *C* to form the next clustering sequence *m*.Increase the sequence number by one: *m*=*m*+1 and set the new level *L[m]=D[cl1,cl2]*;
3. **Update:** Update the distance matrix *D*, by removing the rows and columns corresponding to *cl1* and *cl2* and adding a new row and column corresponding to the new cluster.The distance between the new cluster (*cl1*,*cl2*) and old cluster *k* is calculated as *D[(cl1, cl2),k]=max*{*D[k,cl1], D[k,cl2]*};
4. **Stopping condition:** if *m=K* then **return***L* (*AHCCLH*) || *C* (*AHCCLC*)and stop, otherwise go to step 2.

When we cluster the rules, we need to find the number of clusters. We get the "natural" number of clusters by "cutting" the dendrogram at the point that represents the maximum distance between two consecutive cluster merges. The algorithm that identifies the "natural" number of clusters is presented in Algorithm 2.

Algorithm 2: Computing the optimal number of clusters

```
  Input: an array of cluster heights
  Output: Optimal number of cluster
1:    Max_height_difference=cluster_height[1]-cluster_height[0];
2:    Opt_number_of_cluster= 1;
3:    N=cluster_height.length;
4:    for (k=2; k≤ N; k++) do begin
5:        if (cluster_height[k]-cluster_height[k-1])>Max_height_difference then
6:              Max_height_difference= cluster_height[k]-cluster_height[k-1];
7:                Opt_number_of_cluster=N-k;
8:        end if
9:    end for
10:   return Opt_number_of_cluster
```

The input to Algorithm 2 is a set of cluster distances that are calculated during the building of the dendrogram (so-called cluster "heights"). The output is the "natural" number of clusters. In lines 1-3 the total number of clusters generated by hierarchical clustering is stored. Lines 4-7 outline the main part of the algorithm, *Opt_number_of_clusters*gets to the point where the difference between two consecutive cluster heights will be maximum. Since we start from *0, Opt_number_of_clusters*is equal to *(N-k)*. The last line returns the obtained result.

4.4. Extracting the Representative CAR

Once we found all clusters, our final goal is to extract the representative CAR for each cluster to form our meaningful, compact and descriptive associative classifier. In this research work, we propose two methods of extracting the representative CAR for each cluster.

Representative CAR based on Cluster Center

In this method, we choose the CAR which is closer to the center of the cluster as a representative, that is, the representative CAR must have the minimum average distance to all other rules. Algorithm 3 defines the procedure.

Algorithm 3: A Representative CAR based on Cluster Center (RCC)

```
     Input: a set of class association rules in CARs array
     Output: Arepresentative class association rule
1:     N=CARs.length;
2:     Fill(Dist, 0);
3:     min_avg_distance=Integer.Max.value;
4:     for (i=0; i≤ N;i++) do begin
5:        for (j=0; j≤ N;j++) do begin
6:           Dist[i]=Dist[i]+IBDM(CARs[i], CARs[j]) | WCDM(CARs[i], CARs[j]);
7:        end for
8:        avg_distance=Dist[i]/N;
9:        if (avg_distance<min_avg_distance then
10:           min_avg_distance= avg_distance;
11:           representative_CAR_index=i:
12:       end if
13:    end for
14:    return CARs[representative_CAR_index];
```

The first line gets the number of CARs. We use the distance array "*Dist*" (line 2) to compute the distance from the selected CAR to all other CARs (we use one of the distance measures described in section 4). Initial value of*min_avg_distance* in line 3 is the maximum value of the integer and it is used to store the minimum average distance in line 10. Lines 4-9 find the index of the representative CAR that has the minimum average distance to all other rules and the last line returns the representative CAR.

Representative CAR based on Database Coverage

We decided to propose this method to improve the overall coverage and classification accuracy, while the first method (RCC) suffers to achieve reasonable coverage on some certain datasets. Since we are clustering similar rules having the same class value, it is unnecessary to think about the outer-class overlapping problem (that means some samples from different classes have very similar characteristics), but we should avoid the inter-class overlapping problem (several rules that belong to same class may cover the same samples). We bypass this problem by selecting the representative CAR based on database coverage. First, we find a rule that has maximum database coverage, then we check if the first CAR classifies at least one new example, then we get it as a representative CAR, otherwise we continue. Once we find the representative, we remove all the examples covered by a representative CAR. The procedure is outlined in algorithm 4.

Algorithm 4: A Representative CAR based on Database Coverage(RDC)

```
     Input: a set of class association rules in CARs array, a training dataset D
     andclassified_traindataarray
     Output: Arepresentative class association rule
 1:      N=CARs.length;
 2:   CARs= sort(CARs, coverage);
 3:   Representative_CAR = CARs[1]:
 4:   for i:=1 toN do begin
 5:        for j:=1 toD.length do begin
 6:            if classified_traindata[j]=false then
 7:                if CARs[i] classifies D[j] (e.g. CARs[i].premise⊆D[j].premise) then
 8:                        classified_traindata[j]=true;
 9:                        contribution=contribution+1:
10:                end if
11:            end if
12:        end for
13:        ifcontribution>0 then
14:            Representative_CAR = CARs[i]:
15:            break;
16:        end if
17:   end for
18:   return Representative_CAR;
```

In this approach (RDC), we first sort (line 2) the class association rules in coverage descending order, and we start checking rules from first to last (line 4). If a rule classifies at least one new example (line 13), that is, if CAR premise (left-hand side of the rule) matches the premise of the training dataset (line 7), we return that rule as a representative (line 13-18), otherwise we continue. If any rule cannot be a representative, then, the algorithm returns the first rule (line 3) which has the highest coverage as a representative.

Finally, our proposed approach is represented in algorithm 5.

Lines 1-2 generate the strong CARs that satisfy the user-specified minimum support and minimum confidence constraints from training dataset D by using the “APRIORI” algorithm. The third line sorts the CARs in confidence and supports descending order according to the following criteria:

$R_1$and R_2 are two CARs, R_1 is said to have a higher rank than R_2, denoted as $R_1 > R_2$,

- If and only if, $conf(R_1) > conf(R_2)$; or
- If $conf(R_1) = conf(R_2)$ but, $supp(R_1) > supp(R_2)$; or
- If $conf(R_1) = conf(R_2)$ and $supp(R_1) = supp(R_2)$, R_1has fewer attribute values in its left-hand side than R_2does;
- If all the parameters of the rules are equal, we can choose any of them.

Algorithm 5: Learning the proposed Associative Classifier

```
    Input: A training dataset D, minimum support and minimum confidence
    Output: Associative classifier
1:    F=frequent_itemsets(D, minsup);
2:   R=genCARs(F, minconf);
3:   R=sort(R, minsup, minconf);
4:   G=Group(R);
5:    for (i=0; i≤ number_of_class;i++) do begin
6:       Distance=IBDM(G[i]) | WCDM(G[i]);
7:       Cluster_heights=AHCCLH(Distance, 1);
8:       N=optimal_number_of cluster(Cluster_heights);
9:       Cluster=AHCCLC(Distance, N);
10:      Fill(classified_traindata, false);
11:      for (j=0; j≤ N;j++) do begin
12:           Y= RDC(Cluster[i], D, classified_traindata) | RCC(Cluster[i]);
13:           Associative_Classifier.add(Y);
14:       end for
15:   end for
16:    return Associative_Classifier;
```

CARs are grouped according to their class label in line 4. For each group of CARs (lines 5-15), the distance matrix is constructed by using one of the distance measures defined in subsection 4.2 (line 6), the hierarchical clustering algorithm complete linkage method (*AHCCLH*) computes the cluster heights (distances between clusters) by using the distance matrix in line 7 and these heights (distances) are used to find the optimal number of clusters (line 8). Then, we apply the hierarchical clustering algorithm (*AHCCLC*) again to identify the clusters of CARs (a *Cluster* array stores the list of clustered CARs). Since we are clustering the class association rules class by class, we need a "*classified_traindata*" array to store the information about classified examples, that is, we update this array for the same class only. When we start the clustering of CARs for a new class, we need to initialize the "*classified_traindata*" array. In lines 11-14, the representative CAR is extracted by using one of the methods described in section 4.4 for each cluster and added to our final classifier. The last line returns the descriptive, compact and meaningful classifier. The classification process of proposed methods is shown in algorithm 6.

Algorithm 6 predicts the class label of the test example by using the classifier. The first line files the *class_count* array with 0 (the size of *class_count*array equals to the number of classes). For each rule in the classifier (line 2), if the rule can classify the example correctly, then we increase the corresponding class count by one and store it (lines 3-5).In lines 7-10, if none of the rules can classify the new example correctly, then the algorithm returns the majority class value. Otherwise, it returns the class value that is the most common among the rules that classify the test example.

We built the following different classifiers: "DC" method is built based on direct distance measure (IBDM) and the method for extracting a representative CAR is based on cluster center (RCC), "DDC" method is formed based on direct distance measure (IBDM) and the method for extracting a representative CAR is based on database coverage (RDC), and "CDC" method is formed based on combined distance measure

(WCDM) and the method for extracting a representative CAR is based on database coverage (RDC).

Algorithm 6: Classification process of our proposed approaches

```
Input: A Classifier and a test_example
Output: Predicted class
Fill(class_count, 0);
for each rule $y \in Classifier$do begin
    if y classify test_example then
        class_count[y.class]++;
    end if
end for
if max(class_count)==0 then
    predicted_class=majority_class;
else predicted_class= max_index(class_count);
end if
return predicted_class
```

5. Experimental Results

We evaluated our classifiers by comparing them with 8 well-known rule-based classification algorithms on classification accuracy and the number of rules. All differences were tested for statistical significance by performing a paired t-test (with a 95% significance threshold).

Associative classifiers were run with default parameters *minimum support = 1%* and *minimum confidence = 60%* (on some datasets, however, *minimum support* was lowered to *0.5%* or even *0.1%* and confidence was lowered to *50%* to ensure “enough” CARs (“enough” means at least 5-10 rules for each class value- this situation mainly happens with imbalanced datasets) were generated for each class value). For all other 8 rule learners we used their WEKA workbench [31] implementation with default parameters. Since AR learning does not support numeric attributes, all numeric attributes (in all datasets) were pre-discretized with WEKA’s “class-dependent” discretization method. The description of the datasets and input parameters are shown in Table 1.

Furthermore, all experimental results were produced by using a 10-fold cross-validation evaluation protocol.

Experimental results on classification accuracies (average values over the 10-fold cross-validation with standard deviations) are shown in Table 2.

We can observe from Table 2 that our proposed associative classifiers achieved comparable average accuracies (DC: 80.7%, DDC:81.5% and CDC:82.0% respectively) to other classification models on selected datasets. Interestingly, CDC significantly outperforms all rule-learners on the “Breast Cancer” (except DDC), “Hayes-root” and “Lymp” datasets, while on the “Car.Evn”, “Nursery” and “Monks” datasets, our proposed methods obtained worse accuracy than all other algorithms (except for SA). Standard deviations of accuracy results decrease with an increasing number of examples in a dataset, which is expected behavior.

Table 1. Description ofdatasets and AC algorithm parameters

Dataset	# of attributes	# of classes	# of records	*Min support*	*Min confidence*	# of analyzed rules
Breast Can	10	2	286	1%	60%	1000
Balance	5	3	625	1%	50%	218
Car.Evn	7	4	1728	1%	50%	1000
Vote	17	2	435	1%	60%	500
Tic-Tac-Toe	10	2	958	1%	60%	3000
Nursery	9	5	12960	0.5%	50%	3000
Hayes-root	6	3	160	0.1%	50%	1000
Lymp	19	4	148	1%	60%	1500
Spect.H	23	2	267	0.5%	50%	3000
Abalone	9	3	4177	1%	60%	1000
Adult	15	2	45221	0.5%	60%	5000
Insurance	7	3	1338	1%	50%	722
Monks	7	2	432	1%	50%	800
Laptop	11	3	1303	1%	50%	1480

Table 2. Overallaccuracies with standard deviations:

Dataset	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	DC	DDC	CDC
Breast.Can	70.4±4.1	69.2±6.7	75.0±6.9	74.0±4.0	75.1±5.3	71.8±5.7	71.9±9.8	79.3±4.4	81.2±4.0	81.9±4.1	82.6±4.5
Balance	81.4±8.1	66.7±5.0	64.4±4.3	76.2±5.6	77.5±7.6	68.5±4.3	73.2±3.8	74.0±4.1	72.8±2.4	73.2±2.9	73.2±3.0
Car.Evn	95.4±0.8	91.3±1.7	92.1±1.7	94.3±1.0	91.8±1.1	91.0±1.8	91.2±3.9	86.2±2.1	85.8±1.4	88.5±1.2	87.1±1.6
Vote	94.7±3.4	94.9±3.7	94.7±4.4	94.8±4.2	94.4±2.8	95.6±4.1	94.4±2.6	94.7±2.3	92.9±2.5	93.2±2.8	90.6±2.3
Tic-Tac-Toe	69.9±2.7	74.4±4.4	85.2±2.7	94.3±3.3	94.1±3.1	94.3±2.9	100.0±0.0	91.7±1.5	87.3±1.3	91.8±1.0	92.4±1.1
Nursery	94.0±1.5	93.6±1.2	95.4±1.4	96.7±1.7	91.0±1.4	92.5±1.5	92.1±2.4	91.6±1.2	88.5±1.1	89.3±1.1	92.3±0.9
Hayes-root	75.0±7.2	53.4±8.3	78.7±8.4	73.1±9.7	77.7±8.7	74.3±7.1	75.6±10.9	73.1±6.0	79.9±5.7	77.8±5.2	82.7±6.1
Lymp	72.9±9.0	72.2±8.3	76.2±8.7	81.7±9.0	80.0±8.2	78.3±7.3	79.0±9.7	73.7±5.1	78.4±6.7	80.0±6.1	84.0±6.4
Spect.H	79.3±2.7	79.3±1.6	80.0±9.0	80.4±5.6	80.4±2.2	80.4±2.2	79.0±1.6	79.1±2.1	81.5±0.7	81.3±1.1	82.8±1.3
Abalone	62.1±1.3	61.8±1.5	62.3±1.2	62.3±1.1	61.7±1.6	60.8±0.8	61.1±1.0	61.0±0.9	61.0±1.1	60.7±1.0	60.7±1.2
Adult	73.0±4.1	82.0±2.3	82.4±4.7	82.1±4.7	75.2±3.2	80.8±2.7	81.8±3.4	80.8±2.6	81.9±2.4	82.0±2.6	82.8±3.0
Insurance	74.2±1.1	75.7±1.6	75.8±1.4	75.0±1.8	75.8±1.4	73.4±1.7	75.5±2.0	74.5±1.6	74.0±1.1	74.2±1.1	73.2±1.5
Monks	98.9±0.9	98.9±0.9	98.9±0.9	98.9±0.9	98.9±0.9	97.1±0.7	97.8±1.4	92.1±1.3	92.5±0.8	93.6±0.9	91.1±0.8
Laptop	75.7±2.6	72.9±2.9	75.3±2.3	74.5±2.9	75.4±2.1	73.2±1.8	75.4±2.0	72.0±1.4	71.6±2.1	73.8±2.3	72.6±1.7
Average(%):	80.0±3.5	77.6±3.6	81.2±4.1	82.7±4.0	82.1±3.5	80.8±3.2	82.0±3.9	80.3±2.6	80.7±2.4	81.5±2.4	82.0±2.5

Statistically significant testing (wins/losses counts) on accuracy between DC and other classification models is shown in Table 3. **W**: our approach was significantly better than compared algorithm; **L**: selected rule-learning algorithm significantly outperformed our algorithm; **N**: no significant difference has been detected in the comparison.

Table 3. Statistically significant wins/losses counts of DC method on accuracy:

	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	DDC	CDC
W	6	6	4	2	3	3	3	3	1	1
L	5	4	5	7	5	5	5	2	3	4
N	3	4	5	5	6	6	6	9	10	9

Table 3 illustrates that the performance of DC method on accuracy was better than DTNB, DT and SA methods. Although DC obtained similar result with C4.5 and DDC (there is no statistical difference on 10 datasets out of 14), it is eaten by all other methods according to win/losses counts. However, on average, the classification accuracies of DC are not much different from those of the other 8 rule-learners.

The same experiment on DDC is shown in Table 4. Since DC is compared with DDC in Table 3, it is not included in Table 4.

Table 4. Statistically significant wins/losses counts of DDC method on accuracy:

	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	CDC
W	5	5	4	2	2	3	3	4	2
L	4	3	3	5	4	5	4	1	3
N	5	6	7	7	8	6	7	9	9

It can be seen from the table that DDC's performance on accuracy is better than DC. It outperformed the DTNB, DT, C4.5, SA and DC methods (by win/losses counts).

Table 5. Statistically significant wins/losses counts of CDC method on accuracy:

	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA
W	6	6	6	4	5	5	4	4
L	5	4	6	5	6	3	6	1
N	3	4	2	5	3	6	4	9

CDC achieved the statistically comparable results in terms of classification accuracy with "classical" and "associative" classification approaches. CDC statistically lost to C4.5, FR and CBA methods on 6 datasets out of 14, while it outperformed the rest of the algorithms except PT.

The comparison between our methods and other classification methods on the number of classification rules is shown in Table 6. Since DC and DDC differ in the representative CAR selection process, the number of classification rules generated by both methods stays the same. Thus, DC and DDC methods are merged in Table 6.

Experimental evaluations on the number of classification rules show that DC and DDC significantly outperforms all other rule-learners on 8 datasets out of 14 (except CDC) and it produces classifiers that have on average far fewer rules than those produced by the other 8 rule-learning methods included in the comparison.

Our proposed methods generated a reasonable smaller number of rules on bigger datasets compared to other classification methods. Even though our approaches could not achieve the best classification accuracies on "Car.Evn", "Nursery" and "Laptop" datasets, it produced the statistically smallest classifier on those datasets.

Experimental evaluations on the number of classification rules show that DC and DDC significantly outperforms all other rule-learners on 8 datasets out of 14 (except CDC) and it produces classifiers that have on average far fewer rules than those produced by the other 8 rule-learning methods included in the comparison.

Our proposed methods generated a reasonable smaller number of rules on bigger datasets compared to other classification methods. Even though our approaches could not achieve the best classification accuracies on "Car.Evn", "Nursery" and "Laptop" datasets, it produced the statistically smallest classifier on those datasets.

Table 6. Number of CARs:

Dataset	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	DC&DDC	CDC
Breast.Can	122	22	10	20	13	13	63	20	8	9
Balance	31	35	35	27	44	22	77	45	34	79
Car.Evn	144	432	123	62	100	119	72	160	32	32
Vote	270	24	11	8	17	7	22	30	6	6
Tic-Tac-Toe	258	121	88	37	21	13	23	60	24	17
Nursery	1240	804	301	172	288	141	141	175	79	80
Hayes-root	5	8	22	14	11	10	34	45	19	80
Lymp	129	19	20	10	17	11	23	60	5	7
Spect.H	145	2	9	13	17	12	4	50	8	5
Abalone	165	60	49	71	20	57	131	155	14	14
Adult	737	1571	279	571	150	175	126	130	13	88
Insurance	23	48	21	49	22	22	84	62	18	20
Monks	12	36	14	8	12	10	40	26	14	14
Laptop	101	101	72	60	28	32	41	75	19	18
Average(%):	241	235	76	81	55	46	63	78	21	34

CDC got an unexpected larger number of rules (this is mainly imbalanced and discretized datasets) on “Hayes-root” and “Balance” datasets.

Table 7. Statistically significant wins/losses counts of DC and DDC method on rules:

	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA	CDC
W	11	11	10	10	11	9	12	14	3
L	2	2	0	3	2	4	1	0	2
N	1	1	4	1	1	1	1	0	9

Table 7 shows that C4.5 and SA methods could not produce statistically smaller classifier than DC and DDC methods on any datasets. The most importantly, DC and CDC generated statistically smaller classifiers than all other models on bigger datasets (over 1000 examples), which was our main goal in this research.

CDC statistically got the worse result than all methods on “Balance” (except CBA) and “Hayes.R” datasets in terms of classification rules.

Our main goal in proposing the DDC and CDC methods is to improve the overall coverage (shown in Table 9) and accuracy achieved by the DC method. Experimental results show that we could achieve our goal: DDC and CDC gained better average classification accuracy with 81.5% and 82% (this is still not the best result in terms of average accuracy, but 0.8% and 1.3% higher than the DC method). Average coverage of DDC (90.4%) and CDC (90.5%) increased to 6% compared to DC (84.4%). More precisely, the overall coverage of DDC and CDC was improved on 9 datasets and they achieved better classification accuracies on 8 datasets out of 14 compared to DC. However, DC produced a comparable associative classifier with all other “classical” and “associative” classifiers.

Table 8- Statistically significant wins/losses counts of CDC method on rules:

	DTNB	DT	C4.5	PT	FR	RDR	CBA	SA
W	11	11	9	10	10	8	11	12
L	2	3	2	3	2	4	1	2
N	1	0	3	1	2	2	2	0

Table 9- Overall Coverage:

Dataset	DC	DDC	CDC
Breast Cancer	65.2	72.0	72.7
Balance	74.5	82.8	86.3
Car.Evn	88.7	100.0	100.0
Vote	88.4	86.9	85.1
Tic-Tac-Toe	89.0	92.0	86.0
Nursery	90.4	98.1	100.0
Hayes-root	100.0	100.0	100.0
Lymp	81.0	90.0	88.4
Spect.H	80.9	80.7	79.4
Abalone	74.1	87.6	78.9
Adult	100.0	100.0	100.0
Insurance	81.5	89.5	100.0
Monks	82.4	86.7	90.6
Laptop	86.1	99.0	100.0
Average(%):	84.4	90.4	90.5

On the other hand, accuracy of DC, DDC and CDC was higher than its coverage on “Breast cancer”, “Vote” and “Monks” datasets. This fact is not surprising, since uncovered examples get classified by the majority classifier. When the overall coverage is above 85%, proposed methods tend to get a reasonably high accuracy on all datasets. All of our proposed classifiers achieved the best accuracy on “Breast Cancer” and “Spect.H” datasets among all rule-learner approaches while CDC generated slightly higher number of classification rules comparing to DC and DDC, but on average all of our proposed method achieved the best result in terms of classification rules. Evaluation of our proposed classifiers is shown in Fig 1.

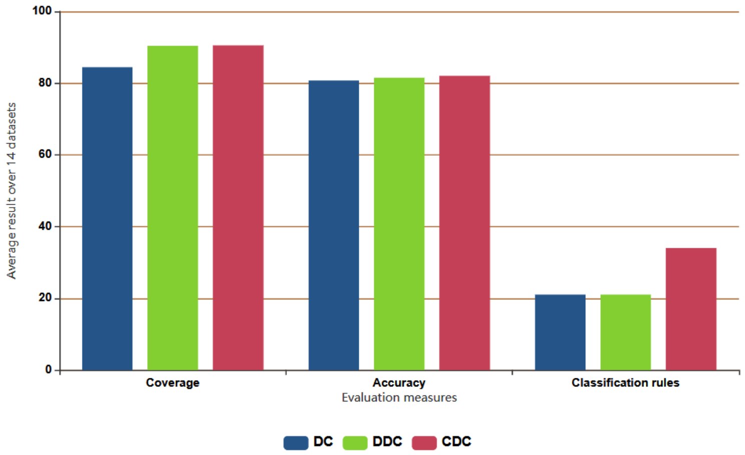


Fig 1. Comparisonof performance of our proposed associative classification models

Fig 1 illustrates that all three methods obtained similar average accuracy. Although CDC gained better coverage than DC, it got worse result in terms of the number of classification rules than that method.

The most important advantage of our proposed methods was to generate a smaller classifier on bigger datasets.

6. Conclusion and Future Work

Experimental evaluations show that we could somehow achieve our intended goal in this research to produce a compact and meaningful classifier by exhaustively searching the entire example space using constraints and clustering. Our DC, DDC and CDC classifiers were able to reduce the number of classification rules while maintaining a classification accuracy that was comparable to state-of-the-art rule-learning classification algorithms. Moreover, we showed in the experiments that our classifiers were able to reduce the number of rules in the classifier by 2-4 times on average compared to the other rule-learners, while this ratio is even bigger on datasets with a higher number of examples.

All three proposed associative classifiers have their advantage on some certain datasets; they are even comparable to each other on the number of generated rules and classification accuracy.

The main drawback of our proposed methods is their time efficiency. In future work, we plan to parallelize DC, DDC, and CDC to bring their time complexity at least a bit closer to state-of-the-art "divide-and-conquer" rule-learning algorithms.

Acknowledgment. The authors gratefully acknowledge the European Commission for funding the InnoRenewCoE project (Grant Agreement #739574) under the Horizon2020 Widespread-Teaming program and the Republic of Slovenia (Investment funding of the Republic of Slovenia and the European Union of the European Regional Development Fund). Jamolbek Mattiev is also funded for his Ph.D. by the “El-Yurt-Umidi” foundation under the Cabinet of Ministers of the Republic of Uzbekistan.

References

1. Lent, B., Swami, A., Widom, J.: Clustering association rules. In: Proceedings of the Thirteenth International Conference on Data Engineering, Gray, A., Larson, P. England, 220–231. (1997)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB '94 Proceedings of the 20th International Conference on Very Large Data Bases. Morgan Kaufmann, Santiago, Chile, 487–499. (1994)
3. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings of the 4th InternationalConference on Knowledge Discovery and Data Mining. Agrawal, R., Stolorz, P. New York, USA, 80–86. (1998).doi: 10.5555/3000292.3000305
4. Hu, L. Y., Hu, Y. H., Tsai, C. F., Wang, J. S., Huang, M. W.: Building an associative classifier with multiple minimum supports. SpringerPlus Vol. 5, No. 528. (2016)
5. Deng, H., Runger, G., Tuv, E., Bannister, W.: CBC: an associative classifier with a small number of rules. Decision Support Systems, Vol. 50, No. 1, 163–170, (2014)
6. Khairan, D. R.: New Associative Classification Method Based on Rule Pruning for Classification of Datasets. IEEE Access, Vol. 7, 157783-157795. (2019)
7. Ramesh, R., Saravanan, V., Manikandan, R.¸: An Optimized Associative Classifier for Incremental Data Based On Non-Trivial Data Insertion. International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol. 8, No. 12, 4721-4726. (2019)
8. Thabtah, F. A., Cowling, P., Peng, Y.: MCAR: multi-class classification based on association rule. In: Proceedings of the 3rd ACS/IEEE international conference on computer systems and applications. Cairo, Egypt, 127–133. (2005)
9. Thabtah, F. A., Cowling, P., Peng, Y.: MMAC: a new multi-class, multi-label associative classification approach. In: Proceedings of the fourth IEEE international conference on data mining. Brighton, UK, 217–224. (2004)
10. Abdellatif, S., Ben Hassine, M. A., Ben Yahia, S., Bouzeghoub, A.: ARCID: A New Approach to Deal with Imbalanced Datasets Classification. 44th International Conference on Current Trends in Theory and Practice of Computer Science, Lecture Notes in Computer Science, Austria, Vol. 10706, 569-580. (2008)
11. Chen, G., Liu, H., Yu, L., Wei, Q., Zhang, X.: A new approach to classification based on association rule mining. Decision Support Systems, Vol. 42, No. 2, 674–689. (2008)
12. Kaufman, L., Rousseeuw, P., J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons, USA (1990)
13. Zait, M., Messatfa, H.: A Comparative Study of Clustering Methods. Future Generation Computer Systems, Vol. 13, No. (2-3), 149-159. (1997)
14. Arabie, P., Hubert, L. J.: An Overview of Combinatorial Data Analysis. In: Clustering and Classification. Arabie, P., Hubert, L. J, Soete, G. D. New Jersey, USA, 5–63. (1996)
15. Ng, T. R., Han, J.: Efficient and Effective Clustering Methods for Spatial Data Mining. In: Proceedings of the 20th Conference on Very Large Data Bases (VLDB). Morgan Kaufmann. Santiago, Chile, 144-155. (1994)
16. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In: Proceedings of the ACM-SIGMOD International conference on Management of Data. Montreal, Canada,103-114. (1996)
17. Theodoridis, S., Koutroumbas, K.: Hierarchical Algorithms. Pattern Recognition, Vol. 4, No. 13, 653-700. (2009)
18. Dua, D., Graff, C.: UCI Machine Learning Repository, Irvine, CA: University of California (2019)

19. Hall, M., Frank, E.: Combining Naive Bayes and Decision Tables. In proceedings of Twenty-First Artificial Intelligence Research Society Conference, Florida, USA, 318-319. (2008)
20. Kohavi, R.: The Power of Decision Tables. In: 8th European Conference on Machine Learning. Lavrač, N., Wrobel, S. Crete, Greece, 174–189. (1995)
21. Hühn, J., Hüllermeier, E.: FURIA: an algorithm for unordered fuzzy rule induction. Data Mining and Knowledge Discovery, Vol. 19, 293–319, DOI: doi.org/10.1007/s10618-009-0131-8, (2019)
22. Frank, E., Witten, I.: Generating Accurate Rule Sets Without Global Optimization. In: Fifteenth International Conference on Machine Learning, Shavlik, J.W. USA, 144–151. (1998)
23. Quinlan, J.: C4.5: Programs for Machine Learning. Machine Learning, Vol. 16, No. 3, 235-240. (1993)
24. Richards, D.: Ripple down rules: a technique for acquiring knowledge. Decision making Support Systems: achievements, trends and challenges for the new decade, Mora, M., Forgionne, G. A., Gupta, J. N. D. USA, 207–226. (2002)
25. Mattiev, J., Kavšek, B.: Simple and Accurate Classification Method Based on Class Association Rules Performs Well on Well-Known Datasets. In: Machine Learning, Optimization, and Data Science, LOD 2019. Nicosia G., Pardalos P., Umeton R., Giuffrida G., Sciacca V, Siena, Italy, Vol. 11943, 192–204. (2019)
26. Dahbi, A., Mouhir, M., Balouki, Y., Gadi, T.: Classification of association rules based on K-means algorithm.In: 4th IEEE International Colloquium on Information Science and Technology. Mohajir, M.E., Chahhou, M., Achhab, M.A., Mohajir, B.E. Tangier, Morocco, 300–305. (2016)
27. Gupta, K. G., Strehl, A., Ghosh, J., Distance based clustering of association rules. Proceedings of artificial neural networks in engineering conference, USA, 759-764. (1999)
28. Kosters, W., Marchiori, E., Oerlemans, A., Mining Clusters with Association Rules. Lecture Notes in Computer Science, DOI:10.1007/3-540-48412-4_4, (1999)
29. Natarajan, R., Shekar, B..: Tightness: A novel heuristic and a clustering mechanism to improve the interpretation of association rules. In Proceedings of the IEEE International Conference on Information Reuse and Integration, USA, 308-313. (2008)
30. Toivonen, H., Klemettinen, M., Ronkainen, P., Hatonen, K., and Mannila, H.: Pruning and grouping discovered association rules. In ECML-95 Workshop on Statistics, Machine Learning, and Knowledge Discovery in Databases, Greece, 47-52. (1995)
31. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I, H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, Vol. 11, No. 1, (2009)
32. Cohen, W., W.: Fast Effective Rule Induction.: In: ICML'95 Proceedings of the Twelfth International Conference on Machine Learning, California, USA, 115-123. (1995)
33. Mattiev, J., Kavšek, B.: A compact and understandable associative classifier based on overall coverage. Procedia computer science, Vol. 170, Warsaw, Poland, 1161-1167 (2020)

Jamolbek Mattiev has a Master degree in Computer Science from National University of Uzbekistan. He was awarded with first degree diploma at “the best Master Dissertation Work of Uzbekistan” competition in his master studies. He obtained his PhD degree at the Department of Information Sciences and Technologies of the University of Primorska, Slovenia. His research fields include Artificial Intelligence, Data Mining, Machine Learning. In particular, the subfields of Supervised and Unsupervised Learning, Frequent Pattern Discovery and Association Rule Learning, Classification, Clustering.

Branko Kavšek has a PhD in Computer Science from the University of Ljubljana. He is an assistant professor at the University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies, a researcher and Vide Head at the Department of Information Sciences and Technologies and member of the Artificial Intelligence Laboratory at the Jozef Stefan Institute in Ljubljana. His research fields include Artificial Intelligence, Data Mining, Machine Learning. In particular, the subfields of Supervised and Unsupervised Learning, Frequent Pattern Discovery and Association Rule Learning, Learning Probabilistic Models and Bayesian Networks Learning, Clustering, and Data Mining applied to Big Data. He is the co-author of several scientific publications and currently involved in national and international projects.

Received: April 30, 2020; Accepted: November 08, 2020.

Computer Science and Information Systems 18(3):813–834 https://doi.org/10.2298/CSIS200308043C

CHEARP: Chord-based Hierarchical Energy-Aware Routing Protocol for Wireless Sensor Networks

Lamia CHEKLAT[1], Mourad AMAD[2], Mawloud OMAR[3], and Abdellah BOUKERRAM[1]

[1] LIMED Laboratory, Faculty of Exact Sciences, University of Bejaia,06000 Bejaia, Algeria.
cheklat.lamia@gmail.com, boukerram@hotmail.com
[2] LIMPAF Laboratory, Computer Sciences Dept, Faculty of Sciences and Applied Sciences,
University of Bouira, Algeria.
amad.mourad@gmail.com
[3] LIGM, ESIEE Paris, Université Gustave-Eiffel, Noisy-le-Grand, France.
mawloud.omar@univ-eiffel.fr

Abstract. Wireless Sensor networks (WSNs) are mostly deployed in hostile environments, where nodes may do not have any information about their location. Hence, the designed routing protocols and applications have to function independently from the nodes location. Moreover, extending lifetime in such networks is a critical and challenging issue, since they consist of miniaturized energy-constrained devices. The motivation of this paper is to design an energy efficient location-independent routing protocol for data delivery in WSNs. Therefore, a Chord-based Hierarchical Energy-Aware Routing Protocol (CHEARP) is developed with the focus on preserving the energy consumption. In contrary to the existing DHT-based protocols that interconnect nodes independently of their physical proximity, this paper proposes an approximate logical structure to the physical one, where the aim is to minimize the average paths' length. Simulation results show that the proposed solution reduces the transmission Load, minimizes the transmission delay, and extends the network longevity.

Keywords: Wireless Sensor Networks, Peer-to-Peer Systems, Chord, DHT, Energy-Efficient, Clustering, Data Communication.

1. Introduction

Wireless Sensor Networks (WSNs) [6] have attracted significant interest in today's human life through their major impact and useful support in the construction of intelligent environments. In fact, new applications are continuously developed in order to meet different requirements and purposes of different fields, including healthcare, industry, environmental monitoring and military domain. However, in WSNs researchers do not only interest to develop new services and applications, but also, to extend the network lifetime [16] and to make the developed applications operational as long as possible. The energy is the most delicate resource in sensor devices due to the limited capacity of the equipped batteries that can be expensive and even impossible to renew, particularly in hostile environments, such as battlefields. Therefore, it is strongly recommended that any designed application for WSNs should be power-efficient to overcome this limitation of energy and to increase the network longevity. In addition, transferring the collected data to the sink represents a core task for any WSN application. For this reason, plenty of research projects are being done, and many mechanisms and routing techniques are ongoing in order to optimize the power consumption of sensor nodes [1] [18]. Applying DHT (Distributed Hash Tables) over WSNs becomes an active branch and provides promising assets to meet WSNs problems [9]. The DHT-based routing solutions achieve better performances in terms of scalability, self-organization, decentralization, network lifetime, fault tolerance and latency [2]; besides that, they guarantee data delivery in a limited time. However, the DHT-based routing schemes build a virtual overlay under a physical network topology, where nodes are connected independently of their physical proximity in the network. Hence, two virtual neighbors in the logical space with two close logical identifiers may be far apart and could not be physical neighbors in the underlying network. In this case, a logical hop transmission in a DHT-routing protocol may go through a several physical hops that may cost many transmission packets and a huge amount of energy consumption, which is unsuitable for energy constrained environments such as WSNs. In this paper, the advantages of DHTs are exploited in favor of WSNs, while the divergence problem of the virtual overlay and the underlying network is faced. Hence, we develop a Chord-based Hierarchical Energy-Aware Routing Protocol (CHEARP) for WSNs, which aims to ensure that two neighbor nodes in the physical network are also neighbors in the virtual overlay. In fact, Chord was introduced to accelerate the lookup and to locate efficiently the node that stores the required data. In the proposed solution, we take the benefits of these two characteristics and the other assets of Chord protocol to accelerate the routing process and to locate efficiently the nodes that are connected to the sink (SCH). This allows to relay the sensed data efficiently in a finite time, while consuming few amount of energy. In this work, the network is organized in a two-tier hierarchical structure, in which only cluster-heads perform routing tasks using a same routing policy of Chord protocol.

The remainder of this paper is structured as follows. The next Section provides a brief overview of Chord protocol and points out the problem statement. Section 3 presents a review of some relevant DHT-based routing solutions. Section 4 describes in details the proposed protocol. Section 5 discusses the performance analysis results. Finally, Section 6 summarizes and concludes this paper.

2. Preliminaries

2.1. Brief View of Chord Protocol

Chord [17] is one of the first and most popular protocols for structured peer-to-peer (P2P) systems, designed to address the issue of lookup in dynamic P2P overlays. It assigns for both peers and resources, unique identifiers in the $m - bit$ key space, using the same hash function, where m represents the identifiers' length. The peers in Chord overlay are organized in $one - dimensional$ virtual ring of modulo 2^m size (from 0 to 2^{m-1}), following the ascendency order of identifiers so that the previous node ID is always lower than its successor, moving in one direction in clockwise. The Peers' identifiers and objects' keys are generated by hashing the peers' IP addresses and data, respectively. To store a pair of $key/value$ of any distributed resource, Chord uses a hash function for generating the resource's key. This latter indicates the node where the pair of $key/value$ will be maintained. Then, the key will be mapped onto the first node whose identifier is equal or follows it in the identifier space.

Chord introduces two variants of lookup schemes; the first one is simple but slow, whereas the second holds additional information but accelerates lookup. In the simple key location scheme, nodes require only to know their immediate successors in the ring. Thus, looking for a given key involves passing through the successors around the ring until finding the node that stores the key. To accelerate lookup, a scalable key location scheme requires each node to maintain a routing table, called finger table, constructed so that the i^{th} entry of the nodes n includes a pointer to the successor of node $n + 2^{i-1}$ in the Chord ring. In other words, in an $m - bit$ identifier space, a finger table includes up to m entries. The first finger in the routing table of a given node represents its immediate successor, and each entry in the table maintains information about the identifier, the IP address, and the port number of the concerned finger. Hence, queries for a researched key are forwarded to the node with the largest identifier that is equal to or precedes the key.

2.2. Problem Statement

The main problem that researchers have to hold in order to develop an energy efficient DHT-based routing protocol for WSNs is how to deal with the divergence between the virtual overlay and the underlying network. To illustrate this problem, an example of Chord topology is shown in Figure 1, where the node $N6$ points to the node $N17$ even though the node $N35$ is closer to it than $N17$, since the ring construction follows the ascending order of identifiers. In case of wired networks, each node can reach any destination through several physical neighbors without influencing the lifetime of the network. However, in case of energy-constraint networks such as WSNs, any virtual hop may cause many physical hops, and hence, many transmission packets, which increases the consumed energy, the end to end delay and the overhead in the network. In this paper, we address these problems and develop CHEARP, a DHT-based energy efficient protocol, which takes the benefits of DHT overlays to be a self-organized and a totally decentralized protocol that suits the random deployed networks, such as WSNs. In the proposed solution, we handle the structure shown in the Figure 1 in such way that the physical structure and the logical overlay are approximated. This allows to guarantee that if two nodes are neighbors in the physical network topology, they will be also neighbors in the virtual overlay.

3. Related Works

The development of routing protocols in WSNs is a subject of many researches. In fact, a myriad of routing protocols are continuously developed with the aim of facing the challenges and the intrinsic constraints of WSNs, in particular, the energy consumption. Since we grant more interest to P2P and DHT-based solutions, in this section, only some DHT-based routing protocols are reviewed.

A survey of DHT-based solutions in WSNs could be found in [9], where the authors study the applicability of DHT over WSNs.

In [3], the authors discuss the application of Chord, a DHT protocol, over WSNs to improve the delivery ratio of the overlay architecture. The proposed scheme introduces CREIDO packets to find out the eventual joining or leaving nodes in order to cope up instantly the network changes by mean of stabilization function. This work focuses only on detecting eventual changes in the Chord network for an instant update of the topology,

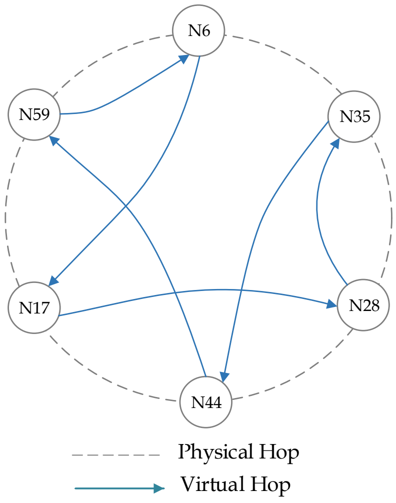


Fig. 1. Illustration of the divergence between physical network topology and the virtual overlay in Chord.

and the authors do not provide any details of the manner in which they apply Chord over WSNs.

The work in [5] consists on a limited energy consumption model for Wireless Sensor Networks, which consists on adapting the Chord protocol for WSNs. The authors organize the random deployed nodes on clusters and elects for each cluster a strongest node in terms of energy as cluster-head. In the developed scheme, the Cluster-heads are interconnected according to Chord topology, and they represent the only nodes that perform data transmission. The solution considers the energy level of nodes and permutes Cluster-heads to save energy. However, it does not consider the physical proximity of nodes.

CLEVER is proposed in [8] as a Cluster-based Energy-aware Virtual Ring Routing protocol [4] for WSNs. It applies a DHT-Virtual Ring Routing protocol for inter and intra cluster communication. In addition, the solution makes a use of clustering mechanism, where the energy powerful nodes are defined as super-peers that take charge of the virtual routing in the network (virtual hops). Besides, for each node is assigned a transmission power considering its energy amount, which seems good for saving energy. However, the super-peers in different clusters could not communicate and require the implication of the weak nodes in each data transmission to perform routing task (physical hops).

The authors in [7] present Coral-based VRR protocol that organizes the network space in multilevel virtual rings. In this work, nodes are categorized according to their residual energy under three classes, namely hyperpeers that represent nodes with a big amount of energy, superpeers that are nodes with more energy than peers, and peers that are nodes having critical energy amount. Indeed, the first Coral-based VRR level in the network regroups all categories of nodes, the second level includes only superpeers and hyperpeers, while the third level includes only hyperpeers. Moreover, this classification of nodes aims to exploit as much as possible the energy powerful nodes in routing. Then, VRR is applied to ensure transmission in each layer, while Coral-based VRR allows transition between the three layers. This technique manages efficiently the network, nevertheless, it shows the same drawback of CLEVER, where two virtual neighbors may be physically far away.

Concisely, from the above reviewed DHT-based solutions, it is noticed that most of DHT-based solutions present the same problem of divergence of the physical and logical neighborhood. We overcome this problem by providing an approximate virtual overlay to the underlying network. More details are given in Section 4

4. Proposed Solution

In this section, the network model considered in this work is described and the details of our proposal are given.

4.1. Network Model

Before giving the details about the principal of the proposed solution, a description of the network model is given. First, we suppose a random deployment environment of WSNs, where sensor nodes are static and organized in clusters according to the physical proximity. Then, a cluster head is elected for each cluster using the objective function, which is given further (Function 2). We consider a scenario of application, where nodes have to sense physical parameters from the zone of interest, and to transmit the collected data to the sink. In this proposal, a Chord-based overlay is built on top of the physical network. Thus, sensors are organized logically in a virtual ring according to the ascending order of their identifiers, as shown in Figure 2. Before the overlay formation, for each node is associated a unique identifier using the same hash function used in Chord protocol. Two kinds of links are distinguished, namely physical and logical. A physical link exists between two nodes if the distance separating them is less than or equal to the maximum radio transmission power ($d(n1, n2) \leq r$), which is supposed to be the same for all the sensor nodes in the network. Whereas, the logical links are defined using Function 1, where for each node n is associated a set of up to m neighbors (fingers). The set of the node n fingers is denoted by $Fingers_n$, the identifiers' length by m, and the identifier of the node n by n_{id}. In this paper $\mathcal{C}$ is considered as the set of clusters, $\mathcal{CH}$ as the set of cluster heads, and a summary of the used notations is given in Table 1.

$$Fingers_n = \{Successor\,[(n_{id} + 2^{i-1})\,modulo\,2^m]\} \tag{1}$$
$$\textit{With } 1 \leq i \leq m$$

4.2. Network Structuring

Preliminary Phase. This phase succeeds the deployment of sensors in the zone of interest and consists on determining a ring band in the network, defined by external and internal

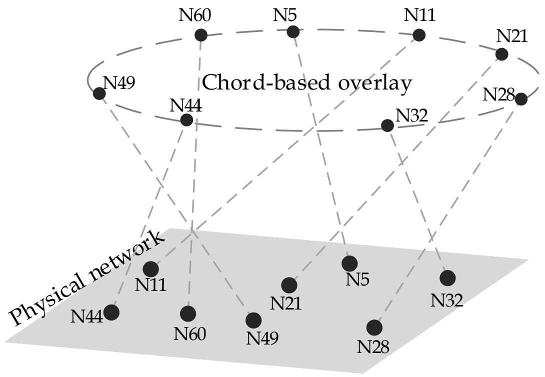


Fig. 2. Example of a Chord-based overlay.

Table 1. Notations.

Notation	Description
$\mathcal{C}$	The set of clusters
$\mathcal{CH}$	The set of cluster heads
$\mathcal{CR}$	The set of CH nodes constituting the CHEARP Ring
CHt	A temporary cluster head
$CH2$	A second degree cluster head
CH_{init}	The CH that initiates the ring creation
CH_{succ}	The CH successor in the ring
CH_{pred}	The CH predecessor in the ring
CN	The number of the formed clusters or cluster heads
Dgr	Node degree
$\mathcal{F}$	The set of finger tables
$Fingers$	A finger table of a node
f	A finger of a node
NN	The variable of the cluster nodes' number
i, j	Loop counters
$idCH$	CH identifier
$idN_{i,j}$	Node N identifier
m	Nodes identifiers length
$N_{i,j}$	The j^{th} node in the cluster C_i
r	The transmission range
SCH	A cluster head that is a one hope linked to the sink (*connected cluster head*).

borders as shown in Figure 3. The goal behind is to avoid nodes at the boundary to be elected as cluster-heads in order to do not waste energy in communication since there are no nodes at the external side of the boundary. On the other hand, the ring band is the most suitable part in the network space where the CHEARP ring has to be created. Indeed, the nodes in this part have more possibility to be directly connected to the sink, which makes them the appropriate nodes for the CHEARP ring creation.

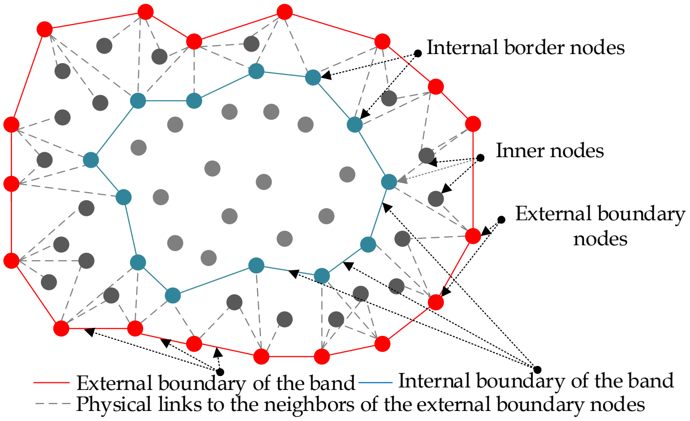

Internal border nodes
Inner nodes
External boundary nodes
External boundary of the band
Internal boundary of the band
Physical links to the neighbors of the external boundary nodes

Fig. 3. Example of a random network after the preliminary phase.

The boundary sensor nodes forming the perimeter of the network represent the external borders of the ring band. There exist many algorithms in the literature to this end, such as in [14][19][12]. To determine the inner nodes that belong to the band, each boundary node sends a broadcast message through the network. The nodes that receive the messages will be part of the ring band. The last nodes that receive the broadcast message along the internal side of the network represent the internal borders of the ring band.

Clustering Phase. Based on clustering mechanisms, sensor space is divided into small zones that regroup sensors considering given properties. The authors use a hierarchical structure since it suites the energy-saving issue. This kind of structures is characterized by a division of the network into clusters, where for each cluster is associated a cluster head (CH). The clustering phase in this work is carried out in three steps, namely the creation of basic clusters, the election of cluster-heads and the cluster expansion.

1. ***Basic Clusters Creation***
   There exist several clustering mechanisms in the literature [10][15][13]; for some of them, cluster heads are firstly elected for each cluster. After that, the sub peers integrate the appropriate clusters. While for some others, sub peers belong first to their appropriate clusters, then, CHs are selected for each cluster. In this first step of the clustering phase, the network is structured considering the second class of clustering approaches. The basic clusters could be formed using the physical proximity of the nodes, where nodes that are geographically close form a cluster. At this stage, the authors assume that the network is organized in a set of basic clusters that include only the nodes in the ring band, as illustrated in Figure 4. Basic cluster creation. For each cluster is associated, randomly, a temporary cluster head (CHt). The cluster heads and the second-degree cluster heads ($CH2$) will be elected in the next step, according to important parameters that are defined further.

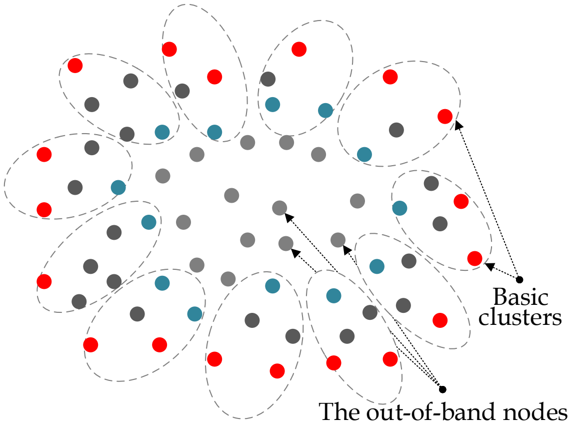


Fig. 4. Basic cluster creation.

2. ***Cluster Heads Election***
   After the formation of basic clusters, the election process of cluster-heads is executed as shown in Algorithm 1. The nodes are chosen to be cluster-heads considering their degrees, which are calculated using the Function 2. Each node N_i calculates its degree Dgr_i based on its residual energy level E_i, its connectivity rate C_i, defined as the number of its neighbors, and its signal strength to the sink P_i. The degrees are sent to the temporary cluster heads $CHts$ that were selected before. The nodes, whose Dgr values are the highest in each cluster, are elected to be cluster-heads and establish connections with the nodes of their clusters.

$$F : Dgr_i \;=\; E_i \,*\, C_i \,*\, (|P_i| \,\vee\, 1) \tag{2}$$

   The degrees of nodes are defined in the Function 2 in such a way that we promote nodes connected to the sink with a good signal strength (a non-null P_i), a good level of energy and that have more neighbors. The logic operator OR inclusive ($\vee$) allows to avoid a null value of Dgr when a given node N_i is not connected to the sink ($P_i = 0$). Thus, if a node is connected to the sink, its degree will be defined as $E_i * C_i * (P_i + 1)$; else, it will be defined as $E_i * C_i * 1$. The elected CHs supervise their clusters and handle the data transmission from their nodes to the sink. While the second-degree cluster-heads take charge of the data aggregation mechanism to save more energy.

Algorithm 1 *Cluster heads election*

```
Input: $\mathcal{C}$: The set of clusters
Output: $\mathcal{CH}$: The set of cluster heads

begin
for $i = 1$ to $CN$ do
    for $j = 1$ to $NN_i$ do
        $CHt_i$ sends broadcast message to the nodes of cluster $C_i$
        Calculation of $Dgr(N_{i,j})$
        Sending $Dgr(N_{i,j})$ to $CHt_i$
    end for
    $CHt_i$ selects the highest degree $Dgr$ and returns the corresponding $CH_i$
    $CHt_i$ selects randomly the second highest degree and return the corresponding $CH2_i$
end for
end
```

3. ***Cluster Expansion***

Once the step of CH election is completed, in the last step (cluster expansion), the elected CHs extend the clusters by sending a multi-hop diffusion messages so that nodes that did not belong to the band join any cluster and be a part of the hierarchical structure of the network, as shown in Figure 5.

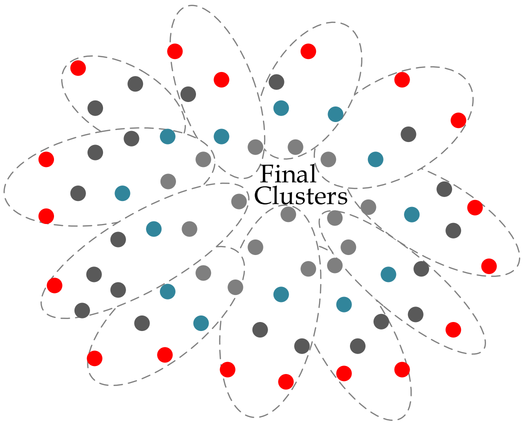


Fig. 5. Cluster Expansion.

As it was previously mentioned, the basic cluster creation includes only the nodes in the ring band that includes effectively the border nodes. Hence, after the basic cluster creation, each border node belongs to one of the created basic clusters. In the next step of clustering phase, a cluster-head is elected for each basic cluster as it is shown in the cluster heads election sub-section. The last step of clustering phase is the cluster expansion that aims to integrate nodes that are not part of the band, i.e. the nodes that are not part of the basic clusters, into the appropriate ones. To do so, each cluster-head (CH) sends a multi-hop broadcast message through the network. The nodes that already belong to one of the basic clusters (including the border nodes) are not interested in the message hence they ignore it. However, the nodes that are outside the band and that did not integrate any cluster, could now belong to one of the basic clusters. The broadcast messages are sent in multi-hop in order to reach nodes that probably could not be reached by cluster-heads. In this case, the connection is ensured by the intermediate nodes of the same cluster.

4.3. Network Routing

Ring Topology Construction and Nodes Re-identification. To approximate the overlay structure to the physical topology, CHEARP protocol proceeds by re-identifying the nodes at the overlay construction step as resumed in Algorithm 2. The first node that initiates the CHEARP ring determines the distances that separate it from other CHs in its vicinity, using the signal strength, and points at the successor node that have the shortest distance. After that, using the source node id, the successor pointer calculates its new id. This procedure continues until the formation of the CHEARP ring is completed. The proposed re-identification mechanism allows to obtain a logical structure (CHEARP ring) close to the physical one. Indeed, the CHs identifiers are substituted after being part of the CHEARP ring, where each new CH identifier is based on the previous CH identifier (the predecessor of the current CH). Hence, this solution guarantees that:

- Each direct successor is the closest in the physical network and in the overlay too;
- $\forall\, CH_i \in \mathcal{CR},\ idCH_{i-1} < idCH_i$.

This allows to shorten the length of paths, which minimizes the transmission response time, reduces the amount of dissipated energy in the network, and hence, extends the network longevity.

Construction of Finger Tables. This step succeeds the CHEARP ring creation phase, and consists on the rooting tables construction. As in the basic Chord, each node in the ring calculates its own finger table using the Function 1. The Algorithm 3 illustrates the way in which these tables are formed.

Data Communication.

1. ***Connection initialization phase***
After the formation of clusters and cluster-heads, connection phase has crucial importance, since it involves the interconnection of the network to the sink. As shown in Figure 6, once the clusters and cluster-heads are elected, the base station sends a

broadcast hello message, including its address, in the network. Since the ring in the CHEARP protocol includes only cluster-heads in routing, the latter nodes are the only ones that are interested in the message sent by the sink. Then, the cluster-heads send back a response message, including their identifiers. After that, the sink will be aware about the cluster heads that could reach it, which is denoted by $SCHs$. In the next step, the sink sends another packet to only the SCH nodes, containing the set of the interconnected SCH identifiers. Then, each SCH relays the packet to its CH neighbors, except of those figured in the packet. Each CH do the same until all the CHs receive the SCH table. Hence, CHs that are not $SCHs$ hold another information besides the routing table, which consists on the table of the $SCHs$. The redundant $SCHs$ ensure the connection availability to the sink and ensure the load sharing.

Algorithm 2 *Ring creation and nodes re-identification*

```
Input: $\mathcal{CH}$: The set of cluster heads
       $CH_{init}$: The initial $CH$
       $m$: The identifiers length
Output: $\mathcal{CR}$: CHEARP Ring

begin
$CH1 \leftarrow CH_{init}$
$\mathcal{CR} \leftarrow CH_{init}$
$\alpha \leftarrow m$
$\beta \leftarrow 1$
while $\mathcal{CH} \neq \emptyset$ do
    if $cardinality(\mathcal{CH}) = 1$ then
        $\mathcal{CH} \leftarrow \mathcal{CH} - CH_{init}$
        $\mathcal{CR} \leftarrow \mathcal{CR} + CH1$
    end if
    for $i = 1$ to $CN$ do
        for $j = 1$ to $NN_i$ do
            $idN_{i,j} \leftarrow idCH_{init} + j$
        end for
    end for
    $\beta \leftarrow \beta + 1$
    $CH_i$ sends a request message to its $CHs$ neighbors
    if $N \in \mathcal{CH}$ then
        $N$ sends a reply message containing the distance that separates it from $CH_{init}$
    else
        $N$ drops the message
    end if
    $CHt_i$ selects the minimal distance and points to the corresponding $CH_{succ}$
    $\mathcal{CH} \leftarrow \mathcal{CH}$ - $CH_{init}$
    $\mathcal{CR} \leftarrow \mathcal{CR}$ + $CH_{succ}$
    $idCH_{pred} \leftarrow idCH_{init}$
    $CH_{init} \leftarrow CH_{succ}$
    $idCH_{init} \leftarrow (idCH_{pred})^{\alpha\beta}$
end while
end
```

Algorithm 3 *Creation of finger tables*

```
Input: $\{idCH\}$: The set of $CHs$ identifiers
       $m$: The identifiers length
       $r$: The transmission radius
Output: $\mathcal{F}$: The set of finger tables

begin
for $i = 1$ to $CN$ do
    $F_i \leftarrow \emptyset$
    $Fingers_i \leftarrow \emptyset$
    for $j = 1$ to $m$ do
        $f_{i,j} \leftarrow Successor\ ((idCH_i\ +\ 2^{l-1})\ mod\ 2^m)$
        if $Dist(CH_i, f_{i,j}) \leq r$ then
            $Fingers_i \leftarrow Fingers_i + f_{i,j}$
        end if
    end for
    $\{\mathcal{F}\} \leftarrow \{\mathcal{F} + Fingers_i\}$
end for
end
```

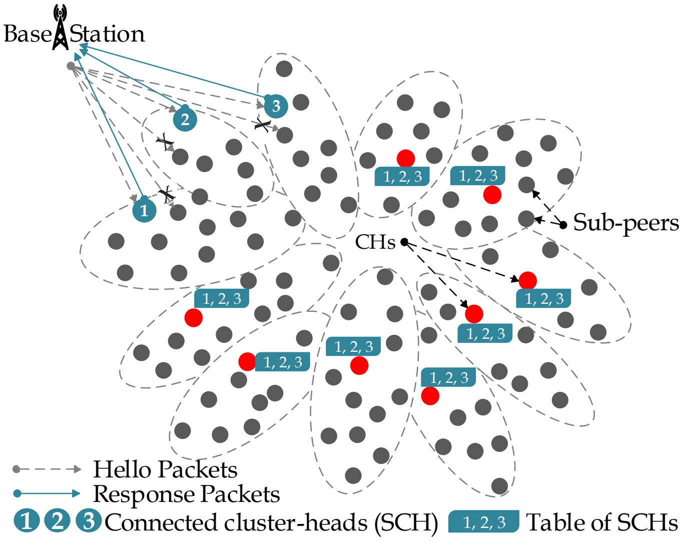


Fig. 6. CHEARP initialization Phase.

2. ***Data delivery phase***

   In order to project the Chord basic objectives to the proposed solution, the $SCHs$ are identified as the unique keys that all the sensor nodes are looking for in order to transmit the collected data to the sink. By exploiting the routing strategy of the basic Chord, CHEARP protocol provides data transmission in an average of $O(logn)$ hops

[2], where n represents the number of nodes in the ring. In this section, a description of how the packets are routed by CHEARP is given as follows:

- ***Intra-cluster routing:*** the communication inside each cluster is in multi-hops. Subpeers send data to the $CH2$ for eventual aggregation, then the $CH2$ relays the data to the CH, where they will be transmitted to the sink.
- ***Inter-cluster routing:*** the communication inter-cluster is in multi-hops, following the routing policy of the basic Chord. When a CH receives data from its subpeers, via the second-degree $CH2$, it selects randomly one SCH among the $SCHs$ set as a key to look for. Hence, the given CH checks in its finger table, the CH with the largest closest identifier to the key and relays the data to it. In this way, the data are passed around the CHEARP ring through the successor pointers until achieving the random selected SCH (the requested key), which takes in charge the transmission of the data to the sink.

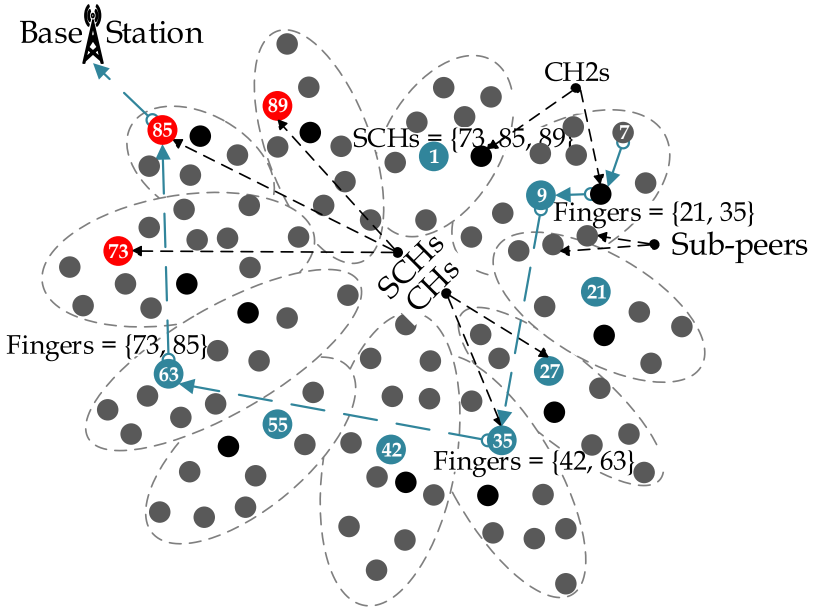


Fig. 7. Example of routing in CHEARP.

An example of routing in CHEARP protocol is given in Figure 7, where a data transmission is supposed from the node $N7$. The latter sends the data packet to the appropriate $CH2$ in its cluster for eventual data aggregation. Then, the $CH2$ sends the packet to the cluster head $N9$ that takes in charge the data transmission. First of all, $N9$ selects randomly one SCH key among the SCH list, for example $N85$. After that, the query will be resolved as follows. $N9$ picks out in its finger table the successor with the closest identifier to $N85$, which is $N35$. This latter points to the finger $N63$ that owns the largest identifier, and next, $N63$ finds out in its finger table an entry to the key $N85$ and relays the data packet to it. Finally, $N85$ transmits the data to the sink.

4.4. Network Updates

To maintain the correctness of the network topology in this work, the basic Chord functions are taken back while providing the necessary modifications.

Joining Process. The CHEARP ring includes only cluster heads. New sensors may integrate the ring, if they satisfy the required conditions during the updating phase. However, they have first to belong to any cluster in the sensor space. To join one of the clusters, a new sensor broadcasts a joining request. The sensor nodes in its vicinity could respond with a joining reply, containing the necessary information about the CH and the $CH2$ of the appropriate cluster. By this way, the new node will join the first node that replays its request, and integrates the same cluster. The new node could replace the current CH if its energy level is higher.

Leaving Process. The network updates for leaving process depends on whether the leaving node is a CH or a subpeer. In the case of subpeer, the node failure does not influence the network topology and the table correctness. However, if the node that leaves the network is a CH, it becomes mandatory to cope up the network changes in order to maintain the network availability and correctness. A failure is detected by physical neighbors as the same as basic Chord, using the function $check - predecessor()$ [17]. If a predecessor of the current node does not response, the current node turns its predecessor to $null$. Since each node in CHEARP ring holds up to $m\ entries$ in its finger table, if the node's successor fails, another finger could be chosen. To distribute the energy consumption over the nodes in CHEARP, we proceed by the distribution of the cluster head role among the ring band nodes in each cluster. Each CH node checks periodically its instant energy level, if it fits under the fixed threshold ($2/3$ of its initial energy), the re-election processes is triggered, and the CH leaves the ring and becomes subpeer. The new elected CHs and the old one permute their identifiers to keep correct the finger tables of the other CHs in the CHEARP ring. As it have been motioned in the preliminary phase of CHEARP, a node could be cluster head if only it belongs to the ring band. Thus, only inner nodes are concerned by the re-election of the new CH.

5. Performance Analysis

5.1. Radio Energy Model

Many energy models have been proposed in the literature, we use for our analysis the model discussed in [11], which is the first order radio model for energy dissipation. According to this, the transmission and the reception energy costs expended for the transfer of an $l - bit$ data message between two nodes over a distance of $d - meter$ are given, respectively, by Equations 3 and 4.

$$E_T(l,d) = l * E_{elec} + l * E_{amp} * d^2 \quad (3)$$

$$E_R(l) = l * E_{elec} \quad (4)$$

Where, $ET(l,d)$ in Equation 3 and $ER(l)$ in Equation 4 denote, respectively, the total energy consumed in the source node transmitter and in the destination node receiver. The parameter E_{elec} represents the required electronic energy to run the transmitter or the receiver circuit. While, E_{amp} characterizes the energy dissipated by the transmitter amplifier.

5.2. Simulation Environment and Parameters

To position the efficiency of the CHEARP protocol, extensive simulation experiments are conducted under Matlab/Simulink environment, and the obtained results versus the recent DHT-based routing protocol: CLEVER [8] are likened. In this regard, a random deployed network of $100 - 500$ nodes is considered on a squared size field of $500 * 500\ m^2$, which means that the abscissa (horizontal) and ordinate (vertical) coordinates of each sensor are randomly selected between 0 and the maximum value of the space dimension. For each node is assigned a transmission range equals to $150\ m$ and an initial energy value of $2\ j$, while the energy values of E_{elec} and E_{amp} are respectively set to $50\ nj$ and $0,0013\ pj$. The summary of simulation parameters used in this model is given in Table 2.

Table 2. Simulation Parameters Value.

Parameter	**Value**
Sensor field	$500 * 500\ m^2$
Network size	$100 - 500\ nodes$
Packet size	$4000\ Bits$
Initial energy	$2\ j$
E_{elec}	$50\ nj$
E_{amp}	0.0013 pj
Node's transmission range	$150\ m$

5.3. Simulation Results

The subsequent sections illustrate the performance evaluation of CHEARP protocol compared to CLEVER. We grant more interest, particularly, to four important performance metrics, namely the transmission load, the end-to-end delay, the average dissipated energy and the network lifetime. The transmission load is measured as the number of packets in function of transmission frequency (number of transmissions per second) and the number of nodes. The end-to-end delay measures the time (S) that takes a transmission to achieve the destination. The average energy dissipation determines in joules the amount of depleted energy of nodes during the network operations.

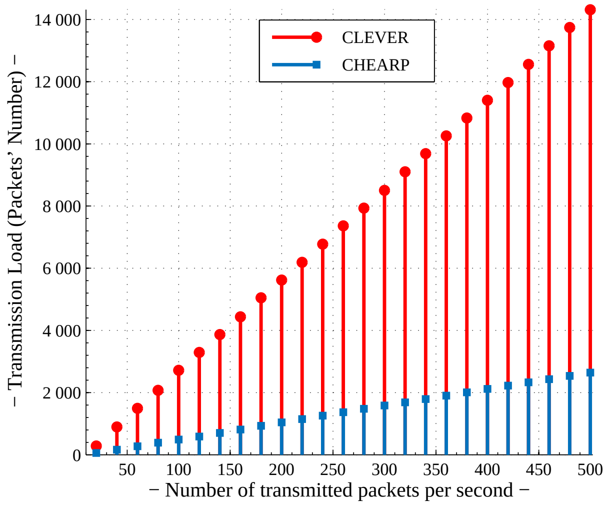


Fig. 8. The impact of transmission frequency on transmission load.

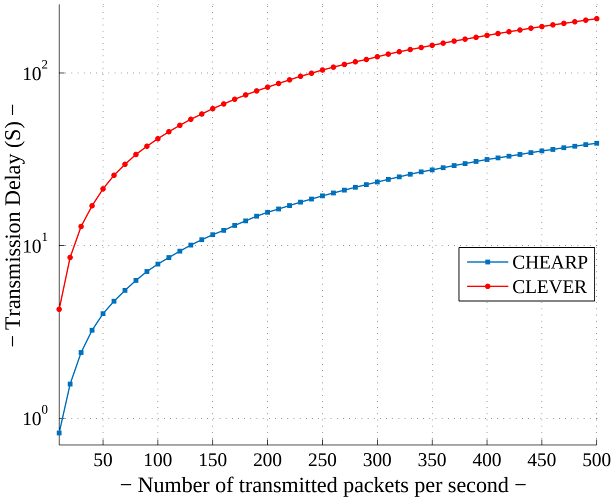


Fig. 9. The impact of transmission frequency on end to end delay

Transmission Frequency Impact. The transmission load of both CHEARP and CLEVER protocols in function of the transmission frequency is depicted in Figure 8. The results are obtained as the number of transited packets in the network versus the transmission frequency ranging from 10 to 500 packet/s. From this figure, CHEARP reveals good results compared to CLEVER. This is justified by the consideration of physical proximity of CH nodes that CHEARP provides. In fact, the proposed protocol shortens the transmission path, since the virtual successor is exactly the physical one, which reduces the

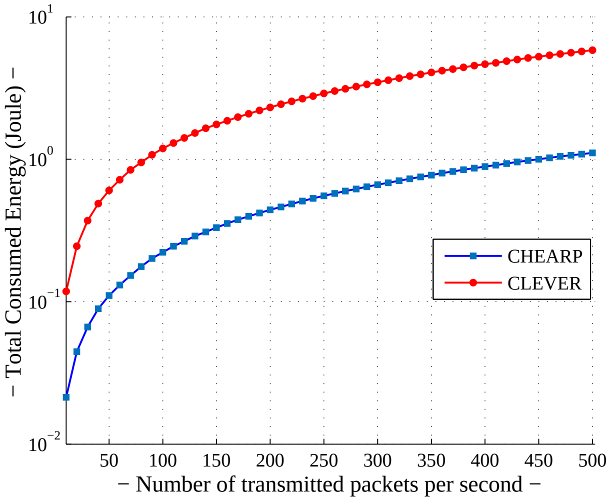


Fig. 10. The impact of transmission frequency on dissipated energy

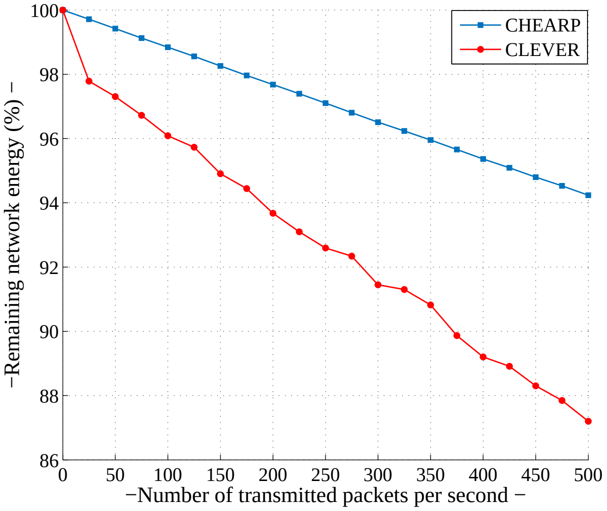


Fig. 11. The impact of transmission frequency on network lifetime

number of visited nodes during transmissions, and hence, the number of transited packets in the network. However, a virtual successor in CLEVER could be reached through several physical hops, which increases drastically the number of transmitted packets in the network, and consequently, the transmission load.

The performance of CHEARP and CLEVER in terms of packet transmission delay are compared in the Figure 9. The obtained results demonstrate clearly that CHEARP outperforms CLEVER with a tolerable increase in front of the excessively high transmission frequency values. Even the use of hierarchical structure and the minimization of the virtual path length, CLEVER shows higher values of end-to-end delay compared to CHEARP due to the high number of packets transmitted through several physical successor nodes before reaching the destination. In contrast, CHEARP allows reaching the destination in short delays, thanks to the approximate physical overlay that it uses.

Figure 10 and Figure 11 compare respectively, CHEARP and CLEVER, in terms of average dissipated energy and network lifetime, where CHEARP proves once more its efficiency against CLEVER. As depicted in Figure 10, CLEAVER consumes more energy than CHEARP since it involves many physical successor nodes in data transmission, which decreases the network lifetime.

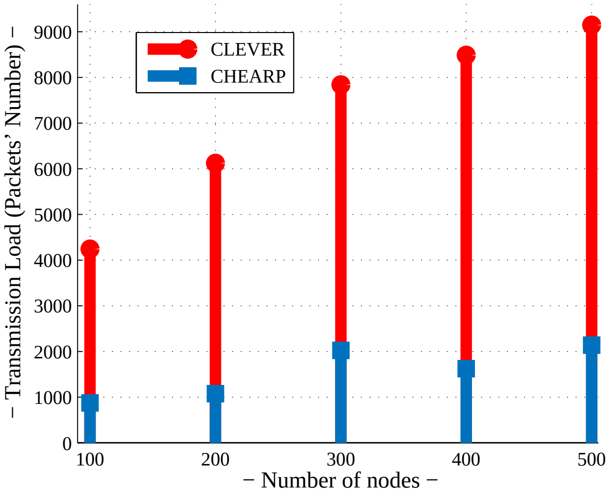


Fig. 12. The impact of network scalability on transmission load

Network Scalability Impact. In the Figures 12, 13, 14 and 15, the authors measure respectively, all of the transmission load, end-to-end delay, dissipated energy as well as the network lifetime, in function of the network size, that ranges from 100 to 500 nodes, where the transmission frequency is fixed at 200 $packets/s$. From these figures, it is noticed that CLEVER generates an increase in terms of load transmission, transmission delay and energy consumption compared to CHEARP. This is due to the lack of the number of transmitted packets through several intermediate sensor nodes (physical hops) before reaching the destination. Besides that, it is noticed that the percentages of the total remaining energy of both CLEVER and CHEARP are almost very close despite the increase of network size, since the total energy of the network increases with the increase of the number of nodes. Hence, even if increasing network size increases the path length, for a same value of transmission frequency, the dissipated energy, in function of network size increase, increases lightly. Through the obtained results, the proposed CHEARP protocol proves its effectiveness as a DHT-based routing solution for WSNs.

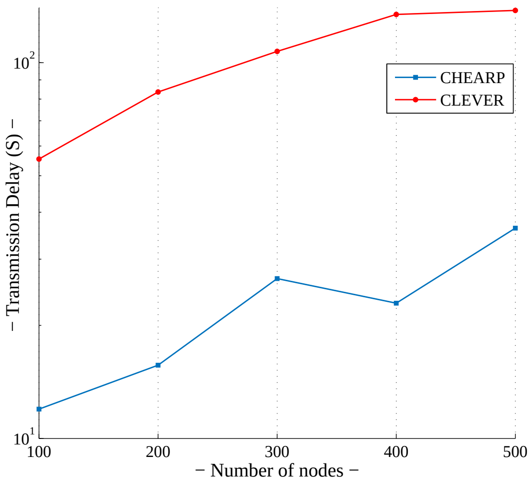


Fig. 13. The impact of network scalability on end to end delay

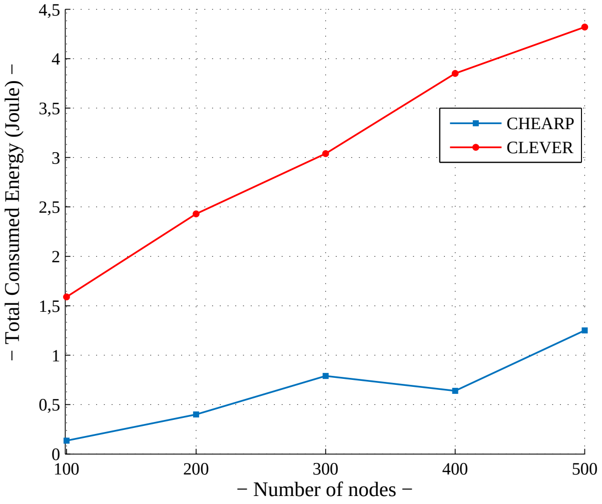


Fig. 14. The impact of network scalability on dissipated energy

6. Conclusion and Future Works

This paper presents a Chord-based Hierarchical Energy-Aware Routing Protocol (CHEARP), which addresses the energy consumption issue in WSNs. In this regard, the proposed solution takes benefits of distributed hash tables, and hierarchical structures to build a hybrid energy aware protocol. Indeed, the proposed protocol copes up the independence between the virtual overlay of the DHT and the physical network topology by re-identifying nodes

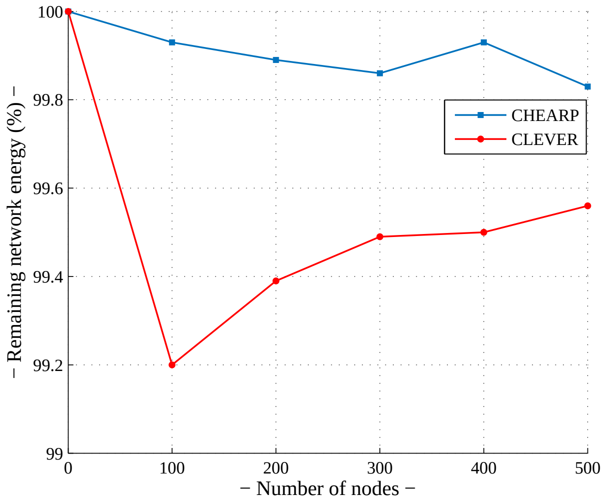


Fig. 15. The impact of network scalability on network lifetime.

during the virtual ring construction. Hence, the proposed solution approximates the two structures and constructs a kind of physical ring, in which CHEARP guarantees for each node, the same set of physical and virtual neighbors. Besides, the proposed protocol suits well the randomly deployed networks, where only a part of sensor nodes could reach the sink. By handling only the identifiers of the interconnected nodes to the sink, any node in the network could transmit the data packets to the sink, following routing principal similar to basic Chord. Furthermore, CHEARP is compared to another DHT-based routing protocol, CLEVER, through which we prove the effectiveness and the good results that CHEARP reveals. In other words, the proposed protocol and its approximate strategy get to reduce the routing path (hop count), which decreases the transmission load and the end-to-end delay, and hence it minimizes the dissipated energy and extends the network longevity, which is the main purpose of this work.

Acknowledgments. This work has been sponsored by General Directorate for Scientific Research and Technological Development, Ministry of Higher Education and Scientific Research (DGRSDT), Algeria.

References

1. Alami, H.E., Najid, A.: Routing-gi: routing technique to enhance energy efficiency in wsns. International Journal of Ad Hoc and Ubiquitous Computing 25(4), 241–251 (2017)
2. Ali, M., Uzmi, Z.A.: Csn: A network protocol for serving dynamic queries in large-scale wireless sensor networks. In: Proceedings of the Second Annual Conference on Communication Networks and Services Research. pp. 165–174. IEEE (2004)
3. Bhalaji, N., Prasanna, S.J., Parthiban, N.: Performance analysis of creido enhanced chord overlay protocol for wireless sensor networks. In: International Conference on Data Engineering and Communication Technology. pp. 489–499. Springer, Singapore (2017)

4. Caesar, M., Castro, M., Nightingale, E.B., O'Shea, G.and Rowstron, A.: Virtual ring routing: network routing inspired by dhts. In ACM SIGCOMM Computer Communication Review 36(4), 351–362 (2006)
5. Cheklat, L., Amad, M., Boukerram, A.: A limited energy consumption model for p2p wireless sensor networks. Wireless Personal Communications 96(4), 6299–6324 (2017)
6. Cheklat, L., Amad, M., Boukerram, A.: Wireless sensor networks, state of art and recent challenges: A survey. Sensor Letters 15(9), 697 – 719 (2017)
7. Fersi, G., Louati, W.and Jemaa, M.B.: Energy-aware virtual ring routing in wireless sensor networks. Network Protocols and Algorithms 2(4), 16–29 (2011)
8. Fersi, G., Louati, W., Jemaa, M.B.: Clever: Cluster-based energy-aware virtual ring routing in randomly deployed wireless sensor networks. Peer-to-Peer Networking and Applications 9(4), 640–655
9. Fersi, G., Louati, W., Jemaa, M.B.: Distributed hash table-based routing and data management in wireless sensor networks: a survey. Wireless networks 19(2), 219–236 (2013)
10. Gilbert, E.P.K., Kaliaperumal, B., Rajsingh, E.B., Lydia, M.: Trust based data prediction, aggregation and reconstruction using compressed sensing for clustered wireless sensor networks. Computers & Electrical Engineering 72, 894–909 (2018)
11. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on wireless communications 1(4), 660–670 (2002)
12. Li, W., Zhang, W.: Coverage hole and boundary nodes detection in wireless sensor networks. Journal of network and computer applications 48, 35–43 (2015)
13. Liu, X.: A survey on clustering routing protocols in wireless sensor networks. sensors 12(8), 11113–11153 (2012)
14. Saoudi, M., Lalem, F., Bounceur, A., Euler, R., Kechadi, M.T., Laouid, A., ..., Sevaux, M.: D-lpcn: A distributed least polar-angle connected node algorithm for finding the boundary of a wireless sensor network. Ad Hoc Networks 56, 56–71 (2017)
15. Singh, S.P., Sharma, S.C.: A survey on cluster based routing protocols in wireless sensor networks. Procedia computer science 45, 687–695 (2015)
16. Stecklina, O., Langendörfer, P., Goltz, C.: A lifetime forecast scheme for a distributed low duty cycle multi-hop routing in wireless sensor networks. International Journal of Business Data Communications and Networking (IJBDCN) 9(4), 1–22 (2013)
17. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Dabek, F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Transactions on Networking (TON) 11(1), 17–32 (2003)
18. Yetgin, H., Cheung, K.T.K., El-Hajjar, M., Hanzo, L.H.: A survey of network lifetime maximization techniques in wireless sensor networks. IEEE Communications Surveys & Tutorials 19(2), 828–854 (2017)
19. Zhao, L.H., Liu, W., Lei, H., Zhang, R., Tan, Q.: The detection of boundary nodes and coverage holes in wireless sensor networks. Mobile Information Systems 2016, 16 (2016)

Lamia Cheklat is an assistant professor at the Computer Science department and a member of RESA research team at LIMED Laboratory (Laboratoire d'Informatique MEDical), Bejaia University (Algeria). She got her PhD and Master degrees in Computer Science from the aforementioned University. Her research interest includes the fields of P2P networks, Wireless Sensor Networks and Internet Of everyThing.

Mourad Amad is an associate professor at Bouira University from October 2016. He is a member LIMPAF Laboratory. He received the engineer degree from the National

Institute of Computer Science (INI-Algeria) in 2003 and the magister degree from the University of Bejaia (Algeria) in 2005. Since January 2012, he is a PhD at the University of Bejaia, in May 2016; he obtained the HDR in computer sciences from Bejaia University. His research interests include: Peer to peer networks (Lookup acceleration, Security, Performance evaluation, Mathematics Modelling ...), Sensor Networks (Clustering, Routing ...), Media Conferencing (architecture, ALM, signalling protocol, P2P-SIP ...), Social Networks, Internet of Things (Architecture, Fault Tolerance . . .)

Mawloud Omar is member of LIGM laboratory and Associate Professor at ESIEE Paris in the University of Gustave Eiffel (France). He got his PhD and Magister degrees in Computer Science from the University of Bejaia (Algeria) with a focus on network cybersecurity. He was Senior Researcher at IRT SystemX (France). Before, he worked for several years as Lecturer and Researcher at the University of Technology of Compiegne and the University of Bejaia. He published several works in journals and conferences and participated actively in national and European projects. His research activities revolve around networking and cybersecurity. He is looking to the challenging issues related in particular to the Internet of things, 5G networks, connected vehicles and industrial environments.

Abdallah Boukerram is professor of Computer Science at University of Bejaia (Algeria). He obtained the PhD degree in Computer Science from University of Louis Pasteur Strasbourg 1991 (French). He is director of Network & Distributed System (LRSD) laboratory at the UFAS1. His current research domains are the networks, the security, the distributed systems, and grid computing.

Received: March 08, 2020; Accepted: November 15, 2020.

Computer Science and Information Systems 18(3):835–865 https://doi.org/10.2298/CSIS200522051P

Decision-Making Support for Input Data in Business Processes according to Former Instances

José Miguel Pérez Álvarez[1], Luisa Parody[2], María Teresa Gómez-López[3], Rafael M. Gasca[3], and Paolo Ceravolo[4]

[1] NAVER LABS Europe
6 Chemin de Maupertuis, 38240 Meylan, France
jm.perez@naverlabs.com
[2] Universidad Loyola Andalucía
Avda. de las Universidades s/n. 41704 Dos Hermanas (Sevilla), España
mlparody@uloyola.es
[3] Universidad de Sevilla
Av. Reina Mercedes s/n, 41012 Sevilla, España
{maytegomez,gasca}@us.es
[4] Universita' degli Studi di Milano
Via Festa del Perdono, 7, 20122 Milano, Italy
paolo.ceravolo@unimi.it

Abstract. Business Processes facilitate the execution of a set of activities to achieve the strategic plans of a company. During the execution of a business process model, several decisions can be made that frequently involve the values of the input data of certain activities. The decision regarding the value of these input data concerns not only the correct execution of the business process in terms of consistency, but also the compliance with the strategic plans of the company. Smart decision-support systems provide information by analyzing the process model and the business rules to be satisfied, but other elements, such as the previous temporal variation of the data during the former executed instances of similar processes, can also be employed to guide the input data decisions at instantiation time.
Our proposal consists of learning the evolution patterns of the temporal variation of the data values in a process model extracted from previous process instances by applying Constraint Programming techniques. The knowledge obtained is applied in a Decision Support System (DSS) which helps in the maintenance of the alignment of the process execution with the organizational strategic plans, through a framework and a methodology. Finally, to present a proof of concept, the proposal has been applied to a complete case study.

Keywords: Business processes, Input Data, Decision-making support, Evolution Models of variables, Constraint Programming, Process Instance Compliance

1. Introduction

The operational plans of organizations are documents that include in detail all technical and organizational aspects related to the development of their products or services. Organizations frequently perform their operations by using *Business Processes* to support the services offered. A *Business Process* consists of a set of activities that are performed in coordination within an organizational and technical environment to achieve an objective [43]. In process orientation, business processes are the main instrument for the organization [18], where commercial *Business Process Management Systems* are incorporated to facilitate the automation and monitoring of their daily processes, and that they are aligned with their objectives and data at the same time [32]. These systems support the implementation, coordination, and monitoring of the business process executions, and produce a great quantity of data that can be stored for its later application in deriving a data evolution pattern temporal variation.

In order to maintain the correct management defined in the strategy plans, companies evaluate the status of the organization by analyzing the suitability of their *Key Process Indicators* (henceforth referred to as KPI) [34]. However, some of the decisions made during the process instances take into account information not available when the decisions are made, because it depends on the activities that will be executed later in the process and how the KPIs will evolve in the future. The process model describes activities that will be executed until the process instance ends. However, to know how the KPIs will evolve, it is necessary to analyze former instances, and the values of the variables that represent the measurements concerning the evolution of the process instances. This data allows a target value to be monitored over specific periods through extraction this value from the execution of business processes. For this reason, this set of observational variables represents the stage of the business, in terms of measurements.

In this paper, we propose a methodology to support the decisions about the values of input variables in business process instances introduced by the business experts at runtime. Frequently, these decisions are made by the expert experience, that can be partial and subjective. But, what can be done when the process instance evolution is not aligned with the KPIs defined. For this reason in this paper, we support the decision-making for the alignment of the execution of the processes with the KPIs, guided by how previous instances of the same process evolved.

To clarify our contribution, the application of our proposal will be illustrated using an example. In the organization of a scientific conference, a set of decisions must be made. Months before the celebration and the attendees are registered, the business experts must decide the early and late registration fees, the venue, or the number of proceedings to print. These decisions affect the successful execution of the conference. Usually, to make these decisions, experts consult previous editions of the same or similar conferences. The analysis of how the number of registrations evolves, and how it is affected by the proximity to the end of the early registration period can avoid making incorrect decisions. Furthermore, the decision-makers can check if the information obtained is aligned with the business goals and organizational constraints.

1.1. Challenges

A methodology to guide decision point during process instantiation was addressed in previous work [15,14]. That methodology was based on modeling the restrictions extracted from the operational plans, by using a special type of *Business Rules* called *Business Data Constraint* [16].

However, the previous approximations fail in the following aspects, which are tackled in the current paper:

- Analyze the temporal variation of the relevant data during the process instances by consistently incorporating the fluctuating patters, their trends and recurring behavior.
- Incorporate into the decision point results obtained by cases with analog contextual patterns. The activities executed during each instance describe the stage of an instance, although other factors can influence the temporal variation of the data involved.
- The degree of uncertainty of the variables handled at the decision points, was not accounted in the previous methodology. It is essential to include this uncertainty since it is not the same to decide the value of a variable in an initial stage of the business process instance, in which its uncertainty is very high, as is in an advanced stage, in which the uncertainty is reduced.

1.2. Objectives

Taking into account the above challenges, in this paper we propose an extension of the previous study to achieve the objective: *The creation of a methodology that supports the decisions made during the process model execution about the input data values. The basis of the methodology is to provide the proper range of values taking into account the stage of the instance and the possible evolution of the variables in the future according to analog models, i.e. related former-instances.*

This objective implies the achievement of the three following sub-objectives:

1. **Problem Modeling:** During modeling time, the different parts of the problem are incorporated using the business *Expert Knowledge*. The elements that must be modeled are: The *Business Process Model*, *Dictionary*, *Stages* and *Business Rules*.
2. **Creation of evolution patterns of variables:** Before starting the instance in which decisions have to be made, it is necessary to create the the patterns of the data temporal variation, by analyzing how they evolved in analog models.
3. **Decision-making support for input data at runtime:** For each decision, the system uses the knowledge extracted from data temporal variation in previous instances, adapted to the current stage of the *Process-Observational Variables* in the decision-making moment. This is performed by using the Constraint Programming paradigm that provides different techniques to solve constraint problems.

Thanks to this new approach, the board and executive team of an enterprise can compare the current status of an instance with previous statuses, thereby helping to maintain the alignment of all business processes with the defined KPIs. On the other hand, thanks to the base of knowledge obtained, we will have the capacity to predict how the business instance will evolve, and how undesirable variation can be detected at an earlier stage.

The paper is organized as follows: Section 2 introduces a real world case study, used throughout the paper to illustrate the various aspects. Section 3 presents a graphical overview of the basic aspects of the methodology. The subsequent three sections describe the three phases of the methodology: Problem Modeling in Section 4, creation of variable evolution pattern in Section 5, and Decision-making support for the input data phase in Section 6. Section 7 describes related work. Finally, conclusions are drawn and future work is proposed in Section 8.

2. Case Study

In order to describe our proposal, a case study is presented to make the methodology accessible. The case study is a sample extracted from an event organization company focused on conferences. The company offers the service by using a web-based application, through which the customers can manage the conferences that they organize. Thanks to this service, the company has a large database of past events that can be consulted in order to improve decisions regarding conferences of the future. Figure 1 shows the process that support a conference organization that follows the operational business plans of the company. The example represents a research conference where participants can present a research work that has been previously reviewed and accepted. The process has been simplified to illustrate the example.

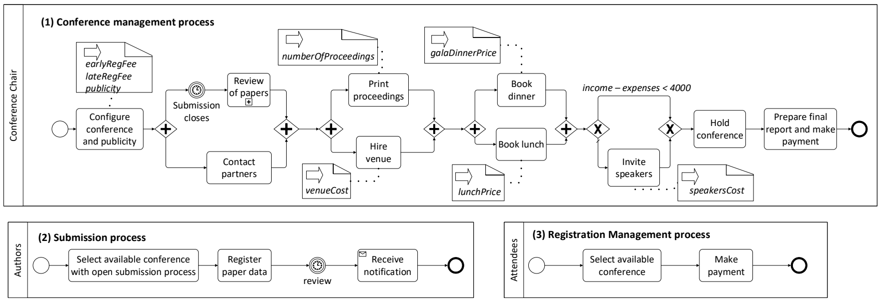


Fig. 1. Example of processes for Conference Organization.

The process *Conference management* is performed by the *Conference Chair*, and supports the operations for the establishment and management of the conference organization. The first task is to *Configure conference and publicity*, where the conference chair decides to set up the parameters of the conference, by defining the initial data values. Some of this data, represented by means of variables in the process, is related to registration fees at different times, available budgets, and the important dates. There is a data element associated with the activity *Configure conference and publicity* where the input variables, whose values must be established, are depicted (e.g. early registration fee, late registration fee, date of submission closes, date of early registration closes, etc.). Once the conference is configured, some values of the input variables cannot be changed, for example, the early registration fee, this is why to take into account how the process instance can evolve is so relevant. Afterwards, *Contact partners* is performed in parallel with the *Review of papers*, as shown in Figure 1. These two latter activities can start once the *Submission time* has expired. Subsequently, the *Print proceedings* and *Hire venue* tasks are executed in parallel. The subsequent steps are *Book gala dinner* and in parallel *Book lunch*. In the case where a profit of more than 4,000 euros is expected, speakers can be invited. This is performed in the task *Invite speakers*. Finally, the conference takes place when the task *Hold conference* is executed, and payments and final reports are performed in the task *Prepare final reports and make payments*.

The second process, *(2) Submission*, is carried out by *Authors*. The first task is *Select available conference with open submission process*, since several conferences can be available for the submission of contributions at the same time. The next task consists of collecting information on the author and the contribution in *Register the paper data*. The author must subsequently wait until the revision is completed. Finally, the author is notified once the task *Receive notification* is executed.

The third process is *(3) Registration management*, which is in a pool assigned to the *Attendees* for their registration in a conference.

This case study is challenging for several reasons: (1) it includes several actors in the process: organizers, authors, reviewers, and attendees; (2) there is a relationship between various processes with a significant number of shared variables, such as conference id, number of attendees, etc.; and (3) there are many decisions about variable values needed throughout the process that are key to the success of the process: registration fees, venue cost, lunch and gala dinner prices, etc. One of the main problematic issues of the conference organization is that several decisions must be made before a specific value is known. Many activities include decisions on the value of variables that are key to the successful celebration of the conference. For example, in the *Configure conference and publicity* activity, the values of registration fees are established without knowing the number of participants. A low registration fee and few attendees could mean that no speakers can be invited to give the keynotes or there may not be enough money to hire the gala dinner. In order to avoid this type of problems, these decisions tend to be made according to the previous experiences of the organizers, but frequently based on subjective aspects. The evolution of the number of papers sent, or the number of early registered can give clues about how the process instance can evolve in the future, making the current decisions more proper and accurate. Without the use of a decision-making support system, the organizer of a conference cannot come ahead of time when the number of attendance are not evolved as expected (in comparison with previous instances), and how the decisions of the future can be made to avoid an unwanted situation.

Thanks to our proposal, the organizers can obtain information of previous instances and make decisions according to this information at runtime. Following with the keynotes problem and let's assume that income is only dependent on registrations. When the organizers have to decide who to invite to give a keynote, they are going to have available not only the number of people registered up to that moment, but also how that value is evolving with respect to previous instances. If the evolution is very negative (many less registered), our framework will let the organizers know how much money they will have available approximately according to that evolution. How the framework is integrated in the business process and in which points the decisions are made are explained in the following sections.

3. Methodology for Decision-Making Support of Input Data

The main objective of our proposal is to assist the business experts in the decision-making during process execution. Figure 2 graphically summarizes our methodology. The summary is introduced in this section, and details are provided in the following sections.

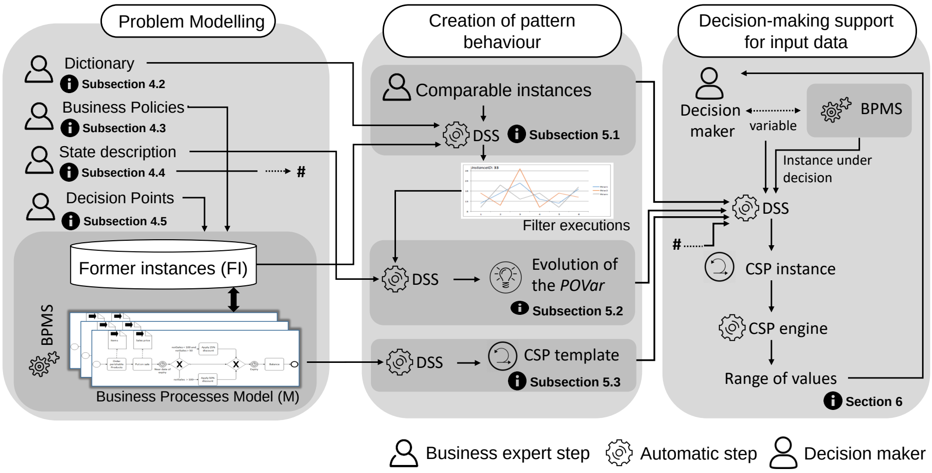


Fig. 2. Steps of the Methodology proposed.

1. **Problem Modeling:** This takes place during the definition of the business process by the business expert as detailed in Section 4:
   - The *Business Process Model* must be included to describe the behavior of the organization (Subsection 4.1).
   - The *Dictionary* of *Process-Observational Variables* is defined, with the objective of maintaining a common language that facilities the following steps of the methodology, explained in Subsection 4.2.
   - *Business Rules* enrich the *Business Process Model* to describe the business rules involving *Process-Observational Variables* as described in Subsection 4.3.
   - The set of relevant *Stages* (*S*), in which the *Business Process Instances* (*BPI*) can be, must be defined. Based on the strategic plans of the company and the business experts' knowledge, in Subsection 4.4 we propose a mechanism to model and compute the *S* that represents the *BPI* temporal variation.
   - Identify activities where decisions will be made, so that different decision points can be pinpointed in the business process (Subsection 4.5).
2. **Creation of the evolution pattern of the variables:** The temporal variation of the relevant variables and how each stage can affect them is fundamental in the later decision-making process. Since the *BPI* works in various ways, the selection of a subset of instances similar to the instance subject to a decision is essential. Section 5 details the following information:
   - A mechanism enables business experts to specify the *Comparable Instance* criteria, to select the former instances that are more related to the instance under decision. These aspects are discussed in Subsection 5.1.
   - According to the *Comparable Instance* and by analyzing the former instances, the temporal variations of the *Process-Observational Variables* are obtained as

detailed in Subsection 5.2. The discovery of the evolution pattern is crucial for the understanding of how the business works, thereby helping in the decision-making process.
- A model template is automatically created by traversing *Business Process Models* and combining the *Business Rule* associated with each activity. Once the evolution patterns are created, we propose the use of the Constraint Programming Paradigm, thereby creating a *Constraint Satisfaction Problem* template that contains the temporal variation patterns obtained and the structure of the process, as detailed in Subsection 5.3.

3. **Decision-making support for input data:** During the execution of each *Decision Point* in an instance under decision, decisions regarding input data can be made. In order to determine the correct input domain, we propose the combination of the current stage with the *Constraint Satisfaction Problem* created in the previous step, as discussed in Section 6.

4. Problem Modeling

The model of the problem is composed of a Business Process Model, a Dictionary of Process-Observational Variables, the Business Rules, the Stage descriptions, and the Decision Points. Each component is described in the following subsections.

4.1. Business Process Model

Business Process Models permit the description of the activities developed by an organization. The standard BPMN 2.0 [31] is usually employed for this purpose. The example of Figure 1 includes three business processes. Business Processes can include: a start event, end events, activities, gateways, and conditions associated with the gateway branches (OR and XOR). These components are combined in a control flow structure that manages the order of execution of the activities.

4.2. Dictionary (D)

In order to describe the relevant control variables that will be involved in the business rules and in the KPIs, the business experts must describe the *Dictionary* of terms. A *Dictionary* is formed of a set of *Process-Observational Variables* $\{\text{POVar}_1, \ldots, \text{POVar}_n\}$, whose values are relevant for the organization to know whether their policies comply and their KPIs are reached or not. Each *Process-Observational Variable* is composed of a name of the variable and its description ($\langle$*name, description*$\rangle$). The description contains information that enables its value to be extracted from the *Business Process Management System* for each instance. In our case study, examples of the *Process-Observational Variables* include:

- `earlyRegFee`: Cost in the early registration period.
- `lateRegFee`: Cost in the late registration period.
- `maxPapers`: Maximum number of papers that can be accepted.
- `submissionOpen`: Date when the paper submission opens.

- `regOpen`: Date when the registration opens.
- `earlyRegClose`: Date when the early registration closes.
- `submissions`: Number of submissions to the conference.
- `earlyReg`: Number of registrations in the early registration period.
- `totalReg`: Number of attendees.
- `sponsorship`: Total amount reached by sponsors.
- `acceptedPapers`: Number of accepted papers.

In order to describe the meaning of the *Process-Observational Variables* and to automatically evaluate and monitor these variables, the use of *Process Instance Query Language (PIQL)* [33] is proposed. PIQL enables business experts to extract information from the process instances that match with specified criteria, and includes a set of operations over the selection. Not only does PIQL allow the selection of instances under execution, but also that of former instances. PIQL is oriented to business experts, offering a natural-language-like specification.

It should be borne in mind that although PIQL is the language selected to specify *Process-Observational Variables*, the study of PIQL itself remains outside of the scope of this paper. Certain details are provided below for a better understanding, but the formal syntax, and in-depth details are available in [33]. The types of operations that can be performed over processes and task instances can be seen in Table 1.

Table 1. Operations over a set of process and task instances.

Operation	**PIQL Syntax**
Count all selected instances	The number of instances of processes
Obtain the value of a variable of the data-flow if just one instance matches	The value of *variable* of the process *p*
Obtain the average of values of a variable of the data-flow for the selected instances of a process	The average value of *variable* of the process *p*
Obtain the maximum value of a variable of the data-flow for the selected instances of a process	The maximum value of *variable* in the process *p*
Obtain the minimal value of a variable in data-flow for the selected instance of a process	The minimum value of *variable* in the process *p*
Count the task instances for all the selected instances	The number of instances of task *t*

Additionally, the attributes of Table 2 can be used for the filtering of processes and task instances. Moreover, certain arithmetical, logical, and comparison operators can be used. Moreover, a set of predicates can also be used as detailed in Table 3.

By using PIQL, several of the *Process-Observational Variables* specified above can be modeled as follows:

- *maxPaper*: The value of *maxPaper* of the process “Conference Management”.
- *regOpen*: The value of *regOpen* of the process “Conference Management”.
- *submissions*: The number of instances of “Submission process” that are finalized.
- *earlyReg*: The number of instances of “Registration Management process” that end before *earlyRegClose*.

- *totalReg*: The number of instances of “Registration Management process” that are finalized.
- *acceptedPapers*: The number of instances of “Submission process” with “accepted” = “true”.
- *notified*: The number of instances of “Author notifications” task of process “Conference Management process”.

Table 2. Attributes of process and task instances.

Attributes	**PIQL Syntax**
idCase	with a case id *idCase*
Process_Name	with a name *process_name*
Task_Name	with a name *task_name*
Start	with a start date *date*
End	with an end date *date*
Cancelled	*cancelled*
Who	executed by the user *user*

Table 3. Predicates allowed

Predicates	**Transformed pattern**
are finalized	end date is not equal to Null
are not finalized	end date is equal to Null
are cancelled	cancelled is not equal to Null
are not cancelled	cancelled is equal to Null
executed by {name}	the user is equal to {name}
start before {date}	a start date is less than {date}
end before {date}	an end date is less than {date}
start after {date}	a start date is greater than {date}
end after {date}	an end date is greater than {date}

4.3. Business Rules

The strategic plan of a company is commonly a set of documents written in natural language. These documents must be translated into something computable in order to know whether a *Business Process Instance* has finalized correctly or not, according to the policies of the companies. Numerous studies propose a variety of taxonomies to classify the definition of business rules [7,19]. One of the most frequent definitions is: A specification, a policy or a standardized procedure, that represents a natural step towards the inclusion of semantic requirements between business functionality and data. However, if the relations between the variables that delineate the *Stages* of the business have to be described, then *Business Rules* have the capacity to describe Business Compliance Rules [5]. This is the focus of this subsection, the modeling of the *Business Rules* by using an adaptation of the *Business Data Constraints* introduced in [13,16].

A *Business Rule* is a Boolean combination of numerical constraints whose evaluation can be true (satisfied) or false (unjustifiable). These numerical constraints can be specified by means of operators of comparison and *Process-Observational Variables* defined in the *Dictionary*. *Business Rules* represent the semantic relation between the data values that are introduced, read, and modified during the execution of the *Business Process Instances* [12].

In our case study, the set of constraints associated with the main process includes: Expenses must be less than or equal to income for every conference; The maximum accepted papers cannot be exceeded; All attendees must possess a copy of the proceedings; It is forbidden to exceed the maximum capacity of the auditory.

Business Rules can be associated with an activity, a set of activities, or as an invariant of the whole process. For example, when the activity *Invite Speakers* is executed, then the constraint *the total cost of bringing a speaker to the conference cannot exceed 30% of the income reached by sponsorship* must be satisfied.

Business Processes describe relations between *Process-Observational Variables*. The scope of these *Process-Observational Variables* can involve various instances of numerous processes. *Business Processes* therefore take into account not only relations between the local status of individual instances of a process, but also the global status of the company. *Business Rules* have been addressed in [15], which is adapted to our proposal, where both the *Process-Observational Variables* and the variables defined in the data-flow of the process can be involved.

These constraints can appear from different sources, such as business strategic plans, business rules, and manager restrictions. The principal aspect here is to determine if these constraints are mandatory for the defined scope, since, in the case where they remain unsatisfied, the process cannot be considered as having been finalized successfully. One example of *Business Process* related to the case study is: "*expenses must be less than or equal to income for every conference*". This is defined by the manager, and in the case where it remains unsatisfied, the instance of the process in which it has been defined, will not finish properly.

In our case study, the set of constraints associated to the main process is:

- Expenses must be less than or equal to income: $expenses \leq income$.
- All attendees must possess a copy of the proceedings: $numberOfProceedings \geq totalReg$.
- It is forbidden to exceed the maximum capacity of the auditory: $venueCapacity \geq totalReg$.
- The cost of inviting a speaker must be less than 30% of sponsorship income: $speakersCost \leq sponsorship \times 0.3$.

On the other hand, there are also other *Business Rules* oriented towards the definition of intermediate variables, such as:

- **Income** is the result of multiplying the number of early registrations by the early registration fee, plus the result of multiplying the number of late registrations by the late registration fee, plus the income by sponsorship: $income = earlyReg \times earlyRegFee + lateReg \times lateRegFee + sponsorship$.

- **Expenses** is the result of adding the total cost of publicity, plus the number of copies proceedings printed multiplied by the price of each copy, plus the total venue cost, plus the gala dinner cost, plus the lunch cost, plus the total cost of inviting speakers to the conference: $expenses = publicity + numberOfProceedings \times proceedingPrice + venueCost + galaDinnerPrice + lunchPrice + speakersCost.$

BRs are stored and queried using a *Constraint Database* as described in [37,11].

4.4. Business Process Stages

The temporal variation of the *Process-Observational Variables* is frequently related to the execution of the activities that form the *Business Process Model*. However, certain external factors may also influence these variables. For instance, conference chairs know that the information about the number of registrations (`TotalReg`) of a conference increases fairly quickly once paper notification has been performed. However, this type of information cannot always be included explicitly in a BPMN model, and cannot be extracted directly from the business process. Since this information is crucial to know the variation at any moment, we propose a simple description of these *Stages*, to be used in the process instance analysis developed for the decision-making process. When a decision has to be made, the user will consult the status of the variables related to this decision.

As can be observed, and according to our case study, the definition of these *Stages* is not exclusively dependent on which activities are being executed, and is related to the variation of the values of some *Process-Observational Variables*. For example, in our case study, there is no specific activity to make a late registration, but there is a period of time where the registration is more expensive (after the *earlyRegClose*). Therefore, even though all registrations are made using the *Registration Management Process*, the variation of the *number of registrations* depends on the stage of the registration (early or late).

Let S be $\langle s_1 \ldots s_n \rangle$, which represents all *Stages* defined by the business experts for a *Business Process Model*, where at least one stage must be defined. In our case study, the set of stages defined are:

- Paper submission: Period of time in which the paper submission is open to authors. The period between `submissionOpen` and `submissionClose`.
- Early registration: Period of time in which the registration is the cheapest. More specifically, the period between `regOpen` and `earlyRegClose`.
- Late registration: Period of time in which the registration is open in a late registration phase, which is the period between `earlyRegClose` and `regClose`.
- Notified: When the activity *Notify* is executed.
- Conference: Period in which the activity *Hold Conference* is executed.

In order to describe the *Stages*, we also propose the use of *Business Data Constraint* [15], by using the *Process-Observational Variables* defined in the *Dictionary*, as shown above. $currentDate$, $currentDay$ and $currentHour$ placeholders are also allowed, and the engine automatically sets their values at runtime.

Listing 1.1 shows an example of *Stages* using *Business Data Constraints*, where the *Stages paperSubmission*, *earlyReg*, *lateReg*, *notified*, and *conference* are defined.

```
[paperSubmission]
   submissions ≥ 0 AND notified = 0
[earlyReg]
   regOpen ≤ $currentDate AND earlyRegClose > $currentDate
[lateReg]
   earlyRegClose ≤ $currentDate AND regClose > $currentDate
[notified]
   notified > 0
[conference]
   holdConference > 0
```

Listing 1.1. Stages defined for the example

There are no restrictions regarding the number of stages defined by the business experts. They can define the set of stages that they consider relevant for the knowledge extraction phase. The overlapping of stages is of no consequence; this issue will be considered in the normalization phase explained in Subsection 5.2.

At moment (t) of the *Business Process Instance*, the possible *Stages* defined by a business expert can be in either of two positions: *Activated* or *Deactivated*. A status is *Activated* when the *Business Process Instance* meets the conditions defined for this *Stages* and the Boolean expression is *true*; it is *Deactivated* and *false* otherwise.

4.5. Decision Points

During the instantiation of a business process model, the activities can include forms to introduce values of variables of the process. The value of the input variables can be provided by the third party not being necessary a decision (e.g. the gathered sponsorship), or by a business expert as a product of decision-making about the most proper value for the variable (e.g. the number of copies of the proceedings according to the estimation of attendance before the final number is known). A *Decision Point* is associated to activity of the business process, where some values of a set of input variables must be introduced at instantiation time [6]. Each *Decision Point* is formed of a tuple ⟨*Decision Variables*, activity⟩.

For the first process of Figure 1, (1) Conference Management Process, there are six *Decision Points* (*Decision Variables*, activity):

- *Decision Point* 1: ⟨{*earlyRegFee*, *lateRegFee*, *publicity*}, Configure conference and publicity⟩
  - `earlyRegFee`: Cost of early registration fee.
  - `lateRegFee`: Cost of late registration fee.
  - `publicity`: Estimation of the cost for publicity actions.
- *Decision Point* 2: ⟨{*numberOfProceedings*, *proceedingPrice*}, Print a copy of the proceedings⟩
  - `numberOfProceedings`: The number copies of the of proceedings to print.
  - `proceedingPrice`: The price of printing a copy of the proceedings of one conference.
- *Decision Point* 3: ⟨{*venueCost*}, Hire venue⟩
  - `venueCost`: Venue cost.

- *Decision Point* 4: $\langle\{$*galaDinnerPrice*$\}$, Book dinner$\rangle$
  - `galaDinnerPrice`: The price of gala dinner.
- *Decision Point* 5: $\langle\{$*lunchPrice*$\}$, Book lunch$\rangle$
  - `lunchPrice`: The price of lunch.
- *Decision Point* 6: $\langle\{$*speakersCost*$\}$, Invite speakers$\rangle$
  - `speakersCost`: The total cost of inviting speakers to the conference.

When a process instance reaches an activity with a *Decision Point*, the DSS must evaluate the range of values for the input data that enable the aim of the *Business Process* to be achieved. This range is derived from the analysis of the former instances of the process to ascertain how the values of the *Process-Observational Variable* can evolve until the process ends according to the stage of the current process instance. The following section analyzes how the *Process-Observational Variable* evolution patterns are created in order to facilitate decision making.

5. Creation of evolution pattern of Process-Observational Variables

In order to make the decisions during process execution, it is important to take into account the context of the decision, in terms of the stages of the *Business Process Instance*. For example, to decide the most appropriate number of copies of the proceedings to print (`numberOfProceedings`), it is necessary to derive the number of attendees (`totalReg`). However, this decision must be made several months before the conference starts. The number of attendees can be derived from the number of those already registered and how this value evolved in previous and similar conferences (e.g. conferences in the same research area, city, or period). Obviously, the final `totalReg` depends on the current value of the variable, and the rest of POVar, or the stage of the instance (before or after the `lateReg` stage). How the POVar can evolve in the future can be derived by analyzing the BP, and how they evolved in previous process executions. However, in order to compare the *Business Process Instance* under decision with previous ones, it is necessary to extract only the *Comparable Instance* (e.g. conference of the same research area) as explained in Subsection 5.1, and to extract the evolution patterns (Subsection 5.2). The subsections below describe how this can be performed.

5.1. Specify Comparable Instances

A *Business Process Instance* (BPI) represents a specific case in an operational business process, and an execution of a *Business Process Model*. All $BPIs$ $\{bpi_1 \ldots bpi_n\}$ of a model, are individually described by the tuple $\langle M, Start, End, UpdateEvents\rangle$: M is the *Business Process Model*, $Start$ is the start data when bpi_i started; End is end date when the bpi_j finalized (empty if not finalized); and $UpdateEvents$ is a set of updated events, that is, the moments at which the value of any variable of the BP instance has been modified. We differentiate between two types of *Business Process Instance*:

- Non-former instances (NFI): *Business Process Instances* that are still under execution, that is, are not finalized. $NFI \in BPI$; $\forall nfi \in NFI, nfi.End = null$. In certain non-former instances, decisions regarding data must be made; these instances are called Instances Under Decision.

- Former instances (FI): *Business Process Instances* already finalized. $FI \in BPI$; $\forall fi \in FI, fi.End \neq null$. The set of former instances constitutes a major part of our proposal, since they represent historical information about the executions. In some of these former instances, decisions have been made.

The analysis of former instances will be very useful to understand the temporal variation of the variables in the various *Business Process Instances*, and the DSS is able to use this information to help business experts make better decisions regarding the instances under study. A subset of the former instances can be selected to create a evolution pattern for the decision point.

Let comparable instances $CI\ \langle ci_1 \ldots ci_n \rangle$ be the set of former instances that share certain characteristics with the instance under study. These related instances are able to analyze the *Process-Observational Variables* in former instances, to improve the decisions. In our case study, examples of characteristics defined by business experts include: (i) Topic: software engineering(ii) End date: 5 days before the current date; and (iii) Number of registrations: the final register will be less than 250 and greater than 100 attendees.

The former instances that comply with the criteria defined above are considered comparable instances to the non-former instances in which decisions must be made. Therefore, the former instances that comply are valid for analysis and for the extraction of useful knowledge to be used in the DSS.

By using the set of variables defined in the *Dictionary*, and by using the grammar defined for PIQL, business experts can define those characteristics as shown in Listing 1.2.

```
topic IS EQUAL TO 'software engineering' AND
holdConference IS GREATER THAN $currentDate − 5 years AND
totalReg IS LESS THAN 250 AND totalReg IS GREATER THAN 100
```

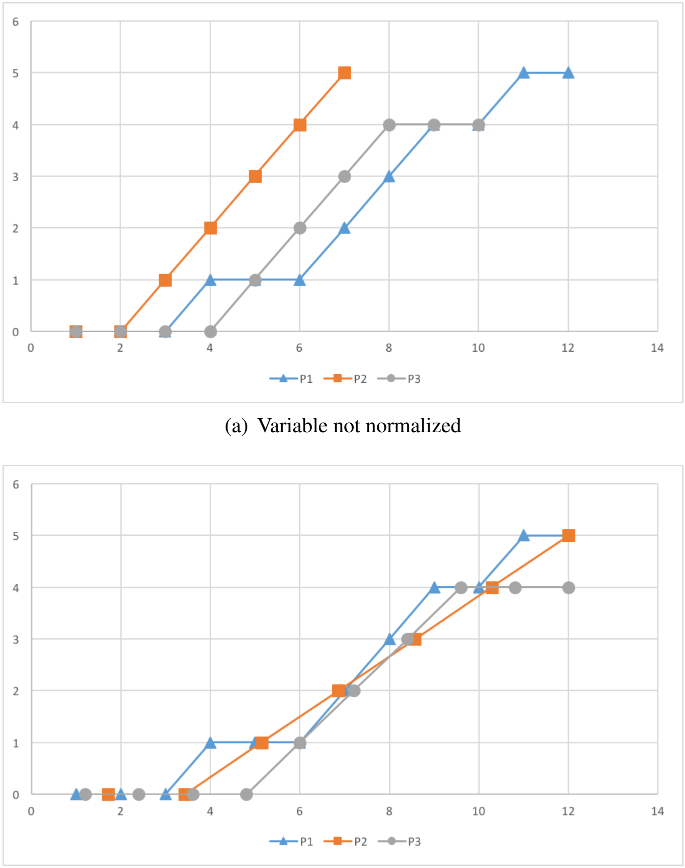
Listing 1.2. Example of the definition of Comparable Instances

5.2. Extracting of patterns of evolution of Process-Observational Variables

This step consists of automatically creating a temporal variation model of the *Process-Observational Variables* of the business processes by analyzing the former instances.

It should be borne in mind that each instance and its aforementioned *Stages*, can have a different duration. For instance, in the example presented in Section 2, conference managements can differ in their duration. Figure 3.a shows the variation of the *Process-Observational Variable* "number of registrations" (`totalReg` for the organization of three conferences (i.e., three different instances of "conference management process"). In order to facilitate the understanding of the example, these three processes start at the same time ($t = 1$). As can be observed, P1 has a duration of 40 units of time, P2 has a duration of 20 units of time, and P3 has a duration of 30 units of time.

The differences in the duration introduce complications into the comparison of the POVar variation between different *Business Process Instances* and into the creation of evolution patterns. However, this comparison is essential to obtain expected data value evolution. For this reason, it is necessary to establish mechanisms that enable the comparison of various instances with different duration and into various periods of activities or stages.



(a) Variable not normalized

(b) Variable normalized

Fig. 3. Normalization Process of the execution time of a *Process-Observational Variable*

Normalization The most evident mechanism of comparison is that of data normalization [10]. A representation of this mechanism is shown in Figure 3, where the execution times are normalized according to the events that occur during the instance-life. Normalization involves the adjustment of the values in different scales to a notionally common scale. Once the *Business Process Instances* are established on the same scale, they can then be compared.

However, the evolution of the values of the *Process-Observational Variables* is not constant throughout the whole life of the *Business Process Instance*. The evolution can differ depending on the *Stages*. Moreover, the duration of each *Stages* in different instances can be different, being necessary the normalization of the *Process-Observational Variables* when they are compared. For this reason, our proposal includes a normalization process that homogenizes the evolution of the values of the *Process-Observational Variables* to be comparable, taking into account the *Stages* and the duration of each business instances.

Therefore, the normalization process consists of two phases: (1) Displace the start instant of all *Comparable Instances* to the same instant, *Start=1*; and (2) Set the different duration of *Business Process Instances* to a common scale to compare the temporal variation of each variable involved in the various instances.

Thanks to this normalization, every instance is scaled to the same duration, where this duration is the maximum duration of the former instances (12 units of time for this example), as shown in Figure 3.b. Once normalized, the *Business Process Instances* can be compared, and the degree of similarity among the temporal variations can be observed. Table 4 shows an application of the *Stages* defined by business experts over the set of data shown in Figure 3.a. In this case, we can observe three possible *Stages* represented with three colors. Three sets of data are obtained (because three stages have been reached), and the results once the stage is started are shown graphically in Figure 4.

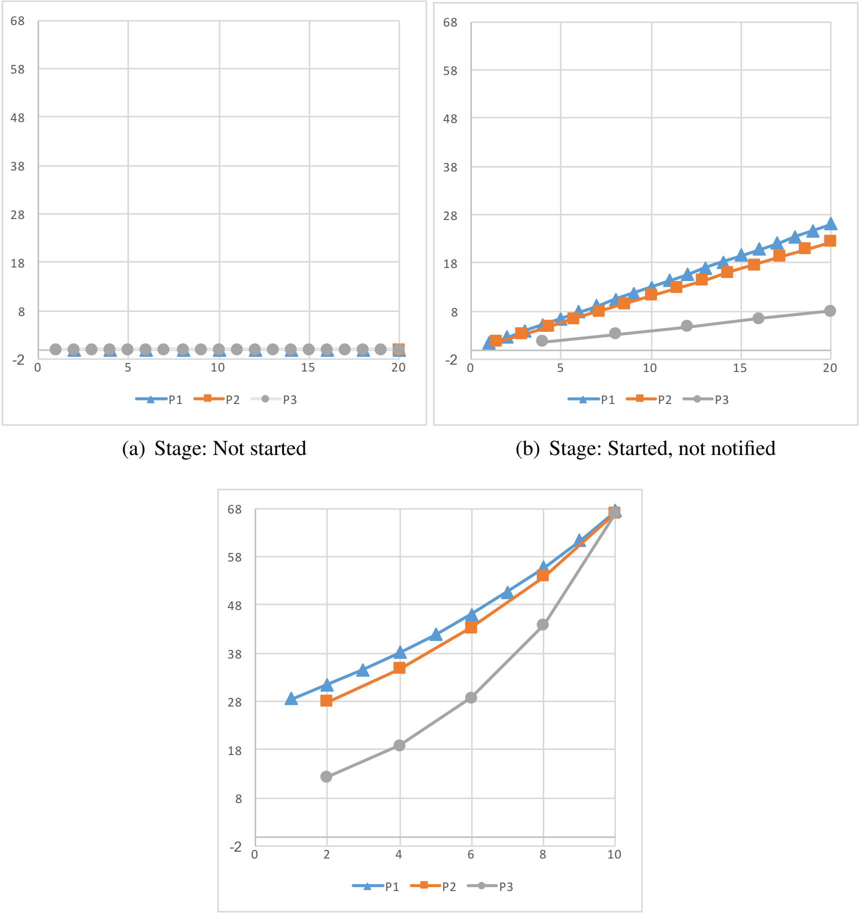
Table 4. Variable *Registrations* with the stages colored in each instance

t	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
P1	0	0	0	0	0	0	0	0	0	0	1	3	4	5	7	8	9	10	12	13	14	16	17	18	20	21	22	23	25	26	29	31	35	38	42	46	51	56	61	67
P2	0	2	3	5	6	8	10	11	13	14	16	18	19	21	22	28	35	43	54	67																				
P3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	5	6	8	12	19	29	44	67										

Finding out the Envelopment Temporal Variation of Process-observational variables Once the temporal variation of the variables is normalized, techniques of knowledge extraction can be applied. For every detected possible combination of *Stages* and all variables in these *Stages*, this step consists of computing the minimum and maximum slope of the *Process-Observational Variable*. It represents the degree by which the variable increases or decreases, from the *Stages* understudy, to the final value of the variable, at the point of finalization. This implies the computing of the maximum and minimum percentage of increment or decrement of the variable from the *Stages* to the end of the process. Slopes for early stages contain more uncertainty than those for late stages; however, this information will be beneficial in the estimation of the potential final range of values of the *Decision Variable*.

There exist numerous references in the literature, where methodologies and techniques are proposed for the creation of models by using the observations, such as curve fitting [24], linear regression [29], and non-linear regression [30]. These models are needed for the computation of the potential increase or decrease of the variable, and hence, our proposal is based on the calculation of the slopes, although other techniques, such as those described below, could be applied.

The slopes are computed by dividing the difference between the final and initial value by the duration of the process. Minimum and Maximum slopes are calculated by taking



(a) Stage: Not started

(b) Stage: Started, not notified

(c) Stage: Started, notified

Fig. 4. Variable normalized according to the stage

minimum and maximum values into account, as shown in Figure 5 where the *Stages* is “started, not notified”. This consists of calculating the slopes by using the lowest value for any instance at the beginning of the stage, and the lowest final value of an instance for the computation of the minimal slope. On the other hand, by using the highest values at the beginning of the stage for any instance and the highest values at the end of the process, the maximum slope is computed.

5.3. Combination of the element of the Models by means of a CSP template

Once the different elements of the model have been described or derived from former instances, they must all be combined for overall reasoning. This implies to link the business rules to the activities, along with the *Decision Points* and the possible envelopment variation according to the stages. We propose to use an approximation similar to [15], where

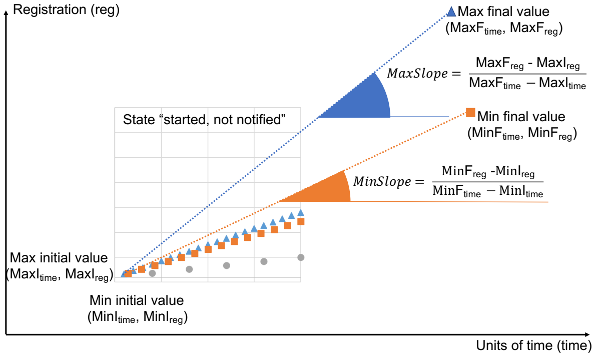

Registration (reg)
Max final value
$(MaxF_{time}, MaxF_{reg})$
$MaxSlope = \frac{MaxF_{reg} - MaxI_{reg}}{MaxF_{time} - MaxI_{time}}$
State “started, not notified”
Min final value
$(MinF_{time}, MinF_{reg})$
$MinSlope = \frac{MinF_{reg} - MinI_{reg}}{MinF_{time} - MinI_{time}}$
Max initial value
$(MaxI_{time}, MaxI_{reg})$
Min initial value
$(MinI_{time}, MinI_{reg})$
Units of time (time)

Fig. 5. Graphical representation of the max and min slope for one stage

the elements were associated to the business process model according to the control-flow structure of the process. The various business rules associated to each activity or set of activities (as explained in 4.3) can be combined in a *Constraint Satisfaction Problem* to ascertain how the *Process-Observational Variables* variation can affect each decision. Since the *Business Process Model* contains elements that route the execution flow of the process (such as gateways), it is necessary to analyze the control-flow structure and its associated business rules. The problem of analyzing the structure of the business process has been addressed in [15], which tackles how to obtain the set of *Business Rules* in accordance with the control-flow. The basic principles of how the Constraints are combined following the control flow are depicted in Figure 6. The set of patterns to combine the constraints are:

- Sequence (Figure 6.a). All the instances must execute these activities, hence all the Business Rules must be satisfied. Therefore they are put together with an AND Boolean relation between them.
- AND Split (Figure 6.b). Similarly with the AND split control flow, all the instances have to execute all the activities of the different branches, although the order is unknown. Therefore, the business rules of all these activities will be combined by means of an AND Boolean combination.
- XOR Split (Figure 6.c). In the case of the XOR control flow, where only one branch can be executed for an instance, the condition associated with each branch will be combined with the Business Constraints of the activities for each branch. The Business Constraints of the activities of a branch have to be satisfied only if the branch is executed. Therefore, the Business Rules of the activities of the different branches will have an OR Boolean relation between them, and the conditions are combined with an AND Boolean relation with the Business Rules of the activities for each branch. A special treatment is performed for the default branch, where the conditional flow of execution implies the non-compliance of the condition for the rest of the branches, thereby implying the negation of the rest of the conditions.

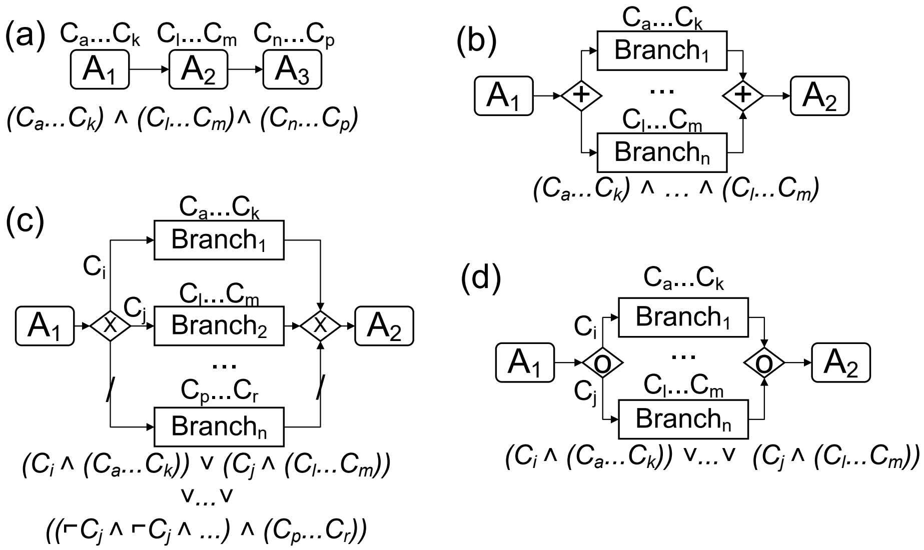

(a)
$C_a \ldots C_k$
$C_l \ldots C_m$
$C_n \ldots C_p$
A_1
A_2
A_3
$(C_a \ldots C_k) \wedge (C_l \ldots C_m) \wedge (C_n \ldots C_p)$
(b)
A_1
$Branch_1$
$Branch_n$
A_2
$(C_a \ldots C_k) \wedge \ldots \wedge (C_l \ldots C_m)$
(c)
C_i
C_j
$Branch_1$
$Branch_2$
$Branch_n$
$C_p \ldots C_r$
$(C_i \wedge (C_a \ldots C_k)) \vee (C_j \wedge (C_l \ldots C_m))$
$\vee \ldots \vee$
$((\neg C_j \wedge \neg C_j \wedge \ldots) \wedge (C_p \ldots C_r))$
(d)
$(C_i \wedge (C_a \ldots C_k)) \vee \ldots \vee (C_j \wedge (C_l \ldots C_m))$

Fig. 6. Combination of *Business Rules* in terms of the control flows and their conditions

- OR Split (Figure 6.d). OR control flow is very similar to XOR, the only difference being that more than one branch can be executed, and hence the default option negating the rest of the branches does not appear since this would make no sense.

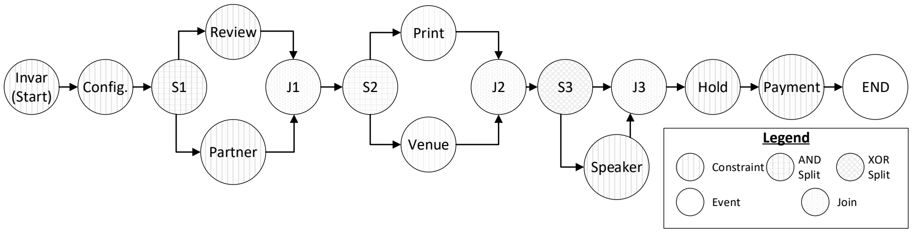


Fig. 7. BPMN Graph for Conference Organization.

These algorithms enable to approach a *Business Processes* as a BPMN-graph that is traversed to build a Constraint Satisfaction Problem that guides during the decision-making process. For example, the BPMN-graph created for the process used in the paper is shown in Figure 7. Each node represents: start event (with the invariant constraints associated); an activity with the associated constraint, if correspond; gateways (and-split, and-join, xor-split, xor-join), and; end event.

The problem obtained has the following form:

$C_{Invariant} \wedge (C_{Config} \wedge (C_{Review} \wedge C_{Partner}) \wedge (C_{Print} \wedge C_{Venue}) \wedge (\neg(Income - expenses < 4,0000) \wedge C_{Speak}) \wedge C_{Hold} \wedge C_{Paym})$

The details about the traverse algorithm can be found in [15].

A *Constraint Satisfaction Problem* (CSP) represents a reasoning framework that consists of of variables, domains, and constraints. Formally, it is defined as a tuple ⟨*X*, *DO*, *C*⟩, where $X = \{x_1, x_2, \ldots, x_n\}$ is a finite set of variables, $DO = \{do(x_1), do(x_2), \ldots, do(x_n)\}$ is a set of domains of the values of the variables, and $C = \{C_1, C_2, \ldots, C_m\}$ is a set of constraints. Each constraint C_i is defined as a relation R on a subset of variables $V = \{x_i, x_j, \ldots, x_l\}$, called the *constraint scope*. The relation R may be represented as a subset of the Cartesian product $do(x_i) \times do(x_j) \times \ldots \times do(x_l)$. A constraint $C_i = (V_i, R_i)$ simultaneously specifies the possible values of the variables in V in order to satisfy R. Let $V_k = \{x_{k_1}, x_{k_2}, \ldots, x_{k_l}\}$ be a subset of X. An l-tuple $(x_{k_1}, x_{k_2}, \ldots, x_{k_l})$ from $do(x_{k_1}), do(x_{k_2}), \ldots, do(x_{k_l})$ can therefore be called an *instantiation* of the variables in V_k. An instantiation is a solution if and only if it satisfies the constraints *C*.

In order to solve a CSP, a combination of search and consistency techniques is commonly used [9]. The consistency techniques remove inconsistent values from the domains of the variables during or before the search. During the search, a propagation process is executed which analyzes the combination of values of variables where the constraints are satisfiable. Several local consistency and optimization techniques have been proposed as ways of improving the efficiency of search algorithms.

In a CSP, the inclusion of a constraint in the set C has the same effect as including this constraint with an AND (∧) relation with the set C. For this reason, the CSP template is composed of:

- *X*: {Related variables defined in the *Dictionary* } ∪ {*Decision Variables*}
- *DO*: Estimated ranges for each variable in *X*, in the *Stages* in which the decisions are going to be taken
- *C*: {*Business Rules* defined for the whole *Business Process*} ∪ {BDC obtained by traversing the *Business Process*}

Not all *Process-Observational Variables* defined in *Dictionary* are included in the CSP template. Only those variables in the intersection between the *Process-Observational Variables* defined in *Dictionary* and the set of variables that have been used to define the *Business Rules*, are included in the CSP template. The reason is that, if *Process-Observational Variable* is defined in *Dictionary* but is not used in any *Business Rules*, then it implies that the variable exerts no affect, and therefore its inclusion can be omitted.

This CSP template represents the whole process since the template contains all relevant *Process-Observational Variables*, and the combined *Business Rules* are in accordance with the control flow. The only element in the preceding template that needs to be specified is the specific value of the *DO* of each *Process-Observational Variable* at the specific moment at which the decision must be made.

For the example, the CSP pattern built to analyze the possible valid values of a *Decision Variable* is presented in Figure 8. The orange parts can be defined with the business process analysis, although the green parts (instantiation of min and max ranges and definition of the input variable to be decided) will depend on the execution decision-making points.

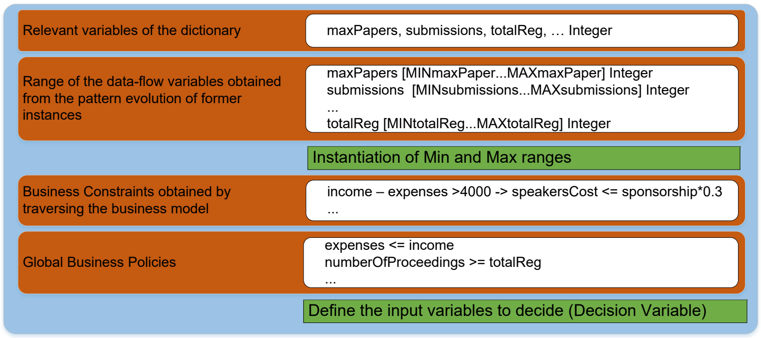

Relevant variables of the dictionary
maxPapers, submissions, totalReg, … Integer
Range of the data-flow variables obtained from the pattern evolution of former instances
maxPapers [MINmaxPaper...MAXmaxPaper] Integer
submissions [MINsubmissions...MAXsubmissions] Integer
...
totalReg [MINtotalReg...MAXtotalReg] Integer
Instantiation of Min and Max ranges
Business Constraints obtained by traversing the business model
income – expenses >4000 -> speakersCost <= sponsorship*0.3
...
Global Business Policies
expenses <= income
numberOfProceedings >= totalReg
...
Define the input variables to decide (Decision Variable)

Fig. 8. Pattern of CSP for input-data decision-dmaking.

Constraint Satisfaction Problems provide possible tuples of variables that satisfy the constraints. On the other hand, Constraint Optimization Problems provide the tuple of values that optimize a function. We can consider the utilization of an optimization in this context, however, it was dismissed for avoiding the reduction of the domain of the decision variables, necessary in uncertain scenarios. For the example, to minimize the outcomes, the person in charge of the decision can always select the lowest quantity of expenses and the highest of revenue. However, these decision could reduce the domain of the future decision provoking possible unsatisfiable business rules. This is why the proposal provides the possible range instead of a single value.

6. Decision-Making Support for Input Data in Decision Points

In this section, the whole DSS process is followed using a real set of *Process-Observational Variable* values applied to the example presented in Section 2. In the example, the number of copies of the Proceedings (`numberOfProceedings`) is an input variable whose value must be decided before the registration process is closed. However, our methodology can help in the decision-making regarding input data once the execution of an instance reaches a *Decision Point*, in this case, to decide the number of copies of proceedings to print in the activity *Print Proceeding*.

6.1. Ascertaining the current Stage

Table 5 shows the temporal variation of the most relevant *Process-Observational Variables* defined in the *Dictionary* of Section 4.2 for one simple former instance. This data has been processed with the aim of showing an understandable dataset, and, for this reason, every row has been summarized into weekly data, and several less relevant weeks have been removed. Several interesting aspects can be seen in this former instance, including aspects such as: The income for sponsorship remains unchanged from week 6; the submissions start in week 6, and their number remain unchanged after week 19; and the total number of registrations is 121 and this remains unchanged from week 36.

Table 5. Temporal variation of the main POVars in the Dictionary

week	sponsorship	submissions	totalReg	earlyReg	lateReg	acceptedPapers
0	0	0	0	0	0	0
1	100	0	0	0	0	0
2	10000	0	0	0	0	0
			...			
6	25000	1	0	0	0	0
7	25000	5	1	1	0	0
8	25000	5	4	4	0	0
9	25000	10	6	6	0	0
10	25000	25	8	8	0	0
			...			
19	25000	50	54	54	0	20
20	25000	50	66	66	0	20
21	25000	50	66	66	0	20
22	25000	50	67	66	1	20
23	25000	50	69	66	3	20
			...			
35	25000	50	114	66	48	20
36	25000	50	121	66	55	20
			...			
40	25000	50	121	66	55	20

6.2. Estimation of the ranges for each *Process-Observational Variable*

The estimation of the ranges will be performed based on the current *Stage* at the moment of the decision, and the knowledge extracted regarding *Process-Observational Variable* temporal variation patterns. These ranges involve the possible ranges of potential final values of each *Process-Observational Variable*, based on the variation of *Process-Observational Variables* extracted from former instances of the business processes, as explained in Subsection 5.

The *Stages* defined by business experts in Section 4.4 are applied with the experimentation data, and the data in each possible set of *Stages* that can be matched together are normalized to be comparable (shown in Section 5). The evolution patterns found are shown in Table 6. In order to create a comprehensible sample, only the most relevant *Process-Observational Variables* are shown.

Table 6. Knowledge extracted from Minimum and Maximum slopes for each POVar.

Stages POVar \	**sponsor.**	**submi.**	**totalReg**	**earlyReg**	**lateReg**	**accepted papers**
earlyReg	0.00 - 625.00	1.25 -1.25	1.55 - 3.00	1.00 - 1.62	0.55 - 1.37	0.50 - 0.75
earlyReg paperSub	0.00 - 405.40	0.00 - 1.32	1.67 - 2.78	1.08 - 1.30	0.59 - 1.49	0.54 - 0.81
earlyReg notified	0.00 - 0.00	0.00 - 0.00	1.68 - 2.68	0.48 - 0.80	0.88 - 2.20	0.00 - 0.00
lateReg notified	0.00 - 0.00	0.00 - 0.00	0.00 - 1.10	0.00 - 0.00	0.00 - 1.10	0.00 - 0.00
lateReg notified confer.	0.00 - 0,00	0.00 - 0.00	0.00 - 0.00	0.00 - 0.00	0.00 - 0.00	0.00 - 0.00

6.3. Instantation of the CSP template

When the possible ranges of the *Process-Observational Variables* have been obtained from the previous step, it is the moment to incorporate the obtained values into the CSP-template and to include the decision variable as the goal of the CSP. The way in which the template is instantiated is presented below.

Table 7. Predicted minimum and maximum values for the related *Process-Observational Variables* in stage "earlyReg and notified"

Variable	Min Slope	Max slope	Current value	Estimated min value	Estimated max value
earlyReg	0.48	0.80	24	35.52	43.20
lateReg	0.88	2.20	0	21.12	52.80
totalReg	1.68	2.68	24	64.32	88.32
acceptedPapers	0.00	0.00	19	19	19
maxPapers	0.00	0.00	20	20	20
sponsorship	0.00	0.00	20000	20000	20000
earlyRegFee	0.00	0.00	300	300	300
lateRegFee	0.00	0.00	400	400	400
publicity	0.00	0.00	20000	20000	20000
venueCost	0.00	160.00	0	0	3840
venueCapacity	0.00	3.20	0	0	76.80
numberOfProceeding	0.00	2.80	0	0	67.20
proceedingPrice	0.00	3.20	0	0	76.80
galaDinnerPrice	1.60	2.80	0	38.40	67.20
lunchPrice	0.60	1.00	0	14.40	24
speakersCost	200.00	400.00	0	4800	9600

Table 6 presents the knowledge regarding the temporal variation of *Process-Observational Variables* that has been extracted by analyzing the *Comparable Instances*. In Table 6, it can be observed how certain variables leave the value unchanged in certain stages, for instance, *earlyReg* has positive slopes when the set of stages defined by business experts is: *earlyReg* (Min: 1.00, Max: 1.65), *earlyReg and paperSubmission* (Min: 1.08, Max: 1.30), *earlyReg and notified* (Min: 0.48, Max: 0.80). This means that, in this situation, the variable has increased minimal and maximal values. However, in the stages *lateReg and notified*, and *lateReg and notified and conference*, the slopes are Min: 0.00, Max: 0.00, which means that, in these stages, the variable *earlyReg* remains unmodified.

Thanks to this information, once a decision regarding data has to be made, it is possible to estimate the final range of values of each variable, and a *Constraint Satisfaction Problem* can be built in order to verify that all constraints in the process are satisfied, and to inform the decision maker of the range of values of the *Decision Variable*.

In the aforementioned conference example, one instance of a decision is given in the establishment of the value of the variable *numberOfProceedings*, which is decided upon in the task *Print Proceedings*. For this example, the decision has been made at moment $t = 16$ weeks, and with the current values for the process following the current *Stages* of the *Process-Observational Variables* defined in the *Dictionary*: {submissions: 53, earlyReg: 24, lateReg: 0, totalReg: 24, acceptedPapers: 19, notified: 1, holdConference: 0, maxPapers: 20, submissionClose: 11, regOpen: 4, earlyRegClose: 23, regClose:

38, sponsorship: 20000, earlyRegFee: 300, publicity: 20000, lateRegFee: 400, venueCost: 0, numberOfProceedings: 0, proceedingPrice: 0, galaDinnerPrice: 0, lunchPrice: 0, speakersCost: 0}.

Therefore, by mapping this information with *Dictionary*, it is possible to observe that the stages activated are *earlyReg* and *notified*. As can be seen in Table 7, by using the previous knowledge extracted from former instances, it is possible to know how these variables will probably evolve. For instance, current *totalReg* is 24, and since the minimum slope and maximum slope calculated for these stages are: Min 1.68 and Max 2.68. Thereby, the final values will probably lie between:

- $EstimatedMinValue = (40t - 16t) \times 1.68\frac{totalReg}{t} + 24totalReg = 64.32$ $totalReg$

- $EstimatedMaxValue = (40t - 16t) \times 2.68\frac{totalReg}{t} + 24totalReg = 88.32$ $totalReg$

Other *Process-Observational Variables*, such as *sponsorship*, have a Min and Max slope of 0.00 in this set of stages and hence we consider that this value is fixed, in this case, to 20,000.

With this information, the CSP-template is instantiated according to the envelopers and the current stage, as shown in Figure 9. The instantiation of the Min and Max Ranges of the variables that evolve, and the input variable under decision is included (green part).

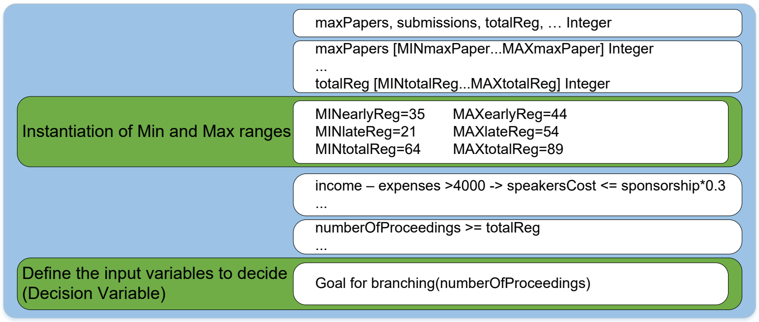

maxPapers, submissions, totalReg, ... Integer
maxPapers [MINmaxPaper...MAXmaxPaper] Integer
...
totalReg [MINtotalReg...MAXtotalReg] Integer
Instantiation of Min and Max ranges
MINearlyReg=35 MAXearlyReg=44
MINlateReg=21 MAXlateReg=54
MINtotalReg=64 MAXtotalReg=89
income – expenses >4000 -> speakersCost <= sponsorship*0.3
...
numberOfProceedings >= totalReg
...
Define the input variables to decide (Decision Variable)
Goal for branching(numberOfProceedings)

Fig. 9. Pattern of CSP for input at decision time.

The domain of the *Process-Observational Variables* depends on the stage of the variables when the decision is made, and how the value of the variables can evolve according to the envelopment temporal variation obtained in the previous subsection. The stage of the variables is known at decision time, and therefore the specific domain of the variables cannot be included in the CSP template, for the example *MINmaxPaper*, *MAXmaxPaper*, *MINsubmission*, *MAXsubmission*, *MINtotalReg* and *MAXtotalReg*.

Since the CSP solver returns all the possible values of the variables, it is necessary to limit it further to present only the values of the *Decision Variables*. To this end, the decision variables are defined as objectives during the propagation process where the variables are instantiated. This limitation enables the search to stop the propagation in those branches where no new values of decision variables can be found, thereby halting the unnecessary combinations of values. For each solution found, each value of the *Decision Variables* is stored in a sorted list. Each of these sorted lists is conditioned to return the list of intervals for each variable of decision. For example, if the values $\{1, 2, 3, 5, 8, 9, 10\}$ are found for the variable x, then the list of intervals built is $\{[1, 3], [5, 5], [8, 10]\}$.

6.4. Solution of the *Constraint Satisfaction Problem*

In order to compute the range of the *Decision Variable* to guarantee a successful execution, a Constraint Programming solver must solve the CSP, and obtain the possible range of the decision variable, *numberOfProceedings* for the example of the CSP above. Numerous commercial solvers are available. In this case, we have selected ChocoSolver [36].

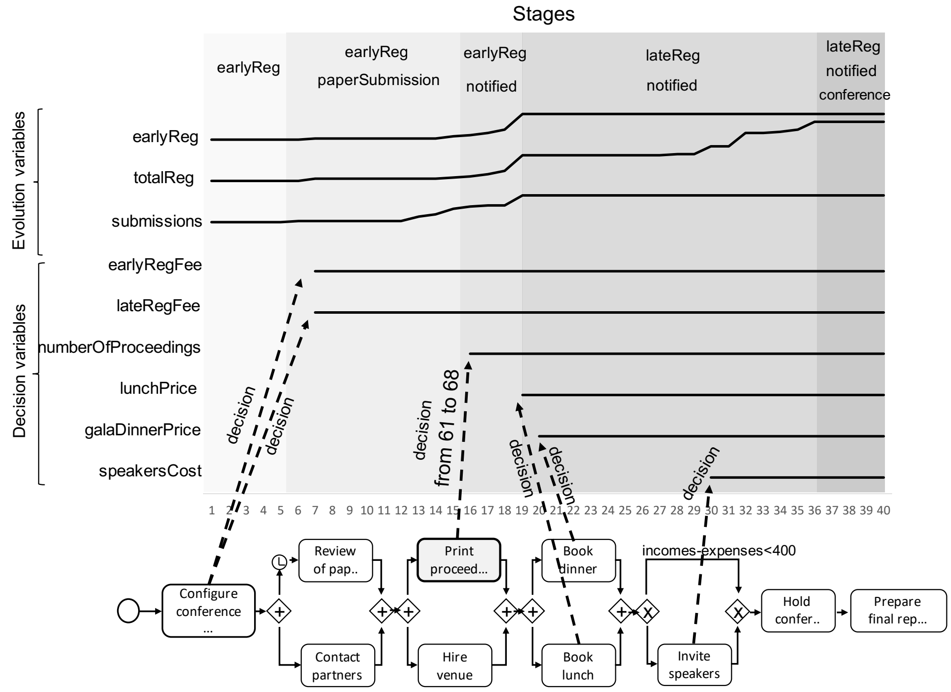


Fig. 10. Decisions and support system.

Figure 10 describes a possible process instance where the activities are executed and the decision are made, in accordance with the evolution of the evolution of the variables. For example, several decisions are made before the task *print proceedings* is executed, where the `numberOfProceedings` to print out has to be decided. Moreover, there is a set of evolution variables that are changing during the process instance. Following the resolution of the CSP, the values of the decision variables, that make possible the successful execution of the instance, are found, and the recommendation to the decision-maker is made. In the case of the `numberOfProceedings` to print out, the system determines that the value should lie between 61 and 68. With this information, the experts makes the last decision, and once it is made, as can be seen in Figure 10, the value of this variable is set. Of course, this value can also affect to future decisions.

As can be seen in Figure 10, for each decision, the system offers a range of values with which the process can be completed correctly, this is, in compliance with all the business rules. To ascertain better how the variable can evolve, the corresponding patter of evolution with the stage is used in each decision (represented with different color tones).

7. Related Work

In business processes, decision-making support contributes towards helping the business process designer choose the best combination of activities, to achieve a given objective. In the literature, **simulation-based** approaches have been proposed, such as [40] for complex dynamic systems and for the inclusion of uncertain data, or methods to optimize processes with fuzzy descriptions [41]. We observe that these types of methods ignore how the process works at runtime and fail to consider the importance of the variables of the data-flow. They are oriented towards the design of the model or the redesign of the business process [23] by analyzing the quality of the process at design time. **Data** has also been involved in other studies related to decision support; for example, [25] proposes operational decision support for the construction of process models based on historical data to simulate processes. That proposal includes a general approach to a business process for operational decision support and includes business process modeling and workflow simulation with the models generated, by using process mining. There are studies related to how to model the processes, such as that in [2], which proposes a framework of **assistance in the creation of models** by taking the necessary resources involved in the process into account. In that paper, the data that describes the resources of the execution of the process is used, but not the data that flows at run-time, nor is it considered how this assistance can function at run-time. Previous studious have faced the problem of optimal execution of decision models [4], where authors minimize and prioritize the acquisition of decision-related data by classifying decision inputs into decision trees, according to the degree of their influence on the outputs. The literature also contains methods for the discovering of the business process from logs, which can be used as a starting point for the business process design.

In general, we found that the literature is largely centered on the activity selection [3], or on the optimization of the process design [35], but not on the **assistance of the user for the input data**. Although work such as [21] is oriented towards auditing the process in order to detect gaps between the information system process flow and the internal control flow in the business process, the quality of the data values at run-time is not a cause of concern for the authors. Errors in the quality of data can be derived from the existence of an oversight in the description of the semantics of data in the business processes. The use of compliance rules has traditionally been used for the validation of the business process, and not for user assistance. The validation of business process traces has provided a field of intense research in recent years using business compliance rules; see [8] as an entry point into this literature. However, these types of proposals cannot be used in the decision-making support for input data since they are focused on compliance with the process model structure [38], [26].

Regarding how to **model data-aware compliance rules**, studies such as [42], [27], [20], and [1], have defined graphical notations to represent the relationship between data and compliance rules by means of data conditions. These types of compliance rules cannot, therefore, be employed to infer the possible values of the variables that are involved in the decisions. In [28], "semantic constraints" and the SeaFlows framework are proposed in order to enable integrated compliance support. Furthermore, in [22], a preprocessing step that allows the efficient verification of data-aware compliance is presented, whereby the data describes under what conditions the activities can be executed. In general, many examples can be found where data objects are used for compliance verification, for instance, the semantically annotating activities with preconditions and effects that may refer to data objects are introduced in [17], and the detection of tuples from different data-sources that refer to the same real-work entity can be found in [39], but none assist the user with this knowledge at run-time.

Summarizing, and to the best of our knowledge, only the preliminary studies [14] and [15] make use of the knowledge of the *Business Process Model* and the *Business Rules* for decision-making support for input data, while all other proposals are focused on the design or redesign of the model (*Business Process Model*). The current paper constitutes an improvement on these two previous papers by including not only *Expert Knowledge* but also the analysis of previous executions, in order to automatically extract knowledge that enables the evolution of the variables involved to be discovered, thereby offering better recommendations regarding input data to the decision maker.

8. Conclusions and Future work

Several decisions must be made at the operational level of an organization. Moreover, the are numerous situations that can affect the evolution of the organization. For this reason, when a decision is made, it is necessary to analyze the process model, the stage of the instance process, and to consider how other instances evolved in the past. All these elements are combined in the proposed DSS to help in the data input during process execution. This system provides a guide as to the correct management as defined in the business plans, by taking advantage of the information regarding former instances and business process knowledge. Thanks to this analysis, the information is set up to help in the decisions concerning input data in current instances, which for exceeds the simple use of this information for *Stages* reports or post-mortem analysis.

In order to develop the DSS, it is necessary to: model the problem, and include the business rules that define the goals of the organizations; create pattern of the temporal variation of the variables according to former instances; and create a CSP model that provides the correct domain for the input data that facilitates the correct finalization of the process.

Regarding future work, we consider three areas where our work could be more helpful: (1) through the improvement of the mechanism to ascertain the behavior of the business and its variables by applying data mining techniques; (2) through enriching the model with further components, such as the inclusion of external constraints related to services; and (3) through assistance in not only satisfying the strategic plan, but also in the optimization.

Acknowledgments. This work has been partially funded by the Ministry of Science and Technology of Spain ECLIPSE project (RTI2018-094283-B-C33) and the European Regional Development Fund (ERDF/FEDER). Thanks to Lesley Burridge for the revision of the English version of the manuscript.

References

1. Awad, A., Weidlich, M., Weske, M.: Visually specifying compliance rules and explaining their violations for business processes. J. Vis. Lang. Comput. 22(1), 30–55 (2011)
2. Barba, I., Weber, B., Valle, C.D.: Supporting the optimized execution of business processes through recommendations. In: Business Process Management Workshops - BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I. pp. 135–140 (2011)
3. Batoulis, K., Baumgraß, A., Herzberg, N., Weske, M.: Enabling Dynamic Decision Making in Business Processes with DMN, pp. 418–431. Springer International Publishing, Cham (2016)
4. Bazhenova, E., Weske, M.: Optimal acquisition of input data for decision taking in business processes. In: Proceedings of the Symposium on Applied Computing. pp. 703–710. SAC '17, ACM, New York, NY, USA (2017)
5. Becker, J., Ahrendt, C., Coners, A., Weiß, B., Winkelmann, A.: Modeling and analysis of business process compliance. In: IFIP International Working Conference on Governance and Sustainability in Information Systems-Managing the Transfer and Diffusion of IT. pp. 259–269. Springer (2011)
6. Borrego, D., Gómez-López, M.T., Gasca, R.M.: Minimizing test-point allocation to improve diagnosability in business process models. Journal of Systems and Software 86(11), 2725–2741 (2013)
7. Cetin, S., Altintas, N.I., Solmaz, R.: Business Rules Segregation for Dynamic Process Management with an Aspect-Oriented Framework, pp. 193–204. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
8. Chesani, F., Mello, P., Montali, M., Riguzzi, F., Sebastianis, M., Storari, S.: Checking compliance of execution traces to business rules. In: Business Process Management Workshops. pp. 134–145 (2008)
9. Dechter, R.: Constraint Processing (The Morgan Kaufmann Series in Artificial Intelligence). Morgan Kaufmann (May 2003)
10. Eck, N.J.v., Waltman, L.: How to normalize cooccurrence data? an analysis of some well-known similarity measures. Journal of the American Society for Information Science and Technology 60(8), 1635–1651 (2009)
11. Gómez-López, M.T., Ceballos, R., Gasca, R.M., Valle, C.D.: Developing a labelled object-relational constraint database architecture for the projection operator. Data Knowl. Eng. 68(1), 146–172 (2009)
12. Gómez-López, M.T., Gasca, R.M.: Run-time monitoring and auditing for business processes data using contraints. In: International Workshop on Business Process Intelligence. pp. 15–25. BPI 2010, Springer (2010)
13. Gómez-López, M.T., Gasca, R.M.: Using constraint programming in selection operators for constraint databases. Expert Syst. Appl. 41(15), 6773–6785 (2014)
14. Gómez-López, M.T., Gasca, R.M., Parody, L., Borrego, D.: Constraint-driven approach to support input data decision-making in business process management systems pp. 457–469 (2011)

15. Gómez-López, M.T., Gasca, R.M., Pérez-Álvarez, J.M.: Decision-making support for the correctness of input data at runtime in business processes. Int. J. Coop. Info. Syst. 23(2), 29 (2014)
16. Gómez-López, M.T., Gasca, R.M., Pérez-Álvarez, J.M.: Compliance validation and diagnosis of business data constraints in business processes at runtime. Inf. Syst. 48, 26–43 (2015)
17. Governatori, G., Hoffmann, J., Sadiq, S.W., Weber, I.: Detecting regulatory compliance for business process models through semantic annotations. In: Business Process Management Workshops, BPM 2008 International Workshops, Milano, Italy, September 1-4, 2008. Revised Papers. pp. 5–17 (2008)
18. Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business (1993)
19. Hay, D., Healy, K.A., Hall, J., et al.: Defining business rules-what are they really. The Business Rules Group 400 (2000)
20. Hoffmann, J., Weber, I., Governatori, G.: On compliance checking for clausal constraints in annotated process models. Information Systems Frontiers 14(2), 155–177 (2012)
21. Huang, S., Yen, D.C., Hung, Y., Zhou, Y., Hua, J.: A business process gap detecting mechanism between information system process flow and internal control flow. Decision Support Systems 47(4), 436–454 (2009)
22. Knuplesch, D., Ly, L.T., Rinderle-Ma, S., Pfeifer, H., Dadam, P.: On enabling data-aware compliance checking of business process models. In: Conceptual Modeling - ER 2010, 29th International Conference on Conceptual Modeling, Vancouver, BC, Canada, November 1-4, 2010. Proceedings. pp. 332–346 (2010)
23. Kock, N., Verville, J., Danesh-Pajou, A., DeLuca, D.: Communication flow orientation in business process modeling and its effect on redesign success: Results from a field study. Decision Support Systems 46(2), 562–575 (2009)
24. Lancaster, P., Salkauskas, K.: Curve and surface fitting. an introduction. London: Academic Press, 1986 1 (1986)
25. Liu, Y., Zhang, H., Li, C., Jiao, R.J.: Workflow simulation for operational decision support using event graph through process mining. Decision Support Systems 52(3), 685–697 (2012)
26. Ly, L.T., Rinderle, S., Dadam, P.: Integration and verification of semantic constraints in adaptive process management systems. Data Knowl. Eng. 64(1), 3–23 (2008)
27. Ly, L.T., Rinderle-Ma, S., Dadam, P.: Design and verification of instantiable compliance rule graphs in process-aware information systems. In: CAiSE. pp. 9–23 (2010)
28. Ly, L.T., Rinderle-Ma, S., Göser, K., Dadam, P.: On enabling integrated process compliance with semantic constraints in process management systems - requirements, challenges, solutions. Information Systems Frontiers 14(2), 195–219 (2012)
29. Montgomery, D.C., Peck, E.A., Vining, G.G.: Introduction to linear regression analysis. John Wiley & Sons (2015)
30. Motulsky, H.J., Ransnas, L.A.: Fitting curves to data using nonlinear regression: a practical and nonmathematical review. The FASEB journal 1(5), 365–374 (1987)
31. OMG: Object Management Group, Business Process Model and Notation (BPMN) Version 2.0. OMG Standard (2011)
32. Pérez-Álvarez, J.M., Gómez-López, M.T., Eshuis, R., Montali, M., Gasca, R.M.: Verifying the manipulation of data objects according to business process and data models. Knowledge and Information Systems (Jan 2020)
33. Pérez-Álvarez, J.M., López, M.T.G., Parody, L., Gasca, R.M.: Process instance query language to include process performance indicators in DMN. In: 20th IEEE International Enterprise Distributed Object Computing Workshop, EDOC Workshops 2016, Vienna, Austria, September 5-9, 2016. pp. 1–8. IEEE Computer Society (2016)
34. Pérez-Álvarez, J.M., Maté, A., Gómez-López, M.T., Trujillo, J.: Tactical business-process-decision support based on kpis monitoring and validation. Computers in Industry 102, 23–39 (2018)

35. Pivk, A., Vasilecas, O., Kalibatiene, D., Rupnik, R.: Ontology and SOA based data mining to business process optimization. In: Information System Development - Improving Enterprise Communication, [Proceedings of the 22nd International Conference on Information Systems Development, ISD 2013, Seville, Spain]. pp. 255–268 (2013)
36. Prud'homme, C., Fages, J.G., Lorca, X.: Choco Documentation. TASC - LS2N CNRS UMR 6241, COSLING S.A.S. (2017), http://www.choco-solver.org
37. Revesz, P.Z.: Constraint databases and temporal reasoning. In: Eigth International Symposium on Temporal Representation and Reasoning, TIME-01, Civdale del Friuli, Italy, June 14-16, 2001 (2001)
38. Sadiq, S.W., Orlowska, M.E., Sadiq, W.: Specification and validation of process constraints for flexible workflows. Inf. Syst. 30(5), 349–378 (2005)
39. Vieira, P.K.M., Lóscio, B.F., Salgado, A.C.: Incremental entity resolution process over query results for data integration systems. Journal of Intelligent Information Systems 52(2), 451–471 (Apr 2019)
40. Völkner, P., Werners, B.: A decision support system for business process planning. European Journal of Operational Research 125(3), 633–647 (2000)
41. Völkner, P., Werners, B.: A simulation-based decision support system for business process planning. Fuzzy Sets and Systems 125(3), 275–287 (2002)
42. Weber, I., Hoffmann, J., Mendling, J.: Semantic business process validation. In: 3rd international workshop on Semantic Business Process Management (2008)
43. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)

José Miguel Pérez-Álvarez received the degree in computer engineeringminoring in systems engineering from the Universidad de Sevilla, Spain, in2010, the M.Sc. degree in software engineering and technology in 2011, andthe Ph.D. degree from the Universidad de Sevilla in 2018. He also received aMaster of Business Administration (MBA) from EOI Business School in 2013. Heis Research Scientist with the Systemic AI group of Naver Labs Europe.Currently, he is also a collaborator of the IDEA Research Group. He hasparticipated in several private and public research projects. He has publishedseveral high impact papers.

Luisa Parody is PhD. in Computer Science and Software Engineering and is Associate Professor at the Universidad Loyola Andalucía since 2018. Her research areas of interest include data awareness in Business Processes, decisions for Business Processes Requirements based on Big Data and Data Quality. She belongs to the IDEA Research Group and she has participated in several private and public research projects. She has published some high-impact papers and she is reviewer in various conferences and international journals.

María Teresa Gómez is Phd. in Computer Science, Lecturer at the University ofSeville and the head of the IDEA Research Group. Her research areas includeBusiness Processes and Data management in Big Data environment. She has ledseveral private and public research projects and has published more than twenty impactpapers(DSS, IS, DKE, IST...). She was nominated as a member of several Program Committees(BPM, ER, EDOC, CAISE Doctoral Consortium...), and she hasbeen reviewing for international journals. She has been invited speaker atvarious conferences and summers schools.

Rafael M. Gasca holds a PhD in computer science from the Universidad deSevilla, in Spain. He is full professor since 2018. He has led the Quivir ResearchGroup since 2000, since 2015, he has been a member of the IDEA ResearchGroup at the Universidad de Sevilla. He has been the leader of different publicand private research projects and has directed twelve PhD theses. He haspublished tens dozens of papers in high-impact factor, including IEEEComputing, IEEE Communications Magazine, Information and SoftwareTechnology, Journal System and Software, Information Systems, Informationand Software Technology, and Data and Knowledge Engineering. He has been areviewer in relevant security conferences and journals and an organizer ofartificial intelligence conferences and an international summer school on faultdiagnosis of complex systems.

Paolo Ceravolo is an Associate Professor at the Università degli Studi di Milano. His research interests include Data Representation and Integration, Business Process Monitoring, Empirical Software Engineering. On these topics, he has published several scientific papers. He is involved in the organisation of different conferences such as the IEEE Services conference, the International Conference on Process Mining, the Conference on Information and Knowledge Management. Since 2014 he is chair and subsequently vice-chair of the IFIP 2.6 Working Group on Database Semantics. As a data scientist, he was involved in several international research projects and innovative startups.

Received: May 22, 2020; Accepted: November 25, 2020.

Computer Science and Information Systems 18(3):867–891 https://doi.org/10.2298/CSIS200206049G

Intrusion Prevention with Attack Traceback and Software-defined Control Plane for Campus Networks

Guangfeng Guo[1,2], Junxing Zhang[1,⋆], and Zhanfei Ma[2]

[1] College of Computer Science, Inner Mongolia University
010021 Hohhot, China
guoguangfeng@163.com, junxing@imu.edu.cn (#Corresponding author)
[2] Baotou Teachers' College, Inner Mongolia University of Science & Technology
014030 Baotou, China
mazhanfei@163.com

Abstract. As traditional networks, the software-defined campus network also suffers from intrusion attacks. Current solutions for intrusion prevention cannot meet the requirements of the campus network. Existing methods of attack traceback are either limited to specific protocols or incur high overhead. To protect the data center (DC) of the campus network from internal and external attacks, we propose an Intrusion Prevention System (IPS) based on the coordinated control between the detection engine, the attack traceback agent, and the software-defined control plane. Our solution includes a novel algorithm to infer the best switch port for defending different attacks of varied scales based on the inverse HSA (Header Space Analysis) and the global view of the software-defined controller. The proposed scheme can effectively and timely block the malicious traffic not only protecting victim hosts from attacks but also preventing the whole network from suffering unwanted transmission burden. The proposed IPS is deployed on the bypass of the DC switch and collects network traffic by port mirroring. Compared with the traditional serial deployment, the new design helps defend the DC internal attacks, reduce the probability of network congestion, and avoid the single point of failure. The experimental results show that the overhead of our IPS is very low, which enables it to meet the real-time requirements. The average defense time is between 10 and 14 ms for the data center internal attacks of different scales. For external attacks, the maximum defense time is about 76 ms for a large-scale network with 100 switches.

Keywords: IPS, Intrusion Prevention System, SDN, Software-defined Network, Attack Traceback, Inverse Forwarding Function, HSA, Header Space Analysis, Campus Networks, DC, Data Center.

1. Introduction

The overall situation of network security is not optimistic in recent years. Intrusions from the Internet have brought serious consequences, which pose a great threat to information security of networked systems. In the first quarter of the year, DDoS attacks rose more than 278 percent compared to Q1 2019 and more than 542 percent compared to the last quarter, according to Nexusguards Q1 2020 Threat Report [16]. In addition, attacks from the internal hosts infected by computer viruses have exhibited great destructiveness in

[⋆] Corresponding author

several security incidents. Worms play an important role in internal attacks. The worm installs itself in the memory of the computer but it has the capability to transfer itself to other hosts automatically even without human intervention, thus making it much more serious than a virus. In recent two years, some enterprises have been attacked by the worm GandCrab, which encrypts victims' files and demands ransom payment in order to regain access to their data. Since launching in January 2018, GandCrabs authors claimed to have brought in over $2 billion in illicit ransom payments [15].

Software-defined networking (SDN) [26] has its roots anchored deeply in education and drives the evolution of the campus network. It demonstrates that the campus network can do more than serving universities, and they are also capable of helping a diverse set of users with varying needs. Therefore, the software-defined campus network is becoming increasingly important. The data center (DC) of the campus network holds the most critical data assets of a university, college, or institute, so it is the key protection unit of the network and also the focus of this paper. Unfortunately, the security of software-defined campus networks is worrying. As traditional networks, software-defined networks also suffer from attacks such as DoS (Denial of Service), U2R (User to Root), R2L (Remote to Local), Probe, etc. Given that SDN is famous for its contributions to the network architecture, we consider taking advantage of its dynamic flow control, network-wide visibility, and network programmability to improve its security, especially its intrusion prevention capability.

The Intrusion Prevention System (IPS) has been widely adopted to enhance network security. Traditionally, it is deployed between the core switch of the campus network and the DC switch in series. Once external malicious traffic attacks DC hosts through the IPS, it can effectively protect them from these attacks. However, before entering the IPS, a large amount of malicious traffic is often forwarded by other switches inside the campus network, which brings additional burden to the network. Moreover, if an internal malicious host attacks other DC hosts, the IPS can no longer protect them because the internal malicious traffic is forwarded by the DC switch without through the IPS. Therefore, we want to build an innovative intrusion prevention system that protects DC hosts from both internal and external attacks. The proposed system is deployed on the bypass of the DC switch and it collects network traffic with port mirroring. The new design prevents the IPS-incurred single point of failure from happening, avoids the network congestion caused by the serial deployment, and defends the DC internal attacks. Further, the new system makes use of the inverse forwarding functions derived by expanding the Header Space Analysis (HSA) [20] framework to accurately trace attack traffic back and find the best switch port for blocking it. Finally, the system harnesses the synergy of the intrusion detection engine, the attack traceback agent, and the software-defined control plane to block intrusion attacks from the source in real time using the OpenFlow protocol and prevents the malicious traffic from soaring at the beginning of attacks.

The contributions of this paper are as follows:

- To protect the data center of the campus network from internal and external attacks, we propose an intrusion prevention system based on the coordinated control between the intrusion detection engine, the attack traceback agent, and the software-defined control plane.

– According to the inverse HSA, we design a novel real-time protocol-independent algorithm to infer the best switch port for preventing different intrusion attacks of varied scales using the forwarding model of the entire network.
– We implement a prototype of the proposed IPS and evaluate its performance in the software-defined campus networks of various scales under the intrusion attacks such as NULL scanning and FIN scanning.

The rest of the paper is organized as follows. Section 2 summarises related work to the IPS in SDN and Attack Traceback. In Section 3 we describe our design principles and system architecture. Section 4 presents our algorithm of finding the best defense switch port. Section 5 presents the implementation details of the proposed IPS. Section 6 details our performance evaluation experiments. In Section 7 we discuss the advantages and drawbacks of various attack traceback methods. Finally, conclusions are drawn in Section 8.

2. Related Work

2.1. IPS in SDN

There is some existing work that leverages SDN for intrusion prevention. Based on the deployment modes of security components, the existing work can be classified into two categories: (1) security applications built on the SDN controller [35,8], (2) security devices that work in cooperation with the SDN control plane [34,9]. Changhoon Yoon et al. [35] implement four types of security functions with SDN in Floodlight applications and evaluate their Floodlight [1] application in real testbeds. For the NIPS (Network Intrusion Prevention System) application, the payload delivery from the data plane to the control plane would incur substantial overhead. Pin-Jui Chen and Yen-Wen Chen [8] propose a defense mechanism, which can find attack packets previously identified through the Sniffer function, and once the abnormal flow is found, the protection mechanism of the Firewall function will be activated. But its evaluation method is simple, and it is difficult to meet the security requirements of campus networks; its real-time performance is not evaluated; the problem of tracing the source of the attacker is not resolved. Xing et al. [34] presented an implementation of Snort IPS for protecting the cloud platform using Snort IDS [10] while sending the blocking action to the SDN controller. The authors get the benefit of snort as open source IDS and adjusted it for integration. But tracing the source of the attacker isn't considered. Yaping Chi et al. [9] propose a scheme for the cloud platform intrusion prevention, and the result shows that the efficiency of the intrusion detection in the new scheme can be improved by two times compared with the traditional intrusion prevention scheme. The solution applies to the cloud environment only; it is difficult to adopt this solution to meet the diverse security needs of the campus network, however.

2.2. Attack Traceback

Attack traceback is not a goal, but a means to defend against great harmful attacks (such as Dos). Identifying the origins of attack packets is the first step in making attackers accountable. Besides, after figuring out the network path which the attack traffic follows, the victim under the attack can apply defense measures such as packet filtering further from the victim and closer to the source.

Some scholars have started research on the attack traceback. IP traceback is a technique for tracing the paths of IP datagrams back toward their origins and also serves as the main technique for attack traceback. its methods can be divided into 5 categories:

1) Link Testing

Link testing [24] is an approach which determines the upstream of attacking traffic hop-by-hop while the attack is in progress. It is compatible with the existing protocols and the network infrastructure, such as routers. However, it is only suitable for tracking the attacks that last for long times.

2) ICMP Trace

This scheme is for each router to come up with an ICMP traceback message [3] or reach directed to the identical destination. The trace message itself consists of consequent and previous hop data and a time stamp. It utilizing the explicitly generated ICMP Traceback message were proposed in [17,31]. It incurs higher overhead in computation, storage and bandwidth.

3) Logging

This solution involves storing packet digests or signatures at intermediate routers and using data-mining techniques to see the trail that the packets traversed [32]. The drawbacks of this technique include significant amount of resources have to be reserved at intermediate routers and hence large overhead on the network, complexity, centralized management.

4) Overlay Networks

CenterTrack [33] is an overlay network, consisting of IP tunnels or other connections, that is used to selectively reroute interesting datagrams directly from edge routers to special tracking routers. The tracking routers, or associated sniffers, can easily determine the ingress edge router by observing from which tunnel the datagrams arrive. The datagrams can be examined, then dropped or forwarded to the appropriate egress point. It need to add special tracking routers, and is easy to find by attackers due to rerouting interesting datagrams.

5) Packet Marking

Packet-marking methods [7,30,4] are characterized by inserting traceback data into the IP packet to be traced, thus marking the packet on its way through the various routers on the network to the destination host. The method is subdivided into Probabilistic Packet Marking (PPM) [29] and Deterministic Packet Marking (DPM) [2]. Each router marks the packet with some probability in the PPM scheme, while every packet passing through the first ingress edge router is only marked with the IP address of the router in the DPM scheme. It requires modifications to the protocol, and cannot handle fragmentation and does not work with IPv6 and is not compatible with IPSec.

To be brief, each type has its advantages and drawbacks. The first type is only suitable for tracking the attacks that last for long times; the second and third type usually incur extra network burdens; the other types of methods require particular routers in the data path.

In this paper, we propose an intrusion prevention system that leverages the coordinated control between the detection engine, the attack traceback agent, and the software-defined control plane. In terms of the SDN design, our solution belongs to the second type of the system layouts described in Section 2.1. We assume that all the forwarding devices in the campus network are stateless and SDN-enabled, and they are controlled by the SDN Controller; the initial network topology is known in advance, and the attacks can be discovered by the detection engine of the IPS.

Current methods of attack traceback in Section 2.2 are either limited by specific protocols or incur high overhead. In this paper, we construct the inverse forwarding functions by expanding HSA framework, and design a novel real-time and protocol-independent algorithm of finding the best switch port for defense. In the later section, we will demonstrate the proposed solution can accurately and timely prevent attacks of varied types and scales.

3. Design Principles and System Architecture

The data center of the campus networks holds the most critical data assets, so it requires an efficient IPS solution for protecting its servers and other hosts from internal and external attacks. In the section, we discuss the design principles of a common intrusion prevention system for software-defined campus networks, propose an innovative IPS architecture to meet these principles, and introduce the intrusion prevention process of our proposal IPS.

3.1. Threat Model

The software-defined campus network not only suffers from the traditional attack (such as Dos, U2R, R2L, and Probe) but also sustains attack for the SDN control plane. Defending the latter attack is our future work, and is out of scope in this paper.

We assume the threats against the data center (DC) server of the campus, except attack for the SDN control plane. The victim may be a DC internal host, an external host or even an Internet host.

3.2. Design Principles

The design principles of a common intrusion prevention system for software-defined campus networks are based on the following key properties:

- **Needing an efficient solution for mainly focusing on protection data center servers of campus networks**: The data centers of campus networks hold the most critical data assets. So the IPS mainly protection data center of campus networks avoiding interior and external attacks in the campus network, and avoids single-point failure and reduces the probability of network congestion.
- **Detecting different types of attack**: the victim host of the data center can suffer from data center internal hosts, other campus network hosts(such as hosts of the office network) and Internet hosts.
- **Finding the best defense switch port for different types of attack**: The IPS can pinpoint the attack source and find the best defense switch port for different attacks and directly block the malicious traffic from injecting switches, which can eliminate dispensable transmission burdens which are raised by malicious attacks.

- **Compatibility with any OF-enabled device and protocol-independent**: The algorithm of finding the best defense switch port can be valid with any OF-enabled device (such as a Layer 2 switch or a router, etc.), regardless of which protocol it belongs to.
- **Real-time**: The IPS can intercept the malicious traffic at the beginning of the attack avoiding the malicious traffic soaring late.

3.3. Overall Architecture

The campus network provides network access services for students and faculties, divided into multiple types of subnets (such as the data center LAN and the office LAN). A minimum software-defined campus network consists of an SDN controller, three switches, several servers and dozens of workstations, as shown in Fig. 1. The border switch links the campus network to the Internet, and the other two switches connect servers and workstations. The DC (data center) switch and application servers comprise the data center LAN of the campus. Similarly, the ON (office network) switch and users' workstations form an office LAN for the campus.

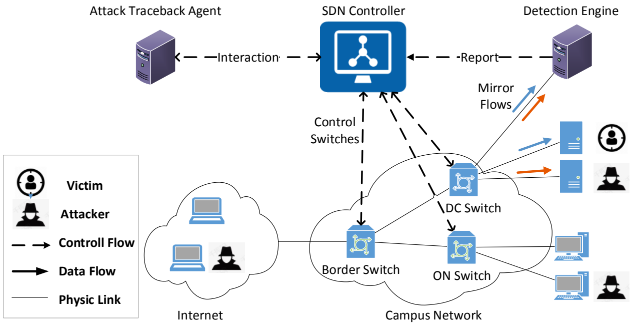


Fig. 1. Intrusion Prevention Architecture on an SDN-based Campus Network

The intrusion prevention architecture for the software-defined campus network is composed of an intrusion detection engine, an attack traceback agent, and the software-defined control plane, which includes all the switches and the controller mentioned above. It realizes the coordinated control between them to detect intrusions as early as possible and prevent attacks as soon as possible. The DC switch regularly mirrors the ingress traffic of the DC servers and sends it to the detection engine. The engine captures and analyses the traffic, and sends alarm messages to the controller if intrusion attempts are detected. In our design, the detection engine is deployed on the bypass of the DC switch. The engine has two NICs, with one interface connected the DC switch and the other joined up to the same LAN with the SDN controller and the trackback agent. The SDN controller transmits network states, such as flow table entries of all switches and topology changes of the total network, to the traceback agent also on a regular basis. According to the received network states, the traceback agent establishes a forwarding model of the entire network and uses it as the global view to infer the best switch port for defending intrusion attacks.

3.4. Intrusion Prevention Process

We have divided the process of detecting intrusion attempts and preventing attacks into four steps, as shown in Fig. 2.

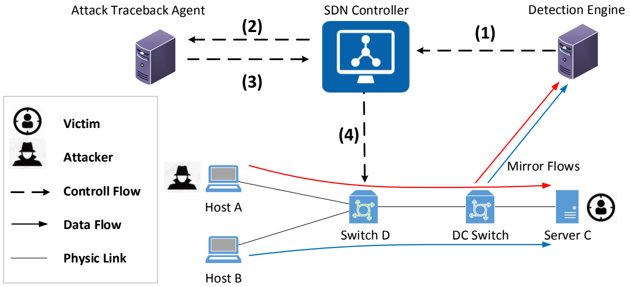


Fig. 2. Attack Detection and Defense Process

1) The detection engine discovers intrusion attempts and sends alarm messages to the controller. The detection engine constantly monitors the mirrored traffic. When it detects an intrusion attempt, it immediately sends an alarm message, which contains the header field of malicious packets, to the controller via the TCP connection.

2) The controller sends a request to the traceback agent asking for the best switch port for defense. When the controller receives the alarm message, it needs to find the best switch port to prevent the intrusion and take the appropriate measures. However, it is hard for the controller to pinpoint the attack source and then determine the best defense port based on the source and the overall situation of the network. The attack source might be a host in the data center network, office network, or Internet. Therefore, the controller queries the traceback agent for the best switch port for defense via the TCP connection.

3) The traceback agent determines the best switch port for defense and responds to the controller. The traceback agent runs the inverse HSA algorithm (details given later in Section 4) to trace attack packets back to their origins and infer the best switch port for defense according to the previously established forwarding model of the whole network, and finally returns the identified switch port to the controller.

4) The controller generates and sends the OpenFlow message to the switch where the port is located to block the malicious traffic. Subsequently, the controller generates a Flow-Mod or Port-Mod message [28] according to the returned switch port and the corresponding defense strategies, and then sends the OpenFlow message to the switch where the port is located. Once the switch receives the message, it updates its flow tables or changes the link state of the port to down preventing the malicious traffic from injecting the network again. As shown in Fig. 2, both Host A and Host B locate outside the data center, their traffic follows Switch D, the DC switch to Server C; the malicious traffic from Host A to Server C is blocked on Switch D preventing it from being forwarded to Server C through the DC switch while the normal traffic from Host B reaches Server C unobstructed.

4. Algorithm Design

In the IPS architecture described in the previous section, it is essential to pinpoint the attack source and infer the best switch port for defense. In this section, we first expand the HSA [20] framework to construct inverse forwarding functions of all switch ports in the network. Then, we design a protocol-independent algorithm to find the best switch port for defense using the inverse forwarding functions and the backtrack technique [5].

4.1. Header Space Analysis

The theory was first proposed by Kazemian Peyman [20], and used for network verification and debugging. The definition of this theory is as follows:

Header Space, H : A packet header is represented as a flat sequence of ones and zeros. Formally, a header is a point, and a flow is a region in the $\{0,1\}^L$ space, where L is an upper bound on the header length.

Network Space, N : The network is modeled as a set of boxes called switches with external interfaces called ports, each of which is modeled as having a unique identifier. If we take the cross-product of the switch-port space (the space of all ports in the network, S) with H, we can represent a packet traversing on a link as a point in $\{0,1\}^L * \{1,...,P\}$ space, where $\{1,...,P\}$ is the list of ports in the network.

Switch Transfer Function, $T()$: As a packet traverses the network, it is transformed from one point in Network Space to another point(s) in Network Space. A node can be modeled using its transfer function T that maps header h arriving on port p:

$$T(h,p) : (h,p) \rightarrow \{(h1,p1),(h2,p2),\cdots\} \tag{1}$$

Network Transfer Function, $\Psi()$: Given that switch ports are numbered uniquely, all the switch transfer functions are combined into a composite transfer function describing the overall behavior of the network. Formally, if a network consists of n boxes with transfer functions $T_1(.),\ldots,T_n(.)$, then

$$\Psi(h,p) = \begin{cases} T_1(h,p) & \text{if} \quad p \in switch_1 \\ \ldots & \quad\ldots \\ T_n(h,p) & \text{if} \quad p \in switch_n \end{cases} \tag{2}$$

Topology Transfer Function, $\Gamma()$: A unidirectional link connects a source port P_{src} to a destination port P_{dst} and delivers packets from P_{src} to P_{dst}. The topology of a network is defined by the set of links in the network, each represented by its source and destination ports. We can model the network topology using a topology transfer function, $\Gamma()$, defined as:

$$\Gamma(h,p) = \begin{cases} \{(h,p^*\} & \text{if } p\, connected\, to\, p^* \\ \{\} & \text{if } p\, is\, not\, connected \end{cases} \tag{3}$$

Inverse of Switch Transfer Function, $T^{-1}()$: For a given switch, the finding application requires working backward from an output header to determine what input (header, port) pairs could have produced it. We define the inverse of switch transfer function as:

$$T^{-1}(h,p) = \{(h',p')|(h,p) \in T(h',p')\} \tag{4}$$

4.2. Inverse Forwarding Functions

To inverse a forwarding function, we need to work backward from a pair of output packet header and output port to infer which pair of input header and input port might have produced it. We expand the HSA theory to add the following two inverse functions.

Inverse of Network Transfer Function,$\Psi^{-1}()$:For a given header at an output port (h,p), $\Psi^{-1}(h,p)$ is the set of all input headers at input port, (h_i,p_i),such that (h,p) $\in$ $\Psi(h_i,p_i)$:

$$\Psi^{-1}(h,p) = \{(h',p')|(h,p) \in \Psi(h',p')\} \tag{5}$$

A transfer function maps each (h,p) pair to a set of other pairs. By following the mapping backward, we can invert a transfer function.

$$\Psi^{-1}(h,p) = \begin{cases} {T_1}^{-1}(h,p) & \text{if} \quad p \in switch_1 \\ \dots & \quad \dots \\ {T_n}^{-1}(h,p) & \text{if} \quad p \in switch_n \end{cases} \tag{6}$$

Inverse of Topology Transfer Function, $\Gamma^{-1}()$: For an arbitral head space h and a given port p, a port p' is connected to p, and the packet reaches from port p' to p.

$$\Gamma^{-1}(h,p) = \{(h,p')|(h,p) \in \Gamma(h,p')\} \tag{7}$$

4.3. Algorithm

In a campus network, there are m OF-enabled switches: S={s_1,...,s_{m-1},s_m}, and the switch s_i has n_i physics ports. In the total campus network, there are n switch ports: P={p_1,...,p_{n-1},p_n}, $n = \sum_{i=1}^{m} n_i$. The IPS captures an attack packet from the mirror traffic of the DC switch j and gets its header space h_{tar}; according to Switch j's Mac-Port table, the port p attached to the victim host is found. According to the principle of network reachability, there is an attack path from the attack source switch port q to p. The path is denoted by:

$$q \rightarrow p_1 \rightarrow ... \rightarrow p_{k-1} \rightarrow p_k \rightarrow p \tag{8}$$

the malicious traffic is injected through Port q, so it is identified as the best switch port for defense. Port q is calculated by:

$$(h,q) = \Psi^{-1}(\Gamma^{-1}(...(\Psi^{-1}(h_{tar},p)))) \tag{9}$$

According to the flow tables of the total switches, Switch Transfer Function $T()$ is calculated, and Inverse of Switch Transfer Function $T^{-1}()$ is calculated by Equation (4), then

Inverse of Network Transfer Function $\Psi^{-1}()$ is calculated by Equation (6). According to the network topology, Inverse of Topology Transfer Function $\Gamma^{-1}()$ is calculated. Finally, we trace the pair (header, port) backward (using the inverse of transfer functions at each step) to find the attack source port q as the best switch port for defense.

To pinpoint the attack source and infer the best switch port for defense, we have designed the algorithm (see Fig. 3). The algorithm makes use of the following three functions: 1) $mac\_src(h_{tar})$ refers to the source MAC address of the attack packet;
2) $mac\_dst(h_{tar})$ refers to the destination MAC address of the attack packet;
3)$find(T, mac\_src(h_{tar}))$ queries the source MAC address of the attack packet from T, which is the set of ($port$,mac) pairs maintained by the DC switch, and returns the corresponding switch port.

The algorithm exploits inverse forwarding functions defined above to infer possible input ports that can forward packets to the target port, then uses the backtrack method to traverse all inferred ports according to the depth-first search strategy, and eventually finds the attack source. The attack source may be either an internal host of the campus network or an external Internet host. In the former case, the algorithm returns the switch port attached to the internal malicious host as the best port for defense. In the latter, the algorithm returns a WAN port of border switches or routers as the best port for defense.

The complexity of our proposed algorithm is O($P \cdot N$), where P is the total number of activating ports which connect to other hosts or switches and N is the maximum number of flow entries in an arbitral switch. For a given header at an output port (h_0, p_0), $\Psi^{-1}(h, p)$ is the set of all input headers at the input port, so the returning (header,port) pairs count of $\Psi^{-1}(h, p)$ is greater than or equal to 1. If the returning (header,port) pairs count of each $\Psi^{-1}(h, p)$ equal to 1, the search path length will less than the diameter of networks. If the returning (header,port) pairs count of each $\Psi^{-1}(h, p)$ is greater than 1, the algorithm needs to traverse the each returning (header,port) pairs in proper order, the search process didn't be terminated until finding a first feasible solution adopting the depth-first search strategy.

5. Implementation

We implement a prototype of the proposed IPS at the central SDN controller using the open-source Ryu SDN Framework [11] and the analysis engine using the open-source IDS Snort [10]. The analysis engine inspects the mirror traffic, and send an alarm to the controller detecting an intrusion attempt. We develop two components: the attack traceback agent, the data forwarding and attack mitigation APP built on the SDN controller.

The attack traceback agent is developed by Python Language, communicate with the SDN controller over a TCP socket, and is a socket-based server that 1) receives flow entries varieties of all switches and topology changes of the entire network, 2) and returns the best defense switch for an attack event. The first function is capturing flow entries varieties and topology changes to establish the forwarding model of the entire network; The second function is tracing the attack packet back to its origin based on the above forwarding model, and calculating the best defense switch port by the algorithm(see Fig. 3). Based on a base class library of the open-source project Header Space Library [19], we add critical codes to support inverse forwarding functions and implement the agent's total functions.

```
Require:
    The header space of a captured attack packet: $h_{tar}$;
    The set of ($port$,$mac$) pairs maintained by the DCN switch: $T$;
    The set of switch ports that attached to Internet: $P_{wan}$;
    The set of switch ports that attached to hosts or Internet: $P_{end}$.
Ensure: The switch port that the attack packet is injected into: $q$.
if $mac\_src(h_{tar}) \in T.mac$ then
    $q \leftarrow find(T, mac\_src(h_{tar}))$
    return $q$
else
    $p_{tar} \leftarrow find(T, mac\_dst(h_{tar}))$
    $history \leftarrow p_{tar}$
    $Stack \leftarrow [header : h_{tar}, port : p_{tar}]$
    while $Stack.size() > 0$ do
        $r \leftarrow Stack.pop()$
        $history.append(r.port)$
        $temp \leftarrow \Psi^{-1}(r.h, r.p)$
        for $(h, p)$ in $temp$ do
            if $p \notin history$ then
                if $p \in P_{wan}$ then
                    $Deque.addLast(p)$
                else
                    $Deque.addFirst(p)$
                end if
            end if
        end for
        while $Deque.size() > 0$ do
            $q \leftarrow Deque.removeFirst()$
            if $q \in P_{end}$ then
                return $q$
            else
                $(h, p') \leftarrow \Gamma^{-1}(h, q)$
                if $p' \notin history$ then
                    $Stack.push(h, p')$
                    break
                end if
            end if
        end while
    end while
end if
```

Fig. 3. Find the Best Defense Switch Port Algorithm

The data forwarding and attack mitigation APP built on the SDN controller is developed by Python Language, and has the following modules:

1) Forward Data. Because RYU's demo applications (such as $simple\_switch\_13.py$ and $rest\_router.py$ etc.) don't support to construct mixed networks which comprise Layer 2 and Layer 3 forwarding devices, in order to emulate the topology of the software-defined campus network, we complete the forwarding APP to support the mixed network architecture.

2) Send Network State Messages. The module sends a series of network state messages to the attack traceback agent via a TCP socket regularly. The network state messages mainly include three types: flow entries varieties messages by each Flow-Mod message, topology changes messages by each related Ryu event, and the MAC address changes messages of servers for updating the set of ($port$,mac) pairs T. We embed the corresponding code into the above forwarding APP. For example, in term of the MAC address changes messages, we capture ARP messages of the DCN switch to get the MAC address change states and send the state change messages to the attack traceback agent.

3) Receive Alert Messages. The module uses an existing library, namely $snort.lib$, which enables the SDN controller as the server to receive Snort alert messages from the analysis engine by a TCP Socket. On the Snort machine, the application Pigrelay [21] running is configured, to enable that the alarm messages generated are sent to the Ryu controller using a TCP socket. Once the controller receives an alarm message from the analysis engine, it then generates an intrusion event.

4) Response to Intrusion Events. Once it monitors an intrusion event, it sends a quest of looking up the best defense switch port to the attack traceback agent via a TCP socket. When the attack traceback agent returns the best defense switch port, according to its position and the appropriate defensive strategy, the App automatically generates a Flow-Mod or Port-Mod message and sends it to the defensive switch. After the switch receives the OpenFlow message, it updates its flow tables or changes the returning port link state to down, blocking the malicious traffic injecting it again.

6. Evaluation

To assess the IPS prototype system depicted above, we choose port scan attacks, i.e. NULL scanning and FIN scanning. We utilize Nmap [23] to conduct malicious scans, Hping3 [14] to generate the background traffic with the constant flow rate, and sflowtool [25] to capture and analyze data traffic.

We used two Lenovo ThinkServer RD440s, each of which has one Inter Xeon CPU E5-2407 and thirty-two GB memory, to install VMWare Exsi 5.1 for instantiating virtual machines and building two testbeds (see Table 1, 2). We conducted four types of assessments. The first type evaluates the effectiveness of our IPS to prevent intrusion attacks, the second type assesses the effectiveness of the attack traceback algorithm, the third type assesses the timeliness of the proposed system in intrusion prevention, and the last type of evaluation compares the system performance of the campus network when equipped with our IPS, existing IPS, or without any IPS. The first three types of assessment are carried out on Testbed I, while the last type is conducted on Testbed II.

Table 1. Testbed I Configuration

No.	Hardware	Software
1	2 CPUs/6GB RAM	Snort
2	4 CPUs/2GB RAM	Ryu/Beacon/Tracing Attack Agent
3	2 CPUs/2GB RAM	sFlow
4	2 CPUs/6GB RAM	Mininet

Table 2. Testbed II Configuration

No.	Hardware	Software
1	2 CPUs/6GB RAM	Snort
2	4 CPUs/2GB RAM	Ryu/Tracing Attack Agent
3	2 CPUs/2GB RAM	sFlow
4	4 CPUs/4GB RAM	LXC/OpenvSwitch
5	4 CPUs/4GB RAM	LXC/OpenvSwitch
6	2 CPUs/2GB RAM	LXC/Linux Bridge
7	4 CPUs/4GB RAM	Linux Bridge

6.1. Effectiveness of Intrusion Prevention

We use Mininet to emulate a small-scale software-defined campus network with a typical topology illustrated in Fig. 4. In the topology, we use two hosts H5, H6, and a router R1 to emulate the Internet, and the OpenFlow switch S2 acts as a border router between the campus network and Internet, and the switches S1, S2, S3 are OpenFlow switches controlled by a Ryu controller.

We carried out three experiments to evaluate the effectiveness of our IPS. The emulated host H3 in the data center network acts as the victim in each experiment. The different experiments assess the IPS effectiveness under various attacks (see Table 3). The first experiment demonstrates the attack from a DC internal host, i.e. H4 acts as the malicious host, which is named Attack I. The second experiment reveals the attack from an ON host, i.e. H1 acts as the attack source, which is called Attack II. The third one focuses on the attack from an Internet host, i.e. H5 acts as the intrusion traffic origin, which is termed Attack III. In each of the three experiments, we assess the effectiveness of the proposed IPS under two scenarios. In the first scenario, our IPS is not configured with attack traceback, while in the second one our IPS has the fully functional attack traceback mechanism.

Table 3. Different Attack Types

Attack Type	Victim	Attacker
I	a DC server	a DC internal host
II	a DC server	an ON host
III	a DC server	an Internet host

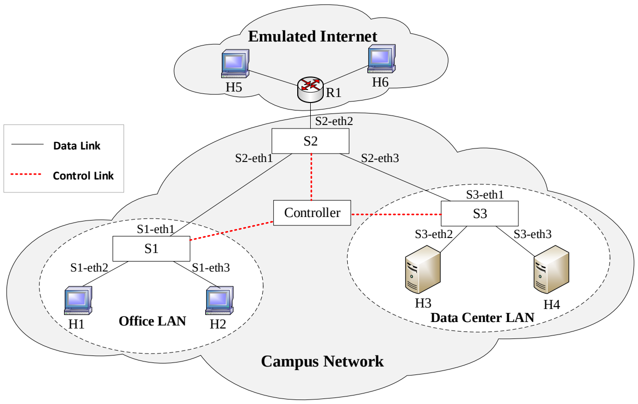
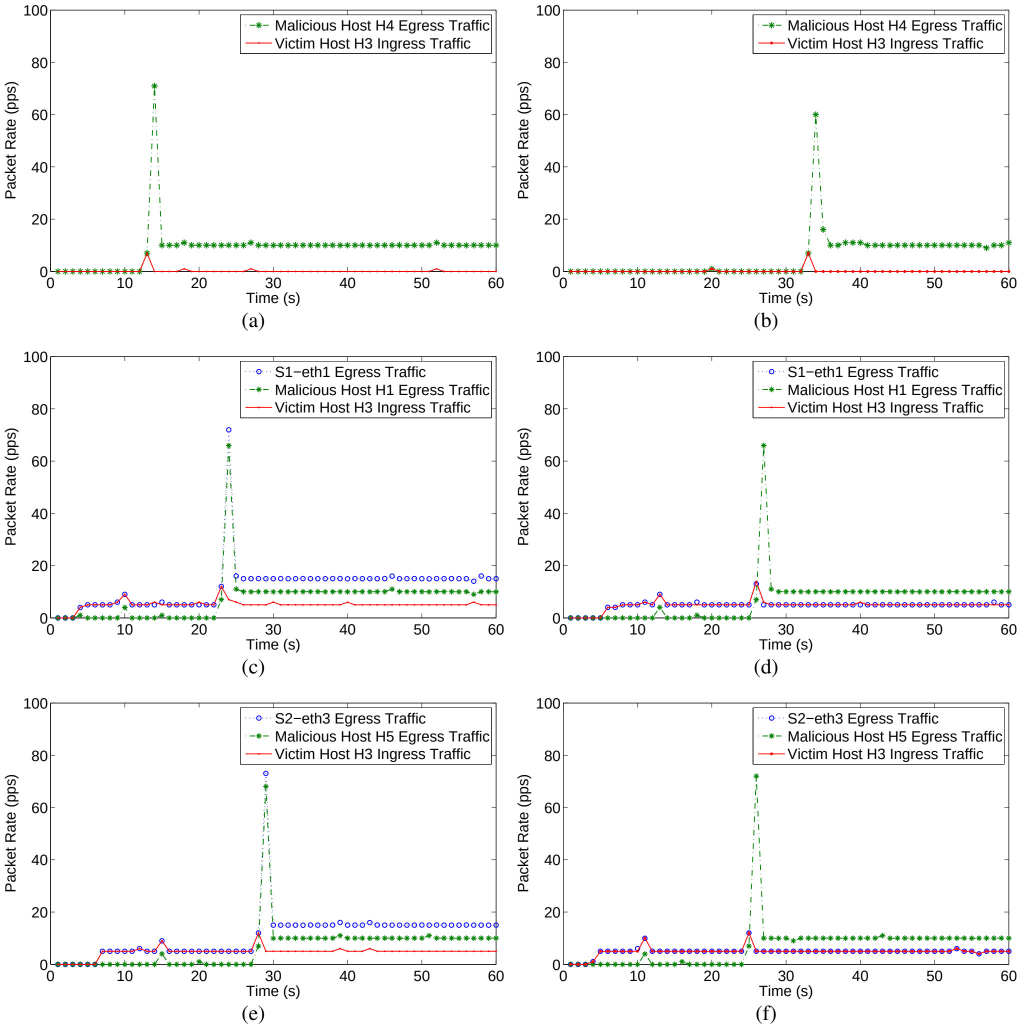


Fig. 4. Emulated Topology of the Software-defined Campus Network

Fig.5a and Fig.5b illustrate the packet rate changes during the process the victim host H3 is attacked by an internal malicious host H4, which is also located in the data center. As depicted by the red solid line, the ingress traffic of the victim H3 increases with the egress traffic of the malicious H4 at the beginning of the attack. However, the ingress traffic of H3 quickly falls to 0 pps at the 13th second and the 33rd second respectively in the two figures, which indicates that the proposed IPS is capable of detecting and stemming the internal attack traffic under both scenarios.

As depicted by the solid red line in both Fig.5c and Fig.5d, the victim host H3 starts to receive the ingress background traffic (from the host H2) at the constant rate of 5 pps since the 5th second. After about 20 seconds, an ON host H1 starts to attack H3. In the “Without Attack Traceback” scenario shown in Fig.5c above, the ingress traffic of the victim H3 begins to decrease at the 23rd second because the IPS has detected and blocked the attack traffic. However, the ON switch S1 continues to forward the attack traffic to its egress port eth1, as indicated by the dotted blue line, which changes with the dash-dotted green line, i.e. the malicious traffic. It reveals that the defense port chosen by the IPS without attack trackback is not on the switch nearest to the attack origin, i.e. S1, so the malicious traffic continues to consume network resources after being detected. In the “With Attack Traceback” scenario in Fig.5d below, after the IPS stems the malicious traffic at the 26th second, both the ingress traffic of the victim and the egress traffic of the ON switch S1 drop to the normal level containing only the background traffic. It indicates that the switch attached to the malicious host, i.e. S1, is employed to directly block the attack.

The experimental results in Fig.5e and Fig.5f exhibit the traffic changes when the victim H3 is attacked by an Internet host H5. Similar to Fig. 5c, the red solid line in Fig.5e shows the ingress traffic of H3 rises at the beginning of the attack and goes down after the IPS detects and blocks the attack traffic, but the dotted blue line exposes the egress traffic of the ON switch S2 still involves the attack traffic. In contrast, Fig.5f demonstrates both the ingress traffic of H3 and the egress traffic of S2 descend after the IPS takes action.



(a) (b) (c) (d) (e) (f)

Fig. 5. Traffic Changes during Three Types of Attacks Using the IPS with or without Attack Traceback: (a) Without Traceback For Attack I; (b) With Traceback For Attack I; (c) Without Traceback For Attack II; (d) With Traceback For Attack II; (e) Without Traceback For Attack III; (f) With Traceback For Attack III

These experiments bring us the following enlightenment. For Attack I, the proposed IPS can avoid malicious traffic from soaring with or without attack traceback. For the other attacks, there are fundamental differences between the two IPS configurations. If the proposed IPS cannot trace attack packets back, malicious traffic can still bring transmission burden to the network even if it is blocked from injecting the victim host. On the contrary, our complete IPS solution can accurately find the best switch port for blocking malicious traffic, so it is able to prevent intrusion attacks more effectively.

6.2. Effectiveness of Attack Traceback

To evaluate the effectiveness of the attack traceback algorithm for large scale campus networks, we use Mininet [27] to replicate the Stanford backbone network, which is a population of more than 15,000 students, 2,000 faculty, and five /16 IPv4 subnets. According to the literatures [36,18], we use Open vSwitch (OVS) [13] to emulate the routers, and install the reserved equivalent OpenFlow rules in the OVS switches with the SDN Controller Beacon [12]. we implement a bundle of the Beacon which can send a series of network state messages to the attack traceback agent regularly. We use emulated hosts to attack the victim host, and perform the attack traceback algorithm to pinpoint the attack source and infer the best switch port for defense. Figure 6 shows the part of the network that is used for experiments in this section. In the entire topology, there are 26 OF-enabled switches and 240 hosts (Due to the limited space, hosts is omitted in Figure 6). we use the two core switches S1, S2 to connect Internet, the two access switches S15, S16 act as the data center LAN switches, and the other access switch S3, S4,..., S14 act as the office LAN switches.

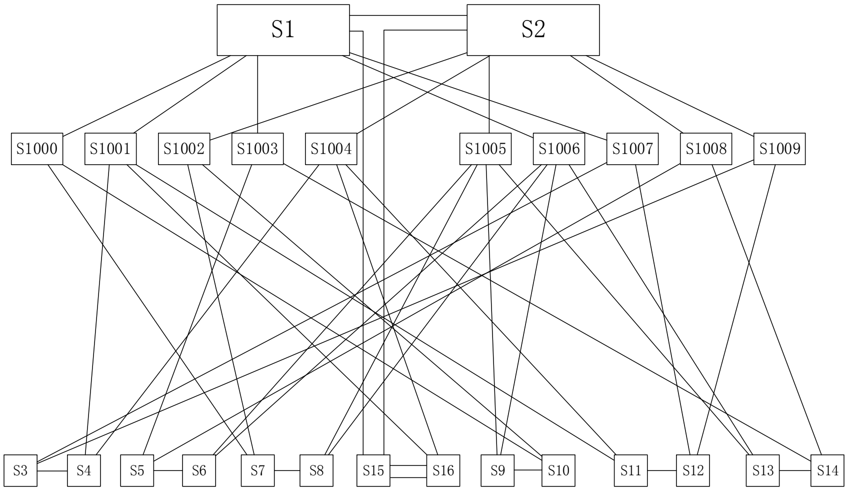


Fig. 6. Topology of the Backbone Network of Stanford University

We carried out three experiments to evaluate the effectiveness of our attack traceback algorithm. The emulated host H227 connected with Switch S15 in the data center network acts as the victim in each experiment. The different experiments assess the algorithm effectiveness under various attacks (see Table 3). The first experiment demonstrates the attack from two DC internal hosts, i.e. H208 and H210 connected with Switch S15 acts as the malicious host, which is named Attack I. The second experiment reveals the attack from ON two hosts, i.e. H77 connected with Switch S4 and H197 connected with Switch S13 act as the attack source, which is called Attack II. The third one focuses on the attack from two Internet hosts, i.e. H17 and H18 connected with Switch H1 act as the intrusion traffic origin, which is termed Attack III. In each of the three experiments, we perform our attack traceback algorithm 10 times for each attack, record the traceback path, the returned switch port and its execution time.

The above experimental results show our algorithm can pinpoint the attack source for three kinds of attacks and infer the best switch port for defense. As depicted by the red dash-dotted line in Fig.7, we use our trackback algorithm to find the attack source host H18 and infer the switch port S1-eth4 connected Internet, as the best switch port for defense, following the trackback path S15 → S1. Also, as depicted by the yellow dash-dotted line, we use the algorithm to find the attacker H197 and infer Port S13-eth9 connected H197, as the blocking port, along the path S15 → S1 → S1006 → S13.

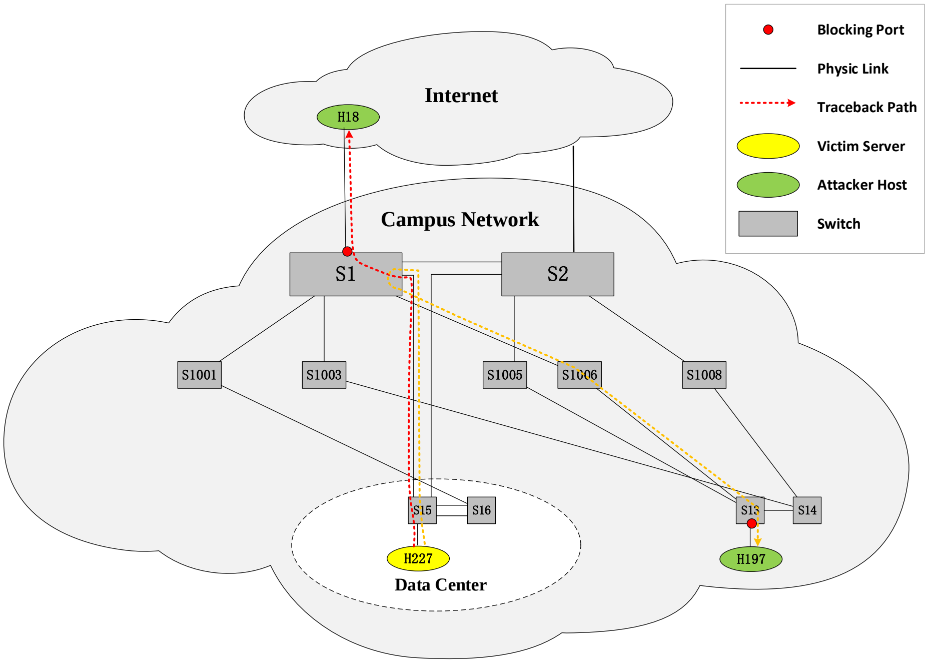


Fig. 7. Traceback Paths and Blocking Ports under Various Attacks

Fig.8 shows the execution time of our trackback algorithm under three types of attacks defined previously in the Stanford campus network. As depicted by the green boxes, it takes about 1 ms to infer the best switch port for defending Attack I. Because the attack is defended within the DC LAN, it only needs to find the Port-Mac table of the DC switch (as line 2 in Fig. 3). As depicted by the blue boxes, their execution time varies greatly for the attackers H77 and H197 although they are belong to the same Attack II. The reason for this difference is that it spent different time to traverse the list of flow entries for the trackbacking switches due to the different number of flow entries for the different switches. As shown in the red boxes, it takes about 2 ms to infer the best switch port for defending Attack III. For Attack I, Attack II, and Attack III, its execution time is 1.1ms, 3.5ms, and 2.2ms, respectively. And the average execution time is about 2.27 ms. In general, the more switches that traceback, the total number of traversal flow entries also increases, and the corresponding execution time will also be longer. Overall, the execution time in the experiments demonstrates that our attack traceback algorithm can work in real-time.

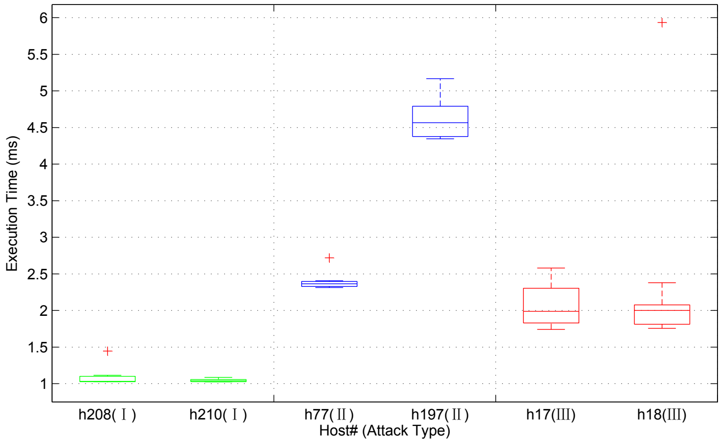


Fig. 8. Execution Time of the Traceback Algorithm under Various Attacks

6.3. Timeliness of Intrusion Prevention

To assess how timely the proposed system can prevent intrusion attacks, we use Mininet to emulate three different scales as manifested in Table 4. In each scale campus network, we attempt to launch three different types of attacks (see Table 3) and measure the defense time spent from receiving the alarm until sending out the OpenFlow message, which is comprised of the computing time of the algorithm (see Fig. 3) and the other time.

Fig.6 shows the defense time under three types of attacks defined previously in networks of different scales. It takes 10-14 ms for the proposed IPS to defend Attack I. Because the attack is defended within the DC LAN, the time varies little with the scale of the campus network. For Attack II and Attack III, with the growth of the network scale, the defense time increases linearly, mainly because the computing time of the attack traceback algorithm, depicted by the lower portion of the bar with horizontal strips, increases when it has to recursively search more switch ports. Thus, the longest defense time, about

Table 4. Three Scale Campus Networks Topologies

Scale Type	Switch Count	LAN Count	Host Count
Small	5	4	40
Medium	50	49	490
Large	100	99	990

76 ms, happens in the large-scale network with 100 switches when it experiences attacks from the Internet. Moreover, the other time, depicted by the upper portion of the bar with diagonal strips, varies litter with an average of 12.2 ms, because time spent on generating OpenFlow messages and exchanging them and other messages is not affected by the network scale and attack type. The defense time in the experiments demonstrates that our IPS and the attack traceback algorithm work in real-time.

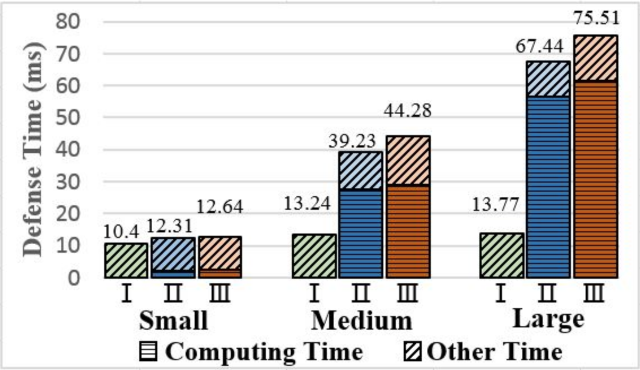


Fig. 9. Defense Time in Each Experiment

6.4. Overhead of Intrusion Prevention

We use LXC [22] to emulate a small-scale software-defined campus network with a typical topology portrayed in Fig. 4. As Table 2 shown, we use three LXC hosts (VM4, VM5, VM6) and VM7 (running Linux Bridge Application as a router in the Internet zone) to emulate the campus network. VM4 has created two containers (H1 and H2) and two OpenvSwitch switches (S1 and S2). VM5 has created two containers (H3 and H4) and one OpenvSwitch switch (S3). VM6 has created two containers (H5, H6). Switch S1, S2 and S3 are controlled by a Ryu controller (VM2). The network is connected to the Internet through the border switch S2 and the router R1 that is emulated with Linux Bridge [6]. To assess the overhead of the proposed IPS, we compare the performance of the network when it is configured without IPS, with the existing IPS, and with our IPS.

When the existing IPS is used, Snort is configured in the inline mode to work as the IPS, and it is deployed between the border switch S2 and DC switch S3 in series. When the proposed IPS is used, as delineated in Fig. 1, the detection engine is deployed on the bypass of the DC switch S3 and Snort is configured in the passive mode. In each case, we evaluate the throughput and RTT of the two paths: one is from an Internet host to a DC host, and another is from an ON host to the same DC host.

As shown in Fig.10a and Fig.10b, the throughput of the network with our proposed IPS is very close to the throughput when the network has no IPS, as revealed by the green dashed line and red solid line. On the contrary, as indicated by the blue dash-dotted line, the throughput with the existing IPS degrades most 50% when the data transmission rate is higher than 20Mbps. Compared with the existing IDS deployed in serial mode, its throughput increases 2-4 times the transmission rate is between 40 Mbps and 100 Mbps. The same trend continues in Fig.11a and Fig.11b. The RTT of our proposed IPS is very close to the RTT of without using IPS when the system workload continuously increases from 20 to 100 Mbps. During the same period, the RTT of the existing IPS is much higher than the other two cases. The experimental results suggest the proposed IPS incurs reasonable overhead, which is much lower than the existing solution.

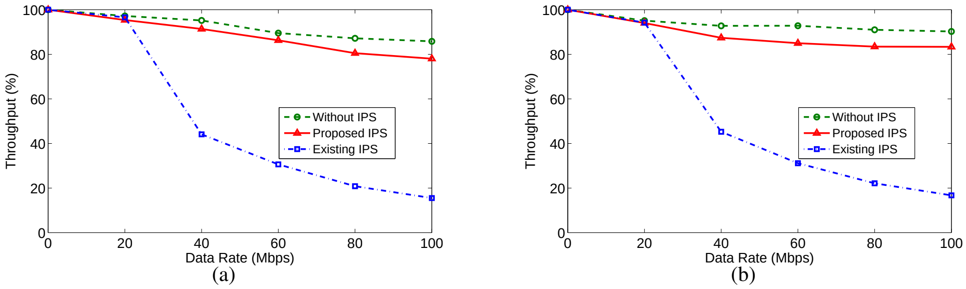


Fig. 10. Throughput of Three IPS Schemes: (a) From Internet Host to DC Server; (b) From ON Host to DC Server

7. Discussion

According to the evaluation results in the literature about attack traceback (details in Section 2.2) and the experimental evaluation results of our proposed traceback method, we compared and analyzed the various methods (see Table 5) in the aspects, such as traceback accuracy, overhead and compatibility, etc.

Compare with other methods, our traceback method based on the inverse HSA performs higher accuracy of locating attack source and lower overhead. Our proposed method only needs to locate the attack proxy host for internal attacks or the border router for external attacks, and block the malicious traffic injecting from their uplink switch port. And it can find the attack path and the attack source when only one packet is detected as an intrusion attempt by its detection engine component, so our proposed IPS can avoid malicious

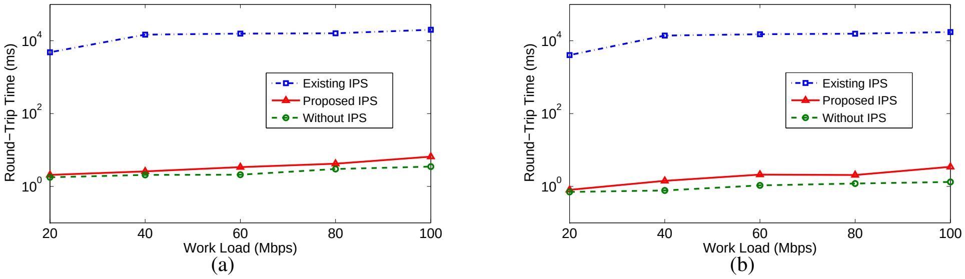


Fig. 11. Round-Trip Time of Three IPS Schemes: (a) From Internet Host to DC Server; (b) From ON Host to DC Server

traffic from soaring and prevent intrusion attacks more effectively. Also, the traceback method and the IPS architecture can be compatible with the existing network infrastructure (such as routers, switches, etc), don't need special devices or modify their protocols, and can support the hybrid SDN networks because the OF-disenabled Middle Boxes can be modeled as the inverse HSA as same as the SDN switches. However, our method has some shortcomings, such as needing the prior knowledge of network topologies and no supporting about the post attack analysis, this will be the next step for us to study in the future.

8. Conclusion

As traditional networks, the software-defined campus network also suffers from intrusion attacks. Current solutions for intrusion prevention cannot meet the requirements of the campus network. Existing methods of attack traceback are either limited to specific protocols or incur high overhead. To protect the data center of the campus network from internal and external attacks, we propose an Intrusion Prevention System (IPS) based on the coordinated control between the detection engine, the attack traceback agent, and the software-defined control plane. The proposed IPS has the following advantages:

First, it can accurately and timely find the best switch port for defense and prevent the malicious traffic from injecting the network at the first time due to our traceback algorithm based on the inverse HSA. We expand the Header Space Analysis (HSA) framework to construct the inverse forwarding functions and design a novel protocol-independent algorithm to trace attack packets back to their origins and infer the best switch port for defending different attacks using the inverse forwarding functions. It can locate the attack host for internal attacks or the border router for external attacks, and block the malicious traffic injecting from their uplink switch port. In this manner, the malicious traffic would not travel through additional switches and increase the transmission burden of the network. Compare with other traceback methods, our method based on the inverse HSA performs higher accuracy of locating attack source and lower overhead. We replicate the Stanford backbone network to verify the effectiveness of our algorithm, it can pinpoint the attack source for three kinds of attacks and infer the best switch port for defense, and its average execution time is about 2.27 ms, which can work in real-time.

Table 5. Comparison of Various Traceback Methods

	Link Test	**ICMP Trace**	**Logging**	**Overlay Network**	**PPM**	**DPM**	**Our Method**
Traceback Accuracy	Medium	Good for less packets	Medium	Good	Medium	Good	Good
Device Overhead	High	High	High	Low	Medium	Medium	Low
Bandwidth Overhead	High	High	High	Nil	Low	Low	Nil
Traceback with number of packets	Huge	Huge	One	Small	Large	Small	One
Device Compatibility	Good	Good	Need special routers	Add special routers	Modify protocols	Modify protocols	Good
Prior Knowledge of Topologies	Need	Not Need	Not Need	Need	Not Need	Not Need	Need
Post Attack Analysis	Not Possible	Possible	Possible	Not Possible	Possible	Possible	Not Possible

Second, it incurs reasonable overhead due to coordinated control design and collecting network traffic with port mirroring. It leverages the coordinated control between the detection engine, the attack traceback agent, and the software-defined control plane. The three components work together to detect intrusion attacks, as well as to plan and enforce the corresponding defense mechanisms swiftly. Our proposed IPS is deployed to bypass the data center switch and collect network traffic with port mirroring. Compared with the existing IDS deployed in serial mode, this design can avoid a single point of failure, reduce the probability of network congestion, and defend the data center's internal attacks. The experimental results show that its throughput increases 2-4 times than the existing IDS deployed in serial mode when the transmission rate is between 40 Mbps and 100 Mbps.

Finally, it can meet the real-time requirements for defense internal attacks and external attacks in different scales, and avoid malicious traffic from soaring and prevent intrusion attacks more effectively. We have implemented a prototype of the proposed IPS and conducted several experiments to evaluate its performance. The experimental results show that the overhead of our IPS is very low, which enables it to meet the real-time requirements. The average defense time is between 10 and 14 ms for the data center internal attacks of different scales. For external attacks, the maximum defense time is about 76 ms for a large-scale network with 100 switches.

The algorithm for finding the best defense switch port is based on stateless forwarding devices and known initial network topologies. In the future, we will improve the algorithm to make it support more stateful forwarding devices regardless of the initial network topology.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China (Grant N0. 61261019 and 61762071), the Inner Mongolia Science & Technology Plan (Grant No. 201802027), the Inner Mongolia Science and Technology Innovation Guidance and Incentive Fund (Grant No. 111-0406041701), and the Inner Mongolia Autonomous Region Natural Science Foundation (Grant No. 2016MS0614 and 2018MS06023).

References

1. Floodlight OpenFlow Controler (2019), `https://github.com/floodlight/floodlight`
2. Belenky, A., Ansari, N.: Ip traceback with deterministic packet marking. IEEE communications letters 7(4), 162–164 (2003)
3. Bellovin, S.M., Leech, M., Taylor, T.: Icmp traceback messages (2003)
4. Bhavani, Y., Janaki, V., Sridevi, R.: Ip traceback through modified probabilistic packet marking algorithm using record route. In: Proceedings of the Third International Conference on Computational Intelligence and Informatics. pp. 481–489. Springer (2020)
5. Bitner, J.R., Reingold, E.M.: Backtrack programming techniques. Communications of the Acm 18(11), 651–656 (1975)
6. Bridge, L.: Linux Bridge (2020), `https://wiki.linuxfoundation.org/networking/bridge`
7. Chao, G., Sarac, K.: Toward a practical packet marking approach for ip traceback. International Journal of Network Security 8(3), 271–281 (2009)
8. Chen, P.J., Chen, Y.W.: Implementation of sdn based network intrusion detection and prevention system. In: International Carnahan Conference on Security Technology. pp. 141–146 (2016)

9. Chi, Y., Jiang, T., Li, X., Gao, C.: Design and implementation of cloud platform intrusion prevention system based on sdn. In: Big Data Analysis (ICBDA), 2017 IEEE 2nd International Conference on. pp. 847–852. IEEE (2017)
10. Cisco: Snort (2018), https://www.snort.org
11. Community, R.S.F.: Ryu SDN Framework (2018), https://osrg.github.io/ryu/
12. Erickson, D.: Beacon (2013), https://openflow.stanford.edu/display/Beacon.html
13. Foundation, T.L.: Open vSwitch (2016), http://www.openvswitch.org/
14. hping3: hping3 (2005), http://www.hping.org/hping3.html
15. Inc., M.: GandCrab ransomware (2020), https://www.malwarebytes.com/gandcrab/
16. Inc., N.: DDoS Threat Report 2020 Q1 (2020), https://blog.nexusguard.com/threat-report/ddos-threat-report-2020-q1
17. Izaddoost, A., Othman, M., Rasid, M.F.A.: Accurate icmp traceback model under dos/ddos attack. In: 15th International Conference on Advanced Computing and Communications (ADCOM 2007). pp. 441–446. IEEE (2007)
18. James Hongyi Zeng, P.K.: Automatic Test Packet Generation(ATPG) (2015), https://github.com/eastzone/atpg
19. Kazemian, P.: Header Space Library (Hassel) (2014), https://bitbucket.org/peymank/hassel-public/
20. Kazemian, P., Varghese, G., McKeown, N.: Header space analysis: Static checking for networks. Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12) pp. 113–126 (2012)
21. Lin, C.W.: Pigrelay (2015), https://github.com/John-Lin/pigrelay
22. LXC: LXC (2018), https://linuxcontainers.org/lxc/introduction/
23. Lyon, G.: Nmap: the Network Mapper (2018), https://nmap.org/
24. Ma, M.: Tabu marking scheme to speedup ip traceback. Computer Networks 50(18), 3536–3549 (2006)
25. McKee, N.: sflowtool (2018), https://github.com/sflow/sflowtool
26. McKeown N.: Software-defined networking (2009), http://infocom2009.ieee-infocom.org/technicalProgram.htm
27. Mininet: Mininet (2018), http://mininet.org/
28. ONF: OpenFlow Spec v1.3.5 [online] Technical report ONF TS-023. Tech. rep., Open Networking Foundation (2015), https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.3.5.pdf
29. Park, K., Lee, H.: On the effectiveness of probabilistic packet marking for ip traceback under denial of service attack. In: Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213). vol. 1, pp. 338–347. IEEE (2001)
30. Satheesh, N., Sudha, D., Suganthi, D., Sudhakar, S., Dhanaraj, S., Sriram, V., Priya, V.: Certain improvements to location aided packet marking and ddos attacks in internet. Journal of Engineering Science and Technology 15(1), 94–107 (2020)
31. Saurabh, S., Sairam, A.: Icmp based ip traceback with negligible overhead for highly distributed reflector attack using bloom filters. Computer Communications (01 2014)
32. Snoeren, A.C., Partridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Kent, S.T., Strayer, W.T.: Hash-based ip traceback. Acm Sigcomm Computer Communication Review 31(4), 3–14 (2001)
33. Stone, R.: Centertrack: an ip overlay network for tracking dos floods. In: Usenixsecurity Symposium, August (2000)
34. Xing, T., Huang, D., Xu, L., Chung, C.J., Khatkar, P.: Snortflow: A openflow-based intrusion prevention system in cloud environment. In: Second Geni Research and Educational Experiment Workshop. pp. 89–92 (2013)

35. Yoon, C., Park, T., Lee, S., Kang, H., Shin, S., Zhang, Z.: Enabling security functions with sdn: A feasibility study. Computer Networks 85(C), 19–35 (2015)
36. Zeng, H., Kazemian, P., Varghese, G., McKeown, N.: Automatic test packet generation. IEEE/ACM Transactions on Networking 22(2), 554–566 (2014)

Guangfeng Guo received two B.S. degrees in Computer Science & Technology and Educational Technology from Inner Mongolia Normal University in 2003 and received an M.S. degree in Computer Application Technology from Tianjin University in 2009. He is a Ph.D. student in Computer Application Technology at Inner Mongolia University. He is also an associate professor in Baotou Teachers' College at Inner Mongolia University of Science & Technology. His research activity is in network security and mobile computing.

Junxing Zhang is a Professor in the College of Computer Science at the Inner Mongolia University. He is also the Director of the Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing. He received a B.S. degree in Computer Engineering from the Beijing University of Posts and Telecommunications, an M.S. degree in Computer Science from the Colorado State University, and a Ph.D. degree from the University of Utah. His research interests include network measurement and modeling, mobile and wireless networking, network security and verification, etc. Prof. Zhang was awarded the title of Grassland Talent by the government of the Inner Mongolia Autonomous Region in 2010. He has published over 40 papers in various internationally recognized journals and conferences, and led several national and provincial research projects. He also served as a peer reviewer for several international journals and conferences, such as IEEE Transactions on Mobile Computing, Wireless Networks, and ICNP.

Zhanfei Ma is a Professor in Baotou Teachers' College at Inner Mongolia University of Science & Technology. He received a B.S. degree in Computer Science and Education from Inner Mongolia Normal University in 1997, received an M.S. degree in Computer Software and Theory in 2002, and received a Ph.D. degree in Computer Application Technology from University of Science & Technology Beijing in 2008. His research interests include network information security and artificial intelligence.

Received: February 6, 2020; Accepted: November 30, 2020.

Computer Science and Information Systems 18(3):893–925 https://doi.org/10.2298/CSIS200530052R

Class Balancing in Customer Segments Classification Using Support Vector Machine Rule Extraction and Ensemble Learning

Sunčica Rogić and Ljiljana Kašćelan

University of Montenegro, 81000 Podgorica,
Montenegro
{suncica, ljiljak}@ucg.ac.me

Abstract. An objective and data-based market segmentation is a precondition for efficient targeting in direct marketing campaigns. The role of customer segments classification in direct marketing is to predict the segment of most valuable customers who is likely to respond to a campaign based on previous purchasing behavior. A good-performing predictive model can significantly increase revenue, but also, reduce unnecessary marketing campaign costs. As this segment of customers is generally the smallest, most classification methods lead to misclassification of the minor class. To overcome this problem, this paper proposes a class balancing approach based on Support Vector Machine-Rule Extraction (SVM-RE) and ensemble learning. Additionally, this approach allows for rule extraction, which can describe and explain different customer segments. Using a customer base from a company's direct marketing campaigns, the proposed approach is compared to other data balancing methods in terms of overall prediction accuracy, recall and precision for the minor class, as well as profitability of the campaign. It was found that the method performs better than other compared class balancing methods in terms of all mentioned criteria. Finally, the results confirm the superiority of the ensemble SVM method as a preprocessor, which effectively balances data in the process of customer segments classification.

Keywords: direct marketing, customer classification, class imbalance, SVM-Rule Extraction, ensemble.

1. Introduction

Direct marketing allows for direct communication with potential customers through various media, such as e-mail, catalogs, social media and the like. It is consumer-oriented, message is sent directly to consumers and, at the same time, it's a "call to action". One of the key issues of this type of marketing is the accurate identification of potential and current customers who will most likely respond to a campaign, i.e. targeting specific customers from an existing database as well as new potential leads. Usually, the customer targeting methods are split up in the literature into segmentation and scoring methods [1, 2]. Segmentation methods, using appropriate explanatory variables, partition the customers into homogenous segments regarding the anticipated response to a direct marketing campaign [3, 4]. Thus, the promotional offers and materials are distributed to such customer segments with highest expected probability of response. On the other hand, scoring methods are used in the customer response models [5, 6], by assigning certain scores to customers, based on the predicted likelihood of the response to the campaign. It is important to state that high probability of response to the campaign does not certainly imply high profits. Hence, methods for customer profitability prediction are included in some of the most important scoring methods [7–10].

Segmentation methods, which are most commonly applied in direct marketing, split a customer data set using Recency, Frequency and Monetary (RFM) attributes [11]. They are based on various techniques, ranging from the simplest cross-tabulation technique, to more complex weighted techniques [4, 12]. These techniques generally require a subjective assessment for the necessary parameters. For this reason, data mining methods, such as K-means or Artificial Neural Network (ANN) clustering, can give more objective results for RFM customer segmentation [13–15].

Recently, classification data mining methods have become very popular, as they can enable the prediction to which segment the customer belongs to, based on the characteristics of the customer [15, 16]. Since the most valuable customer segment is usually the smallest, there is a problem of class imbalance. This problem in most classification methods leads to bias toward small classes and most often to their misclassification [17, 18]. If this problem is ignored, most classification algorithms will not identify the most valuable customer segment at all, or will identify a very small number of customers within that segment, which may lead to an unprofitable campaign.

There are methods in the literature that overcome the class imbalance problem in different ways [19–21]. The main disadvantage of the most commonly used under-sampling method, is that it ignores the large number of examples of the larger class, that may contain significant information for class differentiation. In order to reduce sample bias and minimize the loss of significant information, it can be combined with ensemble techniques (balanced ensemble) [18, 22]. The balanced ensemble approach involves taking random subsets of a larger class (equal in size with a smaller class) in multiple iterations and generating different classification models over those subsets whose results are eventually aggregated to give a final result. Combining multiple classifiers in this way does not only balance classes, but also increases predictive accuracy, reduces sample bias, reduces variance i.e. increases stability of results and avoids overfitting [23, 24].

The previous literature confirms that in case of class imbalance and overlapping the SVM method has a good predictive performance and can be used as a preprocessor that balances classes for other classifiers [25]. However, the SVM classifier is a "black-box", i.e. does not generate a model that can be interpreted, which is very important in the classification of customer segments in order to describe the segments. This deficiency can be solved by a hybrid approach, where the SVM is combined with rules extracting techniques (SVM-RE) [26].

Considering the advantages of the SVM-RE method and ensemble approach noted above, this paper proposes customer segments classification in direct marketing based on a combination of SVM-RE predictive classification and ensemble meta-algorithms. Also, a comparison of the defined method with the standalone data balancing ensemble methods for customer classification was made. Specifically, this study highlights three main research questions (RQ):

RQ1 - Is the ensemble SVM-RE approach adequate for the prediction of customer segments in direct marketing?

RQ2 - Given the unbalanced nature of data in customer segmentation, what is the best class balancing method (ensemble SVM-RE or one of the standalone balancing ensemble methods)?

RQ3 - Is the ensemble SVM-RE method suitable for describing, i.e. explaining segments?

Undoubtedly, the primary success factor of direct marketing predictive models is class imbalance. There are two basic approaches used in previous studies to solve this problem. The first involves modifying the classifier by associating different misclassification costs for each class [27, 28]. Another approach to requires data changes. Balancing is regulated by generating or reducing data using over-sampling or under-sampling techniques [17, 18, 22, 29, 30]. However, both approaches have some drawbacks. Classifier-changing methods require extensive knowledge about the specific learning methods [31], so it is necessary to hire an expert for practical application. With resampling methods, the challenge is removing the data without losing the information necessary to distinguish the classes, as well as knowing whether removing some data would give different results (instability of the solution). When supplementing the data, the main challenge is how to supplement the minority class while maintaining its distribution. Also, the question is what is the optimal class ratio [31]. All this makes the analysis complex in practical applications. A small number of papers for the customer classification use the ordinary SVM method [17, 32, 33], but it has been shown that it is not immune to class imbalance either [17]. The main contribution of the SVM-RE method proposed in this paper is that it automatically eliminates noise and class imbalance. By adjusting the parameters of the SVM as a data preprocessor, the boundaries between the classes are shifted so that the examples of the majority class that are closest (most similar) to the minority class join the minority class. Rule extraction from such balanced data has good classification performance for the minority class as well. The extracted rules provide a description of the segment of the most valuable customers that cannot be obtained if the misclassification of this minority class is not resolved. The performance of SVM-RE methods is further enhanced by combining with ensemble techniques.

Unlike the above-mentioned studies, which mainly use data sets from publicly available repositories, this study uses real-life data that are disordered and have more noise. On such data, the challenge of balancing and achieving good predictive performance is even greater. The final step was using the public dataset to validate the model.

The practical implications of this paper relate to the more accurate and objective planning of direct marketing campaigns, as well as gaining deeper insight into different customer segments, which may lead to more precise targeting and increased profits.

The paper is organized as follows: The second section gives an overview of related papers. Section three shows the proposed methodology, and the fourth section presents the results of the empirical test, which are further discussed in the fifth section. Finally, the sixth section contains concluding remarks.

2. Literature Review

This section provides an overview of previous research related to the customer segmentation and the problem of class balancing in direct marketing.

2.1. SVM Rule Extraction Method

For linearly inseparable classes, Vapnik[34] proposed a SVM method that maps data (viewed as n-dimensional vectors) from the original space into a larger dimension space (feature space), where the classes can be separated by means of a hyperplane. Finding such a hyperplane is realized by minimizing the distance between its end position (so that the gap between the classes i.e. the margin is greater) and the closest points (support vectors). Instead of an explicit mapping function in a larger dimension space, a kernel function is used, which allows calculating the scalar product of the vectors (i.e. the distance of the support vector from the hyperplane) in the original space (kernel trick). Various kernel functions can be used, but Radial Basis Function (RBF) which was used in this paper, is applied most often [35]:

$$K(x_i,x_j) = \exp(-\gamma||x_i-x_j||^2) \tag{1}$$

The SVM algorithm, therefore, strives to maximize the margin in feature space, which boils down to the convex optimization i.e. quadratic programming problem in the original space (2):

$$max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \, \alpha_j y_i y_j K(x_i, x_j) \tag{2}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C \,, \; i = 1, \dots, n$$

where K is kernel function, αi are Lagrange multipliers, n is the number of training examples and C is a parameter, which is adjusted to trade off margin maximization against classification error minimization.

The training of the SVM classifier comes to the selection of the optimized values of the gamma parameter for the RBF kernel, and parameter C, which represents the boundary for the margin, i.e. empty space between classes. Selecting lower values for parameter C reduces over-fitting and increases the generality of the SVM model, i.e. its predictive performance.

In addition to solving the problem of linear inseparability of classes, the advantage of this method is that, in the case of class imbalance, it exhibits better predictive performance than the standard methods, such as logistic regression [25]. The literature confirms that the SVM can successfully remove the noise, i.e. class overlapping from data. Namely, the parameter C can be set so that a number of examples of a larger class, which are close to the example of a lower class (which means that they are similar), are declared as examples of the lower class. For this reason, the SVM can be used as a preprocessor that balances and purifies data, thus providing higher classification accuracy [25, 36].

However, the SVM does not generate an interpretable model, which is usually very important in application. This problem has been solved in the literature by means of rule extraction techniques that enable generating the rules from the SVM results [26, 37]. According to Barakat and Bradley [26] SVM-RE techniques are grouped into two categories: those based on the components of the SVM model, and those that do not use the internal structure of the SVM model, but draw the rules from the SVM output. When the SVM model is interpreted, or SVM is used as a data preprocessor, authors recommend techniques from the second group because they provide more understandable rules. In line with this recommendation, our research uses rule extraction from SVM output. Namely, customer targeting rules are derived from SVM output using a classification Decision Tree (DT) method [38].

The DT method divides the data set by attributes values, so that subgroups contain as many examples of one class as possible, i.e. their impurity is minimized. The criterion by which division is made (measure of quality of division) can be information gain [39], gain index [40], gini index [41], or accuracy of the whole tree. The attributes that provide the best division according to the given criterion are chosen. During the inductive division, a tree-shaped model is formed. The path from the root to the leaves defines if-then classification rules in the terms of the predictive attributes (tree nodes). The complexity and accuracy of the generated model depends on the depth of the tree, the minimum size of the node by which the division can be made (the number of examples in its subgroup), the leaf size, and the defined minimum gain achieved by the node division. The smaller the depth, the larger the minimum size for the split, the larger the leaf size and the higher minimum gain, lead to a less complex tree, but also a tree with smaller accuracy.

SVM rule extraction is not a new method in literature and it was applied in some previous economic studies [36, 38, 42], but for the topic of direct marketing, i.e., to solve the problem of the minor class of the most valuable customers in customer classification, it is applied for the first time in our research in this way.

2.2. Ensemble Methods

Ensemble methods use more learning algorithms to achieve better predictive performance than can be achieved with any of these learning algorithms alone.

There are several different ensemble methods. Thus, Bootstrap Aggregating [43] or shortened Bagging, constructs subsets of the training set using bootstrapping, which implies that the same example can be re-selected in the next iteration (sampling with replacement). The subsets thus obtained generate predictive models whose results are aggregated (the result for which most models voted is taken). If bootstrapping from a larger class takes as many instances as there are in a smaller class, then it is a case of balanced Bagging. Unlike Bagging, which generates samples and models simultaneously, Adaptive Boosting [44] generates the following sample and model based on previous results. Specifically, the succeeding sampling is more likely to select those examples that were previously incorrectly classified because they were given more weight. The result is obtained by weighting, i.e. based on the weight attached to the models depending on their accuracy. Random Forest [45] is an ensemble method that combines Bagging with a random selection of predictors. Each sample generates a forest of DTs that take random subsets of predictors, i.e., a random forest of DTs is generated.

Ensemble methods have been used in some previous studies in direct marketing. Hence, Gupta A. and Gupta G. [46] have compared neural networks and Random Forest to predict clients' response to a term deposit offered at a Portuguese bank. Their findings show that Random Forest performs better. Instead of boosting one learning algorithm, Lessmann et al. [47] proposed an ensemble approach that combines multiple different learning algorithms (decision trees, SVM, random forest, logistic regression, etc.) to create predictive marketing models, such as customer response prediction, profit scoring and churn prediction. In aggregating the results to evaluate the best models, they included profit maximization in addition to predictive model performance. The results showed that this ensemble approach outperformed standalone models in terms of profit. Lawi et al. [30] have combined Adaptive Boosting with SVM and achieved better predictive performance compared to ordinary SVM. In the approach proposed in this study, Bagging was combined with SVM to improve data preprocessing performance, i.e. to eliminate class overlapping, and balanced Bagging with the DT classifier is used to further help solve the problem of class imbalance and ultimately improve the performance of customer classification segments as much as possible.

2.3. Customer Segmentation in Direct Marketing

Hughes [11] defined one of the most commonly used customer segmentation method in direct marketing – RFM segmentation. The RFM model is based on the database of previous customers' purchasing behavior. Recency represents the time period since the last purchase, Frequency marks the number of purchases in the stated period of time and Monetary indicates the value of all customer's purchases during that period [48]. The analysis starts by sorting the available data into five equal segments (each containing 20% of customers), according to recency. Most recent customers receive the score 5, less recent score 4 and so on, following the Pareto principle – 20% of customers account for 80% of sales [49]. Following this procedure, customers are sorted according to their frequency, within the formed quintiles, receiving scores 5 to 1, which results in a database with 25 segments, and finally, database is split according to the monetary value by scoring the customers within the defined groups, which, in turn, results in a database with 125 groups based on the RFM values [4], where the best segment will have a 555 score, and the worst will have a 111 score. However, the choice of segments to be targeted in the future marketing campaigns is subjective.

The resulting segments based on the RFM model can be further analyzed using more objective data mining techniques, taking into account the customer features, their buying behavior or product specific variables [16, 50, 51].

Cheng and Chen [16] used k-means clustering [52] for RFM segmentation. They split the data into segments of 20% each (uniform coding) and created 3, 5 and 7 clusters to test the approach. The disadvantage of this approach is that uniform coding leads to the loss of fine differences between the values of RFM attributes (e.g., customers who have 5 or 9 transactions or those whose revenue is 5000 euros or 7000 euros can be placed in the same rank). Also, the pre-assumed number of clusters does not guarantee optimal RFM segmentation.

In order to develop a set of rules for targeting customers based on their features (the region and credit debt), the authors used a rough set and LEM2 rule extraction method. The predictive attributes also include RFM attributes, aiming to achieve high accuracy rate, as clusters are already formed on the basis of RFM. Hence, extracted rules perhaps do not show some significant customer characteristics for targeting, as they may be absorbed by the effect of RFM attributes. In addition, RFM attributes are unknown for new customers, so this model cannot be used for their prediction.

Additionally, the authors in [16] exclusively used accuracy rate (the percentage of precisely predicted examples within all examples) to determine their classification performance. Since there is usually the smallest number of customers with the highest value, clusters do not contain the same number of customers. Hence, there is a problem of class imbalance in the classification, which may lead to low class precision (the percentage of precisely predicted examples within a predicted class) and / or class recall (the percentage of accurately classified examples within the actual class) for the smallest class, which is the most important customer segment in this case.

In order to overcome the shortcomings mentioned above, a new method for customer segments classification based on data mining techniques will be tested in this paper. Customer segmentation by RFM attributes will be performed automatically using a clustering algorithm instead of manual coding and sorting. Clustering will be applied to the original attributes, so that there is no loss of fine differences that arise due to their uniform coding. Instead of a priori determining the number of clusters, an objective indicator of the optimal number of clusters will be used. Predictive classification will not include RFM attributes, therefore, classification rules describing segments will be defined in terms of customer and product characteristics, which is very important for customer relationship management. In this way, it is possible to classify new customers for whom RFM attributes are not known and available. Finally, and most importantly, the proposed method aims to reduce the misclassification of the most valuable customer segment.

2.4. Class Balancing in Direct Marketing

As pointed out, a major difficulty with predictive models in direct marketing is the class imbalance problem. According to the previously mentioned Pareto principle, the segment of the most valuable customers is the smallest (about 20% of the customers), but also the most important for the success of the campaign. The response rate in a direct campaign is often less than 5%, while non-responders make up as much as 95% of the total number of customers. This leads to very unbalanced datasets for training predictive classifiers in direct marketing [17, 22, 53]. Obviously, the problem of class balancing in this area is very topical, and accordingly, much more recent research deals with methods that effectively address this problem.

According to Sun et al.[31] data-level, algorithm-level and cost-sensitive solutions were developed for the problem when using imbalanced classes in classification models. At the data level, the aim is to balance classes with resampling, while solutions include random or targeted under-sampling and over-sampling. At the algorithm level, solutions try to adapt the algorithm to strengthen small-class learning. Cost-sensitive solutions, at both the algorithm and data levels, assign higher misclassification costs to small-class examples. More recently, there have been several papers that deal with this issue [54–56].

Although resampling eliminates class imbalance, this approach has several limitations and disadvantages, such as unknown optimal class distribution, inexplicit criterion in selecting examples for removal, risk of losing information relevant to class differentiation in majority class under-sampling, and risk of overfitting when over-sampling a minority class. Algorithm-level approaches require extensive knowledge of the algorithms and application domains, while cost-sensitive approaches involve extra learning costs for exploring effective cost setups, when real cost values are not available. However, despite the mentioned shortcomings, most of these solutions are used in recent research in the field of direct marketing.

Thus, Kim et al. [17] compared the efficiency of SVM classifiers with decision tree and neural networks on highly unbalanced data sets in direct marketing. They found that only SVM doesn’t have a complete misclassification of the minor class, but, that positive sensitivity is very small, which means that the class imbalance is also an issue for SVM method. With random under-sampling of majority class with a ratio of 33% (i.e. the class ratio was 2:1), all classifiers improved their performance, while SVM still outperformed the others. However, with a 1:1 class ratio, the performance of SVM model has weakened, suggesting that by removing a large number of examples of the majority class, data relevant to the learning process may be lost. In that sense, it is good to combine under-sampling with ensemble techniques so that random selection is repeated several times and the probability of significant data being completely excluded is reduced, hence some papers dealing with the class imbalance problem in direct marketing go in that direction.

For example, Kang et al. [22] suggested improving customer response models by balancing classes using clustering, under-sampling and ensemble. First, the instances belonging to the non-response class are clustered. In the next step, under-sampling is performed as part of the ensemble procedure by randomly selecting a number of representatives from each cluster, proportional to the size of the cluster, but with the total number of selected instances equal to the minor class (balanced ensemble). In this way, taking a number of representative members of the larger class is achieved and reduces the loss of information relevant to class differentiation. By performing ensemble procedure in *k* iterations, on *k* of such balanced samples, *k* classifiers are generated and their predictions are combined. The results showed that compared to random sampling methods, this approach has more stable predictive performance that decision makers can trust more.

Migueis et al. [18] compared ensemble balanced under-sampling (the EasyEnsemble algorithm that uses sampling without replacement) with an over-sampling method (the Synthetic Minority Oversampling Technique-SMOTE) for direct marketing response prediction in banking and found the EasyEnsemble method gave better results. The sampling model without replacement can compromise the independence of the classifier in the ensemble procedure because the sampling in the next step depends on the one made in the previous step.

Marinakos et al. [29] tested cluster-based under-sampling and distance-based resampling techniques for the bank customer response model (with 12% of respondents and 88% of non-respondents) with several different classifiers, such as linear discriminant analysis, logistic regression, k-Nearest Neighbor (k-NN), decision tree, neural network and SVM. The highest accuracy of the minority class classification was achieved by the combination of cluster under-sampling and k-NN. Cluster under-sampling combined with SMOTE over-sampling proved consistently well, performing across all classifiers.

Peng et al. [27] proposed a solution based on algorithm adaptation in the form of cost-sensitive learning SVM for segmenting credit card users, and showed that this solution gives better results for the smallest class of high-value users than basic SVM with random under-sampling. This approach requires extensive knowledge of the SVM method in order to include misclassifying costs.

Farquad and Bose [25]tested SVM as a class-balancing preprocessor of insurance customers data and found that when classifiers are applied to such a refined set, a much higher sensitivity is obtained i.e. the number of current examples of the minor class that the model accurately classifies. They also found that data balancing with SVM is more efficient than other balancing techniques such as 100% and 200% SMOTE over-sampling or 25% and 50% under-sampling.

In our preliminary research [57], we tested how successfully a hybrid model that combines SVM and decision trees as a rule extraction technique (SVM-DT) solves the problem of the minor class of the most valuable customers. The results showed that with this approach, the segment of the most valuable customers can be predicted with an accuracy of 77%, which is 44% better than the standalone DT. Thus, SVM as a preprocessor has effectively improved the precision of the minor class. The improvement is even higher for the percentage of existing customers who are recognized as members of the most valuable cluster. Standalone DT identified only 4% of them, while SVM-DT managed to identify 63% of such customers. Although the model performed well on a training data set (obtained by cross-validation), it was not tested on an unknown data set, so its actual predictive power was not confirmed in this study.

In a study by Djurisic et al. [58] authors tested how well SVM preprocesses data and enables CRM optimization in banks. The results showed that during the segmentation of credit card users, this method successfully resolves overlapping and unbalanced classes. In this paper either, the model was not tested on a completely unknown data set.

In previous research, in order to overcome the class imbalance problem, balanced ensemble methods in combination with different classifiers, or standalone SVM, as a preprocessor that refines class overlapping and thus balances data, were mainly tested. This paper will test combining ensemble approach and SVM to improve preprocessing performance, as well as balanced ensemble in combination with DT on such a preprocessed dataset to improve rule extraction performance from SVM output, which should ultimately lead to improved performance of customer segments classification.

3. Methodology

The primary goal of predictive customer segmentation in direct marketing is customer value prediction, which determines whether or not a customer is targeted. This section describes the methodological approach for the proposed predictive procedure.

3.1. Data

First step is collection of data on purchasing transactions from previous direct campaigns, which can include customer data (such as: gender, age, region, wealth, etc.), product data (such as: type, category, purpose, etc.), and purchasing behavior data, i.e. recency, frequency and monetary value of purchases. The data were used as a training set for the predictive model.

For the empirical testing in this paper, a data set of on-line purchasing transactions from previous direct campaigns of Sport Vision Montenegro (part of the Sport Vision system - leading sport retailer in the Balkans) was used, for the period from the beginning of September 2018 to the end of January 2019 (fall/winter season). The data set consists of 1605 records (transactions) and has the following attributes: order ID, discount, price, date, gender, product type, product gender, product category, product age and product brand. Product type represents retailer's classification of products into: footwear (sneakers, shoes, boots, etc.), apparel (t-shirts, sweatshirts, joggers, etc.) and equipment (bags, dumbbells, gloves, etc.). On the other hand, product category is another form of classification, based on activities' purpose (for example, running – for running shoes, outdoor – for hiking equipment, etc.). Product gender consists of five values: products for women, for men, for boys, for girls and unisex products. In addition, product age describes the age group that products are intended for (for babies, for kids, for adults, etc.). Finally, product brand splits the products into two major groups – A brands (retailer's distribution brands) and Licence brands (retailer's production and distribution brands) and a small group of "Other" brands. In general, A brands are well known and established sport brands, that are usually more expensive, while Licence brands are more affordable, with not as strong image and brand recognition.

The data was prepared by calculating the RFM attributes as follows: Recency as the date of the last order, Frequency as the total number of orders in the considered period and Monetary as the monetary amount spent by a customer in the considered period expressed in euros. The Recency attribute is encoded so that for 20% of the most recent dates, score 5 is assigned, the next 20% less recent dates are given score 4 and so on until score 1. Attributes Frequency and Monetary are retained in their original form. In the end, all the attributes were normalized with 0-1 range transformation. Table 1 shows the attribute distribution in the starting data set.

For the purpose of testing the predictive performance of the model, the same type of data from the year after were used, but from the same season (fall/winter), when there is a similar sales offer available for consumers. This is because of seasonality, which affects and defines type of current offer. For example, in the fall/winter season, marketing focus is on "back to school" and skiing campaigns, while during the spring/summer season, focus is on summer activities. Hence, it makes sense to only compare the performance of the same seasons and different years, while the same values of attributes are available.

The data was prepared in the same way as the training set (RFM attributes were calculated and all attributes normalized).

Table 1. Attribute distribution in the training dataset

Attribute	Statistics	Range
Order_ID		[42 ; 6278]
Cust_gend	mode = M (891), least = F (714)	F (714), M (891)
Discount	avg = 0.371 +/- 0.107	[0.000 ; 0.500]
Prod_type	mode = Footwear (784), least = Equipment (181)	Footwear (784), Equipment (181), Apparel (640)
Prod_gend	mode = For men (786), least = For girls (67)	For women (399), For boys (210), For men (786), Unisex (143), For girls (67)
Prod_categ	mode = Lifestyle (869), least = Handball (1)	Lifestyle (869), Fitness (231), Running (119), Football (70), Skiing (103), Outdoor (85), Basketball (103), Other (5), Boxing (3), Tennis (12), Accessories (2), Handball (1), Volleyball (1), Skateboarding (1)
Prod_brand	mode = A brands (853), least = Other (74)	A brands (853), Licence (678), Other (74)
Prod_age	mode = For adults (1272), least = For all (23)	For adults (1272), For babies (0-4) (62), For teens (8-14) (127), For younger kids (4-10) (121), For all (23)
R	avg = 3.143 +/- 1.353	[1.000 ; 5.000]
F	avg = 3.616 +/- 3.401	[1.000 ; 17.000]
M	avg = 100.081 +/- 78.211	[9.600 ; 352.000]

3.2. Model Training and Validation

As the first step in model development, a cluster model was generated on the training set and customer segments are identified, i.e. a Customer Value-level (CV-level) for all customers is determined.

As one of the most well-known algorithms for cluster analysis, the k-means method was mostly used for customer segmentation in direct marketing [13–16]and other clustering analysis [59, 60]. This method estimates the centroid cluster model based on the Davies-Bouldin Index (DB) [61], which ensures maximum heterogeneity between clusters and maximum homogeneity within the clusters. DB index calculates the Euclidean distance from the centroid inside and between the clusters. Better quality of clustering is indicated by lower absolute values of the DB index. This study proposes the k-means method because the optimal number of clusters can be determined based on this indicator.

CV-level defines how much the customer is valuable to the company based on purchasing behavior or belonging to the appropriate segment. Thus, the segment of customers who buy most often, who bought the most recently and from whom the largest revenue was made, represents the segment of the most valuable customers for the company. All customers who belong to that segment get the best, that is. first CV level.

In the next step, Bagging SVM is trained, as the data preprocessor. On the training dataset the CV-level is then predicted by the preprocessor. In order to obtain the purest classes possible, with less overlap, and to achieve better class balance, only those results for which more than 90% of the SVM models voted in Bagging procedure are taken, i.e. results for which Confidence is > 0.9. In this way, an under-sampled training set is obtained with a new class label predicted by Bagging SVM. The new class label defines classes that overlap less and are more balanced.

A balanced Bagging DT model is trained on this Bagging SVM output. The model is now trained on much more balanced data and the balanced ensemble meta-algorithm further helps to solve the problem of the minor class (the most valuable customer class) and improves the performance of customer segments classification. In addition, the balanced Bagging DT model extracts rules that better describe customer segments, especially the minor one. Namely, solving the problem of the minor class, significantly more rules are obtained for the most important segment of the customers.

Thus, the final model for customer segments classification was created by combining an ensemble of SVM classifiers and an ensemble of DT rule extractors, so it can be called an ensemble SVM-RE model.

By training the model, the optimal combination of model's parameters is found, which achieves maximum predictive performance. This is attained by combining Grid-Search technique with k-fold cross-validation. More specifically, a grid of possible values is defined for parameters whose combinations are tested using the k-fold cross-validation procedure with stratified sampling. The cross-validation procedure implies that the starting data set is split into subgroups, taking care that percentage of class representation in subgroups corresponds to percentages of class representation in the entire set of data. Then k-1 subsets are used for training the model (training set), while one of the subsets is used for validation, i.e., testing how this model works on an unknown set of data (validation set). The procedure is repeated k times, so that each of the subsets is a validation set. At each iteration, the parameters for classification (accuracy rate, class precision, and class recall), are calculated and finally their average value is found.

For assessment of predictive performance, overall accuracy rate, class precision and class recall are used (these indicators are explained at the end of section 2.3). In addition to predict customer segment with high accuracy, for customer targeting it is important to classify existing consumers more accurately, so class recall is an important indicator of model performance.

3.3. Model Testing

In the testing phase, to assess the accuracy of the model at the test set, the actual CV-level primarily is determined using the cluster model generated in a training phase. Then the CV-level is predicted using the trained Bagging SVM model, while the test set retains examples whose predictions have Confidence > 0.9, and the predicted CV-level now is declared as the actual class label. The trained balanced Bagging DT model is then applied to this preprocessed test set, and thus CV-level predictions are obtained.

In the testing phase, the predictive performance of the model is determined by comparing the CV-level obtained by prediction using trained models with the actual CV-level values in the test set.

3.4. Summary of Predictive Procedure

Figure 1 shows a flow diagram for the training and testing phases of the predictive procedure. The procedure was implemented using Rapid Miner.

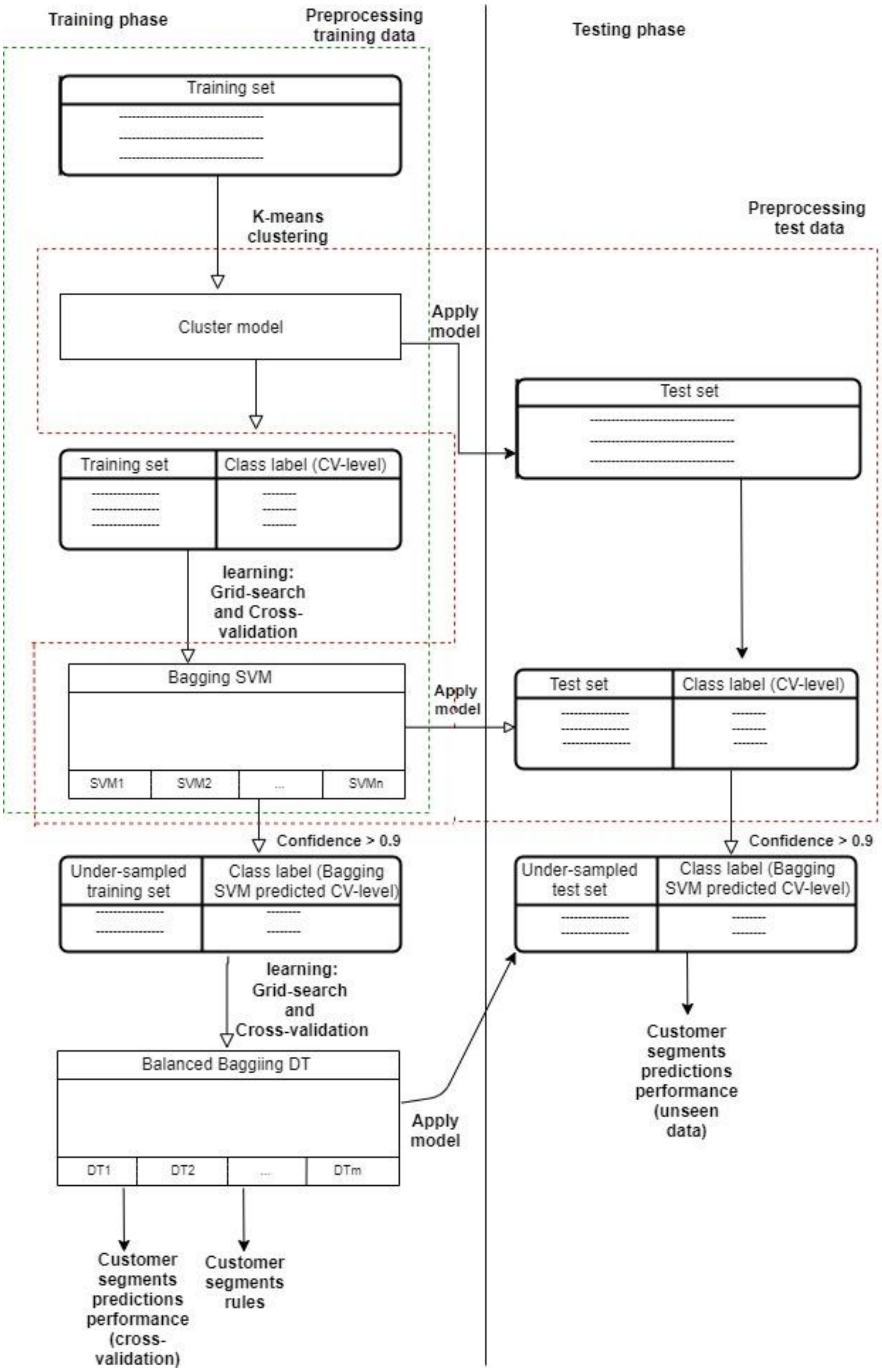

Training phase
Preprocessing training data
Testing phase
Training set
K-means clustering
Preprocessing test data
Cluster model
Apply model
Test set
Training set
Class label (CV-level)
learning: Grid-search and Cross-validation
Bagging SVM
Apply model
Test set
Class label (CV-level)
SVM1
SVM2
...
SVMn
Confidence > 0.9
Confidence > 0.9
Under-sampled training set
Class label (Bagging SVM predicted CV-level)
Under-sampled test set
Class label (Bagging SVM predicted CV-level)
learning: Grid-search and Cross-validation
Customer segments predictions performance (unseen data)
Balanced Baggiing DT
Apply model
DT1
DT2
...
DTm
Customer segments predictions performance (cross-validation)
Customer segments rules

Fig. 1. Predictive procedure

4. Empirical Testing and Results

4.1. RFM Clustering of Training Data Set

By clustering the starting data set using k-means method and normalized RFM attributes, following results are obtained - shown in Table 2. It can be seen that the best DB index (minimum absolute value) is achieved for a 3-cluster model. This cluster model is shown in Table 3.

Table 2. Selection of number of clusters (parameter k) for k-means clustering

K	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
DB	-1.025	**-0.811**	-0.983	-0.958	-0.909	-1.02	-0.98	-0.976	-0.96

Table 3. Centroid Cluster model for RFM segmentation of customers

	R	**F**	**M**	**Items**
cluster_0	0.766807	0.565126	0.677733	238
cluster_1	0.735348	0.088065	0.179499	819
cluster_2	0.137318	0.101734	0.211345	548

Note: Normalized centroid values are shown (0-1 range transformation)

From Table 3, it can be seen that cluster_0 consists of the most recent, most frequent and most profitable customers (CV-Level=1), cluster_1 consists of recent, but less frequent and less profitable customers (CV-Level=2), while cluster_2 is made of non-recent customers, that are less frequent and less profitable (CV-Level=3). The most valuable customer cluster contains significantly less items than the other two clusters (238 versus 819 and 548), so the problem of class imbalance is evident.

Table 4 shows the distribution of customer frequency, recency and profitability across these CV-level segments. It can be observed that customers from the most valuable segment are recent, as well as that they have purchased on average 10 times in the considered period and their average amount of trade is around € 242. In contrast, customers from the CV-Level 3 segment, trade on average at most 3 times, with an average trading volume of around € 82.

Table 4. CV-Level customer segments

CV-Level	**Recency**	**Frequency**	**Monetary**
1	Avg: 4 Min: 3 Max: 5	Avg: 10 Min: 3 Max: 17	Avg: 241.65 € Min: 113.4 € Max: 352 €
2	Avg: 4 Min: 3 Max: 5	Avg: 2.4 Min: 1 Max: 7	Avg: 71.06 € Min: 9.6 € Max: 199.5 €
3	Avg: 1.54 Min: 1 Max: 2	Avg: 2.6 Min: 1 Max: 9	Avg: 81.96 € Min: 12.5 € Max: 239.5 €

4.2. Empirical Testing of Predictive Procedure

Training phase

In order to test proposed predictive procedure, a Bagging SVM model for the CV-level prediction obtained by initial customer clustering was first generated. Using Grid Search parameter optimization and 10-fold cross-validation, optimal combination of parameters for SVM and Bagging is defined as: SVM.C = 400.6, SVM.gamma = 200.006, Bagging.sample_ratio = 0.9, and Bagging.iterations = 10. The model then generated a CV-level prediction which is now taken as the class label of the training set.

In the next step, an under-sampled training set was made by excluding all predictions with confidence <= 0.9, i.e. those results for which only 90% of models and less voted in the Bagging SVM procedure. Table 5 shows the thus obtained new training set.

Table 5. Changes to the training set in the predictive procedure

Training set	Class label	Distribution of class label	Number of examples
Starting	CV-Level	CV-Level 1 (238) CV-Level 2 (819) CV-Level 3 (548)	1605
Obtained at the Bagging SVM output	Bagging SVM predicted CV-Level	CV-Level 1 (132) CV-Level 2 (981) CV-Level 3 (492)	1605
Under-sampled (Conf. > 0.9)	Bagging SVM predicted CV-Level	CV-Level 1 (82) CV-Level 2 (834) CV-Level 3 (360)	1276

Table 6. Results of the training phase (cross-validation performance)

Model	Accuracy	Class Recall	Class Precision
Bagging SVM[4]	61.00%	**21.01%[1]**, 81.07%[2], 48.36%[3]	**50.51%[1]**, 61.37%[2], 62.50%[3]
Balanced Bagging DT on Bagging SVM output[5]	88.71%	**69.51%**, 96.76%, 74.44%	**80.28%**, 87.91%, 93.38%
Standalone DT[6]	60.69%	**4.20%,** 80.34%, 55.84%	**27.78%,** 61.90%, 60.47%

[1] Class performance for CV-Level 1(most valuable customers - minor class);

[2] Class performance for CV-Level 2;

[3] Class performance for CV-Level 3 ;

[4] This model is a data preprocessor;

[5] The performance of this model is actually the performance of the final model for the customer segments classification called the ensemble SVM-RE ;

[6] Standalone DT is generated for comparison purposes.

Following that procedure, the balanced Bagging DT classifier was trained on the training set thus obtained. Grid Search and 10-fold cross-validation determined the optimal combination of parameters: Bagging.sample_ratio= 0.9, Bagging.iteration =

304, balancing_proportion: 82:500:100, split_criterion = gain_ratio, min_size_for_split = 4, min_leaf_size = 2, max_depth = 15, confidence = 0.2, min_gain = 0.01.

For the purpose of comparison, the DT standalone classifier was trained with the optimal combination of parameters: split_criterion = gini_index, min_size_for_split = 4, min_leaf_size = 16, max_depth = 15, confidence = 0.1, min_gain = 0.01.

The classification performance of the trained models are shown in Table 6.

From the Table 6, it can be noticed that the balance Bagging DT on Bagging SVM output model (hereinafter ensemble SVM-RE model) has significantly better classification performance than the standalone DT model. The standalone DT method correctly targeted only 4% of the most valuable customers, while ensemble SVM-RE successfully targeted 69% of them.

Also, all considered classification performances are better with the ensemble SVM-RE model than with DT. The class precision of the most valuable customers for DT is only 28%, which means that the company will have unnecessary campaign costs for 72% of wrongly classified customers. Precision of ensemble SVM-RE model for the class is 80%, which means that only 20% of the offers sent are likely to be unanswered. Therefore, ensemble SVM-RE will, in relation to DT, reduce the cost of the campaign. It can be concluded that, with the high overall accuracy of CV-level prediction (89%), the proposed ensemble SVM-RE method managed to solve the problem of class imbalance.

The results show that Bagging SVM as the preprocessor of data on purchase transactions eliminated noise, so that more precise classification is possible. The DT classification accuracy is increased by 28% - the accuracy for the standalone DT is 61% and for SVM-RE 89%. Mean class recall for standalone DT is 47%, and after data preprocessing and using the ensemble DT it is 80%. Mean class precision has increased from 50% to 87% after preprocessing.

Given the high cross-validation accuracy of rule extraction from the Bagging SVM output (89%), the rules validly interpret the Bagging SVM classification. Table 7 shows some of the 81 derived rules which are recognized as the most important, i.e. rules which cover a large number of examples (Support ~1% and more, except for the minor class where the minimum support is 0.3%), have high accuracy (Confidence > 80%) and good confidence in relation to overall data set (Lift > 1).

On the basis of derived rules, it can be stated that customers with CV-Level = 1 are mostly male customers, who mainly buy: basketball apparel for men from licensed brands (brands for which Sport Vision has licensed production and distribution, such as: Champion, Umbro, Lonsdale, Ellesse, Slazenger, Sergio Tacchini, etc.) and with a discount of 25% to 45%; apparel for adults – men, either for football or lifestyle category from licence brands with a discount between 25% and 35%; as well as men who purchase apparel for teens, with 25-45% discount, or equipment for women with 10-45% discount.

Customers of CV-Level = 2 are mainly women, who either mainly buy clothes from A brands (brands for which the company is a distributor, such as: Adidas, Nike, Under Armor, Reebok, Converse etc.) with a discount from 25% to 35%, or apparel from licensed brands and equipment on a discount from 25% to 45%. Additionally, women who purchase footwear for men on a 25% to 35% discount also belong to this customer segment. Male buyers in this category mainly purchase lifestyle apparel for adult men, either from licensed brands on 35% to 45% discount, or A brands from 25% to 35% discount.

CV-Level = 3 represents the group of least valuable customers. The customers belonging to this group mostly buy products on a discount larger than 45% (sale seekers). Male buyers in this category mainly buy lifestyle or equipment products from A brands. Female buyers in this segment purchase lifestyle footwear for adults from licensed brands.

Hence, with the SVM-RE ensemble, rules are obtained that explain customer segments, which is the answer to the RQ3. Also, solving the problem of the minor class provides a more efficient description of the segment of the most valuable customers with a larger number of important rules.

Table 7. Most significant classification rules derived by ensemble SVM-RE

CV-level	**Rule**	**Confidence (>80%)**	**Support (>1%*)**	**Lift (>1)**
CV-Level 1	if Discount > 45% and Prod_category = Football and Prod_age = For younger kids (4-10)	100%	0.3%	8.33
CV-Level 1	if Discount > 45% and Prod_category = Outdoor and Prod_type = Footwear and Prod_brand = Licence	100%	0.4%	8.33
CV-Level 1	if Discount ≤ 45% and Discount >>35% and Cons_gender = F and Prod_type = Apparel and Prod_brand = A brands and Prod_gender = For women	100%	0.3%	8.33
CV-Level 1	if Discount ≤ 45% and Discount > 25% and Cons_gender = M and Prod_type = Apparel and Prod_age = For adults and Prod_gender = For men and Prod_category = Basketball and Prod_brand = Licence	100%	0.4%	8.33
CV-Level 1	if Discount ≤ 35% and Discount > 25% and Cons_gender = M and Prod_type = Apparel and Prod_age = For adults and Prod_gender = For men and Prod_category = Football	100%	0.4%	8.33
CV-Level 1	if Discount ≤ 45% and Discount > 35% and Cons_gender = M and Prod_type = Apparel and Prod_age = For adults and Prod_gender = For men and Prod_category = Lifestyle and Prod_brand = A brands	100%	1%	8.33
CV-Level 1	if Discount ≤ 35% and Discount > 25% and Cons_gender = M and Prod_type = Apparel and Prod_age = For adults and Prod_gender = For men and Prod_category = Lifestyle and Prod_brand = Licence	100%	4%	8.33
CV-Level 1	if Discount ≤ 45% and Discount > 25% and Cons_gender = M and Prod_type = Apparel and Prod_age = For teens (8-14)	80%	1%	6.67
CV-Level 1	if Discount ≤ 45% and Discount > 10% and Cons_gender = M and Prod_type = Equipment and Prod_brand = Licence and Prod_gender = For women	100%	1%	8.33
CV-Level 2	if Discount > 45% and Prod_category = Skiing	86%	2%	1.17
CV-Level 2	if Discount ≤ 35% and Discount > 25% and Cons_gender = F and Prod_type = Apparel and Prod_brand = A brands	100%	6%	1.37

CV-Level 2	if Discount ≤ 45% and Discount > 25% and Cons_gender = F and Prod_type = Apparel and Prod_brand = Licence	100%	11%	1.37
CV-Level 2	if Discount ≤ 45% and Discount > 25% and Cons_gender = F and Prod_type = Equipment	94%	3%	1.29
CV-Level 2	if Discount ≤ 45% and Discount > 35% and Cons_gender = F and Prod_type = Footwear	98%	7%	1.34
CV-Level 2	if Discount ≤ 35% and Discount > 25% and Cons_gender = F and Prod_type = Footwear and Prod_gender = For men	100%	2%	1.37
CV-Level 2	if Discount ≤ 45% and Discount > 35% and Cons_gender = M and Prod_type = Apparel and Prod_age = For adults and Prod_gender = For men and Prod_category = Lifestyle and Prod_brand = Licence	100%	4%	1.37
CV-Level 2	if Discount ≤ 35% and Discount > 25% and Cons_gender = M and Prod_type = Apparel and Prod_age = For adults and Prod_gender = For men and Prod_category = Lifestyle and Prod_brand = A brands	100%	3%	1.37
CV-Level 2	if Discount ≤ 45% and Discount > 10% and Cons_gender = M and Prod_type = Footwear and Prod_age = For adults and Prod_category = Lifestyle	100%	8%	1.37
CV-Level 3	if Discount > 45% and Prod_category = Basketball	88%	1%	5.83
CV-Level 3	if Discount > 45% and Prod_category = Fitness and Cons_gender = M and Prod_brand = A brands	100%	1%	6.67
CV-Level 3	if Discount > 45% and Prod_category = Lifestyle and Prod_brand = A brands and Cons_gender = M	95%	3%	6.33
CV-Level 3	if Discount > 45% and Prod_category = Lifestyle and Prod_brand = Licence and Prod_age = For adults and Prod_type = Apparel and Prod_gender = For men	100%	1%	6.67
CV-Level 3	if Discount > 45% and Prod_category = Lifestyle and Prod_brand = Licence and Prod_age = For adults and Prod_type = Footwear and Cons_gender = F	100%	1%	6.67
CV-Level 3	if Discount > 45%and Prod_category = Running and Prod_brand = A brands	91%	2%	6.06
CV-Level 3	if Discount ≤ 25% and Discount > 10% and Cons_gender = M and Prod_type = Apparel and Prod_gender = For men	100%	1%	6.67

*note: the criteria for “Support” for chosen rules is ~1% and more, except for the minor class where the minimum support is 0.3%

Testing Phase

In order to determine the predictive performance of the model on the test set, the test set was clustered using a cluster model generated in the training phase (see Figure 1). In this way, each user is assigned an appropriate CV-level to be used to determine predictive performance in the test phase.

The next step is to preprocess the test set using a Bagging SVM trained in the training phase, as well as under-sampling the test set taking examples with a class label voted by more than 90% of the SVM models during the bagging procedure (see Figure 1). The characteristics of the test set before and after preprocessing are shown in Table 8.

Table 8. Changes to the test set in the testing phase

Test set	Class label	Distribution of class label	Number of examples
Starting	CV-Level	CV-Level 1 (286) CV-Level 2 (2906) CV-Level 3 (2027)	5219
Obtained at the Bagging SVM output	Bagging SVM predicted CV-Level	CV-Level 1 (491) CV-Level 2 (3399) CV-Level 3 (1329)	5219
Under-sampled (Conf. > 0.9)	Bagging SVM predicted CV-Level	CV-Level 1 (406) CV-Level 2 (3155) CV-Level 3 (1043)	4604

It is noted that the Bagging SVM complemented the first minor segment so that it has 491 instances after data preprocessing. In addition, this preprocessor has cleared overlaps between segments so that future classification is as accurate as possible.After under-sampling, the trained ensemble SVM-RE model was applied to this test data set and the results shown in Table 9 were obtained.

Table 9. Performance of ensemble SVM-RE model on unseen data

Model	Accuracy	Class Recall	Class Precision
Ensemble SVM-RE	85.79%	**93.84%**[1], 84.44%[2], 86.77%[3]	**79.38%**[1], 94.57%[2], 69.24%[3]

[1] Class performance for CV-Level 1 (most valuable customers - minor class);[2] Class performance for CV-Level 2; [3] Class performance for CV-Level 3;

From the table above it can be seen that the overall accuracy of the ensemble SVM-RE model on unseen data is 85.79% which is good, compared to cross-validation accuracy of 88.71%. Apart from maintaining similar overall accuracy as in model validation, the unknown data also shows good performance for the minor class (class precision of about 80% and class recall of about 94%), which is the most important because the existing and predicted potential most valuable customers are precisely identified.

The ensemble SVM-RE model targets a total of 480 most valuable customers for the campaign. Of these, 381 most valuable customers were correctly targeted (94% of all such customers in the test set) and 99 customers were missed. So about 80% of the offers sent are potentially profitable while for 20% could be in vain (see the confusion matrix shown in Table A1 in the Appendix).

So, as a result of the test phase, it can be concluded that the proposed ensemble SVM-RE model is a quality predictor for CV-level, i.e. adequate model for customer segment classification, so the answer to the RQ1 is positive.

4.3. Comparison with Other Class Balancing Methods

Due to the comparison of the proposed ensemble SVM-DT model with standalone ensemble models, with respect to the efficiency of solving the minor class problem, a combination of DT classifiers with different balanced ensemble techniques were tested. First, on the starting training dataset, a Balanced Bagging DT model was generated with parameters: Bagging.sample_ratio = 0.7, Bagging.iterations = 108, balancing_proportion: 238: 238: 238, criterion = gain_ratio, min_size_for_split = 4, min_leaf_size = 2, max_depth = 15, confidence = 0.2, min_gain = 0.01. Then a balanced AdaBoost DT model with parameters: Ada-Boost.iterations = 3001, balancing_proportion: 238: 238: 238, criterion = gain_ratio, min_size_for_split = 4, min_leaf_size = 2, max_depth = 15, confidence = 0.2, min_gain = 0.01, and finally a balanced Random Forest model with parameters: RandomForest.sample_ratio = 1.0, Random.Forest.iterations = 75, balancing_proportion: 238: 238: 238, criterion = gain_ratio, max_depth = 10, are generated. The optimal parameters were determined using Grid Search and 10-fold cross-validation.

The performance of the class balancing models are shown in Table 10.

Table 10. Classification performance of standalone balanced ensemble models

Model	Accuracy	Class Recall	Class Precision
Cross-validation performance			
Balanced Bagging DT	56.69%	**44.12%**[1], 51.28%[2], 70.26%[3]	**34.20%**[1], 68.74%[2], 56.04%[3]
Balanced AdaBoost DT	55.32%	**41.60%**, 64.22%, 57.30%	**30.43%**, 69.44%, 57.46%
Balanced RandomForest	51.03%	**53.36%,** 41.76%, 63.87%	**28.93%**, 69.94%, 51.70%
Performance on test set (unseen data)			
Balanced Bagging DT	49.55%	**26.22%**, 43.53%, 61.47%	**8.35%** , 66.20%, 51.70%
Balanced AdaBoost DT	46.14%	**35.66%**, 46.73, 46.77%	**7.53%**, 65.48%, 52.93%
Balanced RandomForest	44.51%	**34.62%,** 37.89%, 55.40%	**7.12%,** 67.01%, 51.37%

[1] Class performance for CV-level 1 (most valuable customers - minor class);[2] Class performance for CV-level 2; [3] Class performance for CV-level 3

Comparing the results of ensemble SVM-RE method from Table 6 with the standalone balanced ensemble methods Bagging DT, AdaBoost DT and Random Forest in Table 10, it can be concluded that this method outperforms their capabilities in terms of class balancing i.e. solutions to minor class problems. Namely, while for the ensemble SVM-DT the recall and precision for minor class were 69.51% and 80.28% respectively, the best recall of the minor class was achieved by Random Forest (53.36%) and the best precision for the minor class by balanced Bagging DT (34.20%). Also, the maximum overall accuracy of standalone balanced ensemble models (55.69%) is significantly smaller than the ensemble SVM-DT model (88.71%). When comparing the results at the test set (Table 9), the superiority of the ensemble SVM-RE models is even more pronounced.

Finally, it can be concluded that SVM, in combination with the Bagging ensemble meta-algorithm more effectively solves class imbalance problems than other methods used for this purpose.

4.4. Comparison by Profitability Criterion

For model comparisons in terms of the potentially achievable maximum profit from a campaign based on the minor class prediction (i.e. the segment of the most valuable customers), Table 11 shows the calculation of this indicator individually by models. The profit indicator is calculated by the formula (3):

$$\textit{Profit} = \textit{True Predicted} * R - (\textit{True Predicted} + \textit{False Predicted}) * C \qquad (3)$$

where are: *True Predicted* - number of model's true predicted customers of the most valuable segment; *R*- potential single customer revenue from a campaign; *False Predicted* - number of model's false predicted customers of the most valuable segment; and *C*-estimated campaign cost per single customer.

Table 11. Model comparison by potentially earnable campaign profit

Model	True Predicted[1]	False Predicted[2]	Revenues[3]	Costs[4]	Profit[5]
SVM-RE[6]	150	44	36300	194	36106
Ensemble SVM-RE	165	41	39930	206	39724
Balanced Bagging DT	105	202	25410	307	25103
Balanced AdaBoost DT	126	288	30492	414	30078
Balanced Random Forest	127	312	30734	439	30295

[1] Number of true predicted customers of the most valuable segment
[2] Number of false predicted customers of the most valuable segment
[3] True Predicted *Potential Single Customer Revenue (€ 242)
[4] (True Predicted+False Predicted) * Estimated Single Customer Campaign Cost (€ 1)
[5] Revenues-Costs
[6] Results for standalone SVM-RE are taken from [57]

Potential revenue is assumed to be average revenue generated in previous campaigns at the most valuable segment level (€ 242, see Table 4), while the estimated cost per campaign per customer is € 1. The number of correctly predicted and incorrectly predicted members of the most valuable customer segment is given in proportion to the participation of this class in the initial training set (since the training set obtained at the Bagging SVM output is under-sampled).

Based on the calculation in the table above, it is observed that the maximum profit can be expected based on the ensemble SVM-RE prediction. The improvement of the standalone SVM-RE method by the ensemble meta-algorithm may lead to an increase in profit in campaign for € 3618. From the standalone balanced ensemble method, the highest expected profit of € 30295 is achieved with the RandomForest prediction, which is € 9429 less than the expected profit with the ensemble SVM-RE prediction.

Thus, it can be concluded that the proposed ensemble SVM-RE model out-performs other considered models by profitability criterion. Note that the advantages of the ensemble SVM-RE method according to this criterion would be even more pronounced if the comparison was done on unseen data. Since the predictive accuracy on unseen data was not tested in [57], we compared the cross-validation performance of the models.

Taking into account the comparison according to the criterion of predictive performance from the previous section, as well as based on the criterion of profitability, ensemble SVM-RE is a better class balancing method than other considered methods, which is the answer to the RQ2.

4.5. Validation of the method by testing on a public dataset

Method was also tested on a publicly available *Customer_transaction_dataset*, available on *Kaggle* data science repository (available at: https://www.kaggle.com/archit9406/customer-transaction-dataset), which consists of data regarding cycling equipment sales. The data contains 20,000 sales transactions for 3,500 customers in the period from January to December 2017. The data were refined due to missing values, leaving 19765 items in the set, which were divided into training set (70%) and test set (30%). Recency was calculated based on the date of transactions, Monetary based on the total transactions value, and Frequency as the number of transactions in this period, the same way as in the original dataset. The distribution of these and pre-existing attributes in this dataset is shown in Table A2 in Appendix.

Data were first clustered based on RFM attributes and 3 clusters were obtained as the optimal solution (minimum DB index = -0.879) (Table 12).

Table 12. Centroid Cluster Model for the public dataset

	Recency	**Frequency**	**Monetary**	**Items**
Cluster 0	0.668	0.302	0.282	4264.000
Cluster 1	0.969	0.623	0.549	7034.000
Cluster 2	1.000	0.347	0.301	8467.000

Note: Normalized centroid values are shown (0-1 range transformation)

Cluster 1 of the most valuable customers (CV-Level = 1) contains 7034 items, cluster 2 of the medium valuable customers (CV-Level = 2) has 8467 items, while cluster 0 of the least valuable customers (CV-Level = 3) in this case is the smallest and has 4264 customers. Obviously, the problem of unbalanced classes is also present here.

Repeating the same predictive procedure defined in Figure 1, in the training phase by cross-validation and the test phase by testing on an unknown dataset, the results shown in Table 13 were obtained.

Table 13. Predictive performance of the models for the public dataset

Model	Parameters	Cross-Validation Performance	Test Performance
Bagging SVM	SVM.gamma = 0.0325	accuracy: 46.15%	accuracy: 47.29%
	SVM. C = 1000.0	class recall: 21.94%[1], 44.25%[2], 59.93%[3]	class recall: 23.69%[1], 46.16%[2], 60.12%[3]
	Bagging.iterations = 10	class precis.: 33.87%, 46.65%, 49.12%	class precis.: 37.04%, 47.56%, 49.85%
	Bagging.sample_ratio = 0.8		
Ensemble SVM-RE	DT.criterion = gain_ratio	accuracy: **88.69%**	**accuracy: 90.32%**
	DT.min_size_for_split = 4		
	DT.minimal_leaf_size = 2		
	DT.maximal_depth = 15	class recall: **61.61%,** 85.50%, 93.74%	class recall: **70.07%**, 86.67%, 94.55%
	DT.confidence = 0.1		
	DT.minimal_gain = 0.01		
	Bagging.sample_ratio = 0.9	class precis.: 74.24%, 88.71%, 90.03%	class precis.: 72.03%, 91.57%, 91.41%
	Bagging.iterations = 100 Bagging.Balancig_proportion: 435:1000:2000		
Standalone DT	DT.criterion = gain_ratio	accuracy: **43.01%**	**accuracy: 43.08%**
	DT.min_size_for_split = 4		
	DT.minimal_leaf_size = 2	class recall: **0.50%,** 0.35%, 99.87%	class recall: **0.55%**, 0.33%, 100%
	DT.maximal_depth = 15		
	DT.confidence = 0.1	class precis.: 78.95%, 80.95%, 42.90%	class precis.: 100%, 100%, 42.94%
	DT.minimal_gain = 0.01		

[1] Class performance for CV-Level 3 (minor class); [2] Class performance for CV-Level 1; [3] Class performance for CV-Level 2

The data in the table above indicate that the Ensemble SVM-RE successfully solved the problem of incorrect classification of the minor class on this data set as well. On the training and test set, standalone DT completely misclassified the best customers (class recall is only about 0.3%) and the worst customers (class recall about 0.5%), and the overall accuracy of the model is about 43%. The accuracy of the Ensemble SVM-RE model on unknown data is about 90%, class recall for the best customers about 87% and for the least valuable cluster about 70%. Bagging SVM has a class recall below 50%, not only for the minor (least valuable) class, but also for the non-minor most valuable class. This means that in this data set, besides the problem of the minor class, the problem of class overlap (noise) also exists, which Bagging SVM preprocessor has solved successfully.

5. Discussion

The proposed model aimed to test several improvements of existing methods for predictive classification of customers in direct marketing, such as objective segmentation of customers with an indicator for the optimal number of clusters, description of segments in terms of customer characteristics and products, prediction of value for new customers with unknown purchasing behavior, and finally and most importantly, the reduction of misclassification for the segment of the most valuable customers, i.e. solution of class imbalance problem.

Unlike some previous studies that use hard coding of RFM attributes, sorting based on coded values and subjective selection of segments for the campaign [4, 11, 12], this study suggests a more sophisticated and objective data mining technique - k-means clustering, which achieves segmentation algorithmically using the measure of Euclidean distance, in order to provide maximum similarity within segments and difference between segments. Instead of uniform coding of RFM attributes, which does not treat the customer individually, but identifies them with the group to which they belong, which is a characteristic of many previous studies [13, 14, 16], clustering by un-coded attributes is proposed in this study, because there are numerous values with which the algorithm for clustering works smoothly. This prevents the loss of important information at the level of each individual customer, that may distinguish them from others. Unlike the method proposed in [16], which involves subjective evaluation and testing of the best number of clusters, our method determines the optimal number of clusters objectively based on the DB index, which significantly simplifies the procedure and ensures the accuracy of the model.

Classical RFM segmentation involves the prediction of future customer behavior based only on these three attributes, and is not applicable to the prospecting for new customers because transaction information is not available [4]. In [16], sophisticated data mining techniques are used during customer segmentation, but in addition to customer characteristics, RFM attributes are included as predictive attributes, so the proposed model cannot be used for new customers for whom these attributes are unknown. In our study, only product data and customer characteristics are used as predictive attributes, as it is expected to obtain predictive rules with more suitable information for targeting the potential customers [51].

Furthermore, for predictive customer classification, this study suggests the SVM-RE method in combination with ensemble techniques that enhance the predictive power of the model. The results showed that the SVM ensemble efficiently preprocesses the data, i.e. resolves the noise and class imbalance. First, by moving the margin to the nearest (and therefore most similar) examples of the larger class and classifying them into the smaller class, SVM resolves the noise in the data, i.e. class overlapping and complements the minor class with the most relevant examples. Then, by pooling the results of multiple SVMs in the ensemble procedure, the instances that join the minor class are identified more precisely (the example joins the class that has been voted the most by the SVM model). And in the end, taking only the results for which more than 90% of the SVM models voted, representatives of the classes most likely to belong to the class are selected, i.e. those that are farthest from each other and between which the margin is the widest, leading to maximum separation of classes. Applying a balanced DT ensemble for rule extraction from such pre-processed data set (SVM-RE ensemble) misclassification rate of the most valuable customer segment is reduced by 66%, which is a much better result than the result obtained in [58], where using standalone SVM preprocessor and standalone DT rule extractor this misclassification rate was reduced by 37%.

For the test set, ensemble SVM-RE method achieved Balanced Correction Rate (BCR) (rooted product of class recall of all classes) of 83%, which is 15% better than the best achieved in [22] by applying under-sampling based on clustering and ensemble techniques. Comparing the best class recall of minority class (88%), obtained in [29] using cluster-based under-sampling and k-NN classifiers, with the result achieved by our method (94%), the superiority of our model is obvious. Additionally, the class recall for majority class in [29] is low (63%), while with our method for the other two larger classes it is above 84%. In [17] the best achieved class recall for minority class is 73%, for dataset with moderate degree of class imbalance, and with random under-sampling for class ratio of 2:1, using SVM classifier, which is again lower than our score of about 94% on the test set.

Apart from the proven efficiency, automatic class balancing using the SVM-RE ensemble is less complex for practical application (there are no unknowns regarding the choice of examples to be removed, choice of optimal class ratio, etc.) compared to resampling techniques in similar studies in direct marketing [17, 18, 22, 29, 30]. Balancing the data in this way offers a stable solution that does not suffer from the sampling bias and overfitting that can occur due to resampling [31].

The ensemble SVM-RE method had a misclassification of the most valuable customer segment of about 6% at the test set, which is an excellent result. A similar result, i.e. a misclassification rate of 4% was achieved in [27], where a method based on adapting the SVM algorithm by introducing cost-sensitive learning and random under-sampling was used. However, the advantage of our method is that its application in practice does not require extensive knowledge of the SVM method required for its adaptation.

Comparing the results of standalone SVM-RE method from [57]and the ensemble SVM-RE, it can be seen that the ensemble approach was able to improve overall accuracy by 2.98%, recall for minor class by 6.81%, as well as precision for minor class by 2.83%. While these improvements seem small, considering that identifying the most valuable customers has improved by about 7% and their prediction by about 3%, it can bring about a big increase in the profits generated by the campaign (see Table 11). It should be borne in mind that once an accurately selected or predicted high-profit customer can generate more revenue in a campaign than all other customers combined. Additionally, method was not tested in unseen data in [57], hence, its true predictive power remained unexplored.

The method was additionally tested on a publicly available data set where its superiority was confirmed. The overall accuracy of classification on unknown data was improved from 43% (held by standalone DT) to 90%. The SVM-RE ensemble method at the test set had a misclassification of the most valuable customer segment of about 13%, unlike the standalone DT which had a misclassification of as much as 97% due to the overlap of this segment with the middle value customer segment. As for the problem of the minor class, i.e. least valuable customers in this case, the SVM ensemble reduced its misclassification error from 94.5% to 29% on unknown data. Thus, the method successfully solved the problem of imbalance and class overlap on the validation data set as well.

Contributions to theory/knowledge/literature

Given the above comparison and the highlighted advantages of the proposed ensemble SVM-RE method in relation to previously applied methods, it can be concluded that this study contributes to the existing theory and knowledge in the field of predictive analytics in direct marketing in several ways:

1. Instead of judgment based RFM segmentation, objective k-means based RFM segmentation and estimation of the optimal number of clusters based on DB index is proposed, which simplifies the application and guarantees higher accuracy of the model.
2. Instead of classifying customers into uniformly coded groups, clustering is performed at the level of an individual customer, thus preventing the loss of significant segmentation information.
3. Instead of RFM attributes, customer and product characteristics are used as predictors, so the method can also be used to classify unknown customers.
4. Instead of resampling or adapting the learning algorithm, it is proposed to automatically balance and remove class overlaps using the ensemble SVM method, which leads to a stable solution free of sampling bias, overfitting and extensive knowledge of the learning method by marketers.
5. Instead of preprocessing data using standalone SVM, an ensemble SVM has been proposed that increases the efficiency of balancing and class separation.
6. Instead of rule extraction using the standalone classifier, this study suggests rule extraction using the DT classifier combined with a balanced ensemble meta-algorithm which gives better predictive performance, compared to using standalone DT as the rule extractor.
7. The proposed ensemble SVM-RE method has a smaller misclassification of minority class (segment of the most valuable customers) than the standalone balanced ensemble method, as well as the methods of resampling and adaptation of algorithms used in previous studies, while maintaining high overall accuracy.
8. The proposed method extracts rules that effectively describe user segments (including the smallest one with the most valuable customers, for which rules may be omitted if the minority class misclassification is not addressed). These rules are semantically richer because they contain customer and product characteristics, and are more suitable for targeting existing and new customers in the campaign.
9. Unlike most previous studies, the method is tested on a real-life data set in this study that has not been refined and specially prepared for analysis.Then, the method was validated by testing on a publicly available dataset.

Implications for practice

In addition to the theoretical contribution, the proposed method is important for practical applications and can significantly help marketers in planning direct campaigns. A very creative and innovative offer can result in a low response rate if the targeting is not done precisely, while, on the other hand, a poorly formulated and medium creative offer to the right target group can reduce, but not eliminate, the desired consumer response [62]. Therefore, understanding the preferences and needs of consumers is a more important factor in creating a campaign, than the creative process and the way of communicating the offer. In addition, business intelligence and data mining can enhance the competitive advantage for the companies in contemporary markets [63]. This is in line with the current and ongoing trend of digital transformation in companies, conducted with the aim of keeping up with the competition [64] and improving customer experience [65].

Using the proposed model, it is possible to overcome the impersonal nature of traditional marketing, as it allows companies to treat similar groups of customers in a unique way. The benefits that this model provides to practitioners are reflected through precise targeting, minimizing message waste, and more profitable campaigns. In this way, they are enabled to objectively segment the market, adapt the content to individual segments, and to build a reliable and loyal relationship with customers. In this paper, it is shown that using ensemble SVM-RE model prediction for the most valuable segment results in the highest number of true predicted customers, as well as the lowest number of false predicted customers of that segment. In that sense, this proposed model in direct marketing practice can achieve the highest profit, compared to other considered models and reduce the waste of marketing resources.

Based on the insights from this predictive model, a more elaborate segmentation strategies can be created and more effective targeting can be applied. The rules extracted from our model enable marketers to learn about their most valuable consumers, which is of high importance, having in mind that keeping the current customers is often six to ten times more cost-effective than acquiring new ones [66, 67]. Additionally, explicit rules that describe the most valuable consumers allows for acquisition of precisely those customers that are the most similar to this group, through various targeting strategies. Hence, customers from different clusters and of different value for the company can be targeted in customized and tailored promotional activities. In other words, targeting can be conducted in an objective and precise manner, which improves the profitability of each campaign, as well as the overall effectiveness of direct marketing activities.

Another advantage for the practitioners of direct marketing is the ease of use of our method. There is no need for complex resampling procedures to be carried out, since automatic data balancing is used. Also, practitioners do not have to know the details of the learning algorithm or hire additional experts for that purpose.

6. Conclusions

In this paper an efficient method for customer classification in direct marketing is designed. The presented predictive procedure implies the classification of customer clustering. Using the k-means clustering, customers are divided into segments based on their RFM attributes (past purchasing behavior). Different clusters have different customer value levels, as well as different probability of responding to a marketing campaign. Following this procedure, customer's affiliation with one of the clusters (as well as consumer's appropriate CV-level) is predicted using the ensemble SVM-RE method, using the data on the purchased products and the customer characteristics.

The results of our empirical testing indicate that the class imbalance problem can be overcome, which improves the classification of the minority, and most valuable class.

Combining multiple SVM models with an ensemble meta-algorithm can improve data preprocessing and separate customer segments more efficiently than standalone SVM. Applying balanced ensemble classifiers on such a preprocessed training set improves the predictive indicators for the smaller class and, consequently, the effectiveness of predictive segmentation (especially for the segment of the most valuable customers), as well as the chances of making greater profits in the campaign. Combining ensemble method, based on random (with replacements) under-sampling of larger classes, i.e. on bootstrapping, with SVM data preprocessing and rule extraction, balances classes better than standalone balanced ensemble methods, in customer segment classification.

The main contribution of this study is that the proposed method better deals with the problem of class imbalance that occurs when classifying customers in direct marketing, than the methods of resampling and algorithm adaptation applied in previous papers in this field. Namely, in comparison with the previous results, a smaller misclassification of the minority class (segment of high-value customers) with high overall accuracy was achieved. The class balancing procedure is automated by data preprocessing, thus overcoming the shortcomings of previously applied methods (sampling bias, overfitting, the need for extensive knowledge of learning methods). Ultimately, application is simplified.

In addition to the scientific contribution, this study is of practical importance because the proposed method can significantly help marketers to increase the efficiency and profitability of direct campaigns and to maintain good customer relations. The results of the method can help decide if existing (new) customers should be targeted in the following direct marketing campaigns, as two key elements of the customer relationship management are customer attraction and retention [68]. This method draws out and generates classification rules, which can be used in improving relationships with existing customers and targeting new potential customers, based on their characteristics and the products offered. Ultimately, the method can notably increase the campaign revenues, as well as decrease its costs.

However, this study also has several limitations and drawbacks. First, training sets with relatively small number of instances were used in this study. For a large training set, training of SVM learners, i.e. setting the appropriate parameters, requires high computation time [33]. Secondly, as a preprocessor, SVM is combined only with Bagging, although it is possible that it would balance classes better with some other ensemble technique. Third, the proposed model was tested on only one real data set, so it is unknown what the results would be on another set with a different class distribution. Fourth, as a rule extractor from the SVM ensemble of the preprocessed dataset, only a combination of Bagging and DT classifiers was tested. The bagging technique uses random under-sampling with replacement, which may be less effective than some other techniques such as cluster-based under-sampling. Thus, the question remains whether the rule extraction would yield better results with an ensemble using cluster under-sampling as in [22, 56], or by combining ordinary cluster under-sampling with different classifiers as in [22, 29], as well as by combining some other ensemble technique (e.g. Adaptive Boosting) with different classifiers similar to [30]. In the end, the success of the data mining method largely depends on the quality of the data. The data set used here includes only some customer characteristics. By including more customer attributes, clearer rules for targeting new customers can be obtained.

In future research, this method can be tested on other data sets to verify or improve its efficiency. In this study, the method was tested in the classification of customer segments where the minority class share is about 15%. It would be interesting to test its performance in the customer response model where minority class participation can be significantly lower, even below 5%.

To extract the rules from the ensemble SVM preprocessed dataset, some other ensemble techniques could be tested, such as Random Forest, as well as algorithm-level techniques. Since cluster-based under-sampling was previously confirmed in the literature as a successful class balancing technique [22, 29], a combination of this technique could be tested as a rule extractor (independently or within an ensemble procedure) with different classifiers. Also, although in this study Bagging SVM was confirmed as a preprocessor that successfully balances data from the domain of direct marketing, in future research its results could be compared with cluster-based under-sampling on the same data set. It would be useful to test whether some other ensemble technique, such as Adaptive Boosting, in combination with SVM, would pre-process better, i.e. balance the data more effectively.

References

1. Jonker, J.J., Piersma, N., Van Den Poel, D.: Joint optimization of customer segmentation and marketing policy to maximize long-term profitability. Expert Syst. Appl. 27, 159–168 (2004). https://doi.org/10.1016/j.eswa.2004.01.010
2. Kaymak, U.: Fuzzy target selection using RFM variables. Annu. Conf. North Am. Fuzzy Inf. Process. Soc. - NAFIPS. 2, 1038–1043 (2001). https://doi.org/10.1109/nafips.2001.944748
3. Hughes, A.M.: Strategic Database Marketing: The Masterplan for Starting and Managing a Profitable, Customer-Based Marketing Program. McGraw-Hill (2005)
4. McCarty, J.A., Hastak, M.: Segmentation approaches in data-mining: A comparison of RFM, CHAID, and logistic regression. J. Bus. Res. 60, 656–662 (2007). https://doi.org/10.1016/j.jbusres.2006.06.015
5. Olson, D.L., Cao, Q., Gu, C., Lee, D.: Comparison of customer response models. Serv. Bus. 3, 117–130 (2009). https://doi.org/10.1007/s11628-009-0064-8
6. Olson, D.L., Chae, B.: Direct marketing decision support through predictive customer response modeling. Decis. Support Syst. 54, 443–451 (2012). https://doi.org/10.1016/j.dss.2012.06.005
7. Cui, G., Wong, M.L., Wan, X.: Targeting High Value Customers While Under Resource Constraint: Partial Order Constrained Optimization with Genetic Algorithm. J. Interact. Mark. 29, 27–37 (2015). https://doi.org/10.1016/j.intmar.2014.09.001
8. Kim, D., Lee, H. joo, Cho, S.: Response modeling with support vector regression. Expert Syst. Appl. 34, 1102–1108 (2008). https://doi.org/10.1016/j.eswa.2006.12.019
9. Otter, P.W., van der Scheer, H., Wansbeek, T.: Optimal selection of households for direct marketing by joint modeling of the probability and quantity of response. (2006)
10. Malthouse, E.: Ridge Regression and Direct Marketing Scoring Models. J. Interact. Mark. 13, 19–23 (1999)
11. Hughes, A.M.: Strategic database marketing: the masterplan for starting and managing a profitable, customer-based marketing program. Irwin, Chicago (1994)
12. Drozdenki, R., Drake, P.: Optimal database marketing. Sage Publications, Thousand Oaks, CA (2002)
13. Hosseini, S.M.S., Maleki, A., Gholamian, M.R.: Cluster analysis using data mining approach to develop CRM methodology to assess the customer loyalty. Expert Syst. Appl. 37, 5259–5264 (2010). https://doi.org/10.1016/j.eswa.2009.12.070

14. Sarvari, P., Ustundag, A., Takci, H.: Performance evaluation of different customer segmentation approaches based on RFM and demographics analysis. Kybernetes. 45, 1129–1157 (2016)
15. Khalili-Damghani, K., Abdi, F., Abolmakarem, S.: Hybrid soft computing approach based on clustering, rule mining, and decision tree analysis for customer segmentation problem: Real case of customer-centric industries. Appl. Soft Comput. J. 73, 816–828 (2018). https://doi.org/10.1016/j.asoc.2018.09.001
16. Cheng, C.H., Chen, Y.S.: Classifying the segmentation of customer value via RFM model and RS theory. Expert Syst. Appl. 36, 4176–4184 (2009). https://doi.org/10.1016/j.eswa.2008.04.003
17. Kim, G., Chae, B.K., Olson, D.L.: A support vector machine (SVM) approach to imbalanced datasets of customer responses: Comparison with other customer response models. Serv. Bus. 7, 167–182 (2013). https://doi.org/10.1007/s11628-012-0147-9
18. Miguéis, V.L., Camanho, A.S., Borges, J.: Predicting direct marketing response in banking: comparison of class imbalance methods. Serv. Bus. 11, 831–849 (2017). https://doi.org/10.1007/s11628-016-0332-3
19. Huang, K., Yang, H., King, I., Lyu, M.R.: Learning classifiers from imbalanced data based on biased minimax probability machine. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2, (2004). https://doi.org/10.1109/cvpr.2004.1315213
20. Wu, G., Chang, E.Y.: Class-boundary alignment for imbalanced dataset learning. ICML Work. Learn. from Imbalanced Data Sets II. 49–56 (2003)
21. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357 (2002)
22. Kang, P., Cho, S., MacLachlan, D.L.: Improved response modeling based on clustering, under-sampling, and ensemble. Expert Syst. Appl. 39, 6738–6753 (2012). https://doi.org/10.1016/j.eswa.2011.12.028
23. Dietterich, T.G.: Ensemble learning. Handb. brain theory neural networks. 2, 110–125 (2002)
24. Zhang, C., Ma, Y.: Ensemble machine learning: methods and applications. Springer Science & Business Media, Boston, MA (2012)
25. Farquad, M.A.H., Bose, I.: Preprocessing unbalanced data using support vector machine. Decis. Support Syst. 53, 226–233 (2012). https://doi.org/10.1016/j.dss.2012.01.016
26. Barakat, N., Bradley, A.P.: Rule extraction from support vector machines: A review. Neurocomputing. 74, 178–190 (2010). https://doi.org/10.1016/j.neucom.2010.02.016
27. Zou, P., Hao, Y., Li, Y.: Customer value segmentation based on cost-sensitive learning support vector machine. Int. J. Serv. Technol. Manag. 14, 126–137 (2010). https://doi.org/10.1504/IJSTM.2010.032888
28. Al-Rifaie, M.M., Alhakbani, H.A.: Handling class imbalance in direct marketing dataset using a hybrid data and algorithmic level solutions. Proc. 2016 SAI Comput. Conf. SAI 2016. 446–451 (2016). https://doi.org/10.1109/SAI.2016.7556019
29. Marinakos, G., Daskalaki, S.: Imbalanced customer classification for bank direct marketing. J. Mark. Anal. 5, 14–30 (2017). https://doi.org/10.1057/s41270-017-0013-7
30. Lawi, A., Velayaty, A.A., Zainuddin, Z.: On identifying potential direct marketing consumers using adaptive boosted support vector machine. Proc. 2017 4th Int. Conf. Comput. Appl. Inf. Process. Technol. CAIPT 2017. 2018-Janua, 1–4 (2018). https://doi.org/10.1109/CAIPT.2017.8320691
31. Sun, Y., Wong, A.K.C., Kamel, M.S.: Classification of imbalanced data: A review. Int. J. Pattern Recognit. Artif. Intell. 23, 687–719 (2009). https://doi.org/10.1142/S0218001409007326
32. Bhadani, A., Shankar, R., Vijay Rao, D.: A computational intelligence based approach to telecom customer classification for value added services. Adv. Intell. Syst. Comput. 201 AISC, 181–192 (2013). https://doi.org/10.1007/978-81-322-1038-2_16

33. Cui, D., Curry, D.: Prediction in marketing using the support vector machine. Mark. Sci. 24, 595–615 (2005). https://doi.org/10.1287/mksc.1050.0123
34. Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (2010)
35. Sanderson, M.: Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. Nat. Lang. Eng. 16, 100–103 (2010)
36. Martens, D., Huysmans, J., Setiono, R., Vanthienen, J., Baesens, B.: Rule extraction from support vector machines: An overview of issues and application in credit scoring. Stud. Comput. Intell. 80, 33–63 (2008). https://doi.org/10.1007/978-3-540-75390-2_2
37. Diederich, J.: Rule Extraction from Support Vector Machines: An Introduction. In: Diederich, J. (ed.) Rule Extraction from Support Vector Machines. Studies in Computational Intelligence, vol 80. pp. 3–31. Springer, Berlin, Heidelberg (2008)
38. Martens, D., Baesens, B., Gestel, T. Van, Vanthienen, J.: Comprehensible Credit Scoring Models Using Rule Extraction From Support Vector Machines Credit Risk Modelling , Group Risk Management , Dexia Group. Decis. Sci. 1–21
39. Quinlan, J.R.: Induction of decision trees. Mach. Learn. 1, 81–106 (1986)
40. Quinlan, J.R.: C4.5 - programs for machine learning. Kaufmann, San Mateo, CA (1992)
41. Breiman, L.: Classification and regression trees. Wadsworth International Group, Belmont, CA (1984)
42. Kašćelan, L., Kašćelan, V., Jovanović, M.: Hybrid support vector machine rule extraction method for discovering the preferences of stock market investors: Evidence from Montenegro. Intell. Autom. Soft Comput. 21, 503–522 (2015). https://doi.org/10.1080/10798587.2014.971500
43. Breiman, L.: Bagging Predictors, URL: https://link.springer.com/article/10.1007%2FBF00058655. Mach. Learn. 24, 123–140 (1996). https://doi.org/10.1007/BF00058655
44. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. J. Comput. Syst. Sci. 55, 119–139 (1997). https://doi.org/10.1006/jcss.1997.1504
45. Breiman, L.: Random Forests. Mach. Learn. 45, 5–32 (2001)
46. Gupta, A., Gupta, G.: Comparative study of random forest and neural network for prediction in direct marketing. Springer Singapore (2019)
47. Lessmann, S., Haupt, J., Coussement, K., De Bock, K.W.: Targeting customers for profit: An ensemble learning framework to support marketing decision-making. Inf. Sci. (Ny). (2019). https://doi.org/10.1016/j.ins.2019.05.027
48. Wang, C.H.: Apply robust segmentation to the service industry using kernel induced fuzzy clustering techniques. Expert Syst. Appl. 37, 8395–8400 (2010). https://doi.org/10.1016/j.eswa.2010.05.042
49. Marshall, P.: The 80/20 Rule of Sales: How to Find Your Best Customers
50. Hsieh, N.C.: An integrated data mining and behavioral scoring model for analyzing bank customers. Expert Syst. Appl. 27, 623–633 (2004). https://doi.org/10.1016/j.eswa.2004.06.007
51. Tsai, C.Y., Chiu, C.C.: A purchase-based market segmentation methodology. Expert Syst. Appl. 27, 265–276 (2004). https://doi.org/10.1016/j.eswa.2004.02.005
52. MacQueen, J.: Some methods for classification and analysis of multivariate observations. Proc. fifth Berkeley Symp. Math. Stat. Probab. 1, 281–297 (1967)
53. Bose, I., Chen, X.: Quantitative models for direct marketing: A review from systems perspective. Eur. J. Oper. Res. 195, 1–16 (2009). https://doi.org/10.1016/j.ejor.2008.04.006
54. Liu, J., Zio, E.: Integration of feature vector selection and support vector machine for classification of imbalanced data. Appl. Soft Comput. J. 75, 702–711 (2019). https://doi.org/10.1016/j.asoc.2018.11.045
55. Lopez-Garcia, P., Masegosa, A.D., Osaba, E., Onieva, E., Perallos, A.: Ensemble classification for imbalanced data based on feature space partitioning and hybrid

metaheuristics. Appl. Intell. 49, 2807–2822 (2019). https://doi.org/10.1007/s10489-019-01423-6

56. Wong, M.L., Seng, K., Wong, P.K.: Cost-sensitive ensemble of stacked denoising autoencoders for class imbalance problems in business domain. Expert Syst. Appl. 141, (2020)
57. Rogic, S., Kascelan, L.: Customer Value Prediction in Direct Marketing Using Hybrid Support Vector Machine Rule Extraction Method. Commun. Comput. Inf. Sci. 1064, 283–294 (2019). https://doi.org/10.1007/978-3-030-30278-8_30
58. Djurisic, V., Kascelan, L., Rogic, S., Melovic, B.: Bank CRM Optimization Using Predictive Classification Based on the Support Vector Machine Method. Appl. Artif. Intell. 00, 1–15 (2020). https://doi.org/10.1080/08839514.2020.1790248
59. Kramarić, T.P., Bach, M.P., Dumičić, K., Žmuk, B., Žaja, M.M.: Exploratory study of insurance companies in selected post-transition countries: non-hierarchical cluster analysis. Cent. Eur. J. Oper. Res. 26, 783–807 (2018). https://doi.org/10.1007/s10100-017-0514-7
60. Bach, M.P., Juković, S., Dumičić, K., Šarlija, N.: Business client segmentation in banking using self-organizing maps. South East Eur. J. Econ. Bus. 8, 32–41 (2014). https://doi.org/10.2478/jeb-2013-0007
61. Davies, D.L., Bouldin, D.W.: A Cluster Separation Measure. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-1, 224–227 (1979). https://doi.org/10.1109/TPAMI.1979.4766909
62. Stone, B., Jacobs, R.: Successful Direct Marketing Methods. McGraw Hill (2008)
63. Bach, M.P., Jaklič, J., Vugec, D.S.: Understanding impact of business intelligence to organizational performance using cluster analysis: Does culture matter? Int. J. Inf. Syst. Proj. Manag. 6, 63–86 (2018). https://doi.org/10.12821/ijispm060304
64. Furjan, M.T., Tomičić-Pupek, K., Pihir, I.: Understanding Digital Transformation Initiatives: Case Studies Analysis. Bus. Syst. Res. 11, 125–141 (2020). https://doi.org/10.2478/bsrj-2020-0009
65. Rekettye, G., Rekettye, G.: The Effects of Digitalization on Customer Experience. SSRN Electron. J. 340–346 (2019). https://doi.org/10.2139/ssrn.3491767
66. Colgate, M.R., Danaher, P.J.: Implementing a customer relationship strategy: The asymmetric impact of poor versus excellent execution. J. Acad. Mark. Sci. 28, 375–387 (2000). https://doi.org/10.1177/0092070300283006
67. Verbeke, W., Martens, D., Mues, C., Baesens, B.: Building comprehensible customer churn prediction models with advanced rule induction techniques. Expert Syst. Appl. 38, 2354–2364 (2011). https://doi.org/10.1016/j.eswa.2010.08.023
68. Lazar, E.: Customer Churn Prediction Embedded in an Analytical CRM Model. SSRN Electron. J. 24–30 (2018). https://doi.org/10.2139/ssrn.3281605

Ljiljana Kašćelan is a full-time professor at the Faculty of Economics at the University of Montenegro. She graduated in Computer Science at the Faculty of Natural Sciences and Mathematics, obtained a M.Sc. in Computer Science from the Faculty of Electrical Engineering and a Ph.D. in Business Intelligence from the Faculty of Economics (all at the University of Montenegro). She teaches courses in Business Informatics and Business Intelligence. Ljiljana is the author of several papers in international journals and conferences in the field of data mining and applications.

Suncica Rogić, MSc, is a teaching assistant at the Faculty of Economics at the University of Montenegro. She graduated and obtained a MSc from the same Faculty, and currently is a PhD candidate. The topics of her research interest include direct and digital marketing analytics, customer segmentation and data mining applications.

Received: May 30, 2020; Accepted: December 02, 2020.

Computer Science and Information Systems 18(3):927–955 https://doi.org/10.2298/CSIS200426017D

Real Time Availability And Consistency Of Health-Related Information Across Multiple Stakeholders: A Blockchain Based Approach

Zlate Dodevski[1], Sonja Filiposka[2], Anastas Mishev[2], and Vladimir Trajkovik[2]

[1] iBorn, Skopje, Republic of North Macedonia
zlated@iborn.net
[2] Faculty of Computer Science and Engineering,
Ss. Cyril and Methodious University,
Skopje, Republic of North Macedonia
{sonja.filiposka, anastas.mishev, vladimir.trajkovikj}@finki.ukim.mk

Abstract. Sensitivity of the health-related data and the focus on compliance and security has traditionally emphasized the need for centralized approach while implementing Electronic Health Records (EHR) systems. These one-institutional architectural designs are leading to fragmented and scattered pieces of valuable data across various data warehouses and silos. Interoperability challenges arise due to the absence of unified data management and exchange mechanisms making the social need for fundamental design changes bigger.

The capability of a distributed ledger technology and blockchain to offer immutable, decentralized and cryptographically secured record of transactions throughout a peer-to-peer network can facilitate better collaboration and increased interoperability in the field of health and insurance information exchange processes. The paper examines different approaches and application of blockchain technology and identifies which implementations of components are more suitable and beneficial for the specific eco-system analyzed in the paper.

This paper presents alternative way of dealing with information exchange across multiple stakeholders by justifying the use of decentralized approach, distributed access and solution how to comprehensively track and assemble health related data. We propose an architectural design and overview of a specific use case with focus on information exchange processes between health insurance providers and health care organizations, by using blockchain as an underlying technology.

The architectural overview and data flows, backed up by sequence diagrams from specific use cases offered in this paper, can serve as a guide to the blockchain technology adoption and initial setup.

Keywords: blockchain, health-related data, health information exchange, health insurance, decentralization

1. Introduction

The process of improving the efficiency and quality of the health information exchange is a hot topic in the last period in the field of health informatics. HIE (Health Information Exchange) represents electronic transfer of clinical and / or administrative information between different (mostly competitive) healthcare organizations [2]. When talking about the actors in this ecosystem, they can be of different size and form, including clinical centers, hospitals, laboratories, insurance companies, pharmacies, emergency centers, nursing homes, public health centers, etc. The data exchanged can differ and can be part of a wide range, from a summary of medical examinations, referrals, to laboratory results, and even medical history of specific patient. However, the data that is subject to transfer can be structured in different formats and different terminologies can be used, making the interfaces and integration layers which are part of the HIE process more complex and with high cost. Additionally, the data that is subject to a transfer is scattered across the storages of the organizations that are involved in the process, which often leads to inconsistent data handling processes and erroneous and incomplete medical history records.

Technologists and participants in the e-health industry in the recent period are seeing the blockchain technology as innovation in the attempt to improve the data sharing processes. They are seeing opportunity of using the blockchain infrastructure to create a powerful catalog of health records that references different data sources and connect the patients, healthcare providers, laboratories, researchers, health insurance organizations and many other participants in the eco-system [24].

One of the main goals of this research is to emphasize the need for improving the HIE process and investigate the benefits that HIE systems will have with adopting the blockchain technology.

Blockchain in its essence is a distributed system that stores records for specific transactions. It is a distributed ledger of peer-to-peer transactions, which are grouped in blocks that are connected between themselves. Well-defined cryptographic techniques are fundamental pieces which enable this technology and their efficiency is proven by the first application of blockchain technology in Bitcoin cryptocurrency system presented by Sathoshi Nakamoto [27]. They are the core principles that enable decentralized interactions (processes of storing, exchanging and accessing data) between each participant in the network, bypassing the need for intermediaries and regulatory bodies to acquire trust. In this distributed system, there is no need for central authority, instead of that we have records of transactions which are stored and shared between the involved participants in the network. This characteristic is a foundation for offering innovative solutions in different business use cases [26]. In the first application of blockchain technology, the Bitcoin cryptocurrency system, every participant must be familiar with every interaction in the network and every transaction needs to be verified by all participants to be successful and valid. The verification of the interactions and the distributed state of the network are the principles that enable the collaboration in system in absence of trust between members, while the global log of transactions is immutable. Other approaches that rely on distributed ledger technology and blockchain were introduced in the last period, all of them trying to solve the challenge of achieving consensus in a decentralized environment. At the same time, Blockchain is an emerging technology with unanticipated challenges and the promise of unrealized potential in healthcare [29].

During recent years, blockchain becomes emerging technology offering alternative ways of solving challenges in numerous fields, such as the finance, supply chain management, law and many more [32]. There is a trend towards patient-driven interoperability, in which health data exchange is patient-mediated and patient-driven [16]. Many research works are focusing on exploiting the possibility of decentralization offered by the blockchain as technology in real-life use cases [17].

When it comes to the healthcare and the health-related data, blockchain can help to simplify the way how the parties involved in the health care industry exchange the data and collaborate between themselves. Many research papers and practical application were introduced, with blockchain as foundation technology. All of them are trying to overcome the data sharing challenges and the friction caused by highly sensitive data scattered across different centralized infrastructures [17]. Smart contracts, as an intelligent protocol in the blockchain technology, can be exploited to automatically achieve system confidentiality, integrity, and authenticity [36]

The researchers at the MIT Media Lab have introduced MedRec as prototype system to exploit the benefits of blockchain in the healthcare. The content-management prototype has improved permission mechanism for tracking which organization is able to see which medical records and, in that way, simplify the health information exchange. MedRec utilize the blockchain infrastructure to create an immutable chain of content, supported by decentralized network. They also included the concept called "smart contract" to execute business logic on the distributed layer and to program the representation that connects patients and providers [12].

Iryo Network are trying to consolidate the electronic health records and enrich the patient experience with unified health record system. They are moving towards standardizing health-data and supporting the AI & Big Data research performed on the collected health-related data [28].

A mobile healthcare system for personal health data collection, sharing and collaboration between individuals and healthcare providers, as well as insurance companies is presented in [21]. The presented system enables user to share data with healthcare providers to seek healthcare services, and with insurance companies to get a quote for the insurance policy and to be insured.

These systems reveal the possibility for practical implementation of decentralization in the field of electronic health records systems and they are justifying the decision to incorporate distributed layer and blockchain technology.

From technical point of view, the systems are built in different ways exploiting the technologies that the creators considered to be beneficial for their use cases.

Still, there is a practical need for general analysis of the components of the blockchain technology and the approach for adopting the technology by the participants in the ecosystem. In this paper, we try to explain the technical decisions that need to be considered when this approach is incorporated, by disseminating the components and analyzing their effect. Blockchain solutions must also be adaptive to opportunities and barriers unique to different national health and innovation policy, and regulatory systems [22]. It is crucial to study how blockchain technology can support and challenge the healthcare domain for all interrelated actors (patients, physicians, insurance companies, regulators) and involved assets [19].

The paper proposes an architectural overview of decentralized information exchange system. The focus is put on the information exchange between the healthcare providers and health insurance organization. We analyze different approaches in order to offer health-related data integrity and confidentiality, authentication and permission control mechanism, flexible access control of data by different stakeholders and enriched consent mechanisms. The approach that we present in this paper leads to automatization of different processes in health insurance organizations, related to using and accessing health-related data. With the introduction of “smart contracts”, many of those processes can be regulated and put in place without a human interaction, which can significantly reduce the cost, time and friction.

The paper presents a strategy to address the benefits coming from the distributed layer and blockchain technology and shows how this approach will improve the HIE processes, with focus of sharing health and insurance data between healthcare and insurance organizations.

The paper is organized in several sections. In section 2 we introduce specific health information exchange ecosystem and the challenges that it has, where we are expressing the need for overcoming specific challenges, such as the need for unified medical history records system and better cooperation between stakeholders. In the section 3 we introduce blockchain as a technology and explain how blockchain can help to overcome challenges present in health information exchange ecosystem. In order to analyze different approaches of distributed ledger and blockchain technology and discuss which are more beneficial for our eco-system, in section 4 we present the components of the blockchain approach. In the section 5 we focus on the decision to use Hyperledger Fabric as a specific blockchain platform and in the section 6, we describe the architecture of the blockchain-based approach. This paper should be used as a guide for adopting the blockchain technology, so in section 7 we present the first steps and initial setup that need to be performed by the stakeholders in the eco-system and what they need to perform in order the approach to be useful and beneficial for them. Use case analysis of the approach is offered in the section 8, and section 9 and 10 are dealing with the challenges of the blockchain approach.

2. Health Information Exchange Eco System

The wider ecosystem based on the interchange of health-related data is spreading into multiple sectors that need constant gathering and exchange of data to successfully cope with different problems that arise due to uninformed decisions or fraud attempts [15]. The stakeholders participating in the system include all parties related to creating, storing, or using any health-related data. On the highest level these actors are divided into:

- Individual
  - Service requester and service user, the ultimate beneficiary of the provided services;
  - Personal health information provider, including information from wearable devices such as fitness trackers.
- Healthcare organizations
  - Primary, secondary, tertiary healthcare institutions;
  - Auxiliary health institutions such as laboratories;
  - Emergency healthcare;
- Medication suppliers
  - Pharmacies;
- Health Insurance Organizations
  - Social health plan;
  - Work related insurance plan;

- Life insurance plan;
- Travel insurance plan.

The process of exchanging health information refers to the secure electronic transportation of clinical health information in form which is understandable and usable to both the sender and the receiver. If we try to structure and categorize the transactions, we will ?nd up with two transaction types. The first one is sending information to some registry or other system, and the other one is requesting for and receiving data from other providers and data holders. The goal of the processes can be also divided into two categories, enriching and expanding the patient?s health record and exchanging information between well known healthcare related entities with established business collaboration. The table offers a short overview of the different transactions and processes that one can find in the exchange of health information.

Table 1. Comparison of blockchain categories.

	Goal	**Outbound transactions**	**Inbound transactions**	**Inbound transactions**
Expanding and consolidating patients health record	The goal of this process is to locate scattered patients records, enrich the medical record, aggregate with existing records in the institution and keep them for longer use.	Providers of healthcare, (primary care providers) can broadcast a summary of examination to relevant EHR systems.(if any)	Retrieving a summary of a patient's current conditions from another care provider. Retrieving medication history of patients. Retrieving information from other EHR systems or countrywide registry.	Transfer of care from one primary to another primary healthcare provider.
Exchanging information between relevant healthcare institutions	These processes are initiated by the institution and they are intended to ask for or order some activity or service from another institution. If there is an already established EHR system, the institution can expand the patient?s record with the data of inbound transactions.	Sending referrals to specialists. Ordering tests to the laboratory. Sending prescriptions to medical supplier	Retrieving reports from secondary or tertiary healthcare providers. Retrieving laboratory test results. Retrieving reports of used medical supplies	Patient is referred from a primary care provider to see a specialist.

To be able to use any health service, the individual needs to have the appropriate type of health insurance plan. Multiple health insurance plans can be issued for the individual by different health insurance providers, covering different spectrum of health services. Each health insurance plan is defined using a health insurance policy that defines the terms and conditions including the types of health services covered, the cap expenses covered, and the right to any damage premiums. These health insurance policies may overlap in some areas, in which case the claim should be covered by multiple policies simultaneously. For an example, if the individual was injured during working, the insurance claims activated should be social health, work and life insurance. On the other hand, if the health issue happened while traveling, the medical expenses should be covered by the travel and life insurance plans. The available health insurance plans define the health services that can be received from the health service providers. Based on the treatment defined by the health service providers, the individual may need additional services offered from other medication suppliers, such as pharmacies that work with or without prescriptions depending on the medication necessary.

The actual type of medication that can be acquired is not only dependent on the issued therapy, but also on the active health policy, since some drugs may be covered while others are not covered by the insurance. On the other hand, the health policies are defined based on the overall health history of the individual including all services obtained from the health service providers and medication suppliers, but also individual health data recorded by personal devices such as diet and fitness trackers, blood sugar level and heart-rate monitors.

Since the ecosystem spans over several different institutions that are relatively sparsely interconnected both horizontally and vertically, the exchange of trusted data becomes an issue of high importance.

Normally the individual is tasked with the complete process of information transfer from one institution to the other, usually in the form of printed documentation that is issued by one institution per request and provided in another to obtain the service. This process is not only error prone and tedious for execution, it is also subject to several different fraudulent activities such as false insurance claims, intentional hiding of medical records or failure to provide the most appropriate treatment due to incomplete information.

A system that can support the transparent, yet trustful and confidential, interchange of data between the institutions can help overcome a wide variety of issues.

3. How Can Blockhain Help

To discuss the novelties that the blockchain technology brings into our use case and how we can substitute the centralized model and architecture of implementation of Electronic Health Records (EHR) and Health Information Exchange (HIE) systems we need to fully understand the challenges that those systems have.

The biggest challenge is the complex nature of the health-related data. The reason why this type of data is so special is because they are valuable personal information and subject to numerous security regulations and authorization policies. The systems which are dealing with health-related data must be aware of the consequences of their abuse and that strong access and authorization management mechanism must be applied. Basically, that justifies the reason why systems that electronically manage health related data, are trying to overcome the challenge by putting the data in isolated state on physical storage that is part of the infrastructure of the organization where many security policies and authorization control processes are applied [8]. They are following the centralized approach with keeping the data secured on one place behind firewalls and strong security. Blockchain relies on strong cryptographic techniques and strong security is embedded by default when this implementation approach is used. Blockchain can help in establishing authentication, authorization and membership services through a decentralized network, improving any process that requires permission-control and security mechanisms.

Interoperability is arising as another problem, since there are many different implementations of EHR systems [23]. The process of making systems which are built and implemented differently without unified concept in mind to communicate and collaborate with each other is a difficult and complex task. The process demands the need to build integration layer and put communication protocols in place. Even then, the challenge of portability and secure transfer still exists. Distributed peer-to-peer network is the backbone of the blockchain technology. The EHR systems can have their representative peer included in the eco-system and in that way, they can easily connect and collaborate with the rest of the participants.

Bringing decentralization as a concept close to HIE and EHR systems can sound controversial [9], the appearance of the blockchain as a technology can unlock the true value of this concept. Friction that exists when data with great sensitivity as health-related data is transferred from one place to another, can be significantly reduced and eliminated when decentralization is incorporated in the solution. By putting references to health-related data on the blockchain infrastructure, all participants in the network are able to access and use them with proper permissions and in secure manner. On the other side, the owners of the health-related data can track the changes and have control of which entities should have access to those references by participating in the consensus that proves the validity of the transactions.

The blockchain technology is enabling the use of distributed ledger. The ledger of transactions, in our case the references of health-related data to different data storages, is not stored on centralized server, instead each of the participants holds a copy of the ledger. By owning a synchronized copy of the ledger, the participants in the eco-system are involved in the decision-making processes. Only with their consent, valid transaction in terms of updating or using the references of health-related data can be stored on the ledger. While the control of the centralized databases and storages are task for their owners, in a distributed network implemented by using blockchain, all the interactions to the ledger are synchronized and approved by the participants in the eco-system, eliminating the need for authority that will take care of the integrity and validity.

Blockchain, along with the decentralization concept, brings technological solutions to consolidate the context of transfer and easy access of the data. It can connect widespread particles of data, stored in the storage infrastructure of some centralized implementation and significantly reduce the cost of intermediation. When it comes to our use case, health insurance plans and coverage can be improved and automatized and taken to whole another level. Blockchain can transform the patient records into rich and expanded health history by connecting the different data storages. That rich and easily accessible data portfolio can be used as reference for insurance organizations.

Health insurance organizations can secure accurate claim coverage, they can reduce the effort of coordination between the parties involved and they can apply business logic and policies to reduce the human resource involvement.

4. Components Of The Blockchain Approach

The theoretical definitions of the blockchain, the discussions and research about the potential and opportunity of this technology and the applications in the other fields, such as financial industry, are increasing the awareness of the benefits and the impact. However, to justify the decision to use the blockchain technology we need a general basis of understanding how the infrastructure works underneath. We analyze and cover the fundamental principles and components behind the technology in order to discuss the impacts they will have on the information exchange processes between healthcare providers and insurance organizations. In the following part of the paper, we address the key components of the blockchain approach, that will serve as a reference and lead us to better understanding of the approach to our use case and eco-system.

4.1. Consensus as a Group Decision-making Process

When we are dealing with system that has an absence of central authority, particularly noteworthy is to discuss about how the involved entities can agree on validity of information and how they can agree on the decision to put that information in the distributed ledger and use them as single source of truth. In our case, with the blockchain based approach, it is necessary for the participants which are affected by the process of adding new information to evaluate and agree about the correctness of the information before that information become incorporated and immutable. In other words, there must be some dynamic way of reaching an agreement between the affected participants. They must make a decision based on different parameters, that transaction between two peers in the system is in the best interest for all participants. That general agreement and group decision-making process is defined as a consensus. The process of achieving consensus is important in the blockchain approach and will be subject of discussion in different parts of this paper.

4.2. Categories of Blockchain Approach

With the introduction of blockchain as revolutionary technology and its first real implementation in face of Bitcoin cryptocurrency system, many prototypes are trying to exploit the possibility to gain trust between participants in the system, without the special need for them to know each other. Since the blockchain technology offers a way how to overcome drawbacks of centralized approach in health-related systems, mentioned previously, we must discuss the level of decentralization that is needed [18]. Following the context of this research paper and the use case, before we start analyzing how far should we go with the decentralization approach, we need to define the categories and types of blockchain networks [6].

– The public blockchain as the name implies is shared among anyone in the world and it's open for everyone to join. They are proud representative of the idealistic way that the blockchain brings as concept and in the literature are generally considered as "fully decentralized.";

- Consortium blockchains are modification of the public blockchain, and the main difference is that pre-selected set of nodes/participants are chosen to be carriers of the consensus process instead of having every node to participate in the consensus process as in the case of public blockchains;
- The private blockchain is a blockchain where we can find entity which grants and stores permission to be part of the network. As opposite of the public blockchains these blockchains are only accessible to individuals who has the rights to use it.

Table 2. Comparison of blockchain categories.

	Main Characteristic	**Main advantage in our use case**	**Main disadvantage in our use case**
Public blockchain	Fully decentralized Permission-less peer-to-peer network Representative of the true concept of blockchain	Easy access for any participant to join the community.	Revealing valuable data to the public. There is no built-in component for managing permissions to manipulate with assets. They should be implemented by programming. High and unpredictable transaction fees Performance
Private blockchain	Presence of issuing authority that grants and stores rights of using assets	Permissioned ecosystem suitable for enterprise use cases of blockchain. Restriction on who can participate in the network. Increased performance than public scope, due to centralizing the trust authority	The approach moves away from the decentralized characteristics that blockchain offers.
Consortium blockchain	Pre-selected nodes are carriers of consensus process	Different nodes run by different stakeholders can be part of the decision making and consensus process. Offer more decentralized approach than the private blockchains	Complex hybrid approach with highly trusted entity in the private and power-consuming consensus in the public blockchain.

To explain the decision which of these categories of blockchain are more suitable for our use case, we will depict the characteristics in Table 2. To summarize, each of the categories brings certain advantages and disadvantages when incorporated in the context of our use-case, but this paper intention is to pick only the most suitable one. The decision should consider several main challenges that affect the health information exchange process [11]:

- The performance of the process;
- authorization and permission-to-use the assets due to the nature of the health information;
- involvement of different peers (but under the umbrella of specific organization) in the process of verification and validation
- the subtle closeness of the system due to the need of tightly controlled environment.

Having these challenges in mind, one blockchain scope that is most promising for fulfilling our experiment is the consortium blockchain category. It has increased performance when compared to the public blockchains, and tightly controlled permission environment which consists of multiple organizational authorities [37].

4.3. Channels of Communication and Collaboration

Channels components are responsible for defining collaboration in terms of transactions with privacy and confidentiality and their purpose is to achieve common ground for manipulating with assets provided by the participants in the network. Within the channel of collaboration, each of the participants has proven belonging to specific organization, rights and privileges to act on specific asset. Participants use the collaboration channel for updating the ledger and read or modify the assets in accordance to their rights and permissions. In our specific use case, to simplify the access management and authorization control, and additionally to establish separation of concerns, there will be two channels defined, one for the insurance assets and the other for health-related assets. The participants depending on their intention will be using both channels with different flows. The participant will still be part of the same eco-system, but the separation of channels will reduce the complexity of the data flow and allow us to build business logic tied only to specific channel.

4.4. Ledger

The ledger is component of the system, which is responsible for recording all transactions submitted by the participants in the ecosystem. The ledger consists of immutable sequenced blocks and each block contains multiple transactions. The ledger has these two characteristics in the system that we are analyzing:

- Each participant maintains a copy of the ledger.
- Each channel has only one ledger, which is used to update some asset produced by the participants in the network.

In our approach, to separate the concepts and the data flow, we have two channels of communication and collaboration, one for the insurance assets, the other one for health-related assets. Therefore, there will be two ledgers for each channel. The health-related ledger will be used for assets provided by the individuals (related to diet and valuable information from wearable or other sensor devices), health-care organizations and the medical suppliers. The insurance ledger will be used from the insurance organizations, but it will contain partial health related data and access by healthcare organizations as well. In this way, we can increase the performance of the transactions verification, since the ledgers will have as much amount of data as needed to satisfy the endorsement policy when achieving consensus.

4.5. Participation of Peers

Before we discuss about how participants can join the network and gain better overview of which blockchain category (public, private or consortium) should bring more benefits to the use case let's first analyze some of the main actors.

1. Health insurance organizations with different types of personas (multiple types of participants coming from same organization with/without the same permissions to use the distributed ledger). Different assets can be produced from the insurance organization such as, terms and conditions for specific insurance plan, the coverage options and duration of insurance plan, etc. The insurance organization will use the channel for accessing health-related ledger to query the health condition assets related to potential insurer and define the price of the insurance plans. On the other side, the healthcare organizations can collaborate with the insurance organizations for justifying or initiating claim coverage for their patients.
2. Healthcare Organizations with different types of personas (multiple types of participants coming from same organization with/without the same permissions to use the channel). Different assets can be produced from the healthcare organization such as, medical history of the patient, referrals, lab results, scanning results, diagnosis, prescriptions, etc. The healthcare organization will use the distributed ledger to consolidate all health-related data for the patients from different places and to provide seamless interoperability with other healthcare providers.
3. Individuals can benefit from this eco-system in two ways. First as patients. Blockchain based approach can bring enriched medical history and can transform the health records owned by the healthcare organization with given consent by the patient. Information can be gathered from multiple places, validated and used to make better diagnosis and analysis. Health records can be easily shared and transferred to third parties and guarantee the patient decreased friction in providing details to another healthcare organization. Another role that they can take is the health insurer. If they have active insurance coverage from a health insurance organization that is part of the system, the claim coverage can be taken to another level. The individuals can propose claim coverage and refunding based on the health-related data and insurance plan and if the parties involved agree on the validity, the process is performed automatically.
4. Medical Suppliers such as pharmacies can use the system to override the paper prescriptions (that can be easily altered and subject of a fraud) and communicate directly with the healthcare organization about the validity of the prescriptions. They can plan better, attack the forgery and fraud processes with goal to significantly reduce the cost of medical supplies.

4.6. Permission Management

In the world of electronic health information, health data is not just private secure piece of data, but also personal data related to specific patient's medical history. That's the main reason why the health-related data need to be protected against unauthorized access or corruption. Participants in the HIE process that generate health data can have confidence in the infrastructure of the blockchain system because one of the things it brings as a revolutionary technology is the automation of the data integrity. In addition, giving the possession of personal data should also define the decision-making power of who can manipulate with it. The membership management layer is a component which is responsible for defining the members of specific domain, organization and channel of collaboration and to communicate with other membership providers in order to clarify the ownership of the data. They are also responsible for access privileges, roles and permissions in regards of the context of the network and the channel of collaboration. Specific rights and membership allowance are revoked and handled by specific authorities. In our case:

1. Health Care Institution Authority
2. Health Insurance Institution Authority

Each channel of collaboration is a subject to authorization policy which is using the identity of the participant in order to establish the rights and privileges based on the membership providers. Each of the providers certificate the participant in order to gain appropriate access to the resources.

When it comes to the scope of the blockchain and how the participants can access the ecosystem, accent was put on the need for controlled permission environment and restricted accesses. Each of the organization that can produce assets in some way that are necessary for the functioning of the blockchain based approach, should provide entity or authority that can issue policies for using the assets (in any form, reading, writing or changing). When the peer that is participant in the system, has the security policy and permission to manage the assets, it automatically becomes endorser in the process of verification of validity of specific transaction and carrier of the consensus process. The consensus is important due to the fact that it must be reached in order to initiate update of the change. In our use case, one individual can bring assets to the network in form of health-related documents or personal health information data streams, only if consensus is reached with the other stakeholders, such as healthcare institutions representatives, medical suppliers, etc. One healthcare institution can see health-related information owned by other healthcare institution, only if it has permission to join the system and to read those specific assets.

4.7. Smart Contracts

The smart contracts are carriers of the business logic of the solution. They run on the peers (nodes) in an isolated environment (docker containers) and manage the assets which are hosted on the ledger (world state and blockchain part). Smart Contracts are the executors of the rules and the policies initially accepted by all the participants in the ecosystem.

The power of the blockchain solution is in the fact that each of the participants contains copy of the smart contracts and they can run them on their own ledger and after that compare the result with the results of other peers via the collaboration channel, to achieve consensus. The outcome of executing the smart contract on the ledger must be endorsed by the key participants (every participant executes it successfully in its local world state, signs the proposal and returns it back) in order to be accepted as ledger update. In our use case, the smart contracts will be used in many cases, from read-only medical history queries to decision if one claim proposal should be refunded or not. The endorsement policy which is part of the consensus mechanism is closely related to smart contracts. Every smart contract (chain code) works in concert with its endorsement policy which is specified at the time smart contract (chain code) is instanted.

5. Using Hyperledger Fabric as Blockchain Platform

The first step in designing the architecture of the decentralized platform is to select the blockchain platform which will serve as underlying technology to implement all the components that were discussed previously. The blockchain platform should satisfy the initial requirements of our use case, in most complete manner.

In Table 3, we can see the main characteristics of some of the blockchain platforms that are popular now and which effects should they bring if we are using them as platform for developing our use case [10], [34], [31], [1].

Additionally, there is an information about which consensus algorithm is used by each of the platform and in Table 4 we can notice short explanation about them and some of the crucial features that bring those algorithms to the use cases when they are applied [25], [20], [14], [35].

5.1. Performance and Scalability

As we already mentioned, due to real-time availability of information and health related data included in the system, performance is a requirement that we must consider in our use case. When it comes to Ethereum with the combination of Proof-Of-Work algorithm used for achieving consensus, we stumble upon performance problems due to heavy work and power consumption needed to sustain the permissionless and public network. The price paid in terms of performance drop for permissionless and public characteristics of the potential network is not worth in our use case, since we don't need to apply them. The performance of platform intended for private networks such as Hyperledger Fabric, with pre-defined set of carriers of consensus process, is on satisfactory level for our use case. The disadvantages of consensus algorithm that is used by Hyperledger Fabric are related to its semi-trusted environment, due to existence of permission system and the private nature of the blockchain. Performance decrease will happen if more than 20 peers are included in the consensus achieving process. [30] But, considering the initial requirements, these drawbacks of the platform and consensus algorithms are not playing huge role in our experiment.

5.2. Authorization and permission-to-use the network and assets

The Hyperledger Fabric is intended for building networks where the assets are owned and managed by a group of identifiable and verifiable institutions [26]. In our case, those institutions are healthcare and health insurance companies, as well pharmacies and medical suppliers. The reason why Hyperledger Fabric is better choice than Ethereum for our use-case is the infrastructure of permissioned network that it offers, where all the organization and peers are verified before executing transactions. They can operate in specific collaboration channel if, and only if, they have certificate issued by a membership authority. On the other-side, in Ethereum and public networks, permissions to participate don't exist, everyone can join the network [7]. By using Ethereum source code to implement network, Etherium can be used in a controlled setting as well (this is a rather common approach for Ethereum based start-ups focusing on B2B).

Table 3. Comparison of blockchain platforms

	Main Characteristic	**Smart Contract Code**	**Consensus Algorithm**	**Effects of using in our use case**
Ethereum	Built by Ethereum developers Most mature and first blockchain platform that introduce the smart contracts Permission less approach	Solidity	Proof-of-Work	Mature smart contract programming language that offers smooth development with strong community and broad documentation. Main disadvantage is the use of brute-force look-a-like consensus algorithm that is intended more for public blockchains. The performance and scalability will be the main challenges. The public scope of the platform moves away from our initial idea of closed and controlled private blockchain.
Hyperledger Fabric	Built by Linux foundation Consensus is achieved on transaction level Less mature than Ethereum Complex permission module	Go, Java	Proof-of-Stake based mechanisms Byzantine fault tolerance	The endorsement and consensus are achieved on transaction level meaning that all parties that have permissions and participate in the collaboration channel are responsible for the validity and achieving the trust. Increased focus on permissions system and membership service providers More layers of abstractions in the development process and modular architecture can be easily applied to our use case. Solves performance scalability and privacy issues, perfect for health-related systems.
R3 Corda	Permission based network with strong control on communication points Specialized for financial industry	Kotlin, Java	Transaction level consensus between the participants	Similar and complement to Hyperledger Fabric but more simplified with possibility of easy implementations of out-of-the-box functionalities. It has many unnecessary features that are not part of our initial considerations.
Sawtooth	Supports both permissioned and permission-less networks.	Supports different programming languages	Proof-of-Elapsed Time	The consensus algorithm that is introduced by this platform is not mature and not properly implemented yet. When it comes to security Sawtooth has approach based on roles and permissions
EOS.IS	Built by Block.One and came into the market as a competitor to the Etherium ecosystem. The platform raised 4 billion dollars in the initial coin offerings.	WebAssembly languages like C++, Java and Python was the	Delegated proof-of-stake model	The smart contracts can be written in C++, Java or Python. The platform does not require learning a new programming language. The platform solves a lot of issues that the other platforms experience such as problems with scalability and transaction fees. Main disadvantage is the centralized model of decision-making and achieving consensus. Same as Ethereum, the public scope of the platform moves away from our initial idea of closed and controlled private blockchain.

Table 4. Comparison of some of the most popular consensus algorithms

	Main Characteristic	**Blockchain category (permissioned/ permissionless)**	**Achieving consensus**	**Effects of increasing participants in the consensus**
Proof-Of-Work (PoW)	Fully distributed consensus mechanism The original consensus algorithm introduced by Satoshi Nakamoto. Each of ‘the participants that have the job to secure the network needs to prove their intent by doing some work (to solve a complex mathematical problem that requires huge computational power) in order to mine new blocks of transaction Huge computational power required	Permissionless	Slow	None
Proof-Of-Stake (PoS)	Opposite in the manner of exploiting computational power due to alternative approach. The algorithm is based on coin stakes that node holds to the network as a proof for creating new blocks.	Both	Fast	None
Proof-Of-Elapsed Time (PoET)	The process behind is related to waiting specific amount of time from each participant in order to mine new block The first participant that finish the waiting process is chosen to be the leader	Both	Medium	None
Byzantine Fault Tolerance	Few pre-selected nodes that forms the consortium and they are communicating with each-others to achieve consensus. Hoch transaction throughput	Permissioned	Fast	Decreased performance
Ripple Consensus Algorithm	Byzantine Fault Tolerance based Each channel has its own federated validator that sorts the messages in order to achieve trust	Permissioned	Fast	None

5.3. Achieving Consensus by Using Hyperledger Fabric

Consensus process in Hyperledger technologies, consists of three phases. First phase is called endorsement and it is closely related to “smart contract” component (called chaincode). Before the network is built and put in operating state, endorsement policy must be configured and defined. With other words, by using this endorsement policy we define which peers have rights to execute which transaction. This phase is also called the execution phase, because transactions are executed by using the smart contract layer. Depending on the endorsement policy, in this phase, transaction proposal is sent to some of the peers which are defined as endorsing peers and the channel is waiting for their response. There can be a case, as part of the separation of concerns, that peers even though are part of the blockchain network are not included in the endorsement process and specific business rules (smart contracts) are kept hidden and private from them. In our use case, that can happen when two healthcare organizations are exchanging information. In that case, all of the insurance organizations that participate in the network will not be included in the endorsement process.

The second phase of the consensus process is called ordering phase, because it involves the so-called orderer entity. The responsibility of the orderer as part of the consensus mechanism affects the order of the transactions. It's a keeper of the order and its functionality is related to making sure that each of the participant has the same order of the list of transactions on its ledger. This phase happens only if all endorsement peers signed the transaction.

The third phase comes right after the ordering of the transactions and it's called validation phase. In this phase, all the peers participate in the process because the process involves updating the ledger with new transaction. They validate the results and apply the changes in their copy of the ledger.

6. Architectural Overview

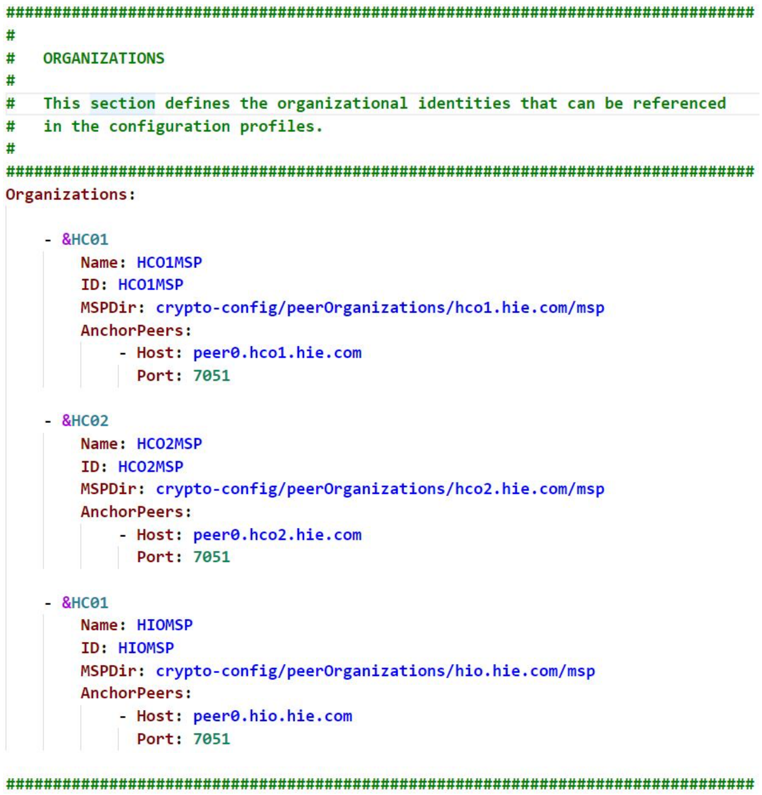
Each participant, defined as node in the network, has two layers included in the architecture. The first one is the blockchain layer, which consists of many components that we already discussed, and this layer serves as trust-less network that can provide agreements and consensus through endorsement of smart contract outcomes [28]. The second layer is the application layer and this layer is responsible for all application logic and data that is not needed to be subject to verification of validity. The application layer can interact with the blockchain through transactions. Defining the components of both layers is crucial for the architectural design [38].

6.1. Defining the Assets, Peers, Organizations and Channels

The starting point of configuring the blockchain based approach is to define the assets, peers, organizations and channels. Hyperledger Fabric technology offers possibility to configure these components with different software tools, scripts and configuration files. After initializing our network, our experiment should consist of:

- healthcare organization 1 with one peer – HCO1;
- healthcare organization 2 with one peer – HCO2;
- health insurance organization with one peer HIO
- individual represented by one peer.

After defining the peers and organizations, we need to configure the channels and their endorsement policies (Figure 1). Every participant should communicate with the others on channel that is supported by related ledger. Each of the peers are configured to be registered to certification authority server, which is responsible for granting them specific permission for performing actions in the system.



```
#################################################################################
#
#   ORGANIZATIONS
#
#   This section defines the organizational identities that can be referenced
#   in the configuration profiles.
#
#################################################################################
Organizations:

    - &HC01
        Name: HCO1MSP
        ID: HCO1MSP
        MSPDir: crypto-config/peerOrganizations/hco1.hie.com/msp
        AnchorPeers:
            - Host: peer0.hco1.hie.com
              Port: 7051

    - &HC02
        Name: HCO2MSP
        ID: HCO2MSP
        MSPDir: crypto-config/peerOrganizations/hco2.hie.com/msp
        AnchorPeers:
            - Host: peer0.hco2.hie.com
              Port: 7051

    - &HC01
        Name: HIOMSP
        ID: HIOMSP
        MSPDir: crypto-config/peerOrganizations/hio.hie.com/msp
        AnchorPeers:
            - Host: peer0.hio.hie.com
              Port: 7051

#################################################################################
```

Fig. 1. Configuration file for organization and peers

6.2. Defining the Data Flow

Considering our eco system, major architectural question that needs to be answered here is the decision which data should be placed on-chain and what should be kept off-chain. That decision will affect performance and flexibility.

Before we present the transaction and data flow, we need to discuss which data are subject to negotiating and verifying by the consensus of the network and should be stored on-chain, meaning that they will be stored in each copy of the ledger of the peers, and which data should be stored off-chain meaning that it will be referenced by the negotiation process. In blockchain approaches, there are two ways how to store the data, the first one is to add data into transactions, like Bitcoin. And the second one is to store it as variables into contract storage as Ethereum. Both ways, store and update data by submitting transactions to the blockchain layer.

What we plan to do in our use case is to find the most suitable way how to connect data stored in traditional relational databases and kept in organizational storage servers to the blockchain network. We designed a way how to establish bridge between the blockchain layer and the off-chain data, by using references, indexes, meta-data, hashes or critical information as on-chain data on the blockchain that will point to the real data needed for achieving consensus. As we mentioned, the real data can be placed somewhere in the infrastructure of the organizations that participate in the blockchain network, no matter if it's cloud solution, physical servers or a public storage.

6.3. Writing the Business Logic

As we mentioned before, the smart contracts, or with the terminology of Hyperledger Fabric, the "chaincode" is responsible for executing the business logic in the blockchain based architecture [30]. As a step towards defining the approach, we should consider developing all the necessary smart contracts that will perform the intended actions in the system. For example, how an injury can be covered by a health insurance organization or how one health care provider can grant permission for using health related data to another health care provider, or how a patient can transfer medical files from one organization to another. All these processes should be covered and implemented by using smart contracts.

6.4. Application and Integration Layer

The blockchain based architecture should contain a layer where the data present at the off-chain storages can be used as a feed to populate the blockchain network with data. Basically, all the assets that can be produced by the participants in the network should be indexed, referenced or hashed and stored to the ledger of the channel as immutable data. This process should be done as initial phase of seeding the blockchain network. The plan in our use case is to build integration layer in form of API calls. That integration layer should serve as a bridge between the organizations and individuals on one side and the web system of our use case together with the blockchain layer on the other side. Individuals can access that integration layer by using distributed application and in that way execute transaction on the blockchain, and organizations can build their own API layer in order to transfer the needed data. The overall architecture and communication can be seen at Figure 2 [39].

7. Initial Setup

After we discussed about what we want to achieve and analyzed how we want to do that and why, the next step is to bring the participants into play and assume their effort to make this blockchain based health information exchange mechanism, feasible and beneficial.

Individuals are the central point of the ecosystem and they are the ultimate beneficiaries from the services, features and functionalities that this decentralized way of health information exchange should provide [4]. Using 2 as reference, we can say that the interaction of the individuals with the blockchain solution is the distributed application. The distributed application offers the individual different types of services, from accessing their health insurance and broad information related to it, to real time access to medical history, lab results, prescriptions, reminders, etc. The distributed application act as a bridge between the individual and the blockchain layer. The individuals can initiate and demand different types of actions, such as claim approval and payment, transfer of data from one institution to another, etc. The system can also detect when specific actions are needed and can initiate them instead of the individual, asking only for a consent.

The efforts of the patient in order to keep the prototype alive and beneficial on highest level can be:

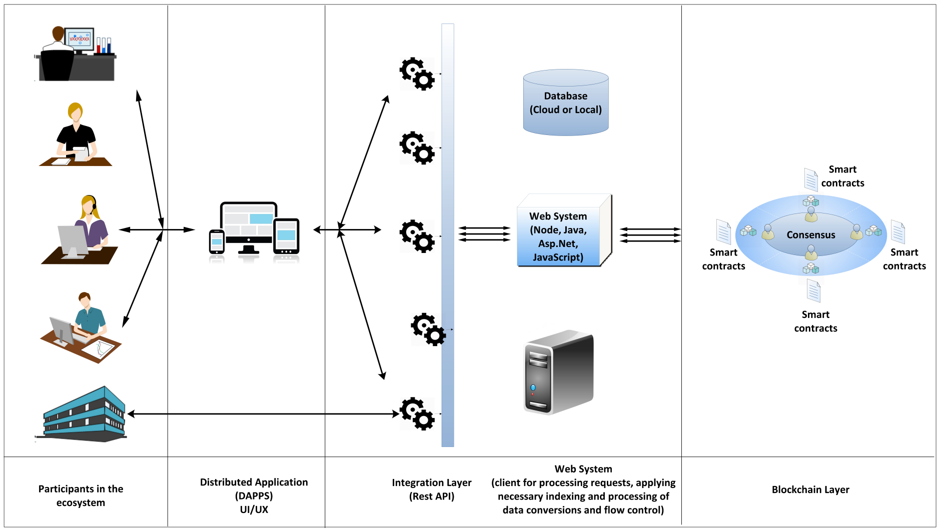


Fig. 2. Blockchain based approach

- Registering an account in the system;
- Feed the network with different types of data. Authorize the prototype to retrieve information from external sources, such as sensors of wearable devices, insert diet and meal plans, period and types of physical activity, etc.
- To seed the blockchain layer with relevant health related data, the healthcare organization that owns the medical records of that specific individual should also be part of the ecosystem. The membership management component of the healthcare organization should authenticate the individual and verify that it is the owner of the health related data. With consent of the individual, that healthcare organization can synchronize the medical records of the individual and feed the blockchain layer with them. From that moment, the individual and the healthcare organization will be always present in the endorsement process and they will always be part of the decision-making process regarding who can control and process the related data stored on the blockchain.
- To seed the blockchain layer with relevant insurance related data, the health insurance organization that owns the insurance policy should also be part of the ecosystem. The membership management component of the insurance organization should authenticate the individual and grant him permission to manipulate with its data. With consent of the individual, that healthcare organization can synchronize the terms and conditions, the duration of the insurance and details about the coverage of the individual and feed the blockchain layer with them.
- After seeding phase, the other actions are related to initiating some service provided by the prototype.

The organizations that are part of the ecosystem, no matter if they are healthcare organizations, health insurance organizations or medical suppliers have two possible ways how to bridge the collaboration gap with the blockchain layer and them. The first way is to access the blockchain layer via the interface of the distributed application. The interaction should be similar as the one which the individual is performing. The actions that can be performed by using the distributed applications are related to registering, setting up membership module and access control mechanism for other participants to ask for permissions to access, adding peers that are part of that organization and feeding the blockchain with relevant data.

To simplify the interoperability between the organizations and the blockchain based solution, the organizations can access the blockchain network directly through the integration layer that is part of the architecture. When it comes to healthcare organization, the Electronic Health Records (EHR) systems can exploit the API calls that are part of the integration layer of the blockchain based solution, to communicate in both directions. Either to retrieve health related data that is not present in the medical record of specific individual, thus enrich the medical history or to feed the blockchain with data that is relevant for the provided services of the blockchain based solution. The same can be done for the other organizations that have software enterprise solution to store and manage health related data.

8. Use Case Analysis

In the previous sections we discussed about the components of the blockchain technology and how can we exploit them in the field of electronic health records. We were considering and evaluating blockchain platforms and consensus algorithms, so we can make the right choice to fulfill the requirements of our use case and satisfy the needs of our eco system. As discussed in section IV, Hyperledger Fabric as a blockchain platform is a good starting point for the practical part of the research. In the next parts, we will analyze two use cases by using sequent diagrams and we will see how the data and the information should move across different components (represented as lifelines) and how the participants can benefit from using this approach. The user scenarios we have chosen to represent the capability of the solution we consider in this paper, focus on the exchange of health data to health insurance institutions and the use of a health insurance plan. The reason for this emphasis comes from the complexity of the workflow, which includes conditions that need to be met and the involvement of multiple actors for a particular request to be approved or a solution to be proposed by the system itself. We should keep in mind that the solution encourages facilitated communication between different entities, if the health insurance companies are not part of that entity set, then the solution will have the same architecture, but with simplified parts of the system in which the conditional logic for decision making is embedded.

8.1. Sequent Diagram for Claim Coverage Proposal Coming From Peer that Represents the Insurer

If the components of the solution described above are initialized and set, the system can execute different types of transaction and data flows, so let's analyze the process of asking for refund of claim covered by insurance plan which is issued by specific health insurance organization insurance. The transaction and data flow is depicted at Figure 3. The process covers proposal for access of health data owned by specific healthcare organization and activating endorsement policy, written in the form of a smart contracts to evaluate the truth about the health conditions and terms and coverages of the insurance plan.

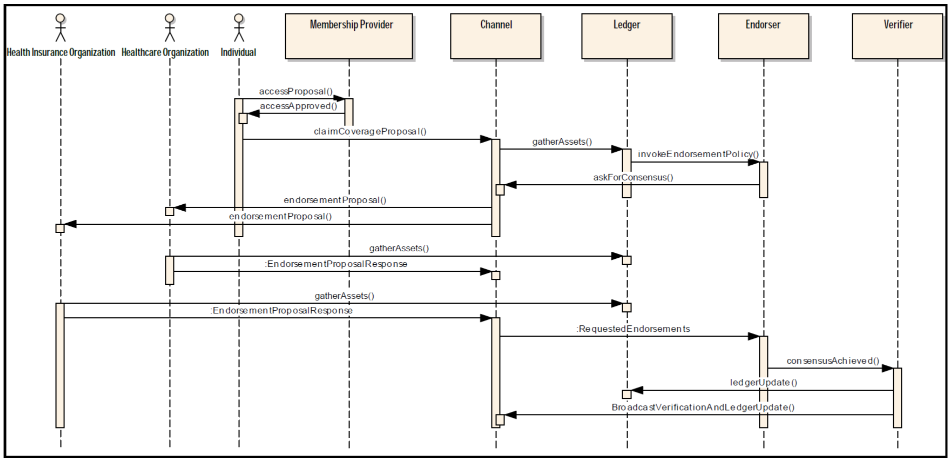


Fig. 3. Sequence diagram of claim coverage proposal

Hyperledger Fabric as a technology platform can help in couple of segments for fulfilling the requirement, as we already discussed in the previous sections. First of all, it's a platform for creating permissioned network of participants. That means that each of the members that participate in the system, should have been pre-configured as valid peers from specific organization and with proven identity. With other words Hyperledger Fabric is a platform that can offer configuration of peers and organizations and permissions that complies with data protection regulations. The healthcare providers, the insurance providers, the patients and the other participants should have their own peers as representatives in the network which will execute their business logic. If satisfying endorsements are given from both parties affected: the healthcare provider owning the health information and health insurance provider owning the insurance plan for the individual, a consensus is made and assets in the ledger are updated, meaning that the claim is reviewed, accepted and refund is approved in a smart and automatic way reducing the need for a human interaction.

When we are talking about endorsement policies, Hyperledger Fabric can offer configuring how many and what kind of combination of endorsers are required for considering one transaction as valid. That is part from the consensus algorithm that is used as a backbone for acquiring distributed trust.

Each of the participant has their own copy of the ledger and own copy of smart contracts, so the process of endorsement is nothing more than executing smart contracts from all parties involved in the process on their ledger and sending the outcome results to the channel in order to verify the truth and achieve consensus.

8.2. Health Information Exchange between Healthcare Providers

In this scenario, shown at Figure 4, an individual that has health information owned by specific healthcare organization asks for health information exchange with other healthcare organization. In other words, this sequence diagram explains the process of adding read permission to healthcare organization which is not an owner of the health information related to the individual. When the individual initiates health information exchange, the channel checks the ownership of the health information and checks which healthcare organization should have permission to read them. Endorsement is given from both parties and the verified proposal is executed as a ledger and membership provider update. From technology aspect, this flow is very similar to the previous one.

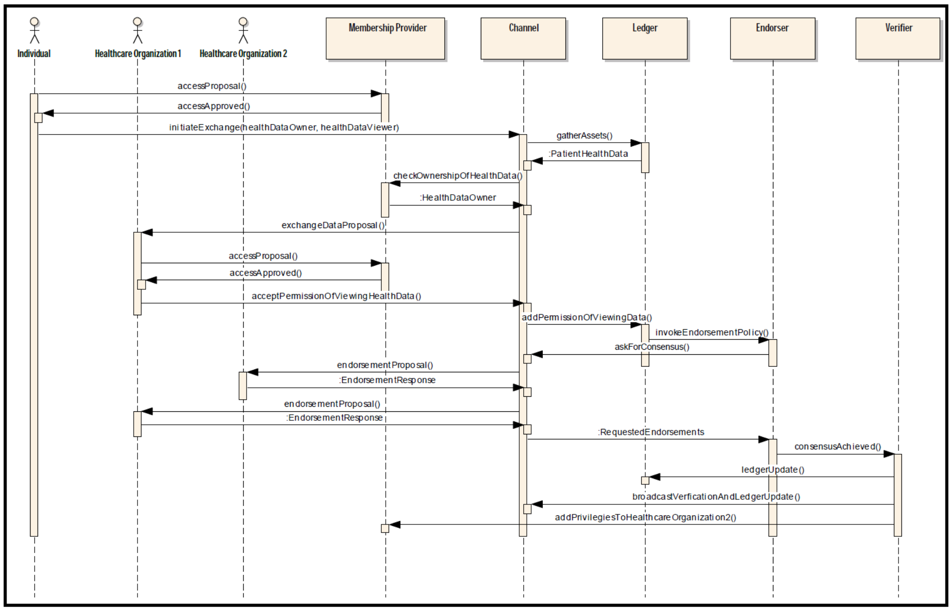


Fig. 4. Sequence diagram of adding read permission to another healthcare provider

Hyperledger Fabric offers configuration of peers and organizations and a channel where they can communicate with each other. The consensus algorithm that includes endorsement policy is helping the process to validate that one peer can have read permissions on some specific data in the blockchain.

9. GDPR

Main topic of this technical paper is the process of exchanging health-related data. When we are discussing about systems where personal data are being processed, then we must consider the compliance with the General Data Protection Regulation (GDPR) [5]. GDPR comes into play when there is some kind of "processing" of personal data. Every information that can identify, directly or indirectly, a data subject which is identifiable natural person, is considered to be personal data. The personal data that can be in form of an identification number, location data, name of the data subject, etc. The "processing" of personal data is defined as operation or set of operations, that can be automated or not, such as storing, structuring, organization, adaptation or alteration, retrieval of data, etc.

There are two very important terms that needs to be mentioned and discussed, the role of the data controller and the role of the data processor in the GDPR. The controller of the data is some entity which states the purpose and the means of the processing of personal data and that statement is determined by legal legislation or laws defined by the legal authority in the country. When there is some entity which process the personal data on behalf of the controller, then that entity is considered to be the data processor.

In the blockchain based approach of defining a system for exchanging health related data, there is no hierarchy, instead every participant is equally responsible for processing of data. In the blockchain system, each of the participant is data controller and it is obligated to comply with the GDP regulation.

One of the main principles that will come into effect with the GDPR is the demanded transparency for the processing, storing and exchanging personal information. With other words, the individual (data subject), can demand the controller of the data, information about all kind of details regarding their personal data. As we are already aware, there is an absence of central authority, so that is certainly a challenge that the blockchain based systems will face to become GDPR compliant. However, the structure of Blockchain technology brings unique possibilities to overcome these challenges and bring transparence and extended logs regarding the access to the distributed data. The transparency and tracing can be achieved by using the characteristics of the blockchain network.

The blockchain network can be used to log every access of the data by the participants in the network. That logging mechanism can serve as an immutable record of health-related data exchanges between parties and the data subject can always control and monitor where their personal information is used, and by whom.

The combination between GDPR and blockchain systems can be a subject to a lot of discussion and probably they do not fit together, keeping in mind the fact that personal information is scattered across peer-to-peer network as a part of blockchain solution. Blockchain is a distributed database with strong encryption and security mechanisms, but still individuals don't know where data is stored (distributed environment) and they don't know who manage their data. On the other side, in many ways the blockchain can be used as an ally and a partner when it comes to overcoming some of the challenges of GDPR obligations.

What blockchain can offer when GDPR comes into play is ways how to solve transparency in data portability, traceability of data usage and many other details related to the use of personal data, improved consent mechanism and management, etc. However, there are many challenges that appear because of the characteristic of the blockchain. Right to be forgotten is one of them. The Table 5 shows details about some of the GDPR obligations that affect the blockchain systems and how can be solved.

One of the most promising thing that can be done in blockchain systems regarding the GDPR compliance is removing the personal identifiable information from the stored data. That can be easily done by encrypting the indicators that identify the data subject and in that way the controller shall not be obliged to maintain, acquire or process additional information [13].

Table 5. Some of GDPR obligations seen through eyes of blockchain solution

GDPR obligations	**Main characteristics**	**Blockchain based systems**	**How can be solved**
Right to be forgotten	The data subject has the right to obtain from the controller the erasure of personal data concerning him.	Key characteristic of blockchain is immutability of stored data. It is the reason why this obligation is a challenge in the blockchain based systems.	Smart contract containing all the data subjects which triggered the right to be forgotten should forbid processing of data related to forgotten subjects. Instead of erasing the data, encrypting the personal data and erase the key used.
Transparent information and data traceability	The controller shall take appropriate measures to provide information about any form of transfer and communication relating to processing to the data subject.	One of the key characteristics of blockchain is absence of central authority which makes difficulty to track details about the controller, propose and the details of the data processing.	Implementing logging mechanism that will utilize the immutability and transparency of the blockchain network. When the personal data is used by specific controller or processor, the access is logged together with all sorts of details. That log can serve as a place for satisfying the needs for information regarding the data flow.
Consent management	Also called lawfulness of processing. The data processing is considered to be lawful if the data subject has given consent for the processing.	In the permissioned blockchain networks, such as those implemented by Hyperledger Fabric platform organizational authorities exist that grant permissions and right to access.	Smart contract can be trigged to forbid the use of data processing that is not lawful. Certification authorities in the permissioned blockchain networks can solve this problem as a condition peer to join the network.

10. Performance

The experiment that we are going to perform, as a result of this research paper, will be implemented by using the Hyperledger Fabric platform. The reason for choosing this platform is already discussed in the previous sections, so it is noteworthy to discuss the performance part since it's important for the final prototype. Hyperledger Fabric is a complex distributed system, so determining the performance will be difficult task, since many parameters can come into play. The performance can vary depending on the type of the distributed application, transaction size, implementation of the ordering service, the network, hardware on which the participant run, number of participants in the system, number of participants that are part of the consensus process, number of channels of collaboration etc [33].

Though there are tools that can measure the performance of the blockchain solution, such as Hyperledger Caliper and some measures that are present in the technical documentations such as 3500 transactions per second with latency less than one second, in this section we are going to focus more on the parameters which affect the performance [30].

When it comes to the performance, since Hyperledger Fabric offers permissioned business blockchain solution, the speed of executing the transactions and validating them through all the participants in the process of achieving consensus can be drastically better than the other implementations of blockchain technology [3]. As we already described, Fabric is using the paradigm execute-order-validate in the transaction flow, where the endorsers are separated from the ordering service giving the possibility transactions to be executed in parallel. Just for comparison, in many other blockchain platforms, such as Ethereum, the nodes must execute transaction sequentially and, in such way, decreasing the performance. Another gain from the platform is the separation of concerns by splitting the blockchain on separate channels of collaboration. Each channel has its own ledger and its own chaincode, giving a huge performance increase, since only specific nodes should involve in executing some business rule.

The constraints that affects the performance can be different and on the highest level can be separated in:

- Block size – measurement for how many transactions can be grouped in a batch which is sent to the peers to form the new block in the blockchain. To maximize the throughput, the blockchain platforms offer possibility to configure the size of the blocks. Block size should be optimized in order the prototype to have the best transaction per second trend. Some experiments with predefined hardware and network parameters, assumes that 2 MB of block size can bring to 3000 transactions per second and latency less than one second [30].
- Number of endorsers – the endorsers are the peers which are defined in the endorsement policy as the special ones which are responsible for the process of achieving consensus. They are responsible for executing process, so logically the performance should drop if the number of endorsers increase.
- Transaction size – we already mention couple of time the importance to include as less amount of data in the transaction as possible, because they affect the performance directly. Additionally, transaction in Fabric are larger because they carry identity and certification data.

11. Conclusion

In this paper, we present a use case analysis in which stakeholders related to healthcare and insurance can distribute health related data in a secure, multi-institutional and multinational way. We analyzed the components of blockchain based architecture that are crucial in prototyping such mechanism for transferring information. After thorough analysis, examination and comparison of the current trends and platforms, we are presenting a combination of approaches related to blockchain technology, that are suitable and we can incorporate it in the overall architecture. The combination consists of development platform that will create private network which is satisfying the needs of our scope, execute-order-validate endorsement policy for enforcing the business rules of the use case, consensus algorithm that offers satisfying performance and scalability to the architecture and membership mechanism that will control the access to the network.

The architecture of the approach depicts and explains how a distributed layer technology can fit into existing Electronic Health Records (EHR) systems or Insurance Content Management systems, bridging the gap between the different implementations and reduce the friction of data distribution and obtaining the best value and performance in such environment.

The paper shows a clear roadmap of the actions and steps that one participant in the system, no matter if it's individual, healthcare provider or insurance company, need to perform to become part of such decentralized health information exchange system. Additionally, it explains the initial setup and efforts that one health care or insurance institution should perform to adopt this alternative approach of health information exchange and the benefits that they will gain from it.

References

1. Ethereum vs Hyperledger. Blockchain Training Alliance (2018), https://blockchaintrainingalliance.com/blogs/news/ethereum-vs-hyperledger
2. What is HIE? — HealthIT.gov. Healthit.gov (2018), https://www.healthit.gov/topic/health-it-and-health-information-exchange-basics/what-hie
3. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference. p. 30. ACM (2018)
4. Bell, L., Buchanan, W.J., Cameron, J., Lo, O.: Applications of blockchain within healthcare. Blockchain in Healthcare Today (2018)
5. Boban, M.: Digital single market and eu data protection reform with regard to the processing of personal data as the challenge of the modern world. Economic and social development: book of proceedings p. 191 (2016)
6. Buterin, V.: On public and private blockchains. Ethereum blog 7 (2015)
7. Cachin, C.: Architecture of the hyperledger blockchain fabric. In: Workshop on Distributed Cryptocurrencies and Consensus Ledgers. vol. 310 (2016)
8. Cardon, D.: Healthcare Databases: Purpose, Strengths, Weaknesses. Health CatalystURL (2018), https://downloads.healthcatalyst.com/wp-content/uploads/2014/08/Healthcare-Databases-Purpose-Strengths-Weaknesses.pdf
9. da Conceição, A.F., da Silva, F.S.C., Rocha, V., Locoro, A., Barguil, J.M.: Eletronic health records using blockchain technology. arXiv preprint arXiv:1804.10078 (2018)
10. Diedrich, H.: Ethereum: Blockchains, digital assets, smart contracts, decentralized autonomous organizations. Wildfire Publishing Sydney (2016)
11. Dixon, B.: Health Information Exchange: Navigating and Managing a Network of Health Information Systems. Academic Press (2016)
12. Ekblaw, A., Azaria, A., Halamka, J.D., Lippman, A.: A case study for blockchain in healthcare:"medrec" prototype for electronic health records and medical research data. In: Proceedings of IEEE open & big data conference. vol. 13, p. 13 (2016)

13. Fabiano, N.: Internet of things and blockchain: Legal issues and privacy. the challenge for a privacy standard. In: Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2017 IEEE International Conference on. pp. 727–734. IEEE (2017)
14. Gervais, A.: On the Security, Performance and Privacy of Proof of Work Blockchains. Ph.D. thesis, ETH Zurich (2016)
15. Goldschmidt, P.G.: Hit and mis: implications of health information technology and medical information systems. Communications of the ACM 48(10), 68–74 (2005)
16. Gordon, W.J., Catalini, C.: Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. Computational and structural biotechnology journal 16, 224–230 (2018)
17. Greenspan, G.: Four genuine blockchain use cases — MultiChain. Multichain.com [Online]. Available (2018), `https://www.multichain.com/blog/2016/05/four-genuine-blockchain-use-cases/`
18. Guegan, D.: Public blockchain versus private blockhain (2017)
19. Kassab, M.H., DeFranco, J., Malas, T., Laplante, P., Neto, V.V.G., et al.: Exploring research in blockchain for healthcare and a roadmap for the future. IEEE Transactions on Emerging Topics in Computing (2019)
20. Krawisz, D.: The proof-of-work concept. Satoshi Nakamoto Institute¡ http://nakamotoinstitute.org/mempool/the-proof-of-work-concept/# selection-17.15-17.19 (2013)
21. Liang, X., Zhao, J., Shetty, S., Liu, J., Li, D.: Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In: 2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC). pp. 1–5. IEEE (2017)
22. Mackey, T., Bekki, H., Matsuzaki, T., Mizushima, H.: Examining the potential of blockchain technology to meet the needs of 21st-century japanese health care: viewpoint on use cases and policy. Journal of medical Internet research 22(1), e13649 (2020)
23. McDonald, C.J.: The barriers to electronic medical record systems and how to overcome them. Journal of the American Medical Informatics Association 4(3), 213–221 (1997)
24. Mettler, M.: Blockchain technology in healthcare: The revolution starts here. In: 2016 IEEE 18th International Conference on e-Health Networking. Applications and Services (Healthcom (2016)
25. Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., Qijun, C.: A review on consensus algorithm of blockchain. In: Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on. pp. 2567–2572. IEEE (2017)
26. Mougayar, W.: The business blockchain: promise, practice, and application of the next Internet technology. John Wiley & Sons (2016)
27. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2009)
28. Network, I.: IRYO. Global participatory healthcare ecosystem. URL (2017), `https://iryo.io/iryo_whitepaper.pdf`
29. Ribitzky, R., St Clair, J., Houlding, D.I., McFarlane, C.T., Ahier, B., Gould, M., Flannery, H.L., Pupo, E., Clauson, K.A.: Pragmatic, interdisciplinary perspectives on blockchain and distributed ledger technology: paving the future for healthcare. Blockchain in Healthcare Today 1, 1–15 (2018)
30. Scherer, M.: Performance and scalability of blockchain networks and smart contracts (2017)
31. Schueffel, P.: Alternative distributed ledger technologies blockchain vs. tangle vs. hashgraph-a high-level overview and comparison (2017)
32. Swan, M.: Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc." (2015)
33. Thakkar, P., Nathan, S., Vishwanathan, B.: Performance benchmarking and optimizing hyperledger fabric blockchain platform. arXiv preprint arXiv:1805.11390 (2018)
34. Valenta, M., Sandner, P.: Comparison of ethereum, hyperledger fabric and corda. Tech. rep., FSBC Working Paper (2017)

35. Vukolic, M.: The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In: International Workshop on Open Problems in Network Security. pp. 112–125. Springer (2015)
36. Wang, R., Liu, H., Wang, H., Yang, Q., Wu, D.: Distributed security architecture based on blockchain for connected health: Architecture, challenges, and approaches. IEEE Wireless Communications 26(6), 30–36 (2019)
37. Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., Pautasso, C., Rimba, P.: A taxonomy of blockchain-based systems for architecture design. In: Software Architecture (ICSA), 2017 IEEE International Conference on. pp. 243–252. IEEE (2017)
38. Zheng, Z., Xie, S., Dai, H.N., Wang, H.: Blockchain challenges and opportunities: A survey. Work Pap.–2016 (2016)
39. Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H.: An overview of blockchain technology: Architecture, consensus, and future trends. In: Big Data (BigData Congress), 2017 IEEE International Congress on. pp. 557–564. IEEE (2017)

Zlate Dodevski is the Chief Operations Officer at iborn.net. He received his MSc in Computer Science, in the field of Intelligent Information Systems, in 2019 at the Faculty of Computer Science and Engineering (FCSE), under the Ss. Cyril and Methodius University in Skopje, Macedonia. Zlate has an extensive background in blockchain technologies, decentralization of systems and secure management of Health Information Exchange. His current research interests are focused on topics such as system security, cryptography, and distributed architectures.

Sonja Filiposka is a full professor at the Faculty of Computer Science, Ss. Cyril and Methodius University in Skopje. Since obtaining her PhD in 2009 from the Faculty of Electrical Engineering and Information Technologies she has been actively taking part in a number of research projects related to e-infrastructure, networking and ICT education. During her professional carrier she has authored over 100 research papers published in conference proceedings and journals. Her main research fields of interest include e-services, orchestration of systems, complex networking and security.

Anastas Mishev, PhD, is a professor at the Faculty of Computer Science and Engineering at UKIM. He obtained his PhD in Computer Science in 2009. The focus of his research is infrastructures for collaborative computing and research, primarily Grid and High-Performance Computing systems. His aim is to get these systems closer to all potential users, mainly the research communities, in order to fully use their enormous potential. He researched in the areas of computer architectures and networks, software engineering, Internet technologies and e-learning, and is co-author of over 70 scientific papers published in international journals and proceedings of conferences. He has participated in the implementation of over 30 international projects funded by TEMPUS, PHARE, DAAD and FP programs, targeting the development of IT infrastructure and IT education.

Vladimir Trajkovik received Ph.D. degree in 2003 from Ss. Cyril and Methodius University in Skopje. He joined the Ss. Cyril and Methodius University, Skopje, R N. Macedonia, in December 1997. His current position is a professor at the Faculty of Computer Science and Engineering. He has published in more than 50 respectable journals and more than 150 conference papers. His research interests include: Information Systems Analyses and

Design, Distributed Systems, ICT based Collaboration Systems and Mobile services with special focus on two areas: Connected Health and ICT in Education.

Received: April 26, 2020; Accepted: December 12, 2020.

Computer Science and Information Systems 18(3):957–978 https://doi.org/10.2298/CSIS200920053R

Predicting Dropout in Online Learning Environments

Sandro Radovanović[1], Boris Delibašić[1], and Milija Suknović[1]

[1] University of Belgrade - Faculty of Organizational Sciences
11000 Belgrade, Serbia
{sandro.radovanovic, boris.delibasic, milija.suknovic}@fon.rs

Abstract. Online learning environments became popular in recent years. Due to high attrition rates, the problem of student dropouts became of immense importance for course designers, and course makers. In this paper, we utilized lasso and ridge logistic regression to create a prediction model for dropout on the Open University database. We investigated how early dropout can be predicted, and why dropouts occur. To answer the first question, we created models for eight different time frames, ranging from the beginning of the course to the mid-term. There are two results based on two definitions of dropout. Results show that at the beginning AUC of the prediction model is 0.549 and 0.661 and rises to 0.681 and 0.869 at mid-term. By analyzing logistic regression coefficients, we showed that at the beginning of the course demographic features of the student and course description features are the most important variables for dropout prediction, while later student activity gains more importance.

Keywords: Education Data Mining, Learning Analytics, Dropout prediction, Lasso, and Ridge Logistic Regression.

1. Introduction

Over the past few decades, education systems had trouble responding to market requirements. Namely, skills and knowledge needed for the industry include technologies that are just developed, thus leaving educational systems no time for full curriculum and syllabus development which could blend into existing study programs. European Commission recognized the problem and developed a term called short cycles of education that are intended for people who want to learn a specific subject, without studying the whole study program [5]. This way students or interested parties can participate and obtain needed knowledge for the task at hand. However, short cycles of education required in house training or supervision of the student which discouraged many of students or professionals. For example, students had to be physically present at the teaching center or they had to study only during the classes. The full bloom of short cycles of education is noted with the development of Massive Open Online Courses (MOOCs) which allows access to learning materials from worldwide renowned Universities and professors on a variety of subjects [29]. Using MOOC platforms one can tailor a learning path to its preferences. Additionally, the learning path can be achieved at any course order, at any pace, without being present and often free of charge [8]. These benefits attracted a lot of students and professionals.

However, newly founded flexibility of learning which includes a diversity of subjects and ease of access to learning materials triggers new problems not observed in traditional learning environments. The major problem in MOOCs is a low percentage of students finishing courses. This phenomenon is called dropout and is defined as a student that unenrolled from course materials before the formal end of the course [44] or a student failing to obtain a passing grade for the course [45]. Although some of the students enroll in the online course just to obtain learning materials, some students interacted with a learning environment, i.e. listened to the lectures, read additional materials, tried quizzes and assignments, and did not obtain enough points to obtain a certificate of accomplishment. Reasons for failing the course can be insufficient background knowledge, lack of time, course design, or one felt discouraged, frustrated, or bored [17].

A lot of research efforts by academia and course providers are invested in answering the question of how and why students dropout. This area of research is studied under a wider discipline called Learning Analytics or Educational Data Mining [35]. Application of data mining or machine learning to education domain is needed [36] because the lack of “negative samples”, i.e. due to the fact that majority of the students are considered as a dropout and that there are a large number of students which in classical statistical analysis results in significant impacts even if it is not. Another issue is a large volume of unstructured data. Although unstructured data is potentially very informative, one must put them into a structure and derive attributes that describe student behavior in a learning environment. The third problem is data variance which is the result of self-paced learning. Namely, students can have many different learning styles which all result in a certificate of accomplishment. One student can interact with the learning environment on a regular basis, do assignments and quizzes, while others can just take assignments, another can just listen to lectures, or some can download learning materials and listen to them on their computer. Each student may finish the course, but they all had different behavior leading to it. This poses a problem to traditional statistical testing so machine learning methods are considered as an appropriate approach. Also, the point of interest that classical statistical analysis fails to address is error analysis. Namely, the course designer wants an analytical model that has a low number of false-negative students, i.e. course designer wants to identify everyone who will fail to pass an exam and contact them as early as possible. This can come with a cost of false alarms (students who are identified as students who are going to fail an exam, but they are going to pass the exam).

In this paper, we used the Open University Learning Analytics dataset [28] to develop models for student dropout prediction. Open University dataset contains 22 courses with different behavior of the student, i.e. interaction with the learning environment (watching videos, reading materials, etc.), scores on quizzes and assignments, and historical enrollments. Due to multiple definitions of dropout, we use two experimental setups with two definitions of dropout (both will be called under umbrella term dropout), one presenting prediction model for students who fail to pass an exam or unenroll from the course (i.e. unenrolled from the course or student did not achieved enough points for the certificate), and the other presenting prediction model for students who unenrolled (i.e. unenrolled from the course). The goal of this paper is to identify how early we can predict dropout. Therefore, besides having two prediction models we will try to predict dropout as early as possible. The dropout models are produced with logistic regression as an algorithm because it provides interpretable models and because it is very suitable to work with datasets with a lot of attributes, and because it tries to fit the distribution of classes (dropouts and successful candidates) in the predictive model respecting the conditional distributions of all attributes w.r.t. the class attribute. To add, coefficients of logistic regression can be interpreted in terms of the logarithm of odds of dropout which can be seen as either a positive or negative influence on dropout.

The contributions of the paper are solving the problem based on two definitions of dropout. Namely, the majority of the papers utilize one definition of the problem which is easier to solve (student will fail to pass the exam or withdraw from the course). Because of that, we developed two predictive models. One, where dropout is defined as a student who fails to pass an exam [36] and another, where a dropout is defined as a student who will withdraw from or fail to pass the exam [43, 40]. Besides using two definitions of the dropout, we created and evaluate logistic regression models in eight different time frames, ranging from the beginning of the course up to the mid-term of the course. Therefore, we provide an answer to how early can we predict a dropout. An additional contribution of the paper is the utilization of the aggregation functions which are not commonly used in learning analytics. In order to gain better results, we used recency [10] and variability seeking index [12] which have shown importance in marketing and sports, respectively. In addition, we utilize counterfactual examples [42] that can aid decision-makers in helping the student by providing causal reasoning on how to reach a positive outcome. Predictive performance is done using the area under the curve (AUC) and area under the precision-recall curve (AUPRC), which can be found in the papers. However, we provide cumulative gain charts and lift curves which are useful for decision making of the predictive model. Finally, we analyzed coefficients of logistic regression to give an insight into why students are becoming dropouts. Since there are eight different time frames we interpret coefficients of the logistic regression and give possible answers to why dropout occurs for a different time period of the course. This finding can be used for course makers and course designers for dropout prevention strategies.

The remainder of the paper is organized as follows. In Section 2 we present a review of the literature. In Section 3 we present methodology. We will present data used in this research, followed by the experimental setup and evaluation of the predictive model. In Section 4 we provide results and interpretation of results, while in Section 5 we conclude the paper.

2. Literature review

From a historical point of view, the first MOOC called “Connectivism and Connective Knowledge” was created by George Siemens and Stephen Downs in 2008 which attracted over 2,000 students who participated free of charge [14]. Today, MOOCs environments such as Coursera, EdX, or Udacity have courses with over 1,000,000 enrolled students coming from over 190 countries [37].

Due to a proliferation of MOOCs in past years and the fact that a minority of students complete course, researchers from the technical field such as statistical analysis, data mining, and machine learning alongside with domain experts in fields of pedagogy, education, and organization tried to tackle the problem of dropout prediction and prevention. The first, main challenge, was the ill definition of the term dropout. The most common, term dropout (or stopout) is defined as a moment when a student unenrolled from the course. From that point, the student does not have access to learning materials anymore. However, many students do not unenroll from the course, but their activity is very low if existing at all. Therefore, the term dropout should be redefined to the last event student participated in, such as a quiz assignment [40] or watching a video [2]. We can define them as students who stopped participating in the course. We can ask ourselves, what about students who participated in the course and yet failed to pass the exam? Are they also dropouts? In some sense, they can be considered as dropouts. They had trouble keeping up with course materials and they needed help. Having in mind that these systems are created to help students gain knowledge and skills needed for tomorrow, one can try to identify them in advance and help them, i.e. give more time for assignments, or provide additional readings. Therefore, many researchers defined dropout as a situation when a student does not earn a certificate of accomplishment within a course [25, 20, 9, 32].

To the best of our knowledge models for predicting fail on the online courses is set as a binary classification task for fixed time periods. Juang et al. [25] used only the performance of the student on the first-week assignment. In their example, only that information was enough to recognize which students will receive a certificate of accomplishment with distinction compared to students who received a certificate of accomplishment with AUC 0.947, and also between students with a certificate of accomplishment and students who failed to pass the exam with AUC 0.851. A similar application can be found in [33, 26]. Namely, student activities on quizzes and assignments are used to predict performance on the final exam. An approach that was used is based on matrix factorization where latent features that describes student cohort and interpret their importance to pass exam with three points of predictions, one at the beginning of the course, one at the mid-point, and final one, a week before the exam. It has been shown that performance increases as more information are added, i.e. more data is available. Namely, predictions are worst at the beginning of the course and increases as more information about student interaction and behavior is added.

However, students do have more activities during the class which can be used for the prediction model. In MOOCs, course providers often have clickstream data, which is considered as an unstructured data set. One can extract features that describe student interaction with the learning environment. For example, interaction with video learning materials, activity on the discussion forum, time spent on a specific page is used. In paper [40] logistic regression with several groups of attributes is used. One group of attributes are attributes that correspond to submission and problem solve such as the number of submissions, a number of distinct problems attempted, a distinct number of correct solutions, the average time needed for submission, etc. Another group is regarded as interaction with other students such as number of forum posts, number of forum responses, number of wiki edits, the total number of collaboration, or time spent of forum and wiki. These features are used for predicting whether a student will withdraw in the fifth week from the reference week (i.e. predicting one month in advance). Similarly, in paper [9] latent Dirichlet allocation is used for behavioral trend identification based on problem sets answers, interaction with questions, videos, forum, etc. It has been shown, as in previous researches, that more information about student performance provides better predictive performance (the longer the course lasts, the model is better at identifying dropout).

Analysis of dropouts on the dataset used in this paper has been already made. Prediction using time series is available in the paper [18]. Namely, student engagement in a virtual learning environment is transformed into time series data which are further classified using time series forest algorithm. The results that are obtained are underperforming at the beginning of the series, i.e. beginning of the course, but improves as more data is available. The role of demographic data is presented on paper [34]. Namely, decision trees are used to predict failure on the exam [13]. This model can be applied before the course starts and it can achieve accuracy between 66% and 83%. One can also find framework Ouroboros [22, 23] that is demonstrated on this data. Finally, one can find the application of Naïve Bayes and Decision trees for course success prediction [3].

Compared to other approaches we will utilize every source of student interaction with the learning environment including demographic data, registration data, video interaction, quiz attempts, and assignments attempts. We will utilize a logistic regression model with lasso and ridge regularization. The reason for this is the fact that coefficients of logistic regression can be interpreted in terms of logarithms of odds of dropout, which will allow us to interpret the influences of each attribute to dropout. Next, we will have two experimental setups regarding the definition of the dropout. In this paper, we, therefore, adopted two definitions that are common in the literature. Further, we used multiple aggregation functions to summarize and describe student behavior which is not used in learning analytics at all, such as the recency of the event and variability seeking index. Finally, we utilized a logistic regression model that allows inspection of the coefficients. Based on the coefficients we can analyze the driving factors of a dropout.

It is noted that as a measure of performance most often Area Under the Receiver Operating Characteristics Curve (AUC) measure is used. AUC measures the probability that a classifier can discriminate between two randomly chosen data points, from which one is a positive outcome, and another negative is negative [1]. The reason why it is commonly used is that it is decision threshold independent, meaning that decision on what confidence or probability threshold predictive model will predict that student will fail an exam is omitted. Besides using AUC as a measure, we will use the area under the precision-recall curve (AUPRC) since it is more appropriate for class imbalance classification problems such as this one.

3. Data and Methodology

Data and Methodology section consists of an explanation of data and feature extraction from the database of Open University Learning Analytics Dataset [28]. Obtained data will be fitted into logistic regression. After an explanation of logistic regression, the whole experimental setup will be provided.

3.1. Data

Open University provided the database for learning analytics [28]. Data contain learning environment interaction, alongside with demographic data, enrollment data, etc. The Open University offers several hundred modules (subjects) from which every single one can be part of a university program or offered as a stand-alone course. Because of that, it suffers from similar problems as MOOCs, i.e. dropout. Namely, a lot of students enroll in a specific subject, but due to various reasons did not finish or unenrolled from the course. However, Open University generates a better completion rate mainly because courses are offered for credit and the length of the course is around 9 months [22].

The database provided for analytics is anonymized and organized in a normalized manner containing seven tables presented in **Fig. 1**.

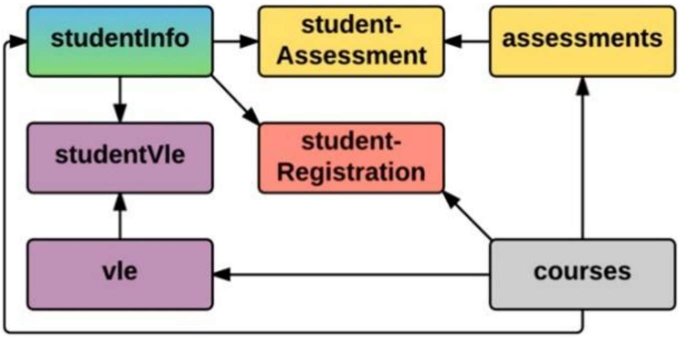


Fig. 1. Dataset structure [28]

In total 32,593 students registered to 22 courses. Information about the students is stored in table *studentInfo*, while information about the course is stored in table *courses*. During the course, the student had multiple assessments. Data about points achieved on the assessment is stored in table *student-Assessment*, while basic assessment information is stored in table *assessments*. There are 173,912 student assessment records. Finally, the student interacted with a virtual learning environment. There are 10,655,280 records of interaction and they are stored in table *studentVle*. Static information about the virtual learning environment is stored in table *vle*. In table *student-Registration* one can find information about registration of the student to the course and there is an indicator of the performance of the student on the course, i.e. withdraw, fail, passed, and distinction which is used for prediction.

Besides taking student demographic information (gender, region, highest education, IMD band, age, and disability), we generated aggregations (sum, count, mean, min, max, median, standard deviation, recency [976] and variability seeking index [976]) for student assessments and interaction with the learning environment. Two aggregation functions that are not common in many applications have been introduced. Namely, the recency of the event has shown to be of great importance in marketing [10]. Idea is to give more importance to events that occurred more recently. In other words, it gives decay to events that occurred long in the past. Variability seeking index is used for aggregation of categorical data, where difference compared to the previous event is calculated. If a student, i.e. changed the grade on the assessment then this deviance from the previous event should be accounted for. Variability seeking index has shown good predictive performance in sports [12].

Recency is calculated using the following formula (1).

$$recency = \sum_{i=1}^{m} s_i * 2^{-\left(\frac{x_i}{halflife}\right)} \tag{1}$$

where $halflife$ present interval for which effect of an attribute should be equal to 0.5 and x_i value to be inserted into the formula (i.e. days passed from the quiz). Since each student can have m events (quizzes or assignments), obtained recency scores are multiplied with the obtained score s_i and summed. This allows exponential decay of the effect of the obtained scores on the quiz or assignment. Half-life is always set to the half of the interval being predicted. For example, if we predict dropout based on the first-month activity, the half-life is equal to 15 days. In terms of educational data mining, this can be interpreted as forgetting term. More specifically, the effects of the previous quizzes and assignments are of less importance compared to the most recent ones.

Variability seeking index is an aggregation measure that calculates the trend of the scores obtained by the student. Although it does not satisfy all properties of the aggregation function (the result depends on the ordering of the data), this function allows identification of the subtle changes in the behavior of the student regarding the property one wants to analyze (i.e. activity on the learning environment, scores on the quizzes and assignments). It is calculated using formula (2).

$$vsi = \sum_{i=2}^{m} (s_i - s_{i-1}) \tag{2}$$

where s_i present the score obtained on the quiz or assignment in the time stamp i. A positive value will indicate an increase in the score values, or a positive trend in scores, while a negative value indicates a negative trend in the score values.

More specifically, an aggregated column from student assessment is a score on the assessment, point obtained from the assessment, and days submitted prior to the deadline. Aggregation is done on the student level and for each assessment type. For interaction with the learning environment number of clicks on the learning materials is aggregated on student level and activity level. In total, for each experiment, we extracted 522 features that describe student behavior.

3.2. Logistic regression

Logistic regression is one of the most popular machine learning algorithm with applications in various fields. It is commonly used in the educational domain, for dropout predictions [40, 25, 20, 23]. The main reason for the usage of logistic regression is the interpretability of the model. Namely, coefficients of the logistic regression model can be interpreted in terms of the odds ratio, and consequently in terms of probability. This property is important, especially for social science applications where each decision needs to be explained.

Logistic regression can be defined as a classifier that models the probability of dependent binary features y given a set of independent features X [19]. The model is defined as presented in the formula (3).

$$\log\left(\frac{p}{1-p}\right) = \theta_0 + \theta_1 x_1 + \cdots + \theta_k x_k. \tag{3}$$

where p represents the probability that dropout is going to occur (y = 1). Values of θ represent weights associated with independent features X. One wants to find the best values of θ such that the model provides the lowest possible error. Error is defined through loss function, called logistic loss (presented in the formula (4)), which has to be minimized.

$$\min \mathrm{L}(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}(y_i \log(\hat{y}_i) + (1-y)\log(1-\hat{y}_i)). \tag{4}$$

One of the problems is that using logistic loss function one can overfit models (learn data at hand, without the power of generalization on the new examples) when working with a large number of attributes. Therefore, extensions of logistic regression have been developed to deal with the problem of overfitting in such situations. One can extend the logistic loss function to regularize the process of learning coefficients. With regularization, one intentionally makes a greater loss with a purpose to create a model that can generalize to new examples [24]. Lasso regularization adds L1 norm in $\mathrm{L}(y, \hat{y})$. L1 norm has the effect that coefficients of logistic regression are forced to zero, i.e. lowers the number of features needed to explain the problem at hand. This way complexity of the problem is reduced. Ridge regularization adds L2 norm which forces coefficients of logistic regression to be lower in general. Both regularization terms are used to prevent the model to explain random noise or error. However, regularization terms introduce hyper-parameter λ which needs to be optimized [41]. In this paper, we utilized inner 10-fold cross-validation to find the best λ that maximizes the AUC measure.

3.3. Experimental Setup

The goal of the paper is to answer two research questions. First, we would like to know how early we can predict whether the student will pass the exam and, second, we want to provide a discussion on what drives the student to fail an exam. In order to answer the first research question, we trained and tested logistic regression models in eight different time periods. The first model is created on day 0 of the course, as seen in [34]. In that period student does not have any assignment interaction. However, a predictive model can be created using demographic features and interaction with learning materials (since some of them are available before the course starts). The next predictive model is created after the first week (seven days) of course. This setup is common in the online course [25]. At this point, student generates data, i.e. interaction with learning materials (videos, readings, etc.) and prediction can already be made. The following time periods are after one month (30 days), 45 days, 60 days, 90 days, and 120 days (approximate middle of the course length). It is expected that the performance of the model will improve as more data about the interaction with the learning environment is available.

In order to answer the second research question, we utilized logistic regression and interpreted the coefficients of the logistic regression. More specifically, used lasso and ridge logistic regression and interpretation of coefficients was performed on the best performing model. Coefficients of the logistic regression can be interpreted in terms of odds ratio and probability of dropout that can be found useful for course designing and decision-making. In addition, we utilize counterfactual examples. This powerful causal explanation finds the most related input attributes that lead to different outcomes [42].

Due to the fact that dropout has multiple definitions, we adopted two definitions which both present problem for any learning system as explained previously. Having in mind that we have eight different time frames of prediction we will have 16 experiments.

In order to have valid results students that unenrolled before the observed time period are dropped from the dataset. General information about a number of examples and the average dropout rate is presented in Table 1. As we can observe, a number of rows are exactly the same in both definitions of dropout. However, the percentage of the dropout is at the beginning two times greater if students that failed the course are included, increasing up to four times greater at the mid-term of the course (experimental setup where the model is created and evaluated after 120 days).

Models are evaluated using AUC because it is commonly used in educational data mining applications and specifically for the problem at hand. AUC can be interpreted as the probability that a random student who will fail to pass the exam has a greater probability that he/she will fail the exam than a random student who will pass an exam. This measure of evaluation is decision threshold independent, meaning that it is calculated for every possible combination of thresholds in data. A random classifier would have an AUC value of 0.5, while the perfect classifier would have an AUC value equal to 1 [11]. We also provide area under the precision-recall curve (AUPRC) which is also a common measure of classification model performance. It is interpreted as how many times a model is better compared to the default model. Values range from 0 to 1, where 1 is the value of the perfect classifier and the random classifier should have an average dropout rate in data at hand. Due to the fact that the model is evaluated using 10-fold cross-validation average value with the standard deviation will be presented. Additionally, we will present the lift curve and cumulative gain curve. Those model visualizations can be used for dropout prevention campaign definition and decision making. Namely, the lift curve presents a gain of a predictive model compared to using no model at all. On the x-axis percentage of students is presented, while on the y-axis present gain (ratio of the percentage of dropout students and the total number of students contacted) obtained using the predictive model. Value 1 on the y-axis presents a situation when the predictive model does not contribute to problem-solving, i.e. predictive model has no gain, while higher values improve the decision-making process (contact strategy). Having this in mind, the value of lift equal to 2 can be interpreted that the predictive model is two times better compared to using no model at all. The cumulative gain curve presents a comparison between dropout students and the total number of students. On the x-axis percentage of contacted students is presented, and on the y-axis percentage of dropout students are presented. If the gain curve is higher than the diagonal line, then the predictive model is usable. Namely, by contacting some percent of the students, we will be able to identify a higher percentage of dropout students.

Hyper-parameter λ for Lasso and Ridge regression was found using grid search with inner 10-fold cross validation.

Table 1. Dataset information

Dropout definition	Experimental setup	Number of rows	% of dropout
Withdraw	0 days	29,496	23.96%
	7 days	29,178	23.13%
	15 days	28,115	20.22%
	30 days	27,446	18.34%
	45 days	26,921	16.69%
	60 days	26,361	14.92%
	90 days	25,562	12.26%
	120 days	24,777	9.48%
Fail or Withdraw	0 days	29,496	47.84%
	7 days	29,178	47.27%
	15 days	28,115	45.28%
	30 days	27,446	43.99%
	45 days	26,921	42.85%
	60 days	26,361	41.64%
	90 days	25,562	39.81%
	120 days	24,777	37.91%

4. Results

After learning the model for both definitions of dropout following results are obtained. In Table 2 we present results on the withdrawal definition of dropout. One can observe that the performance of lasso logistic regression is better compared to ridge logistic regression for every experimental setup. Also, performance improves as more information about student behavior and interaction is available.

Table 2. Performance of logistic regression models on withdrawing students

Experimental setup	Lasso		Ridge	
	AUC	AUPRC	AUC	AUPRC
0 days	0.549 +/- 0.092	0.300 +/- 0.050	0.531 +/- 0.086	0.290 +/- 0.049
7 days	0.542 +/- 0.099	0.296 +/- 0.053	0.519 +/- 0.093	0.279 +/- 0.052
15 days	0.593 +/- 0.126	0.232 +/- 0.053	0.558 +/- 0.115	0.208 +/- 0.048
30 days	0.583 +/- 0.121	0.248 +/- 0.066	0.515 +/- 0.127	0.203 +/- 0.052
45 days	0.607 +/- 0.127	0.247 +/- 0.059	0.566 +/- 0.131	0.196 +/- 0.053
60 days	0.618 +/- 0.089	0.223 +/- 0.042	0.519 +/- 0.134	0.187 +/- 0.055
90 days	0.623 +/- 0.099	0.185 +/- 0.040	0.542 +/- 0.133	0.162 +/- 0.048
120 days	0.681 +/- 0.059	0.162 +/- 0.025	0.569 +/- 0.142	0.131 +/- 0.040

Initial model, i.e. at the beginning of the course, have trouble distinguish between dropouts and non-dropouts with AUC 0.549. This value of AUC means that model is just better than a random model. But, after the interaction of the student with learning materials and the learning environment model captures the withdrawal behavior and manages to discriminate between dropouts and non-dropouts. In the middle of the course (model created after 120 days), AUC is 0.681. A similar conclusion can be made based on AUPRC values. Namely, the initial model has a value 0.300 while the default model should have 0.2396 (percentage of dropouts available in data). Based on these values, the predictive model is better by ~25% compared to a random model at the beginning of the course and ~71% after 120 days. This means that some features make a difference between dropout and non-dropout students.

The second definition of dropout students considers failing on the course and withdraw of the student as a dropout. To some extent, this definition is easier for prediction since some of the events, i.e. quizzes and assignments, directly influence the final grade. Performance in terms of AUC and AUPRC for this definition of dropout is presented in Table 3.

Table 3. Performance of logistic regression models on withdrawing and fail students

Experimental setup	Lasso		Ridge	
	AUC	AUPRC	AUC	AUPRC
0 days	0.661 +/- 0.054	0.635 +/- 0.056	0.651 +/- 0.054	0.628 +/- 0.054
7 days	0.669 +/- 0.062	0.644 +/- 0.061	0.660 +/- 0.059	0.637 +/- 0.059
15 days	0.673 +/- 0.074	0.633 +/- 0.077	0.663 +/- 0.068	0.624 +/- 0.070
30 days	0.693 +/- 0.089	0.646 +/- 0.097	0.707 +/- 0.081	0.658 +/- 0.089
45 days	0.738 +/- 0.087	0.683 +/- 0.096	0.739 +/- 0.084	0.686 +/- 0.090
60 days	0.788 +/- 0.052	0.753 +/- 0.063	0.791 +/- 0.045	0.756 +/- 0.054
90 days	0.820 +/- 0.036	0.791 +/- 0.041	0.817 +/- 0.038	0.785 +/- 0.045
120 days	0.869 +/- 0.025	0.841 +/- 0.030	0.864 +/- 0.033	0.833 +/- 0.040

Considering this definition of a dropout we obtained better predictive performance. Namely, AUC is at the beginning of the course 0.661 and improves up to 0.869. As in previous results, lasso logistic regression is mostly better compared to ridge logistic regression. Based on AUPRC values we can conclude that the predictive model is better than a random model for ~33% at the beginning of the course and ~122% after 120 days.

In order to answer the question of how early can we predict one must ask a question of what good enough performance of the model is? There are no formal guidelines for evaluation AUC and AUPRC values, i.e. what is considered as good performance. All of the models are useful. More specifically, they are better than uninformed decision making. Using the rule of thumb, AUC of 0.700 is considered as a good model. However, this value can be misleading if a class imbalance is present and AUPRC is recommended [11]. Our AUPRC suggests that our model is even at the beginning of the course better ~25% for withdrawing students and ~33% for withdrawing and failed students than a random model. We can say that models are usable, i.e. can be used for course design and decision making. For example, the model can be used for a mass campaign for the prevention of the dropout. Course designers could move the deadline, provide additional readings, or involve more details to students that are more prone to be a dropout.

5. Discussion

After presenting and discussing the results of the predictive model in terms of predictive performance, we present a discussion of the application of the predictive model. First, we need to answer the question of whom to contact. For that purpose, we can use the lift curve and cumulative gain curve presented in **Fig. 2**.

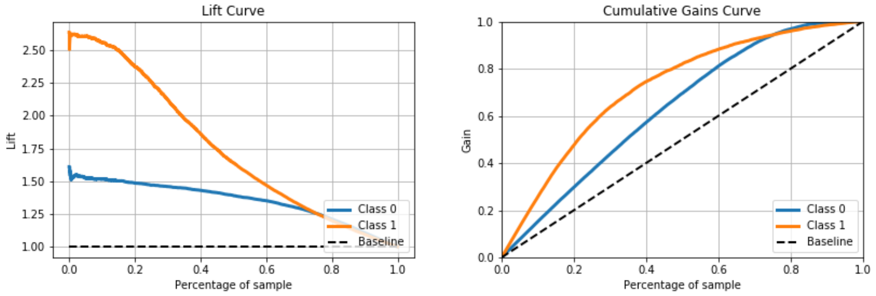


Fig. 2. (a) Lift curve of lasso logistic regression model on withdrawing and fail students, and (b) Cumulative gain curve of lasso logistic regression model on withdrawing and fail students

On the left side of **Fig. 2** one can see the lift curve. On the x-axis percentage of the sample is presented, while on the y-axis lift is presented. Dropout students are presented in orange color and denoted as Class 1, while non-dropout students are presented in blue color and denoted as Class 0. Lift is the ratio of the expected probability of dropout and the probability of dropout in the dataset. Therefore, the baseline value is 1. Examples are sorted according to the probability of dropout and one can make decisions based on it. In addition, one can use the cumulative gain curve (**Fig. 2** (b)). On the x-axis percentage of the sample is presented, and on the y-axis gain, or percentage of students of the corresponding class. Dropout students are presented in orange color and denoted as Class 1, while non-dropout students are presented in blue color and denoted as Class 0.

Having this description in mind, one can contact the top N% of the students sorted by probability of dropout. For the predictive model presented in Fig. 2 (a) it would be beneficial to contact the top 20% of students. This will yield in contacting the students that are more than twice as likely to dropout compared to the overall dropout rate. More specifically, one will contact around 50% of the students that will dropout as seen in Fig. 2 (b).

These figures aid decision-makers by presenting the results of the predictive model that is more interpretable than confusion matrix and predictive performance measures such as AUC or AUPRC. There are multiple reasons for such a statement. Predictive models can use multiple performance measures for the evaluation of the model. First, using too many performance measures can be confusing for the decision-maker, since most of them are similar by definition for experts that are not a data scientist. In addition, the simplest ones, such as accuracy, precision, and recall might be inappropriate due to the selection of the decision threshold. Decision-makers would need prior education on predictive measures and their effects. Finally, the cost of errors is not the same. The cost of contacting and giving incentive to the student that will pass the exam (called false positive) is most probably a lot less than the cost of not contacting the student that will be a dropout (called a false negative). Therefore, usage of the lift curve and cumulative gain curve will give an insight to the decision-maker how many students will be contacted in percentage terms and how likely they to be a dropout is.

Next, we need to discover why the students are prone to dropout. More specifically, we answer the question of what influence dropout. Answering this question will give insight to the decision-maker about the course design, online platform, and student characteristics that can be improved and utilized in further decision-making. This is our second research question in this experiment. For the predictive model created at the beginning of the course, it is expected that demographic attributes influence the prediction model, more specifically age and self-reported social-economic status [34]. As more interaction with the learning environment is available, quiz scores and assignment grades get more influence [40, 22]. We will present several coefficients that contributed most to the prediction model, for four time frames. More specifically, one at the beginning of the course, one after the first week of the course, another after the first month of the course, and last one after 120 days (middle of the course). These time frames are selected because they are common in the literature and it is considered that decision-makers can create a campaign and possibly influence a student in a positive direction.

Coefficients of the most important features of lasso logistic regression for the withdraw students are presented in **Fig. 3**.

At the beginning of the course, the most important features are age between 0 and 35 and between 35 and 55, and “A” level of education. These features have a negative influence on dropout. This means that students that have between 0 and 35 years of age, or between 35 and 55 are less prone to be dropouts. This finding is interesting because in MOOCs this subpopulation is more prone to be dropouts [40, 31]. As a major dropout factor one can find studied credits and disability of the students. It has been noted in the literature that regardless of the type of learning studied credits influence dropout. A number of credits that course offers are correlated with the difficulty of the course. Therefore, the difficulty of the course is one of the factors of dropout [27]. In addition, if the student takes too many online courses, resulting in many studied credits, he/she will have trouble following too many courses and most likely dropout from some of them (or even all of them). It is interesting to note that interaction with a virtual learning environment is present in coefficient even at the model for the beginning of the course. Sum of clicks on content, pages, and forum are of negative influence on dropout [15, 39]. This indicates that students do tend to interact with the learning environment (i.e. reading materials, discuss the forum, etc.) in order to be prepared for the upcoming lecture. These students highly influence to achieve satisfactory results. However, the newly used aggregation measure is of interest. More specifically, the recency of interaction with the content (i.e. videos) learning environment is introduced. Its value is negative, which can be interpreted that more recent interaction with the learning environment is negatively influencing the dropout.

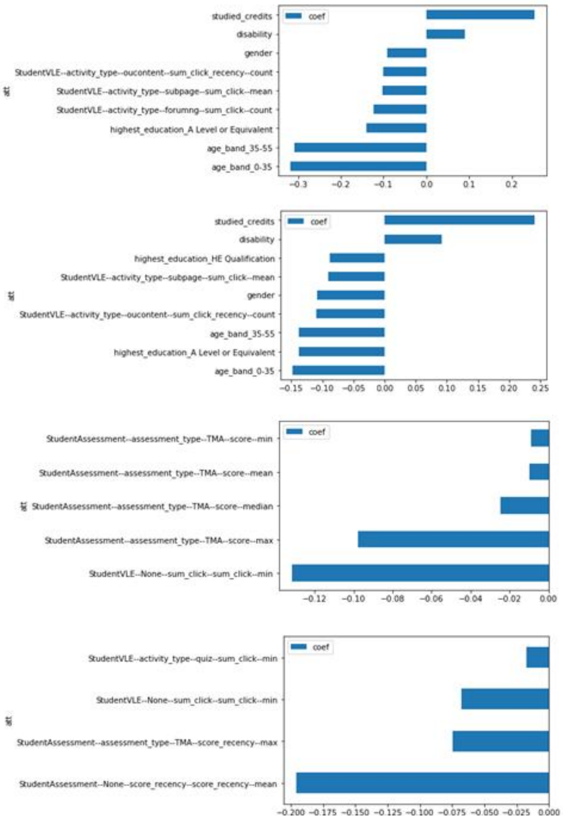


Fig. 3. Bar chart of coefficient weights for (a) beginning of the course, (b) after the first week, (c) after the first month, and (d) at the mid-term, for lasso logistic regression model for the withdraw students

After one week of lectures in the online course, the situation for the data at hand remains similar. Studied credits and disability remain the most important features for logistic regression to recognize dropouts. Also, age and activity in the virtual learning environment are most important at recognizing non-dropouts. It is worth noticing that the recency of interaction with the content (i.e. videos) learning environment remains as one of the most influential attributes in the predictive model. Although it cannot be seen in Fig. 3 variability seeking index is introduced in the predictive model at this point. More specifically, after one week of lectures students generate activities, and the variability of their activity starts making an impact on the predictions. All of the influencing attributes that utilize this aggregation function are related to the interaction with the learning environment and their values are negative. Those attributes are related to the clicking on the URLs in the supplementary materials, posting on the forum, and the overall number of clicks in the learning environment. Therefore, the positive value of variability seeking index (a positive trend in activity) leads to the passing the exam at the final of the course.

After one month of course content, the situation is drastically changed. Students have generated many activities in the learning environment and have had several quizzes and assignments. This yielded in different patterns leading to the dropout, f.e. some of the students were discouraged by the difficulty, or some of them felt unmotivated to proceed. Attributes that were most important for the predictions in previous time periods, i.e. age, studied credits, and disability, are not important that much. They were not present in the most important attributes at all. They are replaced with scores on quizzes and assignments. Scores on tutor marked assignments are the most dominant set of features. However, the most important feature is the number of clicks in total on the virtual learning environment [38]. Recency and variability seeking index are also not present in the most important ones, but they are present in the predictive model. The recency of students' activity on the learning environment, as well as the recency of login on the learning environment and posting on the forum, are still considered as a strong negative influence on the dropout. However, variability seeking index is present only for two attributes, which are the same as for the previous model. More specifically, clicking on the URLs in the supplementary materials and posting on the forum negatively influence dropout of the students'.

After 120 days of the course (mid-course), it becomes clear what influence dropout. The students are familiar with the learning environment, the style of teaching, quizzes, and assignments. Therefore, the amount of effort that is invested in the learning environment is highly reduced, focusing only on the part of the course that results in the certificate (i.e. quizzes and assignments). Having that in mind, scores on quizzes, tutor-marked assignments, and activity on the virtual learning environment are the most important factors of dropout. However, instead of using classical aggregation function, recency is more appropriate, at least for the data at hand. As can be observed, the most influencing attribute was the recency of the obtained score. This indicates that the greater values of the recent scores are negatively influencing the dropout.

We can conclude that at the beginning of the course demographic features of the student and course description features are the most important for dropout prediction. Namely, younger students, with higher education are less prone to dropout, while the difficulty of the course and disability of the student do influence dropout. As the course goes by, student activity gains more importance. Interaction with the virtual learning environment, i.e. spending more time reading materials, posting questions and answers on the discussion forum, as well as scores on the assignments gains more importance for dropout prediction. More specifically, the higher the engagement of the student to the virtual learning environment and the higher the scores on the assignments, there is less chance that student will be a dropout. In addition, proposed aggregation functions are important for the predictive model, as the recency of activity on the learning environment and recency of scores negatively influence dropout. Variability seeking index does influence the predictive model (i.e. positive change in trend in activity on the learning environment leads to fewer dropouts), but not as strong as the recency.

The interpretation is similar, but different if dropout is defined as a student who failed the exam or withdraws from the course. Coefficients are presented in Fig. 4. At the beginning of the course, the most important feature is the “A” level of education. This feature had a negative influence on withdrawing, but in this setting (withdraw and failing the exam), it has a positive influence on dropout prediction. Besides, “A” level of education positive influence on dropout is presented in studied credits and number of previous attempts, as well as lower values of the IMD band. This effect has already been noticed by [21]. IMD band, which represent the socio-economic status of the region of the student could be that lower socio-economic status of the student influences the performance of the student. One should be careful when interpreting this coefficient weight since lower socio-economic status surely does not cause lower performance, but that there is some confounding effect of the performance. As in previous dropout models higher the activity on the virtual learning environment, the less the probability of dropout. Additionally, proposed aggregation functions recency and variability seeking index are not present in the predictive model.

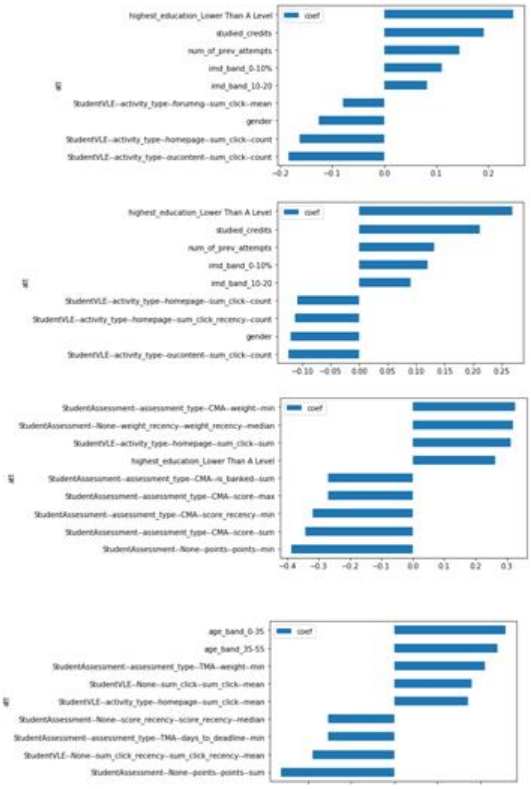


Fig. 4. Bar chart of coefficient weights for (a) beginning of the course, (b) after the first week, (c) after the first month, and (d) at the mid-term, for lasso logistic regression model for the withdraw students and fail students

The coefficients of lasso logistic regression are approximately the same for the model after the first week. More specifically, prior education studied credits, and the number of previous attempts of the course has a positive influence on the dropout predictions, while click counts on the learning materials and being female influence passing exam. However, the newly proposed recency aggregation function applied to the access to the homepage is considered a major factor that influences passing the exam. Recency had a greater influence on the prediction compared to the model predicting only withdraw students. More specifically, recency appeared in this predictive model in ten attributes, all regarding specific interaction with the learning environment (i.e. access to the forum, URLs in the learning materials, and accessing additional learning materials) with a negative value (negatively influencing dropout).

Quizzes and assignments do not influence predictions at the beginning of the course. They are introduced as important features in the model created after the first month of course. It can be observed that the higher the weight of the assignments in the first month greater the probability that students will be a dropout. Also, if a student achieved greater scores on assignments lower the probability of being a dropout. A similar finding can be seen in many dropout predictions model [7, 43]. This is because the cost of failure is low. The students tend to quit after the first several unsuccessful quizzes and assignments mostly to not feel the failure of not achieving the certificate [6]. Our newly added aggregation measures, recency, and variability seeking index have their share in the prediction. Recency, as an aggregation function, appeared as important in many attributes. First, the weight of the test. The attribute coefficient related to the recency of the weight of the test indicates that challenging tasks for the students that account for many points positively influence dropout. More specifically, obtaining a low score on the important test leads to failing the exam or withdrawing the course. Also, the variability seeking index has appeared in several interactions with the learning environment attributes. It indicates that higher usage of the learning environment leads to passing the exam.

Interestingly, after 120 days of course the age of the student returns to the most important features. It is even more surprising that this attribute has the highest positive influence on the dropout. However, it is worth to notice that recency related attributes are very important with a negative influence on the dropout. More specifically, the higher the recent score obtained it is more likely that students will pass the exam.

With the interpretation of the coefficients, we explained why do dropout occurs in general. However, for the decision support system in MOOCs one needs a detailed explanation of why a specific student did not pass the exam or what can this student do in order to pass an exam. For that purpose, we utilized counterfactual examples [30]. Those are examples that are the most similar to the real examples but having a different outcome. In the process of the dropout prediction, counterfactual examples would be students that passed the exam but are most similar to the one we would like to give incentive. By providing this kind of example to the decision-makers one could look at the attributes that can be influenced by the decision-maker in order for a student to pass the exam. Simplified example (due to the high dimensionality of the data we showed only several attributes) of counterfactual examples are presented in Table 4. Suppose we have attributes *gender*, *Forum_post* representing a total number of posts on the forum, *VLE_recency* representing a number of interactions with the virtual learning environment whose score is adjusted to valorize more recent ones, *Quiz_recency* representing average quizzes score adjusted to valorize more recent ones, and output column *dropout* that signal whether the student is a dropout. The first row of the table is the original example, while the remaining rows present counterfactual examples. *Gender*, as well as *Quiz_recency* are attributes that decision-makers cannot influence. Therefore, it will always remain the same (for all counterfactual examples). In other words, *Forum_post* and *VLE_recency* are of the point of influence to the decision-maker as those attributes can be subject to intervention. The decision-maker can request multiple counterfactual examples (in this example three) and they will slightly differ. Column *dropout* represents a signal that a student is a dropout, or probability of a dropout for counterfactual examples. In this example, in a bold letter, we have shown what strategies the decision-maker can take for communicating the student. In this simple example, it can give the incentive to interact using the forum, or interaction with the virtual learning environment, and finally interact with both forum and virtual learning environment, but with less intensity.

Table 4. Counterfactual examples

Student	gender	Forum_post	VLE_recency	Quiz_recency	dropout
Original	1	0	0	57.2	1
CF1	1	**5**	0.57	57.2	0.12
CF2	1	0	**0.93**	57.2	0.08
CF3	1	**1**	**0.75**	57.2	0.25

Finally, once the decision-maker has the predictive model, a graphical tool for selecting the students to be contacted, interpretation of the model, and counterfactual explanation for each student that is predicted to be a dropout, one can generate a decision support system. In this paper, the most suitable solution would be the development of a module similar to a customer relationship manager (CRM). More specifically, decision-makers could create a campaign that will contact a student regularly (i.e. weekly basis) using e-mail messages and/or push notifications.

6. Conclusion

Application of data mining and machine learning in the education domain presents an interesting research area which requires a lot of technical skills (i.e. data visualization, statistics, algorithms, etc.), social skills (i.e. pedagogy, andragogy, communication skills, etc.) in order to make effective and influential decisions. In this paper, we employed lasso and ridge logistic regression for the dropout prediction of the students in the online learning environment. We asked ourselves two questions. How early can we predict dropout and can we explain why dropouts occur? Because of the vague definition of dropout, we developed two experiments where dropout was defined when student unenrolled from the course, and another when a student failed to pass an exam or unenrolled from the course.

In order to answer the first question, we created eight experiments. Namely, we created models at the beginning of the course, after the first week of the course, after 15 days, after 30 days, after 45 days, after 60 days, after 90 days, and after 120 days (mid-term). The results have shown that withdraw from the course is harder to predict. A performance measure that was selected, AUC, was 0.549 at the beginning of the course and arose to 0.681 at the mid-term. For the second definition of dropout, the performance was much greater. More specifically, AUC at the beginning of the course is 0.661 and improves up to 0.869 in the mid-term. These models can be used for informed decision making because improvement compared to uninformed decision making is from ~25% for the model at the beginning of the course, to ~71 at the mid-term.

The second research question (why do the dropout occur) is answered by analyzing the coefficients of the logistic regression model. It has been shown in both definitions that at the beginning of the course demographic features of the student such as age and education influence dropout. More specifically, younger students, with higher education are less prone to dropout. However, the difficulty of the course and disability of the student do influence dropout. Later, as students gain activity in the virtual learning environment, the predictive model gives more importance to those attributes. More specifically, the higher the engagement of the student to the virtual learning environment and the higher the scores on the assignments, the less the probability that students will be a dropout. It is worth noticing that proposed aggregation functions recency and variability seeking index influence predictive models as they were frequently considered as one of the most important ones.

Having answers to the two proposed decision-makers can make an informed decision about contacting the troubled student. Namely, one is given the answers to the question of *who* is at trouble, and *why* is at trouble. The answer to the question of *what* to do or *how* to approach is given using counterfactual examples. In future work, we will try providing explanations using Shapley scores [4] or Lime framework [16]. Counterfactual examples do provide some notion of explanations, but Shapley scores and Lime can more human interpretable explanations.

As we believe that predictive performance is region-specific, i.e. one region has overall better results compared to the other one, we would like as a part of the future research to apply multi-task logistic regression models [46]. This type of analysis would create a predictive model for each region (one region will be one task). However, the multi-task learning framework will tend to have similar coefficients for the attributes throughout the regions. If one region is truly different in the behavior of dropouts then their coefficient for some attribute will differ (i.e. predictive strength will be much greater compared to the penalty imposed by changing the value of the coefficient). This analysis would give us the true value of the driving factors for the dropout based on the region of the student.

Another line of the research should be regarded as the problem of algorithmic fairness. More specifically, we will strive to create predictive models that are non-discriminatory or fair toward socially sensitive groups. As results suggested for the data at hand, gender seems like an attribute that discriminates passing the exam and failing to pass the exam. Since this can be an indicator of disparate impact or even disparate treatment, one should inspect why gender is making a difference in predictions. In order to make a fair predictive model and not including gender (or any other attribute that can be considered as a proxy to gender), we will try to preprocess data to be fair, adjust prediction to seem fair, or adjust the learning algorithm.

Acknowledgment. This paper is the result of the project ONR - N62909-19-1-2008 supported by the Office of Naval Research, the United States: *Aggregating computational algorithms and human decision-making preferences in multi-agent settings*.

References

1. Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., & Roth, D. (2005). Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, *6*(Apr), 393-425.
2. Allione, G., & Stein, R. M. (2016). Mass attrition: An analysis of drop out from principles of microeconomics MOOC. *The Journal of Economic Education*, *47*(2), 174-186.
3. Azizah, E. N., Pujianto, U., & Nugraha, E. (2018, October). Comparative performance between C4. 5 and Naive Bayes classifiers in predicting student academic performance

in a Virtual Learning Environment. In *2018 4th International Conference on Education and Technology (ICET)* (pp. 18-22). IEEE.

4. Biecek, P. (2018). DALEX: explainers for complex predictive models in R. *The Journal of Machine Learning Research*, *19*(1), 3245-3249.
5. Bleiklie, I. (2005). Organizing higher education in a knowledge society. *Higher Education*, *49*(1-2), 31-59.
6. Chen, C., Sonnert, G., Sadler, P. M., Sasselov, D. D., Fredericks, C., & Malan, D. J. (2020). Going over the cliff: MOOC dropout behavior at chapter transition. *Distance Education*, *41*(1), 6-25.
7. Chen, Y., Chen, Q., Zhao, M., Boyer, S., Veeramachaneni, K., & Qu, H. (2016, October). DropoutSeer: Visualizing learning patterns in Massive Open Online Courses for dropout reasoning and prediction. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)* (pp. 111-120). IEEE.
8. Chengjie, Y. U. (2015). Challenges and changes of MOOC to traditional classroom teaching mode. *Canadian Social Science*, *11*(1), 135.
9. Coleman, C. A., Seaton, D. T., & Chuang, I. (2015, March). Probabilistic use cases: Discovering behavioral patterns for predicting certification. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale* (pp. 141-148). ACM.
10. Cui, G., Wong, M. L., & Lui, H. K. (2006). Machine learning for direct marketing response models: Bayesian networks with evolutionary programming. *Management Science*, *52*(4), 597-612.
11. Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 233-240). ACM.
12. Delibašić, B., Radovanović, S., Jovanović, M., Obradović, Z., & Suknović, M. (2018). Ski injury predictive analytics from massive ski lift transportation data. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, *232*(3), 208-217.
13. Delibašić, B., Vukićević, M., Jovanović, M., & Suknović, M. (2012). White-Box or Black-Box Decision Tree Algorithms: Which to Use in Education?. *IEEE Transactions on Education*, *56*(3), 287-291.
14. Downes, S. (2008). Places to go: Connectivism & connective knowledge. *Innovate: Journal of Online Education*, *5*(1), 6.
15. Fei, M., & Yeung, D. Y. (2015, November). Temporal models for predicting student dropout in massive open online courses. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)* (pp. 256-263). IEEE.
16. Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018, October). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 80-89). IEEE.
17. Gütl, C., Rizzardini, R. H., Chang, V., & Morales, M. (2014, September). Attrition in MOOC: Lessons learned from drop-out students. In *International Workshop on Learning Technology for Education in Cloud* (pp. 37-48). Springer, Cham.
18. Haiyang, L., Wang, Z., Benachour, P., & Tubman, P. (2018, July). A Time Series Classification Method for Behaviour-Based Dropout Prediction. In *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)* (pp. 191-195). IEEE.
19. Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC.
20. He, J., Bailey, J., Rubinstein, B. I., & Zhang, R. (2015, February). Identifying at-risk Students in Massive Open Online Courses. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

21. Hlioui, F., Aloui, N., & Gargouri, F. (2018, December). Understanding Learner Engagement in a Virtual Learning Environment. In *International Conference on Intelligent Systems Design and Applications* (pp. 709-719). Springer, Cham.
22. Hlosta, M., Zdrahal, Z., & Zendulka, J. (2017, March). Ouroboros: early identification of at-risk students without models based on legacy data. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference* (pp. 6-15). ACM.
23. Hlosta, M., Zdrahal, Z., & Zendulka, J. (2018). Are we meeting a deadline? Classification goal achievement in time in the presence of imbalanced data. *Knowledge-Based Systems*, *160*, 278-295.
24. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning* (Vol. 112, p. 18). New York: Springer.
25. Jiang, S., Williams, A., Schenke, K., Warschauer, M., & O'Dowd, D. (2014, July). Predicting MOOC Performance with Week 1 Behavior. In *Educational Data Mining* 2014.
26. Jovanovic, M., Vukicevic, M., Milovanovic, M., & Minovic, M. (2012). Using data mining on student behavior and cognitive style data for improving e-learning systems: a case study. *International Journal of Computational Intelligence Systems*, *5*(3), 597-610.
27. Kursun, E. (2016). Does Formal Credit Work for MOOC-Like Learning Environments?. *The International Review of Research in Open and Distributed Learning*, *17*(3).
28. Kuzilek, J., Hlosta, M., & Zdrahal, Z. (2017). Open University Learning Analytics Dataset. *Scientific Data*, *4*, 170171.
29. Mah, D. K. (2016). Learning analytics and digital badges: Potential impact on student retention in higher education. *Technology, Knowledge and Learning*, *21*(3), 285-305.
30. Mothilal, R. K., Sharma, A., & Tan, C. (2020, January). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 607-617).
31. Onah, D. F. (2015, June). Learners expectations and motivations using content analysis in a MOOC. In *EdMedia+ Innovate Learning* (pp. 192-201). Association for the Advancement of Computing in Education (AACE).
32. Radovanović, S., Delibašić, B., Suknović, M. (2019). How early can we predict MOOC performance? In *Proceeding of the Euro mini International Conference on Decision Support System Technology 2019* (pp. 208-214). Madeira, Portugal.
33. Ramesh, A., Goldwasser, D., Huang, B., Daume III, H., & Getoor, L. (2014, June). Learning latent engagement patterns of students in online courses. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.
34. Rizvi, S., Rienties, B., & Khoja, S. A. (2019). The role of demographics in online learning; A decision tree based approach. *Computers & Education*, *137*, 32-47.
35. Romero, C., & Ventura, S. (2013). Data Mining in Education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *3*(1), 12-27.
36. Romero, C., & Ventura, S. (2017). Educational data science in massive open online courses. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(1), e1187.
37. Sanchez-Gordon, S., & Luján-Mora, S. (2016). How could MOOCs become accessible? The case of edX and the future of inclusive online learning. *Journal of Universal Computer Science*, *22*(1), 55-81.
38. Staubitz, T., & Meinel, C. (2018, June). Team based assignments in MOOCs: Results and Observations. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale (L@S2018)* (pp. 47-51).
39. Sunar, A. S., White, S., Abdullah, N. A., & Davis, H. C. (2016). How learners' interactions sustain engagement: a MOOC case study. *IEEE Transactions on Learning Technologies*, *10*(4), 475-487.

40. Taylor, C., Veeramachaneni, K., & O'Reilly, U. M. (2014). Likely to Stop? Predicting Stopout in Massive Open Online Courses. *arXiv preprint arXiv:1408.3382*.
41. Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(3), 273-282.
42. Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, *31*, 841.
43. Xie, Z. (2019). Modelling the dropout patterns of MOOC learners. *Tsinghua Science and Technology*, *25*(3), 313-324.
44. Yang, D., Sinha, T., Adamson, D., & Rosé, C. P. (2013, December). Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-driven education workshop* (Vol. 11, p. 14).
45. Ye, C., & Biswas, G. (2014). Early prediction of student dropout and performance in MOOCs using higher granularity temporal information. *Journal of Learning Analytics*, *1*(3), 169-172.
46. Zhou, J., Chen, J., & Ye, J. (2011). Malsar: Multi-task learning via structural regularization. *Arizona State University*, 21.

Sandro Radovanović is a teaching assistant at the University of Belgrade, Faculty of Organizational Sciences.His main research interests are machine learning, decision support systems, decision theory, and business intelligence. So far, he published over 60 papers in journals and conference proceedings. Since 2018, he is in the Board of Assistants at the EURO Working Group on Decision Support Systems (EWG-DSS).

Boris Delibašić is Full Professor at the University of Belgrade, Faculty of Organizational Sciences (School of Management). Since 2011, he is in the Coordination board at the EURO Working Group on Decision Support Systems (EWG-DSS). His main research interests are decision support systems, machine learning algorithm design, business intelligence, and multi-attribute decision making.

Milija Suknović is Full Professor at the University of Belgrade, Faculty of Organizational Sciences (School of Management).His main research interests are decision theory, decision analysis, decision support systems, machine learning algorithm design, and business intelligence.

Received: September 20, 2020; Accepted: January 10, 2021.

Computer Science and Information Systems 18(3):979–999 https://doi.org/10.2298/CSIS200710055Y

Deep Reinforcement Learning for Resource Allocation with Network Slicing in Cognitive Radio Network[*]

Siyu Yuan[1,2], Yong Zhang[1,2][**], Wenbo Qie[1], Tengteng Ma[1,2], and Sisi Li[1]

[1] School of Electronic Engineering,
Beijing University of Posts and Telecommunication
100876, Beijing, China
{yuanisyu, yongzhang, qwb, mtt, ssl123}@bupt.edu.cn
[2] Beijing Key Laboratory of Work Safety Intelligent Monitoring,
Beijing University of Posts and Telecommunications
100876, Beijing, China

Abstract. With the development of wireless communication technology, the requirement for data rate is growing rapidly. Mobile communication system faces the problem of shortage of spectrum resources. Cognitive radio technology allows secondary users to use the frequencies authorized to the primary user with the permission of the primary user, which can effectively improve the utilization of spectrum resources. In this article, we establish a cognitive network model based on underlay model and propose a cognitive network resource allocation algorithm based on DDQN (Double Deep Q Network). The algorithm jointly optimizes the spectrum efficiency of the cognitive network and QoE (Quality of Experience) of cognitive users through channel selection and power control of the cognitive users. Simulation results show that proposed algorithm can effectively improve the spectral efficiency and QoE. Compared with Q-learning and DQN, this algorithm can converge faster and obtain higher spectral efficiency and QoE. The algorithm shows a more stable and efficient performance.

Keywords: cognitive radio network, network slicing, resource allocation, deep reinforcement learning.

1. Introduction

With the development of wireless communication technology, wireless communication services around the world have shown a trend of rapid movement, huge capacity and mechanism intelligence. The fifth-generation cellular network is the key technology of the current wireless communication technology. The deployment of 5G networks will promote the rapid development of IoT (Internet of Things) and cloud computing services such as 4K video, VR (Virtual Reality), AR (Augmented Reality), driverless cars, intelligent power grids, and telemedicine [14]. 5G network has the characteristics of network virtualization and programmability, and uses a new technology called network slicing [4]. Network slicing is an on-demand networking model that allows operators to separate multiple virtual networks on a unified infrastructure. Each network slice is logically isolated

[*] This paper is an extented version of [27] which is published in International Conference on Human-Centered Computing 2019.

[**] Corresponding author

from the wireless access network to the core network to adapt to various types of applications. The 5G network supports three general service scenarios: eMBB (Enhanced Mobile Broadband), URLLC (Ultra-reliable and Low Latency Communication) and mMTC (Massive Machine-type Communications). eMBB refers to the further improvement of user experience and other performance based on existing mobile broadband business scenarios. The intuitive feeling is that the transmission rate has been greatly improved, which is mainly used for 4K video and large file download. URLLC is characterized by high reliability and low latency, and is mainly used for unmanned driving and remote surgery. In order to provide better performance and cost-effective services, network slicing has a lot of research space in terms of resource management. By using resource management algorithms, the wireless network can effectively increase the total transmission rate of the wireless access network [17], spectrum efficiency [11], and user-perceived QoE [8]. mMTC scenario is mainly used for large-scale IoT services.

At present, people's demand for data rate is higher and higher, and the demand for spectrum resources is also increasing. However, spectrum resources are very scarce. According to current spectrum policies, most of the available spectrum has been allocated or licensed to wireless service providers. In order to solve the problem of spectrum scarcity, cognitive radio technology has become the key to solving this problem [12]. Cognitive radio technology monitors the working conditions of authorized users by sensing the spectrum environment, and dynamically schedules the available idle spectrum under the premise of causing interference within a certain range to the authorized users, thereby improving spectrum utilization. In a cognitive radio network, according to the different ways that cognitive users access idle licensed spectrum, the sharing of licensed spectrum can be divided into two models (overlay and underlay). In the overlay mode, cognitive users can only use authorized spectrum when authorized users are not communicating. Underlay mode allows cognitive users to use the spectrum to which authorized users belong to perform data transmission with authorized users at the same time. Cognitive users will cause certain interference to authorized users, but the interference should be guaranteed within a certain range. In order to restrict the interference caused by cognitive users, the interference temperature constraint plays a key role in the allocation of cognitive radio resources. Interference temperature is a concept defined by the FCC (Federal Communications Commission) in order to improve spectrum utilization efficiency and study the application of cognitive radio [22], which is used to quantify the communication interference of cognitive users.

Currently in China, the 230 MHz frequency band is used for the construction of electric power wireless private networks. It is a dedicated spectrum resource specifically allocated to industries such as power, water power, and geology. Many frequency bands in electric power wireless private networks are licensed frequency bands. Private network users cannot use the licensed frequency bands of other private networks, which makes the 230MHz frequency band have weak transmission capabilities and low spectrum utilization [2]. With the development of wireless communication technology, the current power wireless private network based on the LTE system has begun to evolve to 5G, and the application of multi-slice services needs to be carried out in the spectrum awareness environment. Applying cognitive radio technology to 5G networks can effectively solve the problem of spectrum scarcity, improve spectrum utilization, and provide effective help for the construction of 5G-based power wireless private network systems.

Reinforcement learning algorithms are used to solve decision-making problems and obtain optimal strategies through continuous interaction with the environment. The most widely used reinforcement learning algorithm is Q-Learning [21]. In order to solve complex control problems, deep reinforcement learning combines reinforcement learning with deep learning to learn control strategies from high-dimensional raw data. The basic idea of deep reinforcement learning is to use deep learning to automatically learn abstract features of large-scale input data, and then use reinforcement learning based on deep learning feature representation to learn and optimize problem solving strategies. The DeepMind team first proposed DQN (Deep Q Network) in 2013 for playing Atari video games and obtaining high scores [13]. Later, DQN appeared many variants, such as DDQN (Double Deep Q Network) [19], D3QN (Dueling Double Deep Q Network) [20] and DQN with prioritized experience replay [16].Currently, reinforcement learning has been widely used in the field of wireless communication resource allocation [23,24,26,1].

In this article, we apply a DDQN algorithm and propose a deep reinforcement learning framework called CNDDQN for cognitive radio networks. This deep reinforcement learning framework is used to solve the resource allocation problem in cognitive radio networks with network slicing. Under the cognitive radio network underlay model, this framework jointly optimizes the overall spectrum efficiency of the cognitive network and the QoE of the secondary users by managing the channel selection and power allocation of the secondary users. This framework learns the optimal resource allocation strategy by establishing a mapping between known primary user channel selection and power allocation strategies and secondary user channel selection and power allocation strategies. We first introduce a cognitive radio network model combined with network slicing. Secondly, we introduce the basic concepts of reinforcement learning algorithms, Q-Learning and DDQN algorithms. Subsequently, we show the details of the CNDDQN algorithm. Finally, we conduct simulation experiments on the CNDDQN algorithm to verify the stability and effectiveness of the CNDDQN algorithm.

The key contributions of this article are as follows:

1) This paper proposes a cognitive radio model in the 5G network slicing scenario, which provides effective help for the construction of 5G-based electric power wireless private network system.

2) The resource allocation algorithm proposed in this paper considers user QoE and jointly optimizes the network spectrum efficiency and user QoE to ensure the user experience.

3) This paper proposes a resource allocation algorithm based on DDQN to solve the overestimation problem of DQN algorithm.

The remaining chapters of this paper are arranged as follows. Section 2 introduces some research work related to this article. Section 3 introduces the system model of the cognitive radio network and the formulation process of the resource allocation problem. Section 4 introduces the proposed deep reinforcement learning algorithm (CNDDQN). The simulation results and analysis are in Section 5. We summarize this article in Section 6.

2. Related Work

Resource allocation in cognitive radio networks has been widely studied, [18,6] summarizes these existing studies. The main optimization objectives of resource allocation in cognitive radio network include maximizing throughput, spectrum efficiency and energy efficiency, minimizing interference and ensuring the quality of service of users. [7] proposes a distributed user association and resource allocation algorithm based on matching theory to maximize the total throughput of primary and secondary users. [9] proposes a method based on deep reinforcement learning for cognitive uplink users of cellular networks, and deployed some sensors to help secondary users collect signal strength information at different locations in the wireless environment. Therefore, the secondary user can realize spectrum sharing with the primary user without knowing the power allocation strategy of the primary user. However, [9] does not consider the channel selection strategy of secondary users.

As the key technology of 5G network, network slicing technology is considered in many kinds of resource allocation scenarios. There are some researches on the application of network slicing technology in cognitive radio network resource allocation scenarios [10,3]. In [15], the network slicing technology is classified into spectrum level, infrastructure level and network level network slicing. In [10], the allocation of wireless slicing resources among multiple users is modeled as a bankruptcy game, which realizes the fairness of allocation. [3] proposes a multi-time-scale cognitive radio network slicing resource allocation model. The resource allocation model can be decomposed into inter-slice subchannels pre-assignment in large time period and intra-slice subchannels and power scheduling in same time slot. [3] formulates the inter-slice problem as an integer optimization problem and intra-slice problem as a mixed optimization problem with integer variables, and adopts Lyapunov optimization method with heuristic subchannel assignment procedure and a fast barrier-based power allocation procedure. The above papers use traditional optimization methods, such as game theory and Lyapunov optimization. These traditional optimization methods need to transform the optimization objectives into convex optimization problems to obtain the optimal solution, which has certain restrictions on the communication network scenarios. For example, the locations of users are fixed, and more users will bring higher algorithm complexity and longer calculation time.

In order to solve the problem of resource allocation in complex communication network scenarios, we propose a reinforcement learning architecture to solve the problem of resource allocation optimization in communication networks. The existing reinforcement learning algorithms applied to resource allocation are mainly divided into distributed multi-agent reinforcement learning algorithm [5] and centralized single agent reinforcement learning algorithm [27,25]. The centralized algorithm needs global information, has better utility value, and can balance the whole network users. Distributed algorithm only needs to know local information, so it has less communication cost. [27] proposes a centralized reinforcement learning algorithm based on DQN, which uses underlay access mode to maximize the spectrum efficiency of secondary users under the interference temperature limit acceptable for the primary user. But the network model of [27] does not consider network slicing. [5] proposes a distributed reinforcement learning algorithm based on Q-Learning and SARSA. The secondary users are organized into a random dynamic team in a decentralized and cooperative way, which speeds up the convergence speed of the algorithm, improves the network capacity, and obtains the optimal energy-saving resource allocation strategy. But [5] only considers a single kind of service slice (high rate service slice) in the network model, and due to the use of table-based Q-learning and SARSA algorithm, the state space becomes discrete space, and there is a certain quantization error when segmenting the state space. [25] proposes a graph convolutional network-based reinforcement learning algorithm based on DQN. Secondary users are formed into a graph, and the information features are extracted by graph convolution, and then the DQN algorithm is used for policy learning to maximize the data rate of secondary users on the premise of the quality of service of users. In this paper, we propose a centralized reinforcement learning algorithm based on DDQN, and use DDQN algorithm to solve the problem of over estimation of DQN algorithm, so as to speed up the convergence speed and stability of the algorithm. In addition, in the network scenario, we consider the scenario where multiple service slices are combined with cognitive radio networks, and we consider rate-sensitive eMBB service slices and delay-sensitive URLLC service slices. In terms of optimization goals, if only the overall spectral efficiency of the cognitive network is optimized, this may sacrifice the user experience of some users. Therefore, we jointly optimize the spectral efficiency of the cognitive network and the user-perceived QoE of each user.

3. System Model and Problem Formulation

In this section, the system model and problem formulation are described.

3.1. System Model

This article considers a downlink OFDMA (Orthogonal Frequency Division Multiple Access) cellular cognitive network, as shown in Fig. 1. This network model has one PBS (Primary Base Station) and one CBS (Cognitive Base Station). The PUs (Primary Users) are associated with PBS, and the SUs (Secondary Users) are associated with the CBS. The PBS and CBS share the same spectrum resource. The SU adopts the underlay access model, and within the interference acceptance range of the PU, the SU is allowed to use the licensed frequency band resources of the PU. In Fig. 1, the black line indicates the communication between PU and PBS, the blue line indicates the communication between SU and CBS. The red line indicates the interference from the CBS to the PU, which should be controlled within a certain range.

3.2. Problem Formulation

In this scenario, secondary users are divided into two categories. The two types of secondary users have different service types and communication requirements. One type of secondary users are high-rate users, and the other type of secondary users are low-latency users. For these two types of secondary users, by using network slicing technology, high-rate users are associated with eMBB slices, and low-latency users are associated with URLLC slices. The set of secondary users associated with the eMBB slice is $SU^{eMBB} = \{1, 2, ..., N_{su}^{eMBB}\}$, and the set of secondary users associated with the URLLC slice is $SU^{URLLC} = \{1, 2, ..., N_{su}^{URLLC}\}$. Therefore, the set of all secondary

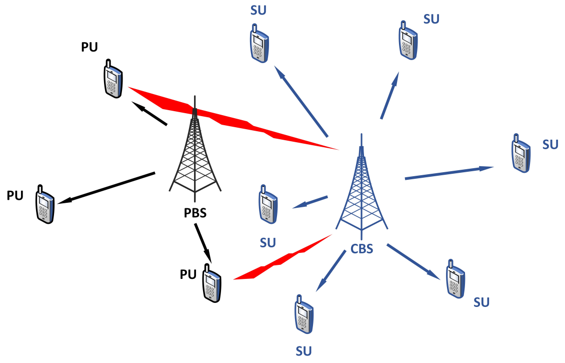


Fig. 1. Cognitive Radio Network Model

users is $SU = \{1, 2, ..., N_{su}\}$, where $N_{su} = N_{su}^{eMBB} + N_{su}^{URRLC}$ is the total number of secondary users. The total primary user set is $PU = \{1, 2, ..., N_{pu}\}$, where N_{pu} is the total number of primary users.

There are k channels for users to use. The channel set is $C = \{1, 2, ..., k\}$ and the bandwidth of each channel is B. Therefore, the total network bandwidth is $W = k * B$. Assuming that each primary user can occupy multiple channels at the same time, the primary user-channel association matrix is $PCA = \{a_{n,k}^{pc}\}_{N_{pu} * k}$. If PU n occupies the channel k, then $a_{n,k}^{pc} = 1$, otherwise $a_{n,k}^{pc} = 0$. Each secondary user can only occupy one channel, and the secondary user-channel association matrix is $SCA = \{a_{n,k}^{sc}\}_{N_{su} * k}$. If the secondary user n occupies the channel k, then $a_{n,k}^{sc} = 1$, otherwise $a_{n,k}^{sc} = 0$.

The channel gain matrix of each primary user and PBS is $PPG = \{g_n^{pP}\}_{N_{pu}}$, and the channel gain matrix of each primary user and CBS is $PCG = \{g_n^{pC}\}_{N_{pu}}$. The channel gain matrix of each secondary user and PBS is $SPG = \{g_n^{sP}\}_{N_{su}}$, and the channel gain matrix of each secondary user and CBS is $SCG = \{g_n^{sC}\}_{N_{su}}$.

Assume that the maximum transmission power of the PBS and CBS are $P_{\max}^{PBS}$ and $P_{\max}^{CBS}$ correspondingly. $P_{n,k}^{pu}$ indicates the transmission power of the primary user n on the channel k, and $P_{n,k}^{su}$ indicates the transmission power of the secondary user n on the channel k.

According to the definition of the signal-to-interference and noise ratio, (1)(2) are the expressions of the signal-to-interference ratio of the PU and SU.

$$\delta^{pu} = \frac{\sum\limits_{k \in C} a_{n,k}^{pc} \cdot g_n^{pu,PBS} \cdot P_{n,k}^{pu}}{\sum\limits_{a \in PU, a \neq n} \sum\limits_{k \in C} a_{n,k}^{pc} \cdot a_{a,k}^{pc} \cdot g_a^{pP} \cdot P_{a,k}^{pu} + \sum\limits_{a \in SU} \sum\limits_{k \in C} a_{n,k}^{pc} \cdot a_{a,k}^{sc} \cdot g_a^{pC} \cdot P_{a,k}^{su} + \sigma^2} . \tag{1}$$

$$\delta^{su} = \frac{\sum\limits_{k \in C} a_{n,k}^{sc} \cdot g_n^{sC} \cdot P_{n,k}^{su}}{\sum\limits_{a \in PU} \sum\limits_{k \in C} a_{n,k}^{sc} \cdot a_{a,k}^{pc} \cdot g_a^{sP} \cdot P_{a,k}^{pu} + \sum\limits_{a \in SU, a \neq n} \sum\limits_{k \in C} a_{n,k}^{sc} \cdot a_{a,k}^{sc} \cdot g_a^{sC} \cdot P_{a,k}^{su} + \sigma^2} . \tag{2}$$

According to the Shannon channel formula $R = B \cdot \log(1 + \delta)$, the transmission rate of the primary user R_n^{pu} and secondary user R_n^{su} can be calculated. Therefore, the total transmission rate of the cognitive network is $R_{cn} = \sum\limits_{n \in SU} R_n^{su}$. (3) is the total spectrum efficiency of the cognitive network.

$$\eta_{cn} = \frac{R_{cn}}{W} = \frac{\sum\limits_{n \in SU} B \cdot \log(1 + \delta_n^{su})}{k * B} = \frac{1}{k} \cdot \sum_{n \in SU} \log(1 + \delta_n^{su}) \,. \tag{3}$$

The user's QoE is mainly reflected by the user's communication needs. The user's QoE is defined as the ratio of the number of packets meeting the communication requirements to the total number of packets. The communication demand of eMBB slice users is that the transmission rate is higher than a certain threshold, and the communication demand of URLLC slice users is that the transmission delay is lower than a certain threshold.

The transmission rate of the data packet is expressed by the user's transmission rate, and the transmission delay of the data packet is composed as shown in Fig. 2.

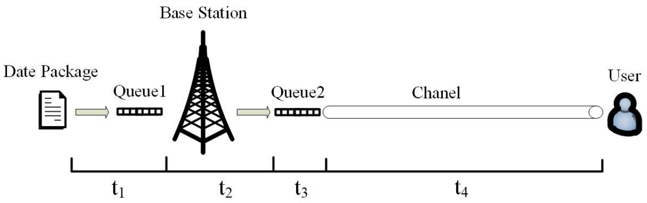


Fig. 2. Transmission Delay of Data Packet

The transmission delay of data packets is mainly composed of the queue delay (t_1) when entering the base station, the operation delay of the channel allocated at the base station (t_2), the queue delay of entering the channel (t_3), and the transmission delay of transmitting in the channel (t_4). In order to simplify the transmission delay model of the data packet, t_1 and t_2 belong to the transmission delay of the base station, the value of t_4 is very small, they are not considered in this paper. Therefore, the transmission delay of the data packet is the queue delay of the data packet entering the channel t_3. We use the M/M/1 queue model to calculate the queue delay. According to the average waiting time formula of the M/M/1 queue model $W_s = 1/(\mu - \lambda)$, where μ is the service rate and λ is the arrival rate. We can get the queue delay $t_3 = 1/(r_{package} - \lambda)$, where λ is the arrival rate of each data packet, $r_{package} = R_n/L$ is the transmission rate of each data packet, R_n is the transmission rate of the user, L is the packet length of the data packet. We assume that the packet length is normally distributed. Therefore, (4) is the transmission delay of the data packet.

$$t = \frac{1}{R_n/L - \lambda} \,. \tag{4}$$

Let $t_{\max}$ and $R_{\min}$ be the threshold for the data packet transmission delay and transmission rate to meet the communication requirements. The expression that meets the communication requirements is shown in (5). The user's QoE is equal to the ratio of the number of packets that meet the inequality requirements to the total number of packets.

$$\begin{cases} R_n \geq R_{\min} & \text{for} \quad \text{eMBB} \quad \text{users} \\ t = \frac{1}{R_n/L-\lambda} \leq t_{\max} & \text{for} \quad \text{URLLC} \quad \text{users} \end{cases} . \tag{5}$$

In order to balance the spectral efficiency and the user's QoE, we set the attention coefficient $\alpha \in [0,1]$ between the spectral efficiency and the user's QoE. $\alpha = 1$ means that the optimization goal is only to maximize the system spectral efficiency, and $\alpha = 0$ means that the optimization goal is only to maximize the user QoE. Therefore, our optimization goal is (6).

$$\max[\alpha\eta_{cn} + (1-\alpha)QoE] . \tag{6}$$

The interference temperature is defined as the ratio of the interference power to the corresponding bandwidth $IT = \frac{P_{\text{interference}}}{k_{cons}W}$, where $P_{\text{interference}}$ is the power of the interference noise in the channel, k_{cons} is the Boltzmann constant, and W is the total bandwidth of the cognitive network. Therefore, the total interference temperature of the cognitive network is (7).

$$IT = \frac{\sum\limits_{a\in SU}\sum\limits_{k\in C} a_{a,k}^{sc} \cdot g_a^{pC} \cdot P_{a,k}^{su}}{k_{cons} \cdot W} . \tag{7}$$

Let the maximum interference temperature caused by the cognitive network acceptable to the PU be $IT^{\max}$.

Constraint C1 indicates that each secondary user can only be associated with one channel. Constraint C2 is the maximum total power constraint of the cognitive base station. Constraint C3 is the main user's interference temperature constraint on the cognitive network.

Therefore, the optimization problem can be expressed as (8- 11).

$$\max[\alpha\eta_{cn} + (1-\alpha)QoE] . \tag{8}$$

$$s.t.C1 : \sum_{k\in C} a_{n,k}^{sc} \leq 1, \forall n \in SU . \tag{9}$$

$$C2 : 0 \leq \sum_{n\in SU} P_{n,k}^{su} \leq P_{\max}^{CBS} . \tag{10}$$

$$C3 : \frac{\sum\limits_{a\in SU}\sum\limits_{k\in C} a_{a,k}^{sc} \cdot g_a^{pC} \cdot P_{a,k}^{su}}{k_{cons} \cdot W} \leq IT^{\max} . \tag{11}$$

Due to the nonlinear constraints of continuous variables (such as $P_{a,k}^{su}$) and binary variables (such as $sca_{a,k}$), the optimization problem is a non-convex problem. Using deep reinforcement learning to solve such non-convex problems is a common method. Therefore, we propose a deep reinforcement learning algorithm to solve this optimization problem.

4. Deep Reinforcement Learning for Optimization Problem

4.1. Reinforcement Learning

Reinforcement learning is a common method for solving decision problems. Reinforcement learning has two basic elements (state and action). Performing a certain action in a certain state is a strategy. Agents need to obtain a good strategy in continuous exploration and learning. If the state is regarded as an attribute and the action is regarded as a mark, reinforcement learning is similar to supervised learning. They are all trying to find a mapping from known attribute/state to the mark/action. In this way, the strategy in reinforcement learning is equivalent to the classifier and regressor in supervised learning. However, in practical problems, reinforcement learning does not have supervised learning as labeled information. Usually results are obtained after trying actions, so reinforcement learning is to continuously adjust the previous strategy through the feedback of the result information, so the algorithm can learn what kind of action to choose in which state to get the best result.

Reinforcement learning is usually described using MDP (Markov Decision Process). The agent is in an environment, and each state is the agent's perception of the current environment. The agent can only affect the environment through actions. When the agent performs an action, the environment will be transferred to another state with a certain probability. At the same time, the environment will feedback a reward to the agent according to the potential reward function. This process is shown in Fig. 3.

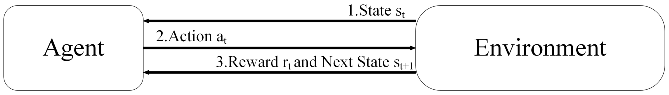


Fig. 3. Basic Process of Reinforcement Learning

Then, we define two value functions—state value function and action value function. The state value function $V(s)$ is defined as the expectation of the long-term reward that the state s can obtain at the moment t. The state value function represents the value of a state, regardless of which action the state chooses. It takes the current state as the starting point to make a weighted sum of all possible actions, the expression is $V_\pi(s) = E_\pi[R_t|S_t = s]$, where π is the strategy, and the expression is $\pi(a|s) = P[A_t = a|S_t = s]$. The action value function $G(s, a)$ is defined as the long-term reward that can be obtained by selecting the action a in state s at the moment t. The action value function represents the value of an action in a certain state. It is the weighted sum of all possible long-term rewards for a given state and action, the expression is $G_\pi(s, a) = E_\pi[R_t|S_t = s, A_t = a]$.

Usually, a limited Markov decision process consists of a quadruple $M = (S, A, P, R)$. Where S represents the limited state set space, A represents the action set space, P represents the state transition probability matrix, and R represents the expected reward value. The Markov decision process relies on the Markov assumption that the probability of the next state S_{t+1} depends only on the current state S_t and action A_t, not on the previous state or action. In the Markov decision process, given a state $s \in S$ and an action $a \in A$, it will transition to the next state $s' \in S$ with a certain probability. $P^a_{ss'}$ is the state transition probability, which means that starting from the state s and taking action a, we will reach the state s' with the probability of $P^a_{ss'}$, the expression is $P^a_{ss'} = P(S_{t+1}|S_t = s, A_t = a)$. $r^a_{ss'}$ is the expected reward, which means starting from the state s, taking action a, and transferring to the state s', the expression is $r^a_{ss'} = E(r_{t+1}|S_t = s, A_t = a, S_{t+1} = s')$.

Because reinforcement learning can be summarized as obtaining an optimal strategy by maximizing rewards. However, if it is only the maximum instantaneous reward, it will only select the action with the largest reward from the action space every time, which becomes the simplest greedy policy. In order to achieve the maximum current reward value including the future, the total reward from the current moment until the end state reaches the goal is maximized. Therefore, the cumulative discount reward function $R(t)$ is constructed with the expression as $R(t) = \sum\limits_{k=0}^{n} \gamma^k r_{t+k+1}$, where $\gamma \in [0, 1]$ is the discount coefficient, which indicates the degree of influence of the current reward in the future. $\gamma = 0$ means that the learned strategy is short-sighted and only considers even rewards and $\gamma = 1$ means that the rewards at all times are equal. Combining the definition of the state value function and the cumulative discount reward function, we can obtain the Bellman equation form of the state value function, as shown in (12- 16).

$$\begin{aligned}
V_\pi(s) &= E_\pi[R_t|S_t = s] && (12)\\
&= E_\pi(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots|S_t = s) && (13)\\
&= E_\pi(r_{t+1}|S_t = s) + E_\pi(\gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|S_t = s) && (14)\\
&= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}\{R^a_{ss'} + \gamma E_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2}|S_{t+1} = s']\} && (15)\\
&= \sum_a \pi(s,a) \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V_\pi(s')] && (16)
\end{aligned}$$

Combining the definition of the action value function and the cumulative discount reward function, we can obtain the Bellman equation form of the action value function through a similar derivation process, as shown in (17). (18) and (19) are Bellman optimality equations.

$$G_\pi(s,a) = \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma \sum_{a'} G_\pi(s',a')]\,. \tag{17}$$

$$V_*(s) = E[R_t + \gamma \max_\pi V(s')|S_t = s]\,. \tag{18}$$

$$Q_*(s) = E[R_t + \gamma \max_{a'} Q(s',a')|S_t = s, A_t = a]\,. \tag{19}$$

The most common reinforcement learning algorithm is the Q-Learning algorithm. By introducing Q-Table, the action value function is described. The update formula of Q-Learning is (20). By constantly updating, we can get an excellent Q-Table to make the decision-making process.

$$Q(s,a) = Q(s,a) + \alpha[R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)]\,. \tag{20}$$

4.2. Deep Reinforcement Learning: from Q-Learning to DQN

Q-Learning is a classic algorithm for reinforcement learning, but there is a problem that Q-Learning uses a Q-Table to store Q values. This makes Q-Learning limited to the action space and the state space are very small, and generally in discrete situations. If there are many types of states and actions in the model, the size of the Q-Table will become very large, even larger than the memory of the computer, and it is also very time-consuming to search in a huge table for each update. However, more complex tasks that are closer to the actual situation often have a large state space and action space. For the field of processing high-dimensional data, deep learning has a good performance. Deep reinforcement learning combines reinforcement learning and deep learning, using neural networks instead of the original table to calculate the value function.

DQN is a representative algorithm for deep reinforcement learning. Based on the original Q-Learning used Q-tables, the Q value (action value function) is calculated using a neural network in DQN algorithm. In the decision-making process, DQN takes the state as the input of the neural network, calculates the Q value of each action through the neural network, and then selects the action according to the principle similar to Q-Learning. Fig. 4 compares the Q value calculation process of Q-Learning and DQN. The original Q value $Q(s,a)$ is replaced by a new form with neural network parameters $Q(s,a;\theta)$, where θ represents the parameters of the neural network.

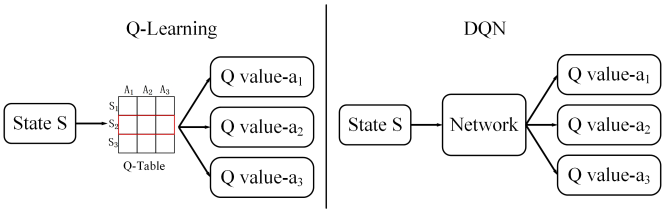


Fig. 4. Comparison of Calculation Process for Q of Q-learning and DQN

In order to reduce the problems caused by the correlation between data, DQN introduced two key technologies of experience replay and fixed target value network.

In supervised learning, each sample is independently identically distribution. However, the samples of reinforcement learning are obtained through the agent's continuous exploration, which makes the samples in reinforcement learning highly correlated and non-stationary, causing the training results difficult to converge. The experience replay technology is used to solve this problem. First put the collected samples into the sample pool, and then randomly select a sample from the sample pool for network training. Random sampling is used to remove the correlation between samples, making the samples independent of each other, thereby improving the stability and convergence of network training.

In the original Q-Learning, as described in (20), when we calculated the TD error, we obtained it by calculating the difference between the target Q and estimated Q. The calculation of the TD target is by using the Bellman equation. The TD target is the reward of the current action plus the highest Q value of the next state through attenuation. However, the same parameters are used when calculating the TD target and estimating the Q value. The correlation between the two makes the model prone to oscillation and divergence. In order to solve this problem, DQN builds an independent target Q network that is slower than the current Q network to calculate the TD target, which makes the possibility of oscillation and divergence during training reduced and more stable.

In Q-Learning, updating the Q value directly changes the value of the corresponding position in the table. In DQN, the Q value is updated by updating the parameters of the neural network. The update of the neural network parameters is based on the reverse transfer of the loss function. The loss function of DQN is defined as the square error form of target Q and estimated Q. (21) is the form of the loss function of DQN.

$$Loss^{DQN} = [r + \gamma \max_{a'} Q(s', a'; \theta^{-}) - Q(s, a; \theta)]^2 . \tag{21}$$

DQN still has the problem of overestimation. Overestimation means that the estimated value function is larger than the real value function, and its root is mainly in the maximization operation in Q-Learning. When calculating target Q, the maximum Q value in the next state is obtained. For real strategies and in a given state, the action that maximizes the Q value is not selected every time, because the general real strategies are random strategies, the selection of the maximum Q value of the action here will often result in the target value being higher than the real value. Double DQN solves the problem of overestimation on the basis of DQN. DDQN implements action selection and action evaluation with different value functions, and in DQN we have proposed two Q networks. Therefore, the step of DDQN calculating target Q can be split into two steps. In the first step, the action to maximize the Q value is obtained by estimated Q network. In the second step, the action value function corresponding to the action is obtained through the target Q network. Combining the two steps together, the loss function of DDQN can be obtained, as shown in (22).

$$Loss^{DDQN} = [r + \gamma Q(s', \underset{\mathrm{a}'}{\operatorname{argmax}}\, Q(s', a'; \theta); \theta^{-}) - Q(s, a; \theta)]^2 . \tag{22}$$

Except for the change of the loss function, the main process of DDQN is the same as that of DQN. Fig. 5 is a flowchart of the operation of the DDQN algorithm.

4.3. CNDDQN (Cognitive Network Double Deep Q Network)

In this paper, we propose a Double DQN algorithm for solving the channel selection and power allocation problems in cognitive networks. In the cognitive network environment, the basic elements of reinforcement learning are set as follows.

The reinforcement learning agent is the overall cognitive network, and the DDQN algorithm runs in the CBS to manage the channel selection and power allocation of all cognitive users. The state of reinforcement learning is the SINR of the PU, which is recorded as $s_t = \{\delta_t^n\}_{1*N_{pu}}$.

The reinforcement learning action $a_t = \left\{\{a_t^{sc,n}\}_{1*N_{su}}, \{P_t^{su,n}\}_{1*N_{su}}\right\}_{1*2\cdot N_{su}}$ is the channel selection of the secondary user and the power allocation of the secondary user. Since the output of the DDQN algorithm is a discrete value, we divide the transmission power of cognitive users into 20 discrete power values on average. (23) is the action space of the transmission power of cognitive users.

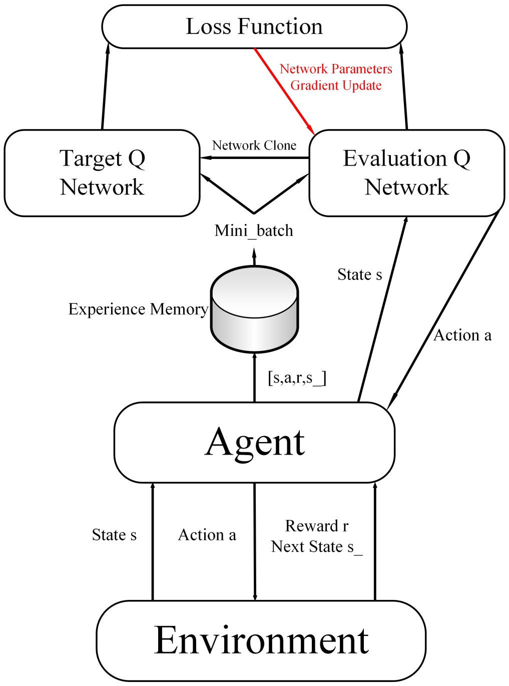


Fig. 5. Flowchart of DDQN Algorithm

$$P_{n,k}^{su} \in \{0,\frac{P_{\max}^{CBS}}{19},\frac{2P_{\max}^{CBS}}{19},...,P_{\max}^{CBS}\}\,. \tag{23}$$

The reward function is modified on the basis of (8). The constraint condition C3 for interference temperature is added to the description of the reward function. If the constraint condition of the interference temperature is satisfied, a normal reward will be obtained. If the constraints of the interference temperature are not met, then only zero rewards can be obtained. The characteristics of the step function meet our expectations. We make the difference between the actual interference temperature and the interference temperature threshold to obtain the interference temperature threshold constraint function (24).

$$f(a_{a,k}^{sc},P_{a,k}^{su}) = \varepsilon(IT^{\max} - \frac{\sum\limits_{a\in SU}\sum\limits_{k\in C} a_{a,k}^{sc}\cdot g_a^p \cdot P_{a,k}^{su}}{k\cdot W})\,. \tag{24}$$

$\varepsilon(x)$ is a step function, its characteristic is $\varepsilon = \begin{cases} 1 & x > 0 \\ 0.5 & x = 0 \\ 0 & x < 0 \end{cases}.$

The step function is discontinuous at 0, making it difficult during gradient descent. Therefore, we use the Sigmoid function to approximate the equivalent step function. Therefore, the interference temperature threshold constraint function is expressed as (25).

$$f(a^{sc}_{a,k}, P^{su}_{a,k}) = Sigmoid(IT^{\max} - \frac{\sum\limits_{a\in SU}\sum\limits_{k\in C} a^{sc}_{a,k}\cdot g_a^{pu,CBS}\cdot P^{su}_{a,k}}{k\cdot W}). \quad (25)$$

The expression of Sigmoid function is $Sigmoid(x) = \frac{1}{1+e^{-x}}$. Combined with the interference temperature threshold constraint function, the reward function can be defined as shown in (26).

$$\text{Reward} = f(a^{sc}_{a,k}, P^{su}_{a,k}) * [\alpha\eta_{cn} + (1-\alpha)QoE]/,. \quad (26)$$

There are two neural networks in this algorithm, the training network and the target network. These two networks have the same structure, but the parameter updates are different. The neural network in this algorithm uses a simple fully connected neural network. The fully connected neural network contains 2 fully connected layers, the structure is shown in Fig. 6.

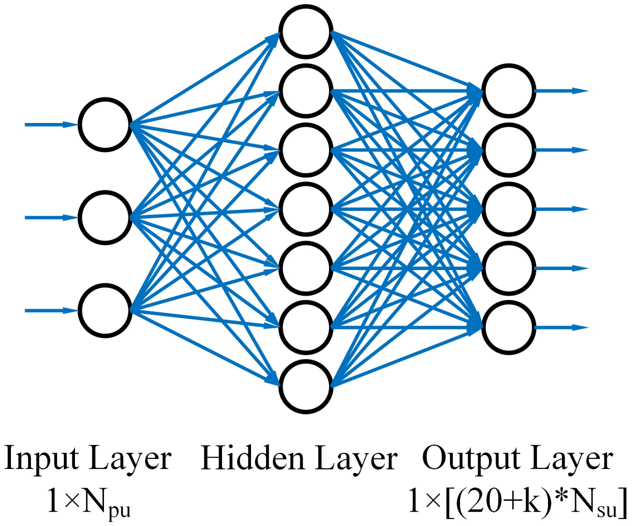


Fig. 6. Neural network structure (N_{pu}: total number of primary users, N_{su}: total number of secondary users, k: total number of channels)

The general steps of CNDDQN is summarized as shown in Algorithm 1.

5. Performance Evaluation

In this section, we first introduce the settings of cognitive network model parameters and DDQN hyperparameters. Then, we provide the simulation performance results.

Algorithm 1 The General Steps of CNDDQN

```
Initialize replay memory $D$ to capacity $N$
Initialize action-value function $Q$ with random weights $\theta$ and target action-value function $\overset{\wedge}{Q}$ with weights $\theta^{-} = \theta$
for each episode, $M$ do
    Initialize network state $s$;
    for each step of an episode, $T$ do
        CBS chooses an action $a_t = \arg\max_a Q(\phi(s_t), a; \theta)$ at state $s_t$ with probability $\varepsilon$ select a random action $a_t$;
        CBS completes channel and power allocation according to the selected action $a_t$;
        CBS calculates the reward $r_t$ according to ( 26) through message passing;
        CBS observes the network state $s_{t+1}$ through message passing;
        CBS stores transition $(s_t, a_t, r_t, s_{t+1})$ in $D$;
        CBS samples random minibatch of transitions $(s_t, a_t, r_t, s_{t+1})$ from $D$;
        $y_j = \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \overset{\wedge}{Q}(s', \underset{\mathrm{a}'}{\operatorname{argmax}} Q(s', a'; \theta); \theta^{-}) & \text{otherwise} \end{cases}$;
        CBS performs a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the network parameters $\theta$;
        Every $C$ steps, CBS resets $\overset{\wedge}{Q} = Q$;
        CBS sets $s_t = s_{t+1}$;
    end for
end for
```

5.1. Simulation Settings

In this model, there are 1 PBS and 1 CBS, and there are 10 PUs and 20 CUs. Among the 20 secondary users, 10 secondary users are associated with eMBB slices and 10 secondary users are associated with URLLC slices. The size of the model is 100m * 100m. The locations of base stations and users are fixed, and the distribution of base stations and users is shown in Fig. 7.

For the PBS and CBS, the maximum transmission power is $P_{\max}^{PBS} = P_{\max}^{CBS} = 46dBm$. For AWGN channels, the noise power is $\sigma^2 = 1e - 7$. The standard deviation of shadow fading is set to 8dB. For model simplicity, channel fading only considers large-scale fading, the expression is $L(d) = 37 + 30\log(d)$, where d is the distance between the base station and the user. The network has a total of 20 channels, the bandwidth of each channel is 180kHz. For cognitive radio networks, the interference temperature acceptable to the primary user is 5dB. For the user's QoE, the rate threshold is set to 0.1Mbps and the delay threshold is 10ms. The hyperparameter settings for the DDQN algorithm are shown in Table 1.

5.2. Simulation Results

First, we show the performance of the DDQN algorithm at different learning rates. In the DDQN algorithm, the setting of the learning rate is very important. In the gradient descent process, the learning rate represents the step size of each update. Fig. 8 is a graph comparing the performance of the DDQN algorithm at different learning rates.

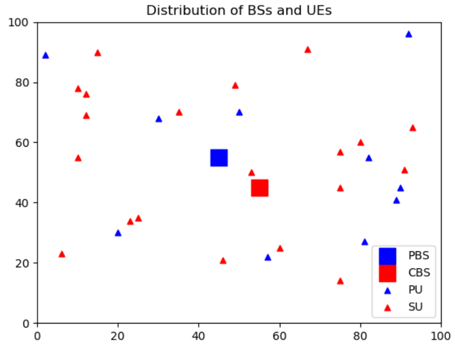


Fig. 7. Distribution of BSs and UEs

Table 1. Hyperparameters of DDQN Algorithm

Papameter	Value
Mini-batch size	32
Discount rate γ	0.995
Learning rate δ	0.005
ε-greedy	0.1
Activation function	Relu
Optimizer	Adam

The three curves in the figure are the images of the DDQN reward function value with the number of iterations when the learning rate is 0.05, 0.005 and 0.0005. When the learning rate value is small ($\delta = 0.0005$), the CNDDQN algorithm converges in about 1500 iterations, and the convergence speed is slow. As the learning rate increases, when $\delta = 0.005$,the convergence speed of the CNDDQN algorithm increases, and the CNDDQN algorithm converges in about 700 iterations. When the learning rate is too large ($\delta = 0.05$), the CNDDQN algorithm converges in about 400 iterations. But the final reward function convergence value is lower than the reward function convergence value when the learning rate is 0.005 and 0.0005. It can be seen that a low learning rate will lead to a slower convergence rate, requiring more iterations to achieve convergence. Too high a learning rate will cause CNDDQN to reach the final reward function convergence value lower than the normal learning rate convergence value. Therefore, the choice of learning rate should be moderate, too high and too low learning rate will make the performance of the algorithm decline. In the simulation of this paper, the learning rate $\delta = 0.005$ is an appropriate value.

We observe the curve of learning rate $\delta = 0.005$ in Fig. 8. We can find that the reward function value is low and unstable at the beginning of the iteration. As the training iteration progresses, the reward function continues to grow. After a certain number of iterations, the reward function completes convergence. This means that the CNDDQN algorithm has learned the optimal action strategy. After the CNDDQN algorithm converges, the jitter of the reward function is caused by ε-greedy in the CNDDQN algorithm.

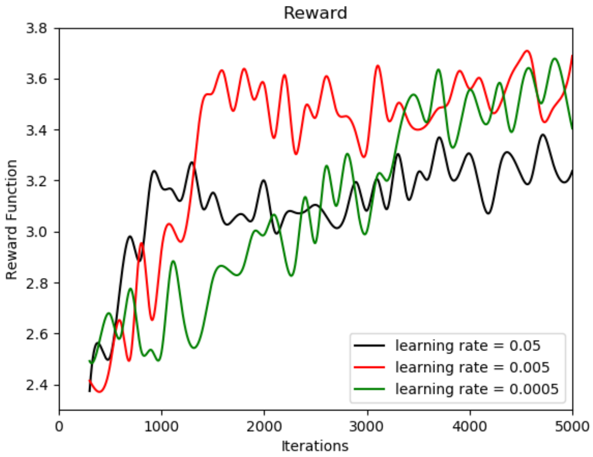


Fig. 8. Comparison of DDQN Performance with Different Learning Rates

The optimization objective in (6) is obtained by the linear combination of system spectral efficiency and QoE, and the attention coefficient is α. Fig. 9 shows the change curves of the average convergence value of user QoE and system spectral efficiency under different coefficients. As the attention factor increases, the CNDDQN algorithm's attention to the system spectral efficiency increases, the final average convergence value of the system spectral efficiency increases, and the final average convergence value of the user QoE decreases. It can be seen that a greater attention to system spectrum efficiency can result in a more superior system spectrum efficiency performance strategy, but at the same time it will cause a certain loss to the user's QoE performance. Similarly, in order to obtain superior user QoE performance strategies, a certain loss will be caused to the system spectrum efficiency.

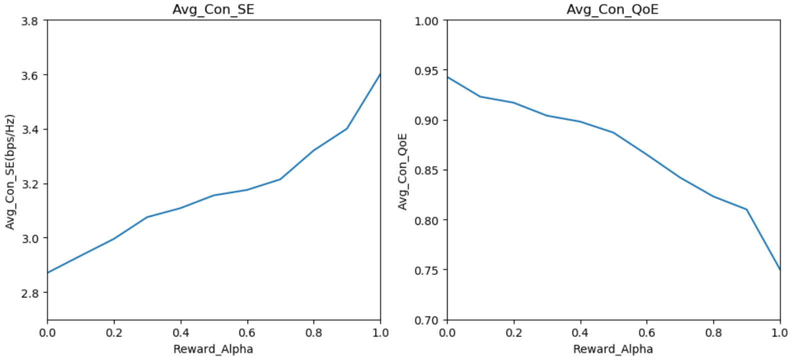


Fig. 9. Influence of Different Attention Coefficients on SE and User QoE

The comparative experiment algorithm selected in this paper is CNDQN algorithm and CNQ-learning algorithm, which is shown in Fig. 10. Compared with the CNDQN algorithm, the convergence speed of the CNDDQN algorithm has been significantly improved. The CNQ-learning algorithm uses a table to store Q values, so that not only the action space is discrete, but the state space is also discrete. This makes CNQ-learning's overall performance far from CNDDQN in complex cognitive radio scenarios.

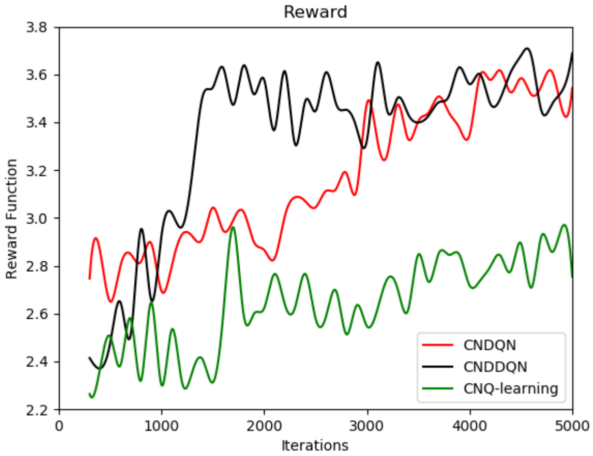


Fig. 10. Performance Comparison of Different Algorithms in Cognitive Radio Networks

6. Conclusion

In this article, we propose a resource allocation algorithm (CNDDQN) for cognitive radio with network slicing. This algorithm is used in cognitive radio scenarios in underlay mode. Under the interference acceptable to the primary user, the secondary user is allowed to access the frequency band authorized to the primary user. In order to quantify the interference caused by secondary users, we introduce the concept of interference temperature. In order to solve the proposed non-convex and NP-hard problem of resource allocation, we use a deep reinforcement learning algorithm (DDQN). The algorithm jointly optimizes the overall spectrum efficiency of the cognitive network and the QoE of the secondary user by managing the channel selection and power allocation of the secondary user. Through continuous iterative learning, the algorithm continuously updates the resource allocation strategy of the secondary users, and finally reaches the optimal resource allocation strategy. Simulation results show that compared with other reinforcement learning methods, the proposed CNDDQN can effectively achieve a near-optimal solution through a smaller number of iterations.

Acknowledgments. The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of this paper. This work is supported by the National Natural Science Foundation of China under Grant No.61971057.

References

1. Chen, J., Chen, S., Wang, Q., Cao, B., Feng, G., Hu, J.: iraf: A deep reinforcement learning approach for collaborative mobile edge computing iot networks. IEEE Internet of Things Journal 6(4), 7011–7024 (2019)
2. Guo, D., Zhang, Y., Xu, G., Hyeongchun, P.: Spectrum aggregation scheme in a wireless broadband data transceiver system. international conference on robotics and automation 33(5) (2018)
3. Jiang, H., Wang, T., Wang, S.: Multi-scale hierarchical resource management for wireless network virtualization. IEEE Transactions on Cognitive Communications and Networking 4(4), 919–928 (2018)
4. Katsalis, K., Nikaein, N., Schiller, E., Ksentini, A., Braun, T.: Network slices toward 5g communications: Slicing the lte network. IEEE Communications Magazine 55(8), 146–154 (2017)
5. Kaur, A., Kumar, K.: Energy-efficient resource allocation in cognitive radio networks under cooperative multi-agent model-free reinforcement learning schemes. IEEE Transactions on Network and Service Management 17(3), 1337–1348 (2020)
6. Kumar, A., Kumar, K.: Multiple access schemes for cognitive radio networks: A survey. Physical Communication 38, 100953 (2020)
7. LeAnh, T., Tran, N.H., Saad, W., Le, L.B., Niyato, D., Ho, T.M., Hong, C.S.: Matching theory for distributed user association and resource allocation in cognitive femtocell networks. IEEE Transactions on Vehicular Technology 66(9), 8413–8428 (2017)
8. Li, R., Zhao, Z., Sun, Q., Chihlin, I., Yang, C., Chen, X., Zhao, M., Zhang, H.: Deep reinforcement learning for resource management in network slicing. IEEE Access 6, 74429–74441 (2018)
9. Li, X., Fang, J., Cheng, W., Duan, H., Chen, Z., Li, H.: Intelligent power control for spectrum sharing in cognitive radios: A deep reinforcement learning approach. IEEE Access 6, 25463–25473 (2018)
10. Liu, B., Tian, H.: A bankruptcy game-based resource allocation approach among virtual mobile operators. IEEE Communications Letters 17(7), 1420–1423 (2013)
11. Ma, T., Zhang, Y., Wang, F., Wang, D., Guo, D.: Slicing resource allocation for embb and urllc in 5g ran. Wireless Communications and Mobile Computing 2020, 1–11 (2020)
12. Mitola, J., Maguire, G.Q.: Cognitive radio: making software radios more personal. IEEE Personal Communications 6(4), 13–18 (1999)
13. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv: Learning (2013)
14. Pengpeng, L.I., Zheng, N., Kang, P., Tan, H., Fang, J.: Overview and inspiration of global 5g spectrum researches. Telecommunication Engineering (2017)
15. Richart, M., Baliosian, J., Serrat, J., Gorricho, J.L.: Resource slicing in virtual wireless networks: A survey. IEEE Transactions on Network and Service Management 13(3), 462–476 (2016)
16. Schaul, T., Quan, J., Antonoglou, I., Silver, D.: Prioritized experience replay. arXiv: Learning (2015)
17. Tang, L., Tan, Q., Shi, Y., Wang, C., Chen, Q.: Adaptive virtual resource allocation in 5g network slicing using constrained markov decision process. IEEE Access 6, 61184–61195 (2018)
18. Tarek, D., Benslimane, A., Darwish, M., Kotb, A.M.: Survey on spectrum sharing/allocation for cognitive radio networks internet of things. Egyptian Informatics Journal (2020)

19. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning pp. 2094–2100 (2016)
20. Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., De Freitas, N.: Dueling network architectures for deep reinforcement learning pp. 1995–2003 (2016)
21. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8(3-4), 279–292 (1992)
22. Yongjun, X., Xiaohui, Z.: Optimal power allocation for multiuser underlay cognitive radio networks under qos and interference temperature constraints. China Communications 10(10), 91–100 (2013)
23. Zhang, Y., Kang, C., Ma, T., Teng, Y., Guo, D.: Power allocation in multi-cell networks using deep reinforcement learning pp. 1–6 (2018)
24. Zhang, Y., Kang, C., Teng, Y., Li, S., Zheng, W., Fang, J.: Deep reinforcement learning framework for joint resource allocation in heterogeneous networks pp. 1–6 (2019)
25. Zhao, D., Qin, H., Song, B., Han, B., Du, X., Guizani, M.: A graph convolutional network-based deep reinforcement learning approach for resource allocation in a cognitive radio network. Sensors 20(18), 5216 (2020)
26. Zhao, N., Liang, Y., Niyato, D., Pei, Y., Jiang, Y.: Deep reinforcement learning for user association and resource allocation in heterogeneous networks pp. 1–6 (2018)
27. Zheng, W., Wu, G., Qie, W., Zhang, Y.: Deep reinforcement learning for joint channel selection and power allocation in cognitive internet of things. In: International Conference on Human Centered Computing. pp. 683–692. Springer (2019)

Siyu Yuan, received the B.E in electronic science and technology from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Electronic Engineering, BUPT, Beijing, China. His research interests include reinforcement learning, cognitive network slicing and wireless network resource allocation. Email:yuanisyu@bupt.edu.cn

Zhang Yong, received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China in 2007. He is a Professor with the School of Electronic Engineering, BUPT. He is currently the Director of Fab. X Artificial Intelligence Research Center, BUPT. He is the Deputy Head of the mobile internet service and platform working group, China communications standards association. He has authored or coauthored more than 80 papers and holds 30 granted China patents. His research interests include Artificial intelligence, wireless communication, and Internet of Things. Email: yongzhang@bupt.edu.cn

Qie Wenbo,received the B.E in Yanshan University, Beijing, China, in 2018. She is currently pursuing the M.S in electronic science and technology, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include reinforcement learning, cognitive network slicing, and resource allocation. Email:qwb@bupt.edu.cn

Ma Tengteng, received the B.S degree from the School of Science, Qufu Normal University, Jining, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China. Her current research interests include network slicing, QoS, cognitive radio and virtual network resource allocation. Email: buptteng@foxmail.com

Li Sisi, received the B.S degree from Beijing University of Posts and Telecommunications (BUPT) in 2019. She is currently pursuing the Ph.D. degree in computer science and technology at BUPT. Her research areas include wireless communication, network slicing, network resource allocation, mobile edge computing. Email: ssl123@bupt.edu.cn

Received: July 10, 2020; Accepted: January 12, 2021.

Computer Science and Information Systems 18(3):1001–1021 https://doi.org/10.2298/CSIS200422016T

Patient Length of Stay Analysis with Machine Learning Algorithms

Savo Tomović

Faculty of Mathematics and Natural Sciences,
Univeristy of Montenegro, 81000 Podgorica, Montenegro
savot@ucg.ac.me

Abstract. In this paper the problem of measuring factor importance on patient length of stay in an emergency department is discussed. Historical dataset contains average patient length of stay per day. Factors are agreed with domain expert. The task is to provide factors' impact measure on specific day that does not belong to the historical dataset (new observation) and average length of stay for that day is higher than specified threshold. Observations are represented as multidimensional numeric vectors. Each dimension represents factor. The basic idea consists of identifying appropriate neighbourhood and measure distances between the new observation and its neighbourhood in the historical dataset with respect to each factor. Impact measure of a factor is derived from the Error Sum of Squares. Factor impact is proportional to distance between the observation and its neighbourhood with respect to the dimension representing that factor. Nearest neighbour and clustering methods for neighbourhood determination are considered.

Keywords: length of stay analysis, nearest neighbours, clustering, SSE

1. Introduction

In this paper the problem of explaining patient length of stay (LOS) elevation in an emergency department is discussed. In the rest of the paper the length of stay explanation problem is referred to as LOSEP.

The task is to identify the most significant factors and objectively measure their impact on patient length of stay. Historical dataset is available. It contains average patient length of stay per day along with a set of features - factors. User is interested in investigating new observations - dates that do not belong to the historical dataset and for which registered average length of stay is higher than a threshold σ. The aim is to identify which factors are the most significant in contributing to longer patient stay in an emergency department, providing the leadership to improve concrete aspects in the organization.

More precisely, let the historical dataset H is given. It contains average length of stay per day along with available features - factors. Features are representing organizational or other aspects that potentially can cause longer patient stay than it is desired or expected. Available features are referred to as *factors* in the rest of the paper. Every record from H is represented with multidimensional numeric vector of the following form $(factor_1, factor_2, ..., factor_n, LOS)$. Let $q \notin H$ represents object $q = ((factor_1(q), factor_2(q), ..., factor_n(q), LOS(q))$ such that $LOS(q) > \sigma$. In the rest of the paper objects like q are referred to as *new observations*. The task is to create a methodology to objectively estimate impact of each factor on the length of stay augmentation registered on the new observation q. Factors can be sorted with respect to the estimated impacts. Such ordering determines the most significant factors causing the situation $LOS(q) > \sigma$.

Length of stay is considered as one of the most important indicators for any hospital department efficiency. The compulsion is to keep length of stay below some value. Such value is not simply average or median, but it is estimated by specific methodology. In large hospitals it is challenging for management to manually analyse every organizational aspect that can affect length of stay. The results of such analysis should assist leaders of hospitals and its staff in understanding what is happening on daily basis and what actions should be taken.

The LOSEP problem is different from problems of length of stay prediction and determining factors influencing patient length of stay where the overall impact of the contributing factors is derived from correlations existing in the training dataset. Such approaches usually create model based on available dataset and use the model to estimate the impact of each factor to the target variable - length of stay in this case. Such estimation is "static" meaning that for every new observation the model will produce the same factor ranking without considering any specificities related to concrete object. For example, the model can detect the highest importance of the number of emergency department visits based on the training dataset. So, for any new observation representing days when average length of stay is higher than σ the answer will be the same: length of stay is elevated due to higher number of visits. Of course, length of stay can be high because number of visits is elevated, but only this factor may not be an issue especially if the patients were of low triage level. Investigating the number of the sickest patients can indicate that this factor is also of significant importance.

The method presented in this paper is able to independently analyse every new observation and provide factors ranking related to specificities of the considered observation. In general, it is possible to obtain different explanations and factors ranking for each new observation. In such manner the solution can cover dynamical aspects of hospital behaviour meaning that on different days different aspects can be of different importance.

The motivation for this study originates from the challenges in designing and implementing system devoted to hospital management in American healthcare system. In the case study that is exposed in the fifth section an emergency department is considered. Data about average length of stay - *AVG_LENGTH_OF_STAY* per day expressed in hours is stored in historical dataset. Factors of interest are defined as follows: number of visits - *ED_VISITS*, number of ambulance visits - *AMBULANCE_VISITS*, average triage level - *AVG_TRIAGE_LEVEL*, number of patients with triage level 1 and 2 (most sick patients) - *TRIAGE_LEVEL_1_2_COUNT* and diversion hours - *DIVERSION_HOURS*. Granularity of the historical dataset is on a day level.

The proposed solution consists of two components. The first component is to find appropriate neighbourhood of a new observation. Nearest neighbour and clustering methods for neighbourhood determination are considered. With the nearest neighbour method the algorithm finds *k* closest objects from historical dataset to construct neighbouring cluster. In clustering method observations from the historical dataset are clustered in preprocessing step and the cluster with the closest centroid to a new observation is considered as neighbouring cluster. Standard Euclidean distance is used to determine the distance between objects.

The second component of the solution is for objective impact estimation of each factor. The proposed procedure calculates increment to the Error Sum of Squares if the new observation was added to the neighbouring cluster and distributes the increment value among factors proportionally.

The paper is decomposed into several sections as follows. The next section is devoted to a number of studies available in the literature, where we try to compare with and signify our contribution. Section 3 presents a motivating example for this research study. Two procedures for a new observation explanation are exposed in the third section. The same section introduces the function for objective impact measurement of each factor. The applicability of the proposed method is demonstrated in section 5 on the case study related to an emergency department. The last section contains concluding remarks and possible extensions to the proposed method.

2. Related work

There is a huge amount of scientific papers dealing with computer applications in medical research. For example, results from almost 300 papers appeared in numerous journals and conferences between 1999 and 2013 were presented in [12]. Specifically, analysis of healthcare services quality occupies significant effort in research community. Length of stay is considered as the most important indicator for any department efficiency, especially from the patient's perspective.

Many studies with objective to predict length of stay and identify and quantify impact of different factors were presented. Number of examined factors is huge and some of them were even more carefully interpreted regrading department specificity.

In [2] authors provide detailed review of length of stay applications and methods to calculate and predict length of stay. Authors classified algorithms into four categories: arithmetic methods, statistical methods, data-driven approaches and multi-stage models.

Arithmetic methods are the simplest. They assume that length of stay is normally distributed [2] and usually calculate average length of stay or median [7]. These measures can be misleading [26] because typically length of stay has an exponential distribution [2].

Statistical methods can be categorized into two subgroups [2]: survival analysis and regression analysis.

Survival analysis [11] uses length of stay as surrogate to estimate the impact of patient data on survival time.

Regression analysis can be considered as a statistical method that identifies factors which possibly predict length of stay. There is a huge number of analysed factors, internal and external, including organizational factors (patient arrival time, physicians and nurse characteristics, physicians and nurse shift changes, admission to specific hospital wards, laboratory performance, imaging, consultation, etc.), demographic data (age, gender, martial status, occupation, place of residence, etc.), information related to hospitalization (diagnosis related group - DRG, specialty of physician, history of admission to hospital, triage acuity level, type of admission, type of treatment, patient condition, method of payment for hospital costs) [8], [1], [21], [9], [20], [4]. Within this type, among others, linear regression and logistic regression and regression trees are found [32], [23], [10]. Percent of length of stay variation that could be explained with this approach vary from about 35% to almost 70%. As it is known from the literature and stated in some of these studies, for regression analysis there are several assumptions that must be satisfied: linearity, normality, homoscedasticity, data must not show multicollinearity, etc. [29].

In [18] authors claim that models based on regression analysis are heavily dependant on available data and even minor change in the data can generate completely different models from which different patterns or rules are extracted. Authors propose the procedure to create diverse regression models through re-sampling of the training data and achieve more stable and accurate models. Other examples of combining and averaging models to reduce prediction error include bagging [5], boosting [13] and random forest [31].

Data driven approaches usually refer to data mining techniques that are used to predict length of stay above or bellow a certain threshold that is for example specific for diagnosis related groups. Authors in [6] apply classification techniques to categorize length of stay in intensive care unit with respect to recommended seven-day norm; authors in [27] try to predict length of stay for post-coronary patients longer than 120 days; authors in [14] consider patients suffering from burns and create a model to predict whether their length of stay will be less than one-week; authors in [19] present research devoted to appendectomy patients and develop a model that recognizes those patients whose length of stay will exceed recommended five-day period. Authors in [28] apart from, testing several classification algorithms, reported results about finding the most significant input variables to predict the target variable - length of stay. Among thirty six input variables, the most significant variables affecting length of stay according to generated classification model were drug categories, co-morbidity (that is the presence of one or more diagnosis co-occurring with the primary disease), gender (men had longer length of stay than women) and age (patient younger than 50 years and older than 80 years had longer length of stay).

Interesting approach is presented in [3]. Authors categorized length of stay into three groups as short, medium and long. Training dataset is constructed by clustering similar claims after which classification is performed using ten different classifiers. For each classifier, using clustering as a preprocessing step gives better accuracy as compared to non-clustering based training dataset.

Wide variety of classification algorithms were applied in previously mentioned studies: decision trees, support vector machines - SVM, artificial neural networks - ANN, Naive Bayesian classifier etc. According to [30], decision tree implementation C4.5 is a classifier that has the best combination in terms of error rate and speed. Authors in [33] found that decision tree R-C4.5s (successor of C4.5) algorithm creates more robust and smaller trees. In [22] authors reported that Naive Bayesian classifier is robust to missing data.

Clustering can be considered as data driven approach, too. In [16], [15] clustering is used to create clinically meaningful groups with respect to length of stay and covariates: gender, age, primary diagnosis, etc.

Multi-stage models are based on modelling patient flow in hospital with Markov and semi-Markov chains. Length of stay is considered as an array of successive stages which the patients go through until they leave the hospital completely. It is possible to include additional variables that may influence patient length of stay as it is suggested in [25],

[24], [17]. The final model represents length of stay based on five of the most important variables, namely age, gender, admission method, Barthel grade and destination on departure from hospital.

As it is stated in the introductory section, all studies treat length of stay and its influencing factors statically in terms of that discovered correlations are interpreted as universal truth. For example, factor *X* predicts *Y%* of the length of stay augmentation. It is not possible to give different explanation for length of stay prolongation on two different observations.

In this study we do not predict length of stay neither do we use training dataset to develop a model from which the most significant factors are detected. The LOSEP problem discussed in this paper consists of defining procedure that is able to objectively estimate impact of available factors on length of stay elevation registered on new observations. Historical dataset H and threshold σ are given. New observations represent dates outside the historical dataset for which registered length of stay is $> \sigma$. Using the set H, the procedure must objectively (mathematically) measure impact of each factor causing higher length of stay than it is expected or desired. It is necessary to consider every new observation independently and provide potentially different explanations for different observations although the same set H is always used.

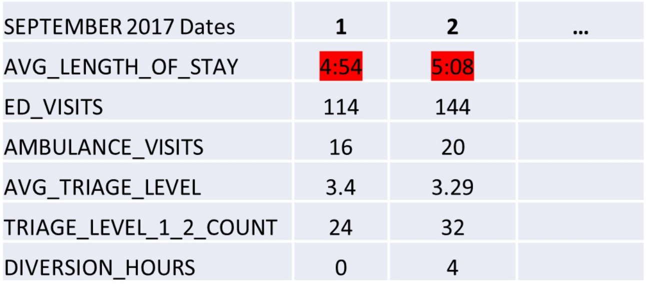
For example, consider two observations $a = (X_1(a), X_2(a), ..., X_n(a), LOS(a))$ and $b = (X_1(b), X_2(b), ..., X_n(b), LOS(b))$ for which $LOS(a) \geq \sigma$ and $LOS(b) \geq \sigma$ and factors $X_1, X_2, ..., X_n$. Explanations why $LOS(a) \geq \sigma$ and $LOS(b) \geq \sigma$ hold can be represented in the following forms $a : \{X_{i1} : impact_{i1}, X_{i2} : impact_{i2}, ..., X_{in} : impact_{in}\}$ and $b : \{X_{j1} : impact_{j1}, X_{j2} : impact_{j2}, ..., X_{jn} : impact_{jn}\}$ where $(i_1, i_2, ..., i_n) \neq (j_1, j_2, ..., j_n)$ are different permutations of the set $(1, 2, ..., n)$. The previous means that the most significant factor for the object a is X_{i1} while the most significant factor for the object b is X_{j1}.

3. A Motivating Example

In this section, an example from the case study presented in section 5, is briefly discussed.

Consider that historical dataset covers the period between January and August 2017 and let $\sigma = 4$ hours. The first two days of September 2017 are depicted on figure Fig. 1. The task is to explain elevation in average length of stay with respect to available factors: *AMBULANCE_VISITS, ED_VISITS, AVG_TRIAGE_LEVEL, TRIAGE_LEVEL_1_2_COUNT* and *DIVERSION_HOURS*. As an example of traditional approach, the Random Forest Regression is fitted with the historical data. Generated model ranks mentioned factors according to their importance to length of stay as follows *AVG_TRIAGE_LEVEL* ≻ *AMBULANCE_VISITS* ≻ *ED_VISITS* ≻ *TRIAGE_LEVEL_1_2_COUNT* ≻ *DIVERSION_HOURS*. It is illustrated on figure Fig. 2. This ranking is learned from the provided dataset and all future observations must be explained in such manner.

The procedure presented in this paper is able to estimate factors independently for each new observation. Results are presented on figure Fig. 3. It can be seen that on the 1st September the most important factors for length of stay elevation are *ED_VISITS* and *TRIAGE_LEVEL_1_2_COUNT*, while for the 2nd September the combination *AMBULANCE_VISITS*, *DIVERSION_HOURS* and *TRIAGE_LEVEL_1_2_COUNT* causes higher length of stay.



SEPTEMBER 2017 Dates	1	2	...
AVG_LENGTH_OF_STAY	4:54	5:08	
ED_VISITS	114	144	
AMBULANCE_VISITS	16	20	
AVG_TRIAGE_LEVEL	3.4	3.29	
TRIAGE_LEVEL_1_2_COUNT	24	32	
DIVERSION_HOURS	0	4	

Fig. 1. Observations that have to be explained

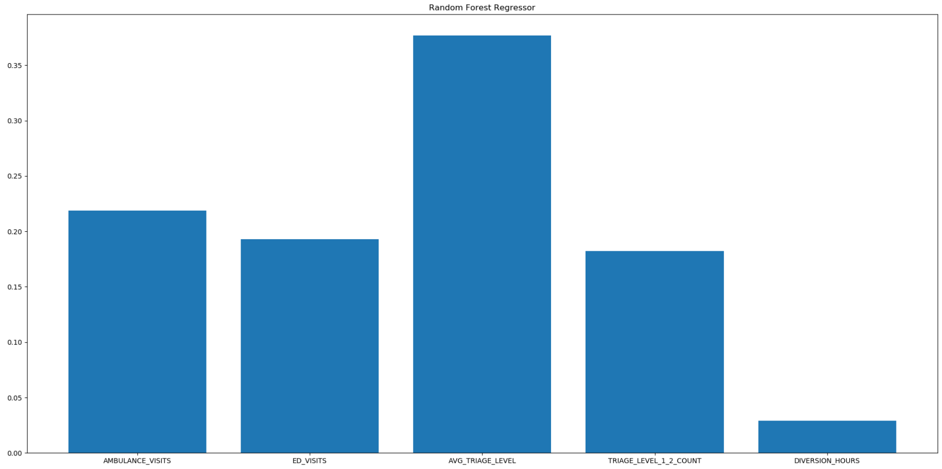


Fig. 2. Contributing factors ranked by Random Forest Regression

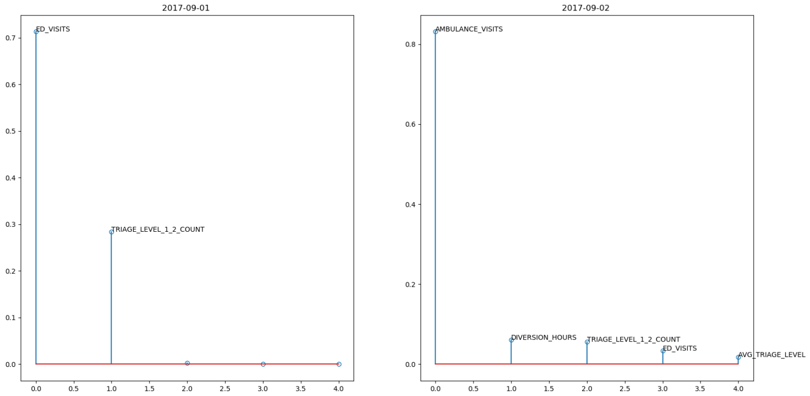


Fig. 3. Contributing factors ranked in LOSEP problem

To achieve clear distinction between the LOSEP problem and problems of determining factors influencing patient length of stay and length of stay prediction in this paragraph more general definition of the LOSEP problem is given. Consider that data matrix $H_{m\times n+1}$ is given. It contains historical data represented in the form of m multidimensional vectors with the following dimensions: $var_1, var_2, ..., var_n, TARGET\_VALUE$. Without loss of generality, we can take that the last column is target variable that should be explained in terms of other columns representing factors of interest. Additionally, threshold value σ is provided. Consider that stream of new observations Q is provided. The task is to objectively measure impact of the defined factors on the target value elevation registered on the new observation $q \in Q$. The result can be represented in the following form $q : \{var_{i1} : impact_{i1}, var_{i2} : impact_{i2}, ..., var_{in} : impact_{in}\}$, where $impact_{i1} > impact_{i2} > ... > impact_{in}$ and $impact_{i1} + impact_{i2} + ... + impact_{in} = 1$. Of course, for different observations $q_1 \neq q$ the explanation is generally different $q_1 : \{var_{j1} : impact_{j1}, var_{j2} : impact_{j2}, ..., var_{jn} : impact_{jn}\}$.

It is obvious that the LOSEP problem is an instance of the previous more general problem. Average length of stay, *AVG_LENGTH_OF_STAY*, on a specific day in an emergency department is *TARGET_VALUE* and factors are *ED_VISITS, AMBULANCE_VISITS, AVG_TRIAGE_LEVEL, TRIAGE_LEVEL_1_2_COUNT, DIVERSION_HOUR*. It can be seen that such selection of factors mostly reflects operational aspects of an emergency department rather than single patient characteristics. The threshold value σ is set to 4 hours. Patient length of stay expressed in hours and averaged on daily basis for each day from the future period forms the stream Q. The task is to estimate impact of each factor on days $q \in Q$ where $AVG\_LENGTH\_OF\_STAY(q) > 4$ and indicate the most important aspect that causes elevation in length of stay.

Also, the LOSEP problem is different from the outlier analysis. The new observations introduced in the above example may or may not be outliers. Instead of finding outlier objects in a dataset, the LOSEP problem consists of finding the factors best explaining why length of stay for the new observation that is not present in the given dataset, is higher than σ. Additionally, in the LOSEP problem it is necessary to introduce the function eligible to objectively measure the impact of the factors on the length of stay value registered on a new observation.

4. Explanation of New Observations

Historical dataset in the LOSEP problem contains data over specific past period of time. The historical dataset H can be considered as a collection of records or data objects. Each object consists of fixed set of variables. So, objects from the historical dataset can be thought of as vectors (points) of the following form $(factor_1, factor_2, ..., factor_n, LOS)$ in a multidimensional space, where each dimension corresponds to exactly one factor. In addition, the last dimension corresponds to the average patient length of stay on a specific day.

In other words, dataset H can be interpreted as $m \times n + 1$ matrix, where there are m rows, one for each object, and $n + 1$ columns, one for each dimension.

Granularity of historical dataset considered in the case study from the fifth section is on a day level. For each $i, 1 \leq i \leq n$, $factor_i(o)$ is a result of some aggregate function (sum, average, count) of variable $factor_i$ registered on every patient processed on a specific day that is represented with o. Accordingly, for each record $o \in H$, $LOS(o)$ is average patient length of stay on a specific day that is represented with o. To emphasize this situation, in the rest of this section LOS is replaced with *AVG_LENGTH_OF_STAY*.

The historical dataset is partitioned on two parts based on *AVG_LENGTH_OF_STAY* dimension and user defined threshold σ. The value for σ can be a combination of looking at historical data and a benchmark leadership wants to hit. Objects $o \in H$ for which $AVG\_LENGTH\_OF\_STAY(o) < \sigma$ holds, are part of a *good partition*. Objects belonging to the good partition are referred to as *good objects*. The other partition contains objects $o \in H$ for which $AVG\_LENGTH\_OF\_STAY(o) \geq \sigma$ is true.

More precisely, good partition H_{GP} is a subset of the given historical dataset H that is determined by the threshold σ as follows:

$$H_{GP} = \{o | o \in H \wedge AVG\_LENGTH\_OF\_STAY(o) < \sigma\}. \tag{1}$$

The LOSEP problem consists of providing explanation of a new observation. New observations belong to the same multidimensional space as the objects from the set H, but they are not known in advance. Actually, new observations represent future observations, dates outside the H, for which *AVG_LENGTH_OF_STAY* is $\geq \sigma$. Explanation of an observation q consists of objective measurement of the factors' impact on the average length of stay elevation and can be represented in the following form $q : \{factor_{i1} : impact_{i1}, factor_{i2} : impact_{i2}, ..., factor_{in} : impact_{in}\}$, where $impact_{i1} > impact_{i2} > ... > impact_{in}$ and $impact_{i1} + impact_{i2} + ... + impact_{in} = 1$ hold. It means that the greatest impact on the value $AVG\_LENGTH\_OF\_STAY(q) \geq \sigma$ has $factor_{i1}$ followed by $factor_{i2}$ and the smallest importance is estimated for $factor_{in}$. Here $(i_1, i_2, ..., i_n)$ is a permutation of the set $\{1, 2, ..., n\}$.

The new observation is explained with respect to the appropriate neighbourhood. The neighbourhood is determined among available good objects from H_{GP}. Two procedures, namely *nearest neighbour based method* and *clustering method* are considered.

Nearest neighbour method finds *k* closest good objects to the given observation $q = (factor_1(q), factor_2(q), ..., factor_n(q), AVG\_LENGTH\_OF\_STAY(q))$. The number of good objects constituting the neighbourhood is a user defined constant. Standard Euclidean distance is used to determine distance between objects.

When *k*-neighbourhood of the new observation q is determined, the algorithm calculates its centroid. The centroid c_q is the mean of all objects in the *k*-neighbourhood. Notice that all neighbouring objects are good objects.

The impact for every factor is estimated based on the formula for calculating the sum of squared errors (SSE). SSE is usually used as an objective function to estimate the quality of clustering. The SSE takes the sum of the squared distances between every object and the closest centroid. Set of clusters with the smallest SSE is considered as the best clustering solution.

Consider that *k*-neighbourhood of the new observation q constitutes the neighbouring cluster C_q, $|C_q| = k$. The SSE of the C_q is given by the following formula:

$$SSE = \sum_{x \in C_q} dist^2(c_q, x). \tag{2}$$

If the observation q is added to the cluster C_q, SSE of the cluster is increased by the amount $dist^2(c_q, q)$. As it is mentioned earlier, $dist$ is a standard Euclidean distance, so impact of i^{th} factor can be estimated by the formula:

$$impact(factor_i) = (c_q[factor_i] - q[factor_i])^2 / dist^2(c_q, q). \tag{3}$$

The algorithm based on the nearest neighbour approach is presented in the following listing. The algorithm is implemented in Python3 using sklearn library.

```
procedure kNN_impact_estimation(X, q, k)
{X is representing good objects}
{q is the new observation}
{k is the user defined size of the new observation neighbourhood}
begin
    {available good objects are fitted to the NearestNeighbors class}
    nnbrs = NearestNeighbors(n_neighbors=k)
    nnbrs.fit(X)

    {cluster and centroid of the new observation neighbourhood}
    $C_q$ = nnbrs.kneighbors(q)
    $c_q = \frac{1}{k} \sum_{x \in C_q} x$

    {impact[i] is impact of the $factor_i$}
    {$q = (factor_1, factor_2, ..., factor_n)$}
    impact = []
    for i in range(0, dim(q)):
        impact[i] = $(c_q[i] - q[i])^2 / dist^2(c_q, q)$

    return impact
```

With the clustering method neighbourhood of a new observation is determined by clustering of all good objects and determining the closest centroid. Theoretically, clusters can be created with any clustering algorithm, but exhaustive experiments that were part of the case study presented in the next chapter indicated that the best choice is Affinity propagation method. Partition-based methods usually require specifying number of clusters in advance, which was impossible to properly estimate in the available dataset. Density based methods during model building declare significant number of good objects as outliers. Consequently, such objects are eliminated from factor estimation that was unacceptable, bearing in mind that the number of good objects is generally limited.

When clusters are created the algorithm determines the closest centroid to the observation q. Let C_q be the cluster with the centroid c_q that is closest to the new observation q. Adding the q to the cluster C_q increases SSE of the cluster by the amount $dist^2(c_q, q)$, where $dist$ is a standard Euclidean distance. As before, the impact of i^{th} factor can be estimated by the formula (3).

The algorithm based on clustering approach is presented in the following listing. The algorithm is implemented in Python3 using sklearn library.

```
procedure clustering_impact_estimation(X, q)
{X is representing good objects}
{q is the new observation}
begin
    {available good objects are fitted to the Affinity Propagation class}
    clustering = AffinityPropagation(X)
```

```
{determine the closest centroid}
$c_q = \min_{c \in clustering.cluster\_centres} dist(c, q)$

{impact[i] is impact of the $factor_i$}
{$q = (factor_1, factor_2, ..., factor_n)$}
impact = []
for i in range(0, dim(q)):
    impact[i] = $(c_q[i] - q[i])^2 / dist^2(c_q, q)$

return impact
```

Nearest neighbour based method and clustering method are not equivalent and in general generate different results as it will be experimentally confirmed (section 5.2). The main difference is due to determining different neighbouring cluster of a new observation. Nearest neighbour based method simply selects the closest *k* good objects from historical dataset. Notice that these objects can be very diverse from each other. On the other hand, clustering method uses clustering as pre-processing step to create clusters of good objects. After that, factor ranking for a new observation is estimated considered good objects from only one cluster, the cluster that is the closest to that observation.

5. Case study - Emergency Department

In this section real life problem, length of stay explanation in an emergency department, is presented and usability of discussed algorithms is demonstrated.

5.1. Modelling the data

The raw data was exported by the author directly from information system of one hospital acquisition and management company. Specific data preprocessing procedures were necessary to be designed and implemented on the raw data to obtain historical dataset H of the form introduced in the previous section.

Eventually, the historical dataset is represented as data matrix that contains columns: *ED_VISITS* - total emergency department visits, *AMBULANCE_VISITS* - number of patients brought in by ambulance, *AVG_TRIAGE_LEVEL* - average triage level of all patients on specific day, *TRIAGE_LEVEL_1_2_COUNT* - number of patients with triage level 1 or 2 on specific day, and *DIVERSION_HOURS* - diversion hours number on specific day. Granularity of the historical dataset is on a day level.

The previous columns represent factors under consideration. The factors considered in this case study are suggested by domain expert.

In addition, there are three more columns: *AVG_LENGTH_OF_STAY*, representing average length of stay in hours in emergency department on a specific day, *DATE*, representing calendar date, and *FACILITY*, representing hospital name. Notice that *DATE* and *FACILITY* constitute primary key.

To conclude, every record from the historical dataset $H_{m \times n}$ is multidimensional vector of the following form *(DATE, FACILITY, LENGTH_OF_STAY, ED_VISITS, AMBULANCE_VISITS, AVG_TRIAGE_LEVEL, TRIAGE_LEVEL_1_2_COUNT, DIVERSION_HOURS)*. It means that $n = 8$. Total number of records after data preprocessing is $m = 602646$.

The raw data was separated among several tables in relational database. All records originated from emergency departments from four different hospitals in California. The covered period was from January, 2013 to April 2018. All hospitals belong to the same hospital management company, so transactional data from every emergency department are stored together in the same database.

Source tables are: Emergency department (ED), Triage level (TL), and Diversion (DV). Details are presented in Table 1. Types and attributes of each table are shown in Table 2, Table 3, Table 4.

Table 1. Source datasets characteristics

Dataset name	Number of records	Starting date	Ending date
ED	603006	2013-01-01	2018-04-14
TL	7474	2013-03-01	2018-04-12
DV	541936	2013-01-01	2018-04-14

Table 2. Types and attributes of ED datasets

Attribute	Type	Explanation
PATIENT_ACCOUNT	string	Unique patient account number
MRN	string	Medical record number of patient
ARRIVAL_TIMESTAMP	datetime	arrival timestamp of patient into the emergency department
DISCHARGE_TIMESTAMP	datetime	
PATIENT_TREATED	boolean	False=Patient registered then left; True=patient was actually treated
ICU_ADMIT	boolean	True=Patient was admitted from ED to the intensive care unit (ICU)
ADMIT	boolean	True=Patient was admitted to the hospital
MEDICARE_ICU_ADMIT	boolean	True=Patient was admitted from ED to the ICU and had Medicare insurance
MEDICARE_TREATED	boolean	True=Patient had Medicare insurance and was treated
UN_INSURED_TREATED	boolean	True=Patient had no insurance and was treated
LEFT_AFTER_TRIAGE	boolean	True=Patient left after being triaged
LEFT_BEFORE_TRIAGE	boolean	True=Patient left before being triaged
LEFT_WITHOUT_BEING_SEEN	boolean	True=Patient left without being seen (LWBS)
ELOPED	boolean	True=Patient left and being assessed by a nurse
AMA	boolean	True=Patient left against doctor's orders
TRANSFER	boolean	True=Patient was transferred to another hospital within the same system
AN_OTHER_HOSPITALS	boolean	True=Patient was transferred to a hospital outside system
EMS	boolean	True=Patient brought in by ambulance
EMS_ADMIT	boolean	True=Patient brought in by ambulance and was admitted
UN_INSURED_ADMIT	boolean	True=Uninsured patient was admitted to the hospital
TRIAGE_START_TS	timestamp	initial triage started
BED_ASIGN_TS	timestamp	bed was assigned to the patient
NURSE_TS	timestamp	the nurse saw and triaged the patient
PHYSICIAN_TS	timestamp	the physician has come in and done their assessment
ARRIVAL_MODE	string	the method of arrival: walk-in, ambulance, police, etc.
PAYER_CODE	string	initial payer code description
FACILITY_NAME	string	

Emergency department dataset contains high level data such as medical record number, patient account, insurance type, payer code, arrival mode etc. Most importantly, ED dataset contains all timestamps identifying starting and ending points of treatment procedure: arrival (*ARRIVAL_TIMESTAMP*), arrival to triage (*TRIAGE_START_TS*), triage to bed (*BED_ASIGN_TS*), bed to nurse (*NURSE_TS*), nurse to doctor (*PHYSICIAN_TS*), doc-

Table 3. Types and attributes of TL datasets

Attribute	Type	Explanation
CPT4_CODE	integer	
PATIENT_ACCOUNT	string	
ARRIVAL_TIMESTAMP	datetime	
HOSPITAL_NAME	string	

Table 4. Types and attributes of DV datasets

Attribute	Type	Explanation
HOURS_OF_DIVERSION	integer	number of hours the diversion occurred
DATE	date	
HOSPITAL_NAME	string	

tor to disposition (*DISCHARGE_TIMESTAMP*). Overall length of stay is calculated as the difference between arrival time-stamp and departure time-stamp $LENGTH\_OF\_STAY = DISCHARGE\_TIMESTAMP - ARRIVAL\_TIMESTAMP$. Of course, $LENGTH\_OF\_STAY = arrivaltotriage + triagetobed + bedtonurse + nursetodoc + doctodisposition$. Also, records from ED dataset contain information if patient was treated in ambulance, $EMS = True$. Based on it, number of ambulance visits per day is calculated.

To conclude, for the purpose of the analysis ED dataset is projected on a schema of the form *ED(FACILITY, ARRIVAL_TIMESTAMP, DISCHARGE_TIMESTAMP, EMS)*.

Triage level dataset, among others, contains information about CPT4 codes from which triage level for each patient visit can be extracted. Triage level is an integer value that describes degree of patient sickness. Value 1 for triage level means *very sick*. Value 5 means *not sick*.

To conclude, for the purpose of the analysis TL dataset is projected on a schema of the form *TL(FACILITY, ARRIVAL_TIMESTAMP, TRIAGE_LEVEL)*. The *TRIAGE_LEVEL* column is derived from the original *CPT4_CODES* column with the mapping provided in Table 5.

Table 5. TRIAGE_LEVEL mapping

CPT4_CODE	TRIAGE_LEVEL
99281	1
99282	2
99283	3
99284	4
99285	5
99291	1

Diversion dataset contains records about how many hours the emergency department had to shut down because it was full. Each record summarizes number of hours the diversion occurred for every day and every hospital.

For the purpose of the analysis DV dataset is projected on a schema of the form *DV(FACILITY, DATE, HOURS_OF_DIVERSION)*.

Finally, data matrix $H_{602646\times 8}$ with columns *(DATE, FACILITY, (AVG_LENGTH_OF_STAY, ED_VISITS, AMBULANCE_VISITS, AVG_TRIAGE_LEVEL, TRIAGE_LEVEL_1_2_COUNT, DIVERSION_HOURS)* can be obtained. The aim of defining the unique schema was to enclose necessary data from the three before mentioned datasets. To achieve the previous schema some specific transformations must be done on the original tables.

Length of stay for one visit is calculated as the difference between arrival time-stamp and departure time-stamp, $LENGTH\_OF\_STAY = DISCHARGE\_TIMESTAMP - ARRIVAL\_TIMESTAMP$. Both time-stamps are present in the ED dataset. Average length of stay, *AVG_LENGTH_OF_STAY*, is obtained by grouping records with the same date and facility name. Similarly, with grouping by day and facility and counting $DISCHARGE\_TIMESTAMP - ARRIVAL\_TIMESTAMP <= 24$ hours *ED_VISITS* - number of patients in ED per day and facility is found. It means that this study takes into account patients who spent less than one day in an emergency department. From available data (column *ADMIT* in ED dataset) it is possible to separate patients who were admitted to the hospital for further treatment - INPATIENT, from those who were only treated in emergency department - OUTPATIENT. But, in this case study only patients who spent less that 24 hours in an emergency department are covered, regardless of their admission to the hospital for further treatment (INPATIENT or OUTPATIENT).

Finally, by counting $COUNTIF(EMS = True)$ the number of ambulance visits per day is obtained.

The previous can be concisely expressed with the following pseudo SQL query:

```
 SELECT date_part(ARRIVAL_TIMESTAMP)
   , FACILITY
   , COUNT(DISCHARGE_TIMESTAMP - ARRIVAL_TIMESTAMP <= 24 hours)
     AS ED_VISITS
   , COUNTIF(EMS = True) AS AMBULANCE_VISITS
   , AVG(DISCHARGE_TIMESTAMP - ARRIVAL_TIMESTAMP) AS AVG_LENGTH_OF_STAY
FROM ED
GROUP BY date_part(ARRIVAL_TIMESTAMP), FACILITY
```

In the previous query *date_part* stands for a function that can extract date part from *ARRIVAL_TIMESTAMP* time-stamp. In that way roll-up is performed by climbing up to the day level in the time dimension.

From the TL dataset *AVG_TRIAGE_LEVEL* and *TRIAGE_LEVEL_1_2_COUNT* are calculated with the following pseudo SQL query:

```
 SELECT date_part(ARRIVAL_TIMESTAMP)
   , FACILITY
   , AVG(TRIAGE_LEVEL) AS AVG_TRIAGE_LEVEL
   , COUNTIF(TRIAGE_LEVEL in (1, 2)) AS TRIAGE_LEVEL_1_2_COUNT
FROM TL
GROUP BY date_part(ARRIVAL_TIMESTAMP), FACILITY
```

On the DV dataset the following pseudo SQL query is run:

```
 SELECT DATE
   , FACILITY
   , SUM(HOURS_OF_DIVERSION) AS DIVERSION_HOURS
FROM TL
GROUP BY DATE, FACILITY
```

At the end, the data matrix $H_{602646 \times 8}$ is generated by calculating natural join on the results of the previous three SQL queries.

5.2. Factor impact estimation

In the following experiments historical dataset H is filtered to contain records between January and August 2017, about 115K objects. The set Q contains observations between September and December 2017 for which *AVG_LENGTH_OF_STAY* is $\geq \sigma$. The threshold is set to $\sigma = 4$ hours. For this case study the value for σ was suggested by domain expert. In general, σ is a combination of looking at historical data and a benchmark leadership wants to hit.

Two methods for new observation explanation presented in the fourth section were applied. Because of the lack of space, in this section results of explaining only subset of Q is reported. The subset $Q_{exp} \subset Q$ contains the top 5 observations having the highest values for patient length of stay, 5 objects with the smallest patient length of stay that is $> \sigma$ and 5 objects around the median value of the objects from Q.

The results obtained from the method $kNN\_impact\_estimation$ are presented in figures Fig. 4, Fig. 5, Fig. 6. The new observation is identified with date that is shown as chart title. It can be seen that the method is able to clearly identify the most significant factor for every observation from the set Q_{exp}.

The results obtained from the method $clustering\_impact\_estimation$ are presented in figures Fig. 7, Fig. 8, Fig. 9. Titles of the charts are dates of the observations. It can be seen that the method is able to clearly identify the most significant factor for every new observation from the set Q_{exp}.

The results from the previous experiments indicate that the $clustering\_impact\_estimation$ provides clearer distinction between the most significant factor and the others. For example, the difference between the impacts of the most significant factor and the following factor is in average 0.43 for $clustering\_impact\_estimation$. The method $kNN\_impact\_estimation$ achieves 0.38 as the average difference between impact of the two most important factors.

Execution time of the proposed methods is measured on machine with Intel(R) Core(TM) i7-55000U CPU at 2.40GHz and 8GG of RAM memory. The

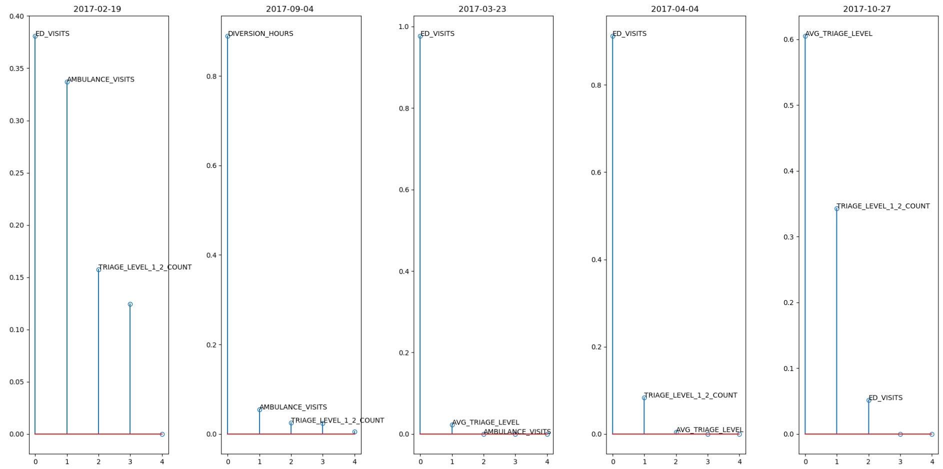


Fig. 4. The method $kNN\_impact\_estimation$ for the top 5 new observations

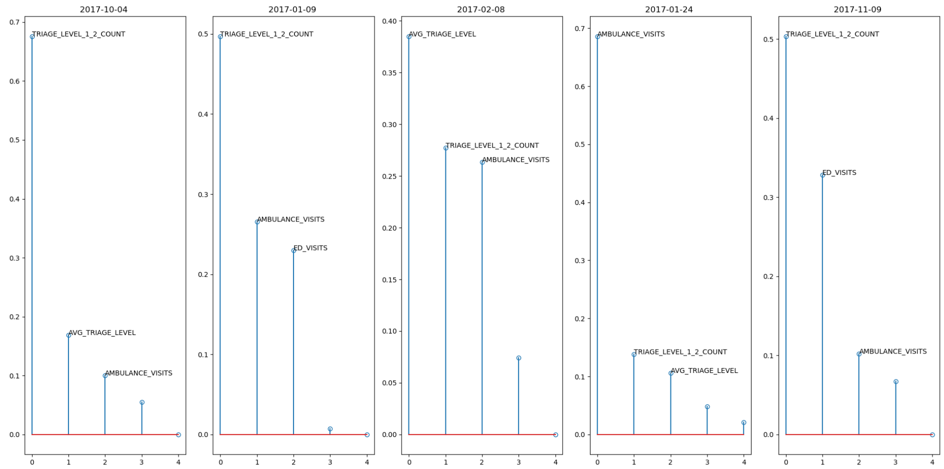


Fig. 5. The method $kNN\_impact\_estimation$ for 5 objects around the median

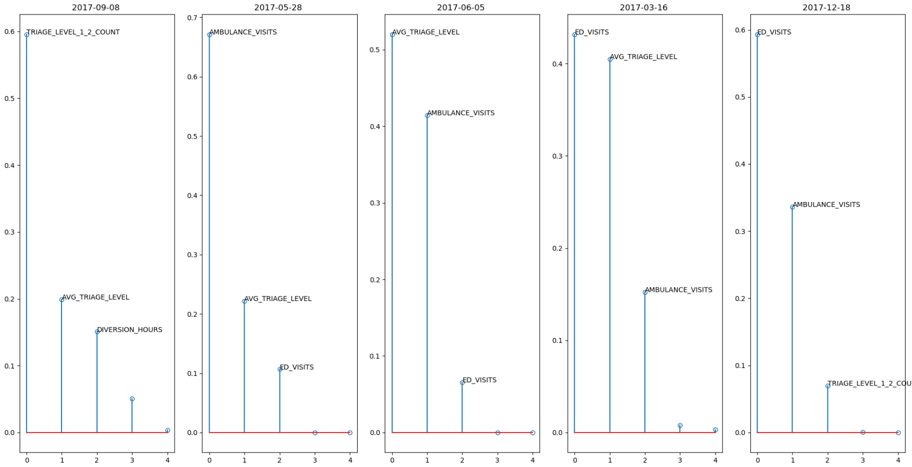


Fig. 6. The method $kNN\_impact\_estimation$ for the smallest 5 new observations

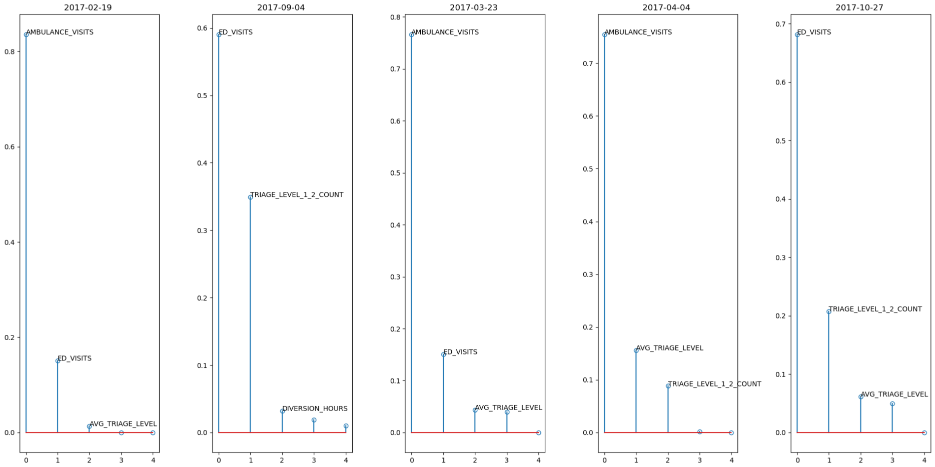


Fig. 7. The method $clustering\_impact\_estimation$ for the top 5 new observations

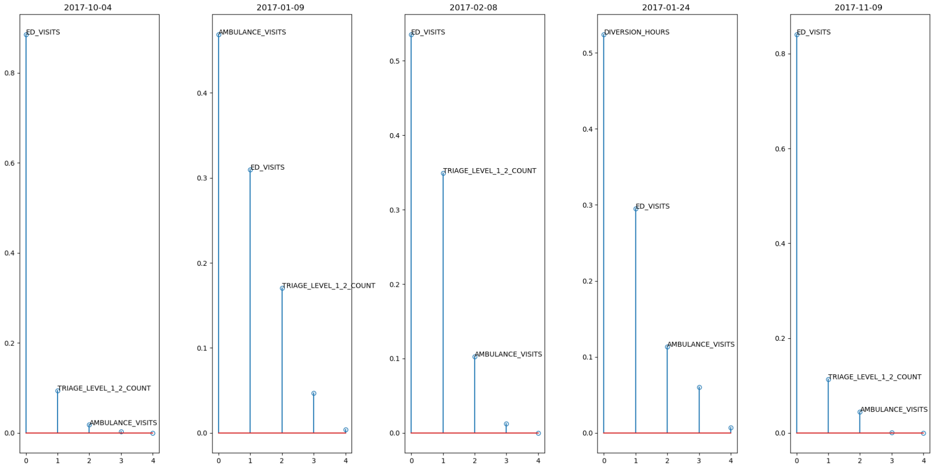


Fig. 8. The method $clustering\_impact\_estimation$ for 5 objects around the median

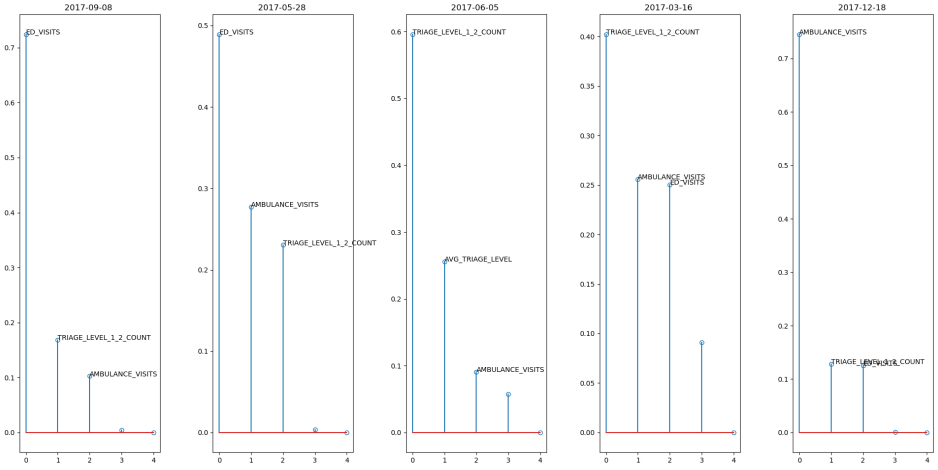


Fig. 9. The method $clustering\_impact\_estimation$ for the smallest 5 new observations

$clustering\_impact\_estimation$ requires 2.0005 seconds comparing to 6.8509 seconds that are necessary for $kNN\_impact\_estimation$ to finish.

Numerical and objective estimation of factors' impact allows user to determine major factors to specific anomaly (elevation in length of stay) and to express how much in % of total elevation this factor is contributing to the problem. For example, consider the observation representing 8th September from the figure Fig. 9. It can be seen that the most important contributing factor in this case is *ED_VISITS* contributing more than 70% to length of stay elevation registered on 8th September. Similarly, explanations for all other cases can be generated.

Length of stay is especially important parameter because it can be interpreted as efficiency. The results of such analysis should assist leaders of hospitals and its staff in understanding the "story" or "narrative" of their organization.

6. Conclusions

In this paper length of stay explanation problem - LOSEP is introduced. The problem consists of estimating impact of available factors on length of stay values that are higher than the threshold σ. Historical dataset is given. Objects from the historical dataset representing cases when registered length of stay is $\leq \sigma$ are referred to as good objects. The set of good objects can be considered as the knowledge database for the proposed methods. Observations of interest are new observations, possibly coming from a stream, for which length of stay is higher than σ. The system is queried to provide explanation about length of stay elevation on a new observation in a form of estimated importance of each factor.

The paper presents two methods: $kNN\_impact\_explanation$ and $clustering\_impact\_explanation$. In both approaches a new observation o is explained based on its neighbourhood.

The essential difference between $kNN\_impact\_explanation$ and $clustering\_impact\_explanation$ is in the process of finding the appropriate neighbourhood. The neighbourhood consists of good objects. In the method $kNN\_impact\_explanation$ the algorithm finds k-neighbourhood consisting of the closest k good objects. The algorithm from $clustering\_impact\_estimation$ finds neighbourhood of the q by clustering good object and determining the cluster with the closest centroid. Standard Euclidean distance is used to determine the distance between objects.

When the neighbourhood of the new observation q is determined, the impact for every factor is estimated. New observation q can be represented as $q = (factor_1(q), ..., factor_n(q))$. The procedure for objective impact estimation of each factor calculates increment to the SSE if the observation was added to the neighbouring cluster and distributes the increment value among factors proportionally.

Also, results of the case study in which proposed methods were applied on length of stay explanation in an emergency room are discussed. The historical dataset contains above 600K data objects. The following factors are considered: *ED_VISITS* - total emergency department visits, *AMBULANCE_VISITS* - number of patients brought in by ambulance, *AVG_TRIAGE_LEVEL* - average triage level of all patients on specific day, *TRIAGE_LEVEL_1_2_COUNT* - number of patients with triage level 1 or 2 on specific day, and *DIVERSION_HOURS* - diversion hours number on specific day. Granularity of the historical dataset is on a day level.

Results of the analysis show that proposed methods are capable to recognize the most important factor and additionally to express how much in % of total elevation every factor is contributing to the specific observation.

Experiments show that the proposed two methods are not equivalent. In general, they assign different factors' impacts for the same observation. As future work, it can be interesting to implement voting schema and combine these two methods. Additionally, these methods potentially can be extended towards expert system that will be able to independently construct narrative explanation of the problem and propose possible actions regarding the situation.

References

1. Aghajani1, S., Kargari, M.: Determining factors influencing length of stay and predicting length of stay using data mining in the general surgery department. Hospital Practices and Research 1, 53–58 (2016)
2. Awad, A., Bader-El-Den, M., McNicholas, J.: Patient length of stay and mortality prediction: a survey. Health Services Management Research 30, 105–120 (2017)
3. Azari, A., Janeja, V., Mohseni, A.: Healthcare data mining: Predicting hospital length of stay (phlos). International Journal of Knowledge Discovery in Bioinformatics (IJKDB) 3, 44–66 (2012)
4. Bashkin1, O., Caspi, S., Haligoa, R., Mizrahi, S., Stalnikowicz, R.: Organizational factors affecting length of stay in the emergency department: initial observational study. Israel Journal of Health Policy Research 4, 1–7 (2015)
5. Breiman, L.: Bagging predictors. Machine learning 24, 123–140 (1996)
6. Buchman, T.G., Kubos, K.L., Seidler, A.J.: A comparison of statistical and connectionist models for the eprediction of chronicity in a surgical intensive care unit. Crit Care Med. 22, 750–762 (1994)
7. Castillo, M.G.: Modelling patient length of stay in public hospitals in mexico. Thesis (Doctoral), University of Southampton, Southampton Business School 1, 318pp (2012)
8. Chaou, C.H., Chiu, T.F., Yen, A.M.F., Chip-Jin, Chen, H.H.: Analyzing factors affecting emergency department length of stay—using a competing risk-accelerated failure time model. Medicine 95, 1–7 (2016)
9. Chua, J.M.: Factors associated with prolonged length of stay in patients admitted with severe hypoglycaemia to a tertiary care. Endocrinology, Diabetes Metabolism 1, 1–5 (2019)
10. Combe, C., Kadri, F., Chaabane, S.: Predicting hospital length of stay using regression models: Application to emergency department. In: MOSIM'14. vol. 124, pp. 672–674 (2014)
11. DO, D.R.C.: Analysis of Survival Data. Chapman HAll (1984)
12. Esfandiari, N., Babavalian, M.R., Moghadam, A.M.E., Tabar, V.K.: Knowledge discovery in medicine: Current issue and future trend. Expert Systems with Applications 41, 4434–4463 (2014)
13. Friedman, J.: Greedy function approximation: a gradient boosting machine. Annals of statistics 29, 1189–1232 (2001)
14. Frye, K.E., Izenberg, S.D., Williams, M.D.: Simulated biologic intelligence used ot predict length of stay and survival of bums. J Burn Care Rehabil 17, 540–546 (1996)
15. Garg, L., Mcclean, S., BJ, B.M., Millard, P.: Phase-type survival trees and mixed distribution survival trees for clustering patients hospital length of stay. Informatica 22, 57–72 (2011)

16. Garg, L., McClean, S., Barton, M., BJ, B.M., Fullerton, K.: Intelligent patient management and resource planning for complex, heterogeneous, and stochastic healthcare systems. Systems, Man and Cybernetics, Part A: Systems and Humans 42, 1332–1345 (2012)
17. Golouke, N., Huibers, C., Stalpers, S., Taekema, D., Vermeer, S., Jansen, P.: An observational, retrospective study of the length of stay, and its influencing factors, among elderly patients at the emergency department. European Geriatric Medicine 6, 331–335 (2015)
18. Grubinger, T., Kobel, C., Pfeiffer, K.: Regression tree construction by bootstrap: Model search for drg-systems applied to austrian healthdata. BMC Medical Informatics and Decision Making 10, – (2010)
19. Hu, P.: A data-driven approach to manage the length of stay for appendectomy patients. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 39, 1339–1347 (2009)
20. Jus, E.: Factors influencing length of stay in the emergency department in a private hospital in north jakarta. Endocrinology, Diabetes Metabolism 27, 165–173 (2008)
21. Khosravizadeh, O., Vatankhah, S., Bastani, P., Kalhor, R., Alirezaei, S., Doost, F.: Factors affecting length of stay in teaching hospitals of a middle-income country. Electronic Physician 8, 3042–3047 (2016)
22. Liu, P., Lei, L., Yin, J., Zhang, W., Naijun, W., El-Darzi, E.: Healthcare data mining: predicting inpatient length of stay. Proceedings of the 3rd International IEEE Conference on Intelligent Systems Los Alamitos 1, 832–837 (2006)
23. Liu, Y., Phillips, M., Codde, J.: Factors influencing patients' length of stay. Australian Health Review 24, 63–70 (2001)
24. Marshal, A.H.: Conditional phase-type distributions for modelling patient length of stay in hospital. International Transactions in Operational Research 10, 567–576 (2003)
25. Marshal, A.H., McClean, S.I., Shapcott, C.M., Millard, P.: Modeling patient duration of stay to facilitate resource management of geriatric hospitals. Helath care management science 5, 313–319 (2002)
26. Marshall, A., Vasilakis, C., El-Darzi, E.: Length of stay-based patient flow models: recent developments and future directions. Health Care Management Science 8, 213–220 (2005)
27. Mobley, B.A., Leasure, R.: Artificial nerual network predictions of lengths of stay in a post-coronary care unit. Heart Lung 24, 251–256 (1995)
28. PR, P.H., Ahmadi, M., Alizadeh, S., Sadoughi, F.: Use of data mining techniques to determine and predict length of stay of cardiac patients. Healthcare informatics research 19, 121–129 (2013)
29. SixSigma: Box plot diagram to identify outliers (2019), available from: `https://www.whatissixsigma.net/box-plot-diagram-to-identify-outliers/`
30. TS, T.L., Loh, W., Shih, Y.: A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Machine Learning 40, 203–228 (2000)
31. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (2000)
32. Yoon, P., Steiner, I., Reinhardt, G.: Analysis of factors influencing length of stay in the emergency department. Can J Emerg M 5, 155–161 (2003)
33. Z, Z.Y., Liu, P., Lei, L., Yin, J.: R-c4. 5 decision tree model and its applications to health care dataset. Proceedings of ICSSSM 2005 2, 1099–1103 (2005)

Savo Tomovic received his PhD in computer science from University of Montenegro in 2011. During his PhD studies he was involved in the project Linear Collider Flavour Identification (LCFI) with the aim to compare different data mining and classification algorithms as well as to understand the relative importance of the various input variables for the resulting tagging performance. He is currently an associated professor in the Faculty of Science at University of Montenegro and Head of the Computer Science Department. He teaches a wide variety of undergraduate and graduate courses in several computer science disciplines, especially data mining, machine learning and data warehousing. In addition, he is currently engaged as consultant in several software companies on projects for design and implementation of cognitive systems and data warehouse models.

Received: April 22, 2020; Accepted: January 27, 2021.

Computer Science and Information Systems 18(3):1023–1040 https://doi.org/10.2298/CSIS201120032L

Multimodal Encoders and Decoders with Gate Attention for Visual Question Answering

Haiyan Li[1] and Dezhi Han[2]

[1] School of Information Engineering, Shanghai Maritime University
Shanghai,201306, China
1977115781@qq.com
[2] School of Information Engineering, Shanghai Maritime University
Shanghai,201306, China
dezhihan88@sina.com

Abstract. Visual Question Answering (VQA) is a multimodal research related to Computer Vision (CV) and Natural Language Processing (NLP). How to better obtain useful information from images and questions and give an accurate answer to the question is the core of the VQA task. This paper presents a VQA model based on multimodal encoders and decoders with gate attention (MEDGA). Each encoder and decoder block in the MEDGA applies not only self-attention and cross-modal attention but also gate attention, so that the new model can better focus on inter-modal and intra-modal interactions simultaneously within visual and language modality. Besides, MEDGA further filters out noise information irrelevant to the results via gate attention and finally outputs attention results that are closely related to visual features and language features, which makes the answer prediction result more accurate. Experimental evaluations on the VQA 2.0 dataset and the ablation experiments under different conditions prove the effectiveness of MEDGA. In addition, the MEDGA accuracy on the test-std dataset has reached 70.11%, which exceeds many existing methods.

Keywords: Deep Learning, Artificial Intelligence, Visual Question Answering, Gate Attention, Multimodal Learning.

1. Introduction

Deep learning has been extensively applied in the domains of Computer Vision (CV) and Natural Language Processing (NLP), such as object detection, image segmentation, machine translation, and has shown excellent performance in these domains. Tasks based on language and vision are attracting more and more researchers' attention. Inspired by the multimodal task of image captioning, people began to study Visual Question Answering(VQA). VQA [3] is a complete artificial intelligence task which takes images and questions as input and combines their information to output an answer using human language, but some questions cannot be answered directly from the picture, which requires certain knowledge reasoning, so it requires not only a detailed understanding of questions but also the analysis of the visual elements of images [39][28]. How to predict suitable answer is one of the most challenging tasks in VQA. The VQA has the practically applied value in helping the blind [18][19] and image retrieval [16][27], etc. Blind people can input photos they photo and questions into the VQA system to solove their questions, which can help them ”see” the world. VQA has been applied to medical images recently, CG-MVQA can better assist doctors in clinical analysis and diagnosis [35]. VQA can also be combined with wireless sensors [4][42] [34][1]. Wireless sensors can be used in military, agriculture, ecological environment, medical treatment [20] [8][9], etc. The data collected by wireless sensors in these scenarios can be processed into picture data as input to the VQA system. We ask questions about the sensors for related questions, the VQA system will give corresponding answers. The development of CV and NLP produces endless VQA models which base on deep learning. Many previous models use VGG-NET[37], ResNet [21] for the extraction of global features information, and then VQA models learn from Faster RCNN [2] in object detection to obtain the region of interest of image which applies an object detector to obtain image categories accurately; from BoW to Long-Short Term Memory(LSTM),Gate Recurrent Unit(GRU), GloVe, Bert [10], these technologies have significantly improved the VQA accuracy.

On the other hand, we utilize the attention mechanism[44][40][7]to improve the accuracy of the VQA model, which proved to be one of the most effective methods. Attention in the human vision refers to obtain an object region by quickly scanning the entire picture. The global features extracted by the early VQA models contain a lot of irrelevant information or noise information. To circumvent this problem, people apply attention mechanism to the VQA. Yang et al. [46] apply it in their model which makes the VQA more conducive to fine-grained visual understanding. But early attention models ignore interactions in different modal and the links between image areas and the words in the question. In order to avoid this defect, recent studies have also proposed the co-attention model [31][45][32], which can learn image attention and text attention simultaneously.

Although the results of experiment indicate that their models have a good improvement on accuracy, their results still contain some irrelevant information. We guess that the VQA model can further filter out irrelevant information and perform better based on the following conjectures: 1) The VQA model can analyze the correlation between image feature informaion or question feature information and the attention results. 2) The model can model the relationship between different visual objects in the image. Experiments verify our ideas. Specifically, inspired by the AoA network [22] in the image captioning task, we apply gate attention to achieve this. We have designed MEDGA, which can acquire the information in the images and questions more effectively to make more accurate reasoning and give more accurate answers. The entire framework is shown in Figure 1. From Figure 1, we can see that MEDGA consists of several encoder and decoder blocks.

The contributions of this article can be summarized as follows:

(1) A VQA model based on multimodal encoders and decoders with gate attention is designed. Self-attention is employed to describe the inter-modal interactions and use cross-attention to better describe the intra-modal interactions of multi-modal data. The proposed MEDGA makes multi-modal reasoning more accurate by stacking multiple encoders and decoders.

(2) We design a new encoder block and decoder block. Gate attention is introduced in the new blocks, that is, make use of self-attention, cross-modal attention results, and queries to better model the contextual relationship between different objects in the picture so that it is conducive to give fine-grained answers to relational reasoning questions.

(3) This paper has proved the effectiveness of MEDGA based on a great deal of experiments and ablation studies. The accuracy on the VQA v2 dataset outperforms many advanced methods.

The rest structure of this paper is arranged as follows: Chapter 2 introduces the related work of Visual Question Answering, Chapter 3 introduces the overview framework of MEDGA; Chapter 4 introduces related experiments and the comparison of MEDGA with other advanced methods; The conclusion of this paper is shown in Chapter 5.

2. Related Work

2.1. Multimodal Features Fusion

The visual question answering task needs to input images and questions at the same time. The image exists in the form of pixels and contains a lot of rich information while the question exists in the form of text and contains limited information. Therefore, the VQA task requires a complex interaction between visual features and language features to obtain fused features that contain abundant information. The visual features and language features are processed into a form of vector, and the two types of features are merged for gaining a joint representation. In vector fusion, the traditional ways are dot produce, dot addition, and full connection. Akira Fukui et al. believed that the outer product of vectors is more expressive, so they raise a Multimodal Compact Bilinear pooling (MCB) model [14], but it may cause a sharp increase in dimensions. J.-H. Kim et al. proposed a Multimodal Low-rank Bilinear Attention Networks(MLB) model [24]. In this method, the tensor of three used for bilinear combination is decomposed into three 2-dimensional weight matrices and applies matrix decomposition to reduce the rank based on the two-dimensional tensor. The method reduces the dimensionality of the tensor and can make the output feature dimensions low to a certain extent. The parameters of experiment are small. However, MLB is sensitive to hyper-parameter and converges slowly. MUTAN [6] proposed by Ben-younes et al. promotes MCB and MLB, which has stronger expressiveness. This method is based on the decomposition of Tucker tensor, including decomposition into three matrices and core tensor, which effectively parameterizes vision and textual representation.

2.2. Attention Mechanism

In recent years, a wide variety of tasks take advantage of the attention model. The attention mechanism is introduced to the image field [44] by Xu et al. and they calculate the probability distribution of attention to highlight the impact of a key input on the output. Their model can identify salient regions in the image and generate subtitles based on these regions. This idea is applied to visual question answering, making the model focus on the image area related to the question. The visual question answering task not only needs to understand the image feature information, but also the question features. Therefore, understanding the image attention features guided by the question and understanding the question attention features guided by the image features is very important in the task. The introduction of attention mechanisms in VQA tasks is of great help in improving accuracy. Lu et al. proposed to focus on images and problems through parallel co-attention and alternative attention [30], and proposed that in addition to visual attention which is "where to look," the "what words to listen to" is also important in question attention. The co- attention model was used to infer image and text attention jointly. Peng Gao et al. proposed a Dynamic Fusion with Intra- and Inter- modality Attention Flow (DFAF) model [15]. This method can be used to pass information dynamically between visual and linguistic modalities and can well capture the high-level interaction between language and visual area, and the performance of VQA tasks has been significantly improve. Yu et al. proposed a multi-level attention network (MLAN) [11]. The attention in this system includes semantic attention,attribution attention, and visual attention while paying attention to the semantic attributes and image regions related to question. MLAN reduces the semantic gap between vision and language.

2.3. Other Works for Computer Vision and Language

In addition to the methods mentioned above, there are many other methods for VQA. The method proposed by Qi Wu et al. [43] combines external knowledge to make inferences. Shi Yang et al. proposed using question type feature for solving VQA task [36]. The model is able to predict the type of question in advance before answering the question, reducing the search space for the answer, and achieving a good result in the TDIUC dataset. In [48], they propose a neural network component, allowing count objects from the object proposals. In the cases of not affecting other types of questions, this method improves the accuracy of the number category on the VQA v2 dataset.

3. Multimodal Encoders and Decoders with Gate Attention

This section introduces the MEDGA architectures. The overall framework of MEDGA is shown in Figure 1. MEDGA includes the following four components: 1) Basic visual and language features extraction 2) The Encoder to model intra-modal interactions within language modality and the Decoder to capture intra-modal interactions within visual modality and inter-modal interactions across two modality simultaneously. 3) Feature fusion. 4) An answer prediction layer with multi-label classification. The details of these sections will be described next.

3.1. Image and Question Representations

To obtain visual features, this paper uses the top-down and bottom-up model proposed by Peter Anderson et al. [2]. Faster R-CNN uses ResNet-101 for initialization, and then perform fine-tuning on the Visual Genome dataset[26].The object detection in the pictures by Faster R-CNN includes the following two steps: First, object proposals in the image are predicted through the RPN and select the proposals with the highest score as input to the next step. Second, use RoI pooling to select smaller feature maps for each boxing proposal. In this article, we take the top 100 detected objects with the highest probability. Given image I, the obtained vision feature can be expressed as $V \in R^{\mu \times 2048}$,where μ represents the total number of object regions.The i^{th} region feature can be expressed as $r_i \in R^{2048}$.Input a question Q of length L, Q is first tokenized into a sequence of words. These words are represented by one-hot vector and then they are transformed

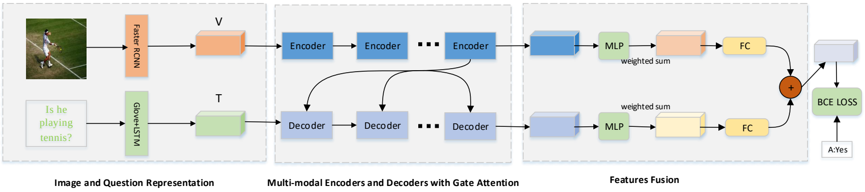


Fig. 1. The overall framework of MEDGA. MEDGA stacks multiple encoders and decoders with self-attention, cross-modal attention, and gate attention. Through MEDGA, we obtain visual features and question features. Input the fused features to answer prediction layer to get the answer to the question.

into a 300-dimensional word embeddings by GLoVe. The resulting word sequence size is $n \times 300$,where n is the number of words in the question. It is then sent to the LSTM. The word vector is encoded into 1024-dimensional features. The process of obtaining visual features V and language features T can be expressed by Equations (1) and (2),where θ_{lstm} and $\theta_{faster\,rcnn}$ re the parameters of the visual and language features.

$$V = Faster\ RCNN(I; \theta_{faster\,rcnn}) \tag{1}$$

$$T = LSTM(Q; \theta_{lstm}) \tag{2}$$

3.2. Encoder and Decoder

This article designs encoders and decoders block. MEDGA takes question representation V and image representation T as input, and then outputs their features with attention learning. Every encoder and decoder includes self-attention, cross-modal attention and gate attention modules.

In short, the attention mechanism is the process of mapping a query and a set of key-value pairs to output [41]. Both self-attention and cross-modal attention use multi-head attention [41]. Multi-head attention is calculated by scaled dot-Product attention h times respectively and it can make the model learn relevant information in different representation subspace. The scaled dot-Product attention mechanism is depicted in the left of Fig 2. The input to it is the query matrix (Q) with d_q dimension, the key matrix (K) with d_k dimension, and the value matrix (V) with d_v dimension. The dot product of the query and all keys are calculated. To prevent the dot product get too large, we adjust by $\frac{1}{\sqrt{d_k}}$ which called scaling factor, and then apply the softmax function to obtain the weight on the values. The formula for the scaling dot-Product attention is shown in Equation (3).

$$f_d(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

Multi-head attention consists of h parallel heads. Each head is an independent scaled dot-Product attention. The parameters attention structure is shown in the right of Fig

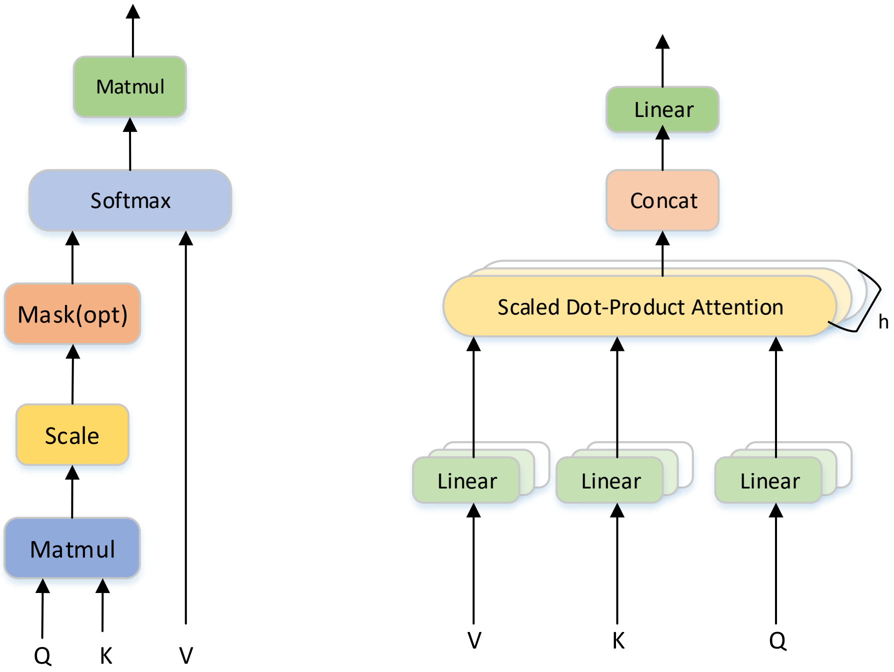


Fig. 2. Scaled dot-Product and Multi-Head Attention

2. For the sake of simplicity,$d_k = d_v = \frac{d_{model}}{h}$ in each attention layer. The specific calculation process is described by Equation (4), where $W_i^Q, W_i^K, W_i^V \in R^{d \times d_h}$ and $W^o \in R^{hd_v \times d_{model}}$ are parameter matrices.

$$\begin{aligned} f_m(Q,K,V) = Concat(head_1, head_2, ..., head_h)W^o \\ head_i = f_d(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{4}$$

The gate attention learns from the AoA model proposed in [22], which take the results of multi-head attention calculations and queries as input. We can get a more accurate attended information A from the following Equation (5).

$$A = \sigma(FC_q^g(Q) + FC_f^g(f_m)) \bigodot (FC_q^g(Q) + FC_f^g(f_m)) \tag{5}$$

Encoder Fig 3 illustrates the details of the encoder. The encoder is concerned with the single modal features of the question. When encoding each word, it also pays attention to other words in the input sentence. The encoder can capture some syntactic or semantic

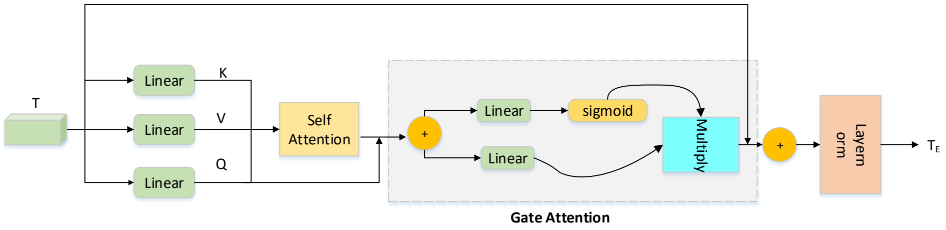


Fig. 3. The Encoder Architecture

features between words in the same sentence. It is easier to capture long-distance interdependent features in sentences, ensuring that important words in the question will be given greater weight. For example, for the question, "Where is the child sitting?", more attention should be attended to the words "child" and "sitting" and therefore the model has the ability to focus on the relevant regions and infer the correct answer. The input of the encoder is language feature $T = [t_1, t_2, t_3, t_4, ...t_m] \in R^{n \times d_t}$ and output a group of attended features $T_s a \in R^{n \times d}$.First, T is transformed into keys, values, and queries which are of the same shape through three independent fully connected layers. They are represented by $T_K, T_V, T_Q \in R^{n \times d}$. The multi-head attention in Equation (2) is used in the self-attention in the encoder.

$$T_K = Linear_k(T) \tag{6}$$

$$T_Q = Linear_q(T) \tag{7}$$

$$T_V = Linear_v(T) \tag{8}$$

Where Linear represents a fully connected layer, and d represents the same dimension of the transformed features in the language modality. The process of obtaining T_{sa} is as Equation (9):

$$\begin{aligned} T_{sa} &= Concat(head_1, head_2, ..., head_h)W^o \\ head_i &= f_d(T_Q W_i^Q, T_K W_i^K, T_V W_i^V) \\ f_d(Q, K, V) &= softmax(\frac{T_Q T_K^T}{\sqrt{d_k}})T_V \end{aligned} \tag{9}$$

Then T_{sa} is fused with the query Q in the original feature, and the fused feature is T'_{sa}.

$$T'_{sa} = Concat(T_{sa}, T_Q), T'_{sa} \in R^{n \times d} \tag{10}$$

The result T'_{sa} reflects the relationship between the words in the question, but in the process of learning self-attention, even if there is no related vector in the query, self-attention will generate a weighted vector, which makes the model confusing. Therefore, this article inputs the results of self-attention into the gate attention layer to get expected useful attended information.

The fused feature T'_{sa} is used as the input of the gate attention layers, which is input to the two linear layers to calculate the attended information vector A_E, which can filter out irrelevant results from attention. The calculation of it is as following, where σ is a non-linear sigmoid function and $FC_q^g, FC_f^g \in R^{d \times d}$.

$$A_E = \sigma(FC_q^g(T_Q) + FC_f^g(T_{sa})) \bigodot (FC_q^g(T_Q) + FC_f^g(T_{sa})) \tag{11}$$

After the gate attention layer, it is then fused with the original features, and Layernorm [5] is used to obtain the question features $T_E \in R^{n \times d}$ that after the encoder module. T_E is obtained by Equation (12). Layernorm plays a role of regularization and the model applying Layernorm is more stable.

$$T_E = Encoder(T) = Layernorm(A_E + T_k) \tag{12}$$

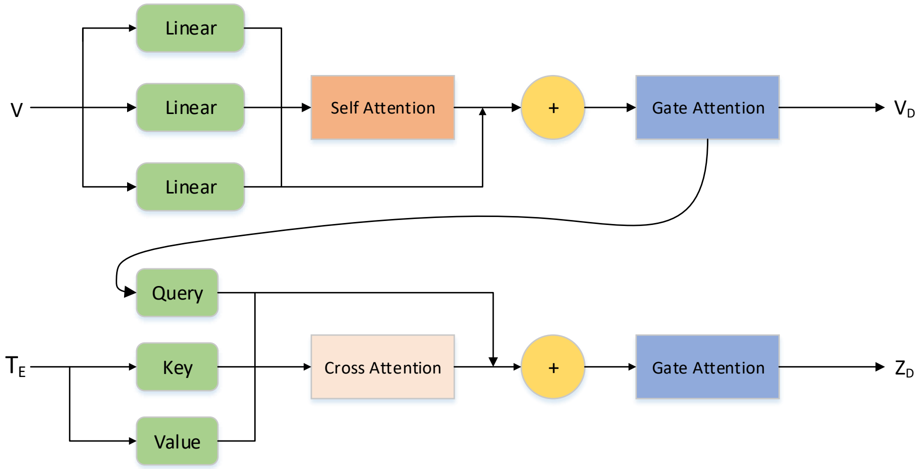


Fig. 4. The Decoder Architecture

Decoder The details of the decoder are shown in Fig 4. The decoder not only pays attention to the relationship between each visual area of each image, but also the connection between each image area and the words in the question, such as the question "What is the man holding with the left hand?", the decoder should focus on the visual area of the individual's left hand, and the relationship between the visual region of the left hand and the object.

The input of self-attention in the decoder is the visual feature $V = [v_1, v_2, v_3, v_4, ..., v_m] \in R^{m \times d_v}$. For an image, the relationship between different visual regions is different, so the weights should be different. Through stacking several decoders, we can model the relationship of objects in images. Similar to the process of calculating the language feature T_E, we can get V_D which gets through the self-attention layer and gate attention and use it as part of the cross-modal attention input.

Cross-modal attention uses a question's semantic features to guide the attention distribution of each region of the image while using gate attention to filter out unrelated attention results. Cross-modal also uses multi-head attention. Different from self-attention is that its input is the text feature T_E obtained by the encoder and the visual feature V_D obtained by the self-attention and gate attention calculations. The feature Z_{Ca} obtained by the cross-modal attention is calculated by Equation (16). The matrix Z_Ca captures the importance between each object region and the word. The computational procedure the gate attention resembles that of the encoder. After the decoder module, the multi-modal feature Z_D can be obtained.

$$Z_K = Linear_k(T_E) \tag{13}$$

$$Z_Q = Linear_q(T_E) \tag{14}$$

$$Z_V = Linear_v(T_E) \tag{15}$$

$$\begin{aligned} Z_{ca} &= Concat(head_1, head_2, ..., head_h)W^o \\ head_i &= f_d(QW_i^Q, KW_i^K, VW_i^V) \\ f_d(Q,K,V) &= softmax(\frac{Z_Q Z_K^T}{\sqrt{d}})Z_V \end{aligned} \tag{16}$$

3.3. Feature Fusion and Answer Generation

After stacking several encoders and decoders, the visual features $V' = [v_1, v_2, v_3, ..., v_x]$ and language features $T' = [t_1, t_2, t_3, ..., t_y]$ contains rich image and text information. For the two features, first apply a multi-layer perceptron (MLP) with ReLu nonlinear activation function, then the softmax function is devoted to obtain the attention weights which is relevant to the image and the question and finally weight the image and question features from all regions through these attention weights. The Relu activation function makes the output of some neurons zero, which makes the neural network sparse, reduces the interdependence of parameters, and relieves the occurrence of the over-fitting problem. The final weighted sum as the final visual and text features V_{attd}, T_{attd} can be described by the following formulas. V_{attd}, T_{attd} are projected to the same dimension by linear layer. Fusing such features adopts concatenation, or element-wise product, or addition. We use feature addition to obtain the final fused feature H, which can get best performance.

$$\tau_v = softmax(MLP(V')) \tag{17}$$

$$\tau_t = softmax(MLP(T')) \tag{18}$$

$$V_{attd} = \sum_{i=1}^{x} \tau_{v_i} v_i \tag{19}$$

$$T_{attd} = \sum_{i=1}^{x} \tau_{t_i} t_i \tag{20}$$

$$H = Layernorm(W_v^T V_{attd} + W_t^T T_{attd}) \tag{21}$$

In this paper, like other existing methods of visual question answering, we treat the VQA task as a multi-label classification task. The fused multi-modal feature H is input into the answer classifier in the answer prediction layer, and the score is standardized using the sigmoid function. It is between 0 and 1, which is used as the probability of the candidate answer. The final answer is the first 5 answers that appear most often and are used as classification labels.

The loss function in this paper refers to the strategy proposed in [38]. A BCE calculation function is used. The binary cross-entropy calculation function is described in Equation (20), where M is the number of training samples, s represents the probability of answer prediction and N is the number of candidate answers.

$$L = -\sum_{i}^{M} \sum_{j}^{M} s_{ij} \log(s'_{ij}) - (1 - s_{ij}) \log(1 - s'_{ij}) \tag{22}$$

4. Experiments

In this part, specific experiments for evaluating the effectiveness of MEDGA are presented.

4.1. Dataset

The experiments in this paper are performed on the VQA v2 dataset [17]. VQA v2 is a human annotated dataset for open-ended VQA. Each image from Microsoft COCO dataset [29] contains question-answer pairs. Compared with the VQA v1 dataset, it contains more examples for training, verifying, and testing. To prevent the improvement of model accuracy caused by over-fitting, each question corresponds to two images in VQA v2, so each question has two different answers. VQA v2 also minimizes some language biases in the VQA v1 dataset. VQA v2 contains 204721 images from the MSCOCO dataset an abstract scene dataset including 50,000 clipart. Each image in the dataset corresponds to three questions, and every question has 10 answers. This article evaluates MEDGA on 200,000 real images, including about 80,000 pictures in the training set, about 40,000 pictures in the validation set, and about 80,000 pictures in the test. 25% of the data in the test set is called test-dev. Examples of questions and answers on VQA v2 are shown in Figure 5. All questions fall into three categories: Yes / No, Number, Other. As in previous studies, this paper trains on the training and validation sets, and the results are verified on test-dev and test-standard. The experimental results include the accuracy of the three categories and overall accuracy.

Q: How many zebras are there?
A:Three

Q:Is he playing tennis?
A:Yes

Q:What color are these bananas?
A:Yellow

Q: What is the red food?
A:Tomato

Fig. 5. Typical example in VQA v2. The question types are Number, Yes / No, Other

4.2. Experimental Setup

Faster R-CNN is used to extract 2048-dimensional visual features. LSTM encode the language features into a 512-dimensional vector , then the two features are embedded into a 512-dimensional vector through a fully connected layer. The dimension of the fused feature is 1024. The self-attention in the encoder and the cross-modal attention in the decoder have 2 multi-head attention with 256 dimensions for each head. The batch size in every epoch is 64. Each training is 13 epochs. During the process of training, Adam optimization algorithm [25] is used with the parameters $\alpha = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.98$. Gradient clipping and dropout are also used in the experiment. The number of encoders and decoders is from 1 to 10. The candidate choices for the answer only retain the answers that appear more than 8 times in the training set, and the size of the answer vocabulary is 3129. All ablation experiments are performed on the validation set. The experiments in this article are implemented using Pytorch. All initialization is the default initialization in Pytorch.

4.3. Evaluation Metric

The answers to questions of the dataset are given by 10 different people which causes every question have different answers, such as “cat” has the same meaning as “kitty”. It cannot be determined which answer is correct. Therefore, to solve the inconsistency of answers, this paper uses the evaluation method proposed by Antol et al[3]. That means a prediction is right if and only if at least three persons have the same answer. The method can be described as Equation (21) where are the answers given by different annotators, a is the predicted answer, C = 10.

$$Accuracy(accuracy) = \frac{1}{C}\sum_{c=1}^{C}\min(\frac{\sum_{1\leq j\leq c, j\neq c}\prod a = a_j}{3}, 1) \tag{23}$$

4.4. Ablation Studies

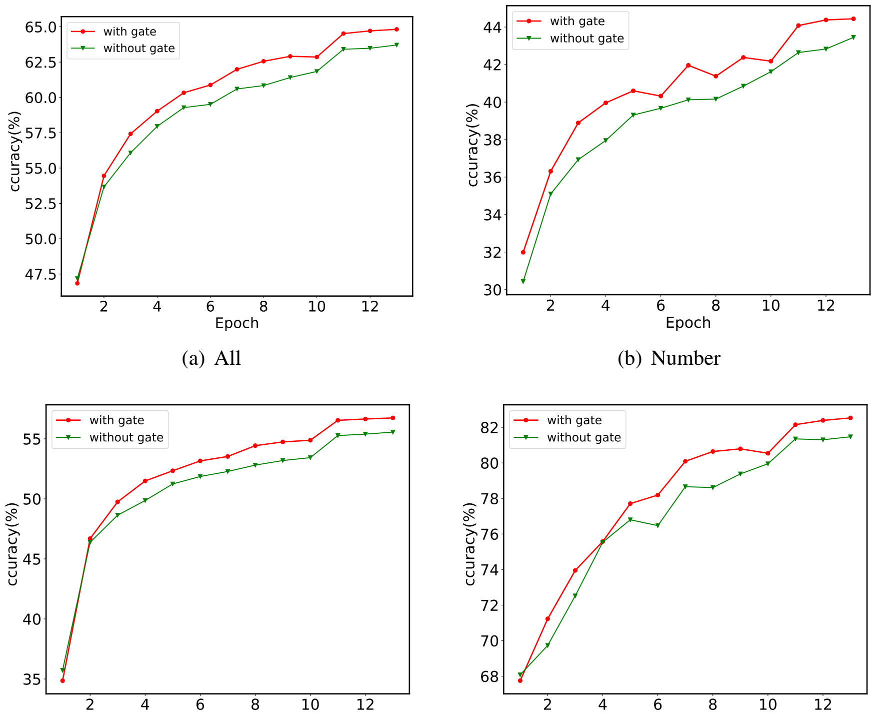
The MEDGA consists of several modules. For testing the impact of each component on the accuracy, this paper conducts several ablation studies under different conditions. With different parameters and settings, different versions of MEDGA are trained on the training

Table 1. Effect of Attention Head

head	All	Y/N	Num	Other
1	64.59	82.22	44.29	56.56
2	**64.82**	82.53	**44.44**	**56.74**
4	64.71	82.55	43.86	56.67
8	64.73	82.44	44.37	56.65

Table 2. The effect of the number of encoders and decoders on the experimental accuracy

Number	All	Y/N	Num	Other
1	64.82	82.53	44.44	56.74
2	65.58	83.13	46.82	57.39
4	66.10	83.41	47.08	57.97
6	66.46	83.94	47.86	58.10
8	**66.63**	**84.10**	**48.42**	**58.16**
10	66.63	84.09	48.83	58.05



(a) All

(b) Number

(c) Other

(d) Y/N

Fig. 6. The overall and per-type accuracies of the MEDGA along with the variants with gate attention and without gate attention

set of VQA v2, and the effects are displayed on the verification set to verify the effectiveness of MEDGA.

First, we conduct the ablation studies with different number of attention head. To save the run time, the default encoder and decoder are 1. Table 1 shows the influence of multi-head attention. It is found in the experimental results that when h = 2, the overall accuracy is the highest. Too fewer or too many heads reduces the overall accuracy.

Next, we explore the effect of the number of encoders and decoders. Table 2 shows the effect on the overall accuracy when the number of encoders and decoders is 1,2, 4, 6, 8,10. In this experiment, we use 2 head with the best performance As clearly appears from the table 2, when the number is 8, the overall accuracy is the highest. As the number of encoders and decoders continues to increase, the accuracy has not changed much, probably due to over-fitting. Considering the running time, we choose 8 in the final model.

Finally, the effectiveness of gate attention is verified. The default number of encoder and decoder is 1 and head is 2. Figure 6 shows the accuracy of each type in the model with and without gate attention. The model without gate attention only use self-attention and cross-attention. From the results, we can see that MEDGA with gate attention has a better performance than the model without gate attention on overall accuracy. Furthermore, for three categories questions, MEDGA has higher accuracy than without gate attention.

4.5. Visualization of Attention

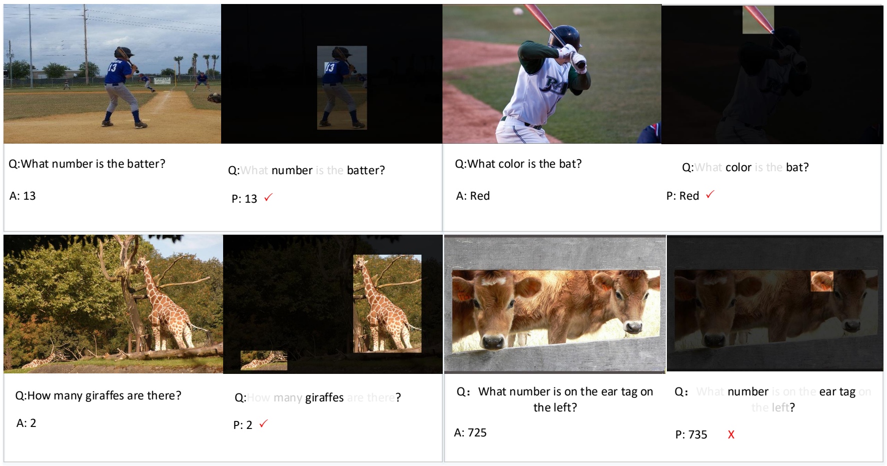

Q:What number is the batter?
A: 13
Q:What number is the batter?
P: 13 ✓
Q:What color is the bat?
A: Red
Q:What color is the bat?
P: Red ✓
Q:How many giraffes are there?
A: 2
Q:How many giraffes are there?
P: 2 ✓
Q： What number is on the ear tag on the left?
A: 725
Q： What number is on the ear tag on the left?
P: 735 X

Fig. 7. Typical example in VQA v2. The question types are Number, Yes / No, Other

In Figure 7, four examples of attention visualization are shown, involving types of questions such as color and counting. The left side of each diagram is the input image of the model, and the right side is the learned attention visualization. We observe from Figure 6 that MEDGA highlights the most relevant regions to the question and the most important words in the sentence. Q is the question asked for the picture, A represents the correct answer, while P denotes the predicted answer by the MEDGA. Among the four visualization examples given, there is an error example. The reason why the model predicts wrong is the model did not give the word "left" enough weight when assigning weights, which caused the model to mislocate in the image. Instead of focusing on the left object, it focused on the right object so that the model gives the wrong answer.

4.6. Comparison with Existing Advanced Methods

Table 3. Compared with some advanced methods on test-dev and test-std. all models are tested on vqa v2 dataset

Model	test-dev				test-std
	Y/N	Num	Other	All	All
BUTD	81.82	44.21	56.05	65.32	65.67
MFH	/	/	/	66.12	/
Graph	82.91	47.3	56.22	/	66.18
ODA-GCN	83.73	47.02	56.57	66.67	66.87
BU+QAA	/	/	/	66.70	67.0
DCN	83.51	46.61	57.26	66.87	66.97
Counter	83.14	51.62	58.97	68.09	68.41
MFH+BUTD	84.27	49.56	59.89	68.76	/
BAN	85.31	50.93	60.26	69.52	/
MDAnet	/	/	/	/	69.74
MEDGA(ours)	**85.97**	51.56	60.09	**69.78**	**70.11**

Table 3 shows the performance of the proposed MEDGA and existing advanced methods on VQA v2, where / indicates that the model has not tested the accuracy of the question type or the dataset. In Table 3, BUTD [2] won the champion in the VQA challenge 2017. It puts forward to use Faster R-CNN to extract features not using ResNet [21]. MFH [47] is a state-of-the-art bilinear pooling method. Graph [33] builds a graph in all the regional propose boxes and conditions this graph on the question.ODA-GCN [49] is also a graph-based visual question answering method, and they introduce a soft attention layer. QAA [12] proposes a question-agnostic attention mechanism that complements existing attention mechanisms. The Dense Symmetric Co-Attention Model (DCN) [32] stacks multiple co-attention modules. Although it does not use Faster RCNN but uses ResNet to extract image features, the experimental results are better than previous advanced methods. Counter [48] makes full use of bounding box information to make it highly accurate in counting type questions. Bilinear Attention Network [23] has 12 stacked bilinear attention modules.MDAnet [13] is a method that uses a multi-modal encoder to replace the RNN in the traditional method, so that the position can be reserved. MEDGA outperforms the advanced models above on both test-dev and test-std datasets, and it achieves 70.11% accuracy on test-std. On Y/N type, MEDGA perform the best. Although our model is a little worse than Counter on num type, MEDGA outperforms other methods on this type due to the advantages of our model in modeling The overall accuracy of MEDGA on test-dev exceeds the current advanced method BAN by 0.26 percentage points and exceeds DCN by 2.91 percentage points, verifying the performance of cross-modal attention and gate attention. Through the comparison with other methods, we can prove the effectiveness of MEDGA.

5. Conclusion and Future Work

This paper raises and designs a VQA model on the basis of a multi-modal encoder and decoder with gate attention. This model solves the visual question answering task by stacking multiple encoders and decoders. The core of the model is to use self-attention, cross-modal attention, and gate attention. MEDGA pay close attention to major words of the questions and model the relationship between various visual regions in the images. The attention maps obtained only by self-attention and cross-modal attention may have a lot of irrelevant information. The introduced gate attention can solve this problem and make the final attention result more accurate and more conducive to the final answer prediction. The MEDGA presented in this paper is simple and efficacious. The experimental results on the VQA v2 dataset prove the performance of the model. But our model has certain defects in counting and other types. In the future, we will focus on how to make the model more accurate in object detection so that the accuracy of counting is higher. On the other hand, we will study the performance of our model on other datasets, as well as its application in medical images, satellite image recognition, etc. Besides, we will conduct more in-depth research on wireless sensors and combine their wide range of applications with VQA, so that VQA will have a broader application prospect and benefit mankind.

References

1. Ahutu, O.R., El, H.: Centralized routing protocol for detecting wormhole attacks in wireless sensor networks. IEEE Access 8, 63270–63282 (2020)
2. Anderson, P., He, X., Buehler, C., Teney, D., Mark Johnson, S.G., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6077–6086. IEEE Computer Society, Salt Lake City, UT, USA (2018)
3. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2425–2433. IEEE Computer Society, Santiago, Chile (2015)
4. B, H., H., Z.: Obstacle-aware fuzzy-based localization of wireless chargers in wireless sensor networks. Electrical and Computer Engineering 43(1), 17–24 (2019)
5. Ba, L.J., Kiros, J.R., Hinton, G.E.: Layer normalization. CoRR (2016)
6. Ben-younes, H., Cadène, R., Cord, M., Thome, N.: Mutan: Multimodal tucker fusion for visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2631–2639. IEEE Computer Society, Venice,Italy (2017)
7. Chen, C., Han, D., Wang, J.: Multimodal encoder-decoder attention networks for visual question answering. IEEE Access 8, 35662–35671 (2020)

8. Cui, M., Han, D., Wang, J.: An efficient and safe road condition monitoring authentication scheme based on fog computing. IEEE Internet Things J. 6(5), 9076–9084 (2019)
9. Cui, M., Han, D., Wang, J., Li, K.C., Chan, C.C.: Arfv: An efficient shared data auditing scheme supporting revocation for fog-assisted vehicular ad-hoc networks. IEEE Transactions on Vehicular Technology PP(99), 1–1 (2020)
10. Devlin, Jacob, M.W.C.K.L., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language, NAACL-HLT. pp. 4171–4186. Association for Computational Linguistics, Minneapolis, MN, USA (2019)
11. Dongfei, Y.: Attention mechanism and high-level semantics for visual question answering. University of Science and Technology of China (2019)
12. Farazi, M.R., Khan, S.H., Barnes, N.: Question-agnostic attention for visual question answering. CoRR (2019)
13. Feng, J., Gong, P., Qiu, G.: Mdanet: Multiple fusion network with double attention for visual question answering. In: Proceedings of The 3rd International Conference on Video and Image Processing. pp. 143–147. ACM, Shanghai, China (2019)
14. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 457–468. The Association for Computational Linguistics, Austin, Texas, USA (2016)
15. Gao, P., Jiang, Z., You, H., Lu, P., Hoi, S.C.H., Wang, X., Li, H.: Dynamic fusion with intra- and inter- modality attention flow for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6639–6648. Computer Vision Foundation / IEEE, Long Beach, CA, USA (2019)
16. Gordo, A., Larlus, D.: Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5272–5281. IEEE Computer Society, Honolulu, HI, USA (2017)
17. Goyal, Y., Khot, T., Agrawal, A., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6325–6334. IEEE Computer Society, Honolulu, HI, USA (2017)
18. Gurari, D., Li, Q., Stangl, A.J., Guo, A., Lin, C., Grauman, K., Luo, J., Bigham, J.P.: Vizwiz grand challenge: Answering visual questions from blind people. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3608–3617. IEEE Computer Society, Salt Lake City, UT, USA (2018)
19. Han, D., Pan, N., Li, K.C.: A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. IEEE Transactions on Dependable and Secure Computing PP(99), 1–1
20. Han, D., Yu, Y., Li, K., de Mello, R.F.: Enhancing the sensor node localization algorithm based on improved dv-hop and DE algorithms in wireless sensor networks. Sensors 20(2), 343 (2020)
21. He, Kaiming, X.Z.S.R., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778. IEEE Computer Society, Las Vegas, NV, USA (2016)
22. Huang, L., Wang, W., Chen, J., Wei, X.: Attention on attention for image captioning. In: Proceedings of the International Conference on Computer Vision. pp. 4633–4642. IEEE, Seoul, Korea (South) (2019)
23. Kim, J., Jun, J., Zhang, B.: Bilinear attention networks. In: Proceedings of the Conference and Workshop on Neural Information Processing Systems. pp. 1571–1581. Montréal, Canada (2018)

24. Kim, J., On, K.W., Lim, W., Kim, J., Ha, J., Zhang, B.: Hadamard product for low-rank bilinear pooling. In: Proceedings of the International Conference on Learning Representations. OpenReview.net, Toulon, France (2017)
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations. San Diego, CA, USA (2015)
26. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. 123(1), 32–73 (2017)
27. Li, H., Han, D.: A novel time-aware hybrid recommendation scheme combining user feedback and collaborative filtering. Mob. Inf. Syst. 2020, 8896694:1–8896694:16 (2020)
28. Li, H., Han, D., Tang, M.: A privacy-preserving charging scheme for electric vehicles using blockchain and fog computing. IEEE Systems Journal pp. 1–12 (2020)
29. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proceedings of the European Conference on Computer Vision. pp. 740–755. Springer, Zurich,Switzerland (2014)
30. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: Proceedings of the Conference and Workshop on Neural Information Processing Systems. pp. 289–297. Barcelona, Spain (2016)
31. Nam, H., Ha, J., Kim, J.: Dual attention networks for multimodal reasoning and matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2156–2164. IEEE Computer Society, Honolulu, HI, USA (2017)
32. Nguyen, D., Okatani, T.: Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6087–6096. IEEE Computer Society, Salt Lake City, UT, USA (2018)
33. Norcliffe-Brown, W., Vafeias, S., Parisot, S.: Learning conditioned graph structures for interpretable visual question answering. In: Proceedings of the Conference and Workshop on Neural Information Processing Systems. pp. 8344–8353. Montréal, Canada (2018)
34. Qadir, J., Ullah, U., de Abajo, B.S., Bego: Energy-aware and reliability-based localization-free cooperative acoustic wireless sensor networks. IEEE Access 8, 121366–121384 (2020)
35. Ren, F., Zhou, Y.: Cgmvqa: A new classification and generative model for medical visual question answering. IEEE Access 8, 50626–50636 (2020)
36. Shi, Y., Furlanello, T., Zha, S., Anandkumar, A.: Question type guided attention in visual question answering. In: Proceedings of the European Conference on Computer Vision. pp. 158–175. Springer, Munich,Germany (2018)
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations. San Diego, CA, USA (2015)
38. Teney, D., Anderson, P., He, X., van den Hengel, A.: Tips and tricks for visual question answering: Learnings from the 2017 challenge. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4223–4232. IEEE Computer Society, Salt Lake City, UT, USA (2018)
39. Teney, D., Wu, Q., van den Hengel, A.: Visual question answering: A tutorial. IEEE Signal Processing Magazine 34(6), 63–75 (2017)
40. Tian, Q., Han, D., Li, K.C., Liu, X., Castiglione, A.: An intrusion detection approach based on improved deep belief network. Applied Intelligence (3) (2020)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the Conference and Workshop on Neural Information Processing System. pp. 5998–6008. Long Beach, CA, USA (2017)
42. Venugopal, K.R., T., S.P., Kumaraswamy, M.: Qos routing algorithms for wireless sensor networks. Springer (2020)

43. Wu, Q., Shen, C., Wang, P., Dick, A.R., van den Hengel, A.: Image captioning and visual question answering based on attributes and external knowledge. IEEE Trans. Pattern Anal. Mach. Intell. 40(6), 1367–1381 (2018)
44. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: Proceedings of the International Conference on Machine Learning. pp. 2048–2057. JMLR.org, Lille, France (2015)
45. Yang, C., Jiang, M., Jiang, B., Zhou, W., Li, K.: Co-attention network with question type for visual question answering. IEEE Access 7, 40771–40781 (2019)
46. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.J.: Stacked attention networks for image question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 21–29. IEEE Computer Society, Las Vegas, NV, USA (2016)
47. Yu, Z., Yu, J., Xiang, C., Fan, J., Tao, D.: Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. IEEE Trans. Neural Networks Learn. Syst. 29(12), 5947–5959 (2018)
48. Zhang, Y., Hare, J.S., Prügel-Bennett, A.: Learning to count objects in natural images for visual question answering. In: Proceedings of the International Conference on Learning Representations. OpenReview.net, Vancouver, BC, Canada (2018)
49. Zhu, X., Mao, Z., Chen, Z., Li, Y., Wang, B.: Object-difference drived graph convolutional networks for visual question answering. Multimedia Tools and Applications (2020)

Haiyan Li is currently pursuing the M.S degree in Shanghai Maritime University, China. Her research interests include visual question answering and deep learning.

Dezhi Han received the B.S. degree in applied physics from Hefei University of technology, China, in 1990, the Ph.D. degree in computing science from the Huazhong University of Science and Technology, China, in 2005. He is currently a Professor with the Department of Computer, Shanghai Maritime University, China. in 2010. His research interests include reinforcement learning and deep learning, wireless communication security, network and information security. He is a member of IEEE.

Received: November 20, 2020; Accepted: February 10, 2021.

Computer Science and Information Systems 18(3):1041–1056 https://doi.org/10.2298/CSIS200620019L

Identifying Key Node in Multi-region Opportunistic Sensor Network based on Improved TOPSIS

Linlan Liu[1], Wei Wang[1], Guirong Jiang[2], and Jiang Zhang[1]

[1] School of Information Engineering, Nanchang Hangkong University,
330063 Nanchang, China
765693987@qq.com
919269210@qq.com
zhangjiangky@163.com
[2] School of Software, Nanchang Hangkong University,
330063 Nanchang, China
1132153564@qq.com

Abstract. The topology of multi-region opportunistic sensor networks is evolving, and it is difficult to identify the key nodes in the networks by traditional key node identification methods. In this paper, a novel method based on the improved TOPSIS method is proposed to identify the key node from the ferry node. The dynamic topology information is represented by the graph model which is modeled by the temporal reachable graph. Based on the temporal reachable graph, three attributes are constructed to identify the key node, which are average degree, betweenness centrality and message forwarding rate. The game theory with a combination weighting method is employed to combine the subjective weight and objective weight, so as to obtain the combined weight of each attribute. The TOPSIS method is improved by the combined weight. The key node is identified by the improved TOPSIS. The experiments in three simulation situations show that, compared with the TOPSIS method and MADM_TOPSIS method, the proposed method has better accuracy for the key node identification in the network.

Keywords: multi-region opportunistic sensor network, key node, combination weight, TOPSIS.

1. Introduction

Multi-region opportunistic sensor networks (MOSNs) are a type of self-organizing network which can collect sensor data through the movement of nodes and encounters between nodes. Part of the concept of MOSNs is derived from mobile ad hoc networks (MANETs) and delay-tolerant networks (DTNs), such as the Intermittent links, temporal paths, real-time messages, etc. [1] MOSNs consist of nodes and links between nodes. The node that has the greatest influence on the network structure and function is called a key node. The events that the key node is attacked or failed may lead the networks to be paralyzed. By identifying the key node, MOSNs can be optimized in advance to improve its security and robustness. Hence how to accurately and efficiently identify the key node in MOSNs is a hot topic.

Aim at solving this problem, lots of indicators have been proposed [2], such as degree centrality [3], betweenness centrality [4], eigenvector centrality [5], Katz centrality [6], etc. Although these methods can identify the key nodes in complex networks from different perspectives, the adaptability and accuracy are easily affected by the factors such as the network structure and scale. On this basis, some researchers combine multiple indicators to identify the key nodes, thereby improving applicability and stability. The reference [7] defines four parameters to represent the influence of a node in social networks. The direct influence spread and indirect influence spread are used to indicate the influence of a node on other nodes. The direct overlaps and indirect overlaps reflect the conflict between nodes. Then, the technique for order preference by similarity to an ideal solution (TOPSIS) is used to combine these parameters to obtain the influence of each node. Fei et al. [8] believe that the interaction between nodes follows the inverse-square law, and the node importance is evaluated by combining the degree centrality of nodes and the distance between nodes. The experiments show that the accuracy of the method is higher than some well-known centrality indicators.

The location of a node in the network determines its importance. Korn et al. [9] are inspired by the fact that the H-index quantifies the contribution of scholars in informatics and use the H-index to evaluate the node importance. If a node has n neighbor nodes whose degree is not less than n, then the H-index of the node is n. Kitsak et al. [10] use K-shell to judge the location of nodes in the network and think that the nodes in the core location usually are more important. The Ks of each node is determined by separating the nodes from the network according to the order of residual degrees from small to large, and it considers fully the global characteristics of nodes. But the K-shell is not suitable for some special networks such as the tree networks and star networks. When the node remaining degree is less than the current number of iterations, the iteration cannot be carried out properly. Lü et al. [11] propose that the H-operation based on the degree centrality converges to the Ks of the node. The view can avoid possible errors in the K-shell process and improve the monotonicity of the evaluation results. Based on the reference [11], Shao et al. [12] propose an important node identification method based on the H-operation in dynamic networks. This method takes the smaller value in the past H-index and the present Ks as the initial value of the H-operation, so that the important nodes can be found quickly at every moment.

However, the topology of complex networks usually is changing, the nodes and edges may appear or disappear at any time, and the network is called a temporal network [13]. In recent years, some researchers have begun to identify the key node in the temporal network. Zhang et al. [14-18] model MANETs as time-varying graphs to represent the topology of temporal networks. Based on the previous research, Zhang et al. [19] define a new metric called criticality that can measure node importance accurately in MANETs, and the experiments show that attacking the key node identified by the criticality has a greater impact on network performance than some centrality indexes. Based on the view that the node importance depends on their neighbors, the reference [20] proposes a temporal information aggregation process to identify the key node in temporal networks. Arrigo et al. [21] utilize the sparse version of dynamic communicability matrix to estimate node importance and rank for nodes, and the experiments show that this method can rank the list of highly central nodes accurately with a lower level of storage, and the cost is only linearly with the number of time points. Abbas et al. [22] divide the data into past time window and future time window based on user-object binary networks, to identify and predict the key node (the popular or important objects in the future) in e-commerce networks and social networks. Xiao et al. [23] predict the most powerful persuaders based on machine learning in social networks. The reference [24] proposes coverage centrality in temporal networks. It is found that the most of nodes with high centrality are located in a small time window near a certain time. The majority of information in temporal networks is only transmitted by the minority of nodes, and there is a bottleneck period in the transmission process.

Different from the general temporal networks, there are three types of nodes in MOSNs, which are sink node, ferry node, and sensor node. The sink node collects all of the messages generated from the sensor region and sends them to the server. The ferry node walks along fixed or random routes in the sensor region and forwards the messages from the sensor region to the sink node. The sensor node is fixed in the sensor subregion and generates the messages that contain sensor data. The key node must be found from the ferry node. The main contributions of this paper are as follows:

(1) The dynamic topology information is represented by the graph model which is modeled by the temporal reachable graph. Based on the temporal reachable graph, three attributes are constructed to identify the key node, which are average degree, betweenness centrality and message forwarding rate.

(2) The TOPSIS method is improved by the combined weight. The game theory with a combination weighting method (GTCW) is employed to combine the subjective weight and objective weight, so as to obtain the combined weight of each attribute.

(3) This paper uses the simulator ONE to conduct simulation experiments in three experimental scenarios. The simulation results show that compared with methods such as the TOPSIS method and MADM_TOPSIS method, the method proposed in this paper has better accuracy for the key node identification in MOSNs.

The paper is organized as follows: The problem description and definitions about the temporal reachable graph are presented in section 2. The key node identification method based on the improved TOPSIS (GTCW_TOPSIS) is introduced in section 3. The experiments and results are shown in section 4. The conclusion and prospect are given in section 5.

2. Problem Description and Definitions

2.1. Problem Description

Due to various reasons such as node damage, signal attenuation, and geographical environment, the sensor region of MOSNs may be divided into multiple subregions in practical application. MOSNs consist of one sink node, several sensor nodes, and several ferry nodes. The sink node is fixed and used to collect the sensor data from sensor nodes. The sensor node is placed in the sensor region and senses environmental conditions. The ferry node moves between the sink node and the sensor region, and forwards the messages from the sensor region to the sink node by the "carry-store-forward" mechanism. The process of communication is as follows: Firstly, the ferry node receives the messages from the sensor region when it passes through the sensor region. Then, the ferry node saves the messages in the cache unit it encounters with other ferry nodes or the sink node. Finally, the ferry node sends the messages to the sink node.

In order to reduce the complexity of the research, a sensor subregion is regarded as a region node instead of considering every sensor node separately. Compare with the sensor node, the ferry node plays a vital role in the communication between the region node and the sink node, and the topology of MOSNs changes with the location of the ferry node. It is obvious that the ferry node is the most valuable node besides the sink node, so the key node is selected among the ferry nodes. As shown in fig. 1, some sensor nodes collect data in the region node R_1, the ferry node F_1 serves as a bridge to deliver messages between the region node R_1 and R_4, and when F_1 encounters with F_2, it will send the messages received from R_1 to F_2. The event that F_1 is damaged is likely to cause the region node R_1 to be separated from the network.

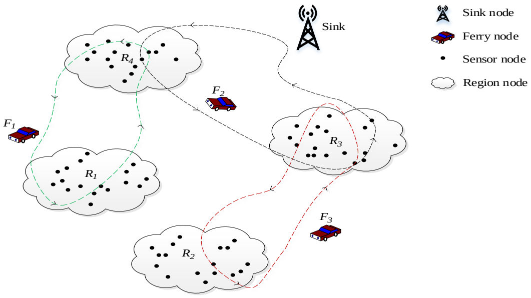


Fig. 1. The scenario graph of MOSNs

2.2. Definitions

In order to realize the effective transmission of sensor data, through the mechanism that location of the ferry node changes, the network structure of MOSNs is frequently changed. Aiming to represent the dynamic topology information and reduce the temporal information loss, the temporal reachable graph is used to model MOSNs.

Definition 1: The temporal reachable graph $G = \{G_1, G_2, G_3, \cdots, G_L\}$ is a set that is composed of several ordered graphs during the observation period $[0, T]$, where L is the number of temporal reachable subgraphs, $G_l = (V_l, E_l, W_l), l = 1,2,3, \cdots, L$ is the l_{th} temporal reachable subgraph. V_l is the node set in G_l, and it consists of the sink node S, the ferry node set F and the region node set R, E_l is the edge in G_l, W_l is the edge weights set in G_l. The key node will be found in F.

Definition 2: The set W_l in $G_l = (V_l, E_l, W_l)$ is defined in (1):

$$W_l = \left\{ w_{ab}^l \,|\forall a, b \in V_l \; and \; (a, b) \in E_l \right\} \tag{1}$$

In which w_{ab}^l is the number of connections between the node a and the node b in G_l. According to the Definition 1 and 2, considering the changes of the network structure in each time window, the subnet in the interval $[t_{l-1}, t_l]$ is aggregated as G_l, the temporal reachable subgraph sequence $\{G_1, G_2, G_3, \cdots, G_L\}$ are shown in fig. 2.

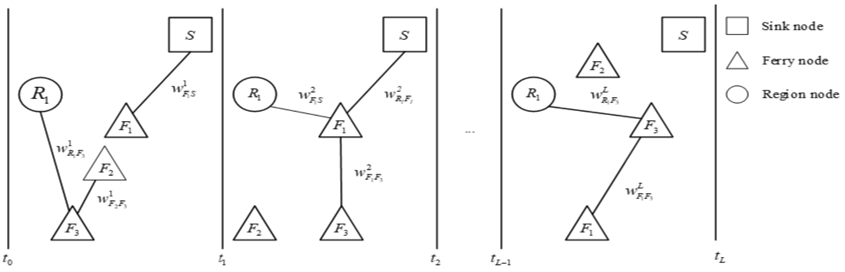


Fig. 2. Temporal reachable subgraphs

As shown in Fig. 2, in G_1 and G_L, there are two edges between R_1 and F_3, $w^1_{R_1F_3}$ and $w^L_{R_1F_3}$ are the edge weight between R_1 and F_3, and there is not edge between R_1 and F_3 in G_2. In summary, the edge weight is aggregated at every temporal reachable subgraph to construct the temporal reachable graph, and as shown in Fig. 3.

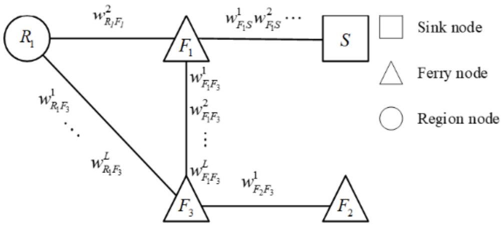


Fig. 3. Temporal reachable graph

Definition 3: In MOSNs, the messages eventually are aggregated to the sink node along the temporal reachable path. As for the region node R_i and the sink node S, if there is an edge sequence $[R_i, x_1], [x_1, x_2], [x_3, x_2], \dots, [x_{n-1}, S]$ that exists in the network G, the sequence is a temporal reachable path from R_i to S. In Fig. 3, the messages can be transmitted from R_1 to S by two temporal reachable paths: $R_1 \to F_1 \to S$ and $R_1 \to F_3 \to F_1 \to S$. Different from the traditional path, the edges that make up a temporal reachable path must follow the order of time, which results in lower usability of paths than that in static networks. Hence, the temporal reachable path is applied to representing the information of message transmission. The temporal reachable paths are shorter, the communication and interaction are easier between a pair of nodes.

3. Identifying Key Node

In this section, we define three attributes of the ferry node as indicators, then calculate the combination weight by the game theory with a combination weighting method [26]. Based on the above model, we propose a method named GTCW_TOPSIS to identify the key node.

3.1. Three Attributes of Ferry Node

Definition 4: In the subgraph G_l, the average degree AD_{F_i} of the ferry node F_i is defined in (2):

$$AD_{F_i} = \frac{\sum_{j=1}^{N}\sum_{l=1}^{L}\omega_{F_iR_j}^{l}}{L} \quad (2)$$

In (2), L denotes the number of the temporal reachable subgraph, N is the number of nodes in the network. The average degree reflects the relation with the surrounding nodes. In general, the greater the average degree, the more important the ferry node.

Definition 5: The betweenness centrality BC_{F_i} of F_i is defined in (3):

$$BC_{F_i} = \sum_{j=1}^{|R|}\frac{g_{R_jS}^{F_i}}{g_{R_jS}} \quad (3)$$

Where R_j is a region node in the network, S is the sink node, g_{R_jS} denotes the number of the temporal reachable path from R_j to S, $g_{R_jS}^{F_i}$ denotes the number of the the temporal reachable path through F_i in g_{R_jS}. The betweenness centrality reflects the ability that the ferry node affects the message transmission paths from the region node to the sink node. The larger the betweenness centrality, the more important the ferry node.

Definition 6: The message forwarding rate MFR_{F_i} of F_i is defined in (4):

$$MFR_{F_i} = \frac{m_{F_i}}{\sum_{j=1}^{|R|} m_{R_j}} \quad (4)$$

Where m_{R_j} denotes the total number of the messages forwarded from F_i to S, M_b denotes the total number of the messages generated by R_j, the message forwarding rate reflects contribution of the ferry node to message delivery in MOSNs.

3.2. Attribute Weights

The subjective weight ω_1 and objective weight ω_2 of the attributes are obtained by the analytic hierarchy process (AHP) and entropy method respectively. By constructing a basic weight set $W = \{\omega_1, \omega_2\}$, the combination weight is defined in (5):

$$\omega = \sum_{k=1}^{2}\alpha_k\omega_k^T, \omega_k \in W \quad (5)$$

The combination weights consist of ω_1 and ω_2, where α_k is the weight coefficient of different ω_k. Then, we minimize the deviation of ω and ω_k, as show in (6):

$$min||\sum_{k=1}^{2}\alpha_k\omega_k^T - \omega_h^T||_2 \quad (6)$$

According to the differential nature of the matrix, equation (7) is the condition that optimizes the first derivative of (6):

$$\sum_{k=1}^{2}\alpha_k\omega_h\omega_k^T = \omega_h\omega_h^T \quad (7)$$

The corresponding linear equation as shown in (8):

$$\begin{bmatrix} \omega_1\omega_1^T & \omega_1\omega_2^T \\ \omega_2\omega_1^T & \omega_2\omega_2^T \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \omega_1\omega_1^T \\ \omega_2\omega_2^T \end{bmatrix} \tag{8}$$

The weight coefficient vector (a_1, a_2) is obtained in (8), and we normalize (a_1, a_2) according to [27]:

$$\alpha^* = \frac{\alpha_k}{\sum_{h=1}^{2} \alpha_h} \tag{9}$$

Finally, the weight coefficient normalized vector α^* is substituted into (5), and the combination weight ω^* is calculated in (10):

$$\omega^* = \sum_{k=1}^{2} \alpha_k^* \omega_k^T \tag{10}$$

3.3. Estimation Algorithm

In this paper, the key node is identified by a new method called GTCW_TOPSIS. The steps used to identify the key node in MOSNs are as follows:

(1) Construct normalized decision matrix

It is assumed that there are n ferry nodes in the network so that the corresponding solution set denoted by $F = \{F_1, F_2, F_3, \dots, F_N\}$, the attributes of each ferry node are denoted by the attribute set $A = \{\alpha_1, \alpha_2, \alpha_3\}$, where α_1 is AD, α_2 is BC, α_3 is MFR. The decision matrix is expressed as (11):

$$X = \left(x_{ij}\right)_{n\times 3} \tag{11}$$

Where x_{ij} $(i = 1,2,3, \cdots, n; j = 1,2,3)$ is the j_{th} attribute of the i_{th} ferry node.

The normalized decision matrix is obtained by the vector normalization method according to (12) ~ (13):

$$Y = \left(y_{ij}\right)_{n\times 3} \tag{12}$$

$$y_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}} \tag{13}$$

(2) Construct weighted normalized decision matrix

The combination weight of each attribute is calculated by (5) ~ (10), and the weight of j_{th} attribute is denoted as ω_j^*. The weighted normalized decision matrix E is denoted as (14):

$$E = \left(e_{ij}\right)_{n\times 3} = \left(\omega_j^* y_{ij}\right)_{n\times 3} \tag{14}$$

Where $e_{ij} = \omega_j^* y_{ij}$ $(i = 1,2,3, \cdots, n; j = 1,2,3)$.

(3) Determine the positive ideal solution A^+ and the negative ideal solution A^-

The maximum value of each attribute constitutes a positive ideal solution A^+, and the negative ideal solution A^- is composed of the minimum value of each attribute. As shown in (15) ~ (18).

$$e_j^+ = \max_i \left\{ e_{ij} \,\middle|\, i = 1,2,3, \dots, n; j = 1,2,3 \right\} \tag{15}$$

$$e_j^- = \min_i \{e_{ij} \,|\, i = 1,2,3, \dots, n; j = 1,2,3\} \tag{16}$$

$$A^+ = \left(e_j^+ | j = 1,2,3\right) \tag{17}$$

$$A^- = \left(e_j^- | j = 1,2,3\right) \tag{18}$$

(4) Calculate the Euclidean-Distance from each solution to A^+ or A^-

The Euclidean-Distance from each solution to A^+ or A^- is the deviation between them.

$$d_i^+ = \sqrt{\sum_{j=1}^{3} \left(e_{ij} - e_j^+\right)^2} \tag{19}$$

$$d_i^- = \sqrt{\sum_{j=1}^{3} \left(e_{ij} - e_j^-\right)^2} \tag{20}$$

(5) Calculate the closeness from each solution to A^+ and A^-

The closeness of the node i indicates how close the solution is to the positive ideal solution.

$$c_i^+ = \frac{d_i^-}{d_i^+ + d_i^-} \tag{21}$$

Where $0 < c_i^+ < 1$.

(6) Construct closeness set $C^+ = \{c_1^+, c_2^+, c_3^+, \cdots, c_n^+\}$, and the node with the largest c_i^+ is suspected to be a key node.

(7) The steps 1~6 is repeated for k times, the time that each ferry node is identified to be a suspected key node is recorded, and the ferry node with the most times is the key node.

3.4. Verification of Results

The node removal method is utilized to verify the experiment result, and the whole network delivery success rate (WNDSR) of the network which a node is removed is compared with that of the complete network, the process is repeated until all ferry nodes are removed. The WNDSR can be used to reflect the performance of MOSNs. If the event that a node is removed makes the greatest reduction in the WNDSR, the node is the key node.

Definition 7: In MOSNs, the whole network delivery success rate is defined as:

$$WNDSR = \frac{m_S}{\sum_{i=1}^{|R|} M_{R_i}} \tag{22}$$

Where m_S denotes the total number of messages received by the sink node S during the observation period $[0, T]$.

4. Experiments and Analysis

In this section, we use the WNDSR as a basis for identifying the key node and compare the accuracy that the key node is identified by the GTCW_TOPSIS method the TOPSIS method [27] and MADM_TOPSIS method [28] under three different scenarios.

4.1. Experiments

Three scenarios are simulated by the simulator ONE. ONE is an opportunity network simulator developed by the University of Helsinki in Finland. The parameters of the three scenarios are shown in Table 1.

Table 1. The parameters in the three scenarios.

parameter name	value
Radius of region node	50 m
region node cache	20 M
Radius of ferry node	100 m
ferry node cache	50 M
Date transfer rate	250 kB/s
Router	Epidemic Router
Message survival time	10 min

As shown in fig. 4, there are one sink node (s), six ferry nodes (fa, fb, fc, fd, fe and fg) and five region nodes (ra, rb, rc, rd and re) in scenario 1. There are 20 sensor nodes in each region node. Among the six ferry nodes, the movement model of fa, fg, fd and fe is the random way point model, they walk randomly between the sensor region and the sink node, and the movement model of fb and fc is the movement based map model. Without considering the node whose movement model is the random way point model, the movement track of fb passes through re, the movement track of fc passes through ra, fb and fc can communicate with s.

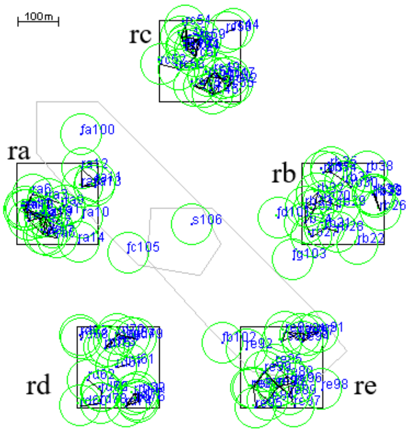


Fig. 4. Scenario 1

As shown in fig. 5, there are one sink node (s), six ferry nodes (fa, fb, fc, fd, fe and fg) and five region nodes (ra, rb, rc, rd and re) in scenario 2. There are 20 sensor nodes in each region node. The movement models of the six ferry nodes are the movement based map model. fc, fe and fg can connect with s directly, fa, fb and fd cannot communicate with s directly, and the movement track of fa and fd passes through ra, rb and rc, the movement track of fb passes through re and rd, the movement track of fc passes through re, the movement track of fe and fg passes through ra.

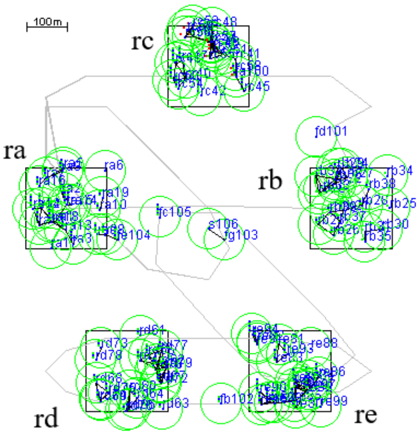


Fig. 5. Scenario 2

As shown in fig. 6, there are one sink node (s), six four nodes (fa, fb, fc and fd) and three region nodes (ra, rb, and rc) in scenario 3. There are 20 sensor nodes in each region node. The movement models of the four ferry nodes all are the movement based map model. Among them, the movement track of fa and fb passes through rb and ra, fc walks between ra and rb, and fb walks between ra and rc.

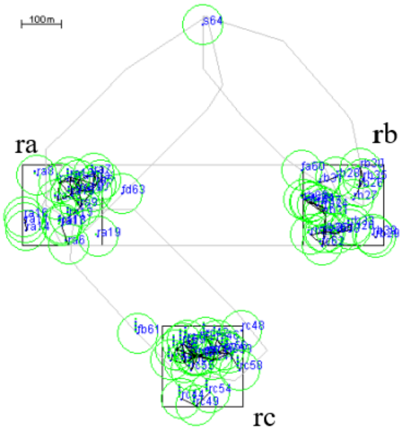


Fig. 6. Scenario 3

4.2. Results

The time window is set to 10 minutes for each experiment, and the experiments are repeated 200 times in each scenario, the results obtained are shown in fig. 7~9.

From fig. 7-9, we imply that the key nodes of scenario 1-3 are fb, fc and fd. The number of identifying the suspected key node by the GTCW_TOPSIS method is more than the others. Although the key node in MOSNs can be identified by the three methods, the effectiveness of the three methods needs to be verified by the WNDSR. If the WNDSR is the lowest, after the key node identified by the GTCW_TOPSIS method is removed, the method proposed in this paper is effective.

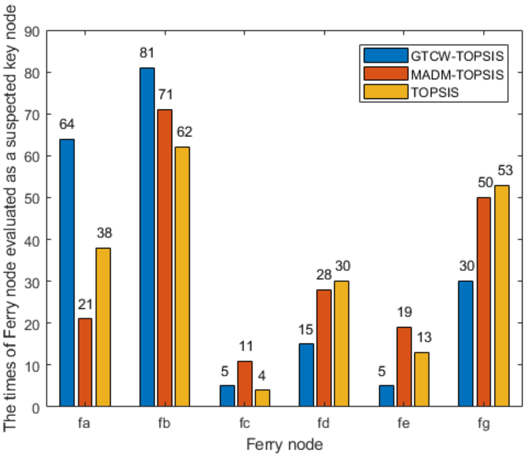


Fig. 7. The results of scenario 1

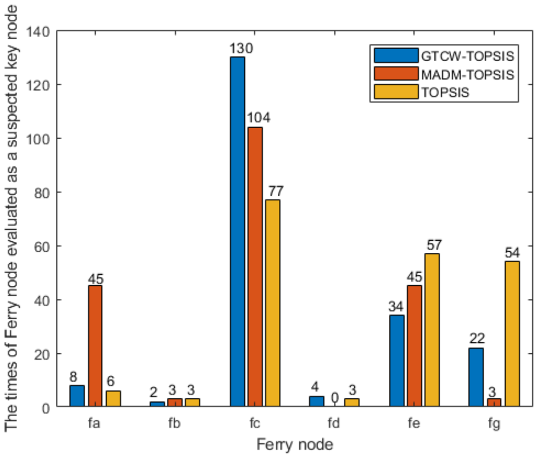


Fig. 8. The results of scenario 2

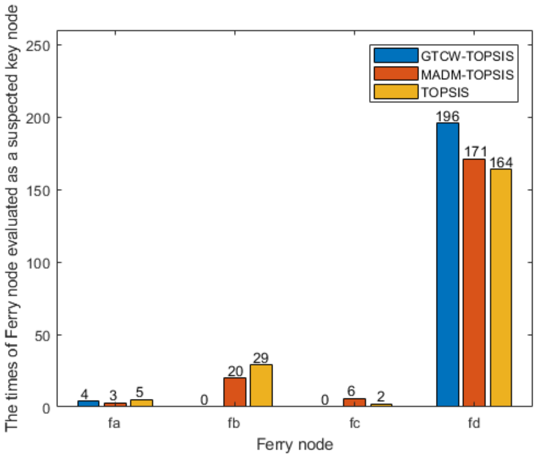


Fig. 9. The results of scenario 3

4.3. Verification

In the experiments, we remove a ferry node every 200 minutes and compute the WNDSR of the remaining network until every node is removed once. Aiming to reduce the error caused by randomness, the simulation experiments are repeated 10 times. The WNDSR is shown in fig. 10-12, wsall denotes the WNDSR before removing ferry nodes, wsda denotes the WNDSR after the fa is removed, wsdb denotes the WNDSR after the fb is removed, the significance of wsdc, wsdd, wsde and wsdg is same as wsda and wsdb.

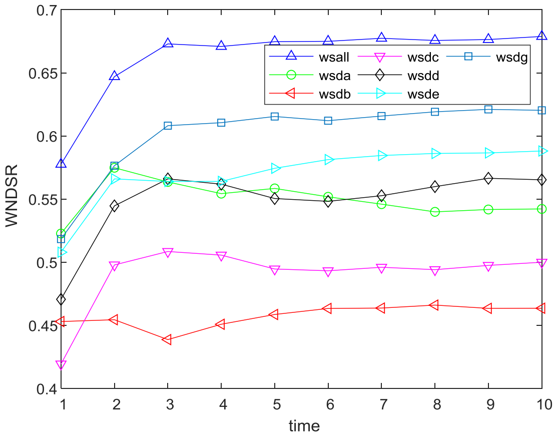


Fig. 10. WNDSR of scenario 1

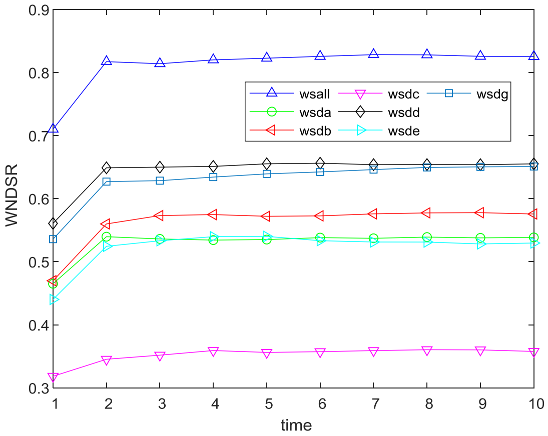


Fig. 11. WNDSR of scenario 2

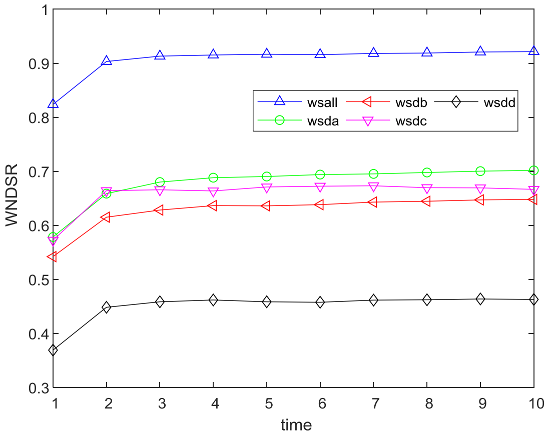


Fig 12. WNDSR of scenario 3

In fig. 10, the removal of fb notably reduces the WNDSR, therefore fb is the key node. As in fig. 11, the removal of fc leads to a crucially decrease of the WNDSR, therefore fc is the key node. From fig. 12, it shows that the removal of fd significantly reduces WNDSR, therefore fd is the key node. In summary, the results show that the key nodes identified by the method proposed all align with the real key node in the scenario 1~3. It can verify that identifying the key node in MOSNs by the method proposed is feasible. It can be seen from Fig. 4-6, the time that the key node is identified by the TOPSIS method and MADM_TOPSIS method is significantly less than the proposed method, which indicates that the proposed is the best in the three methods.

4.4. Accuracy

According to fig. 7~9, the accuracy of the GTCW_TOPSIS method, TOPSIS method and MADM_TOPSIS method are shown in fig. 13. There are some ferry nodes that move with the random way point model in scenario 1, so the accuracy of the three methods is similar. In scenario 2 and scenario 3, the accuracy of the GTCW_TOPSIS method is 65% and 98%, which is obviously higher than the TOPSIS method and MADM_TOPSIS method.

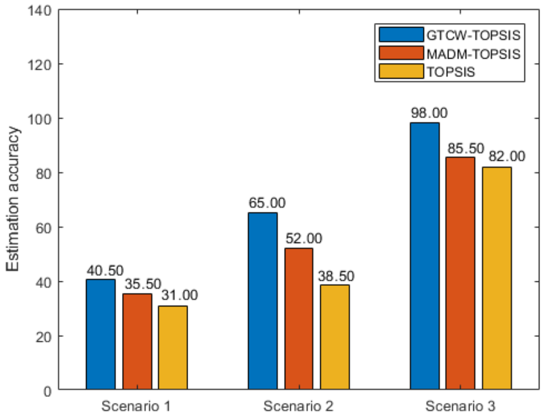


Fig. 13. Estimation accuracy

5. Conclusions

Aiming to identify the key node in MOSNs, first of all, we focus on the characteristics that the topology changes frequently, and use the temporal reachable graph to model MOSNs. Based on this model, the average degree is defined to reflect the activity of the ferry node in the network. The betweenness centrality is defined to reflect the ability of the ferry node to control the path between the region node and the sink node. The messages forwarding rate is defined to reflect the contribution on delivering messages generated by the region node to the sink node. Secondly, we use the average degree centrality, the betweenness centrality and the messages forwarding rate as the attributes of identifying the key node, the subjective weight and objective weight of each attribute are obtained by the AHP method and the entropy method respectively, and the GTCW method is used to calculate the combination weight of each attribute. Thirdly, we use the combination weight to construct the decision matrix and identify the key node in MOSNs by the TOPSIS method. Finally, the experiments in three scenarios are used to verify the effectiveness and evaluate the performance of the GTCW_TOPSIS method. In the future, we will further analyze the characteristics of MOSNs to propose more node importance attributes and apply the method based on the GTCW_TOPSIS method to other dynamic networks.

Acknowledgment. This paper is supported by the National Natural Science Foundation of China (62062050, 61762065, 61962037), the Natural Science Foundation of Jiang Province (20202BABL202039), and the Innovation Foundation for Postgraduate Student of Jiangxi Province (YC2019028).

References

1. Xiong, Y. P., Sun, L. M., Niu, J. W.: Opportunistic networks. Journal of Software, Vol. 20, No. 1, 124-137. (2009)
2. Lü, L. Y., Chen, D., Ren, X. L.: Vital nodes identification in complex networks. Physics Reports, Vol. 650, No. 13, 1-63. (2016)
3. Yang, F., Li, X. W., Xu, Y. Q.: Ranking the spreading influence of nodes in complex networks: An extended weighted degree centrality based on a remaining minimum degree decomposition. Physics Letters A,Vol. 382, No. 34, 2361-2371. (2018)
4. Barthélemy, M.: Betweenness centrality in large complex networks. European Physical Journal B, Vol. 38, No. 2, 163-168. (2004)
5. Solá, L., Romance, M., Criado, R.: Eigenvector centrality of nodes in multiplex networks. Chaos, Vol. 23, No. 3, 033131. (2013)
6. Nathan, E., Sanders, G., Fairbanks, J.: Graph Ranking Guarantees for Numerical Approximations to Katz Centrality, Procedia Computer Science, Vol. 108, 68-78. (2017)
7. Zareie, A., Sheikhahmadi, A., Khamforoosh, K.: Influence maximization in social networks based on TOPSIS[J]. Expert Systems with Applications, Vol. 108, No. 15, 96-107. (2018)
8. Fei, L. G., Zhang, Q., Deng, Y.: Identifying influential nodes in complex networks based on the inverse-square law. Physica A: Statistical Mechanics and its Applications, Vol. 512, No. 15, 1044-1059. (2018)
9. Korn, A., Schubert, A., Telcs, A.: Lobby index in networks[J]. Physica A: Statistical Mechanics and its Applications, Vol. 388, No. 11, 2221-2226. (2009)
10. Kitsak, M., Gallos, L. K., Havlin, S.: Identification of influential spreaders in complex networks. Nature Physics, Vol. 6, No. 11, 888-893. (2010)
11. Lü, L. Y., Zhou, T., Zhang, Q. M.: The H-index of a network node and its relation to degree and coreness. Nature Communications, Vol. 7, 10168. (2016)
12. Shao, H., Wang, L. W., Zheng, J.: Important node identification method for dynamic networks based on H operation. Journal of Computer Applications, Vol. 39, No. 09, 2669-2674. (2019)
13. Holme, P.: Modern temporal network theory: a colloquium. European Physical Journal B, Vol. 88, No. 9, 1-30. (2015)
14. Zhang, D. S., Gogi, S. A., Broyles, D. S.: Modelling attacks and challenges to wireless networks. International Congress on Ultra Modern Telecommunications and Control Systems and WorkshopsSt. Petersburg, Russia, 806-812. (2012)
15. Nicosia, V., Tang, J., Mascolo, C.: Graph Metrics for Temporal Networks. Understanding Complex Systems, Vol. 2013, No. 4, 15-40. (2013)
16. Casteigts, A., Flocchini, P., Quattrociocchi, W.: Time-Varying Graphs and Dynamic Networks. international journal of parallel emergent & distributed systems, Vol. 27, No. 5, 387-408. (2010).
17. Wu, H., Cheng, J., Huang, S.: Path problems in temporal graphs. Proceedings of the Vldb Endowment, Vol. 7, No. 9, 721-732. (2014)
18. Nicosia, V., Tang, J., Musolesi, M.: Components in time-varying graphs. Chaos, Vol. 22, No. 2, 175-R. (2011)
19. Zhang, D. S., Sterbenz, J. P. G.: Modelling Critical Node Attacks in MANETs. The Proceedings of the 3rd International Workshop on Self-Organizing Systems. Palma de Mallorca, Spain, 127-138. (2013)

20. Qu, C., Zhan, X. X., Wang, G. H.: Temporal information gathering process for node ranking in time-varying networks. Chaos, Vol. 29, No. 3, e033116. (2019)
21. Arrigo, F., Higham, D. J.: Sparse matrix computations for dynamic network centrality. Applied Network Science, Vol. 2, No. 1, 17-35. (2017)
22. Abbas, K., Shang, M., Abbasi, A.: Popularity and novelty dynamics in evolving networks. Scientific Reports, Vol. 8, No. 1, e6332. (2018)
23. Fang, X., Hu, P. J. H.: Top persuader prediction for social networks. Management Information Systems Quarterly, Vol. 42, No. 1, 63-82. (2018)
24. Taro, T., Yosuke, Y., Yuichi, Y.: Coverage centralities for temporal networks. European Physical Journal B, Vol. 89, No. 2, 1-11. (2016)
25. Whitbeck, J., Amorim, M. D. D., Conan, V.: Temporal reachability graphs. The 18th Annual International Conference on Mobile Computing and Networking. Istanbul, Turkey, 377-388. (2012)
26. Zhu, C., Ju, J. B., Wang, P.: Anti-submarine patrol decisions based on grey incidence decision and combination weighting method. Command Control & Simulation, Vol. 39, No. 2, 10-14. (2017)
28. Liu, L. L., Zhang, J., Su, J.: Multiple attribute decision making-based prediction approach of critical node for opportunistic sensor networks[J]. Journal of Computer Research and Development, Vol. 54, No. 9, 2021-2031. (2017)
27. Chen, Q. F., Liu, L. L., Yang, Z. Y.: Prediction approach of critical node based on multiple attribute decision making for opportunistic sensor networks. Journal of sensors, Vol. 2016, No. 4, 1-8. (2016)

Linlan Liu born in 1968, and received the Bachelor degree in computer science from the National University of Defense Technology, Changsha, China, in 1988. Currently. She is a full Professor, School of Information Engineering, Nanchang Hangkong University, Nanchang, China. She was a Visiting Scholar at Wilfrid Laurier University, Waterloo, Ontario, Canada. She has authored/coauthored more than 70 papers. Her research interests include wireless sensor networks and embedded system (765693987@qq.com).

Wei Wang born in 1995, and is a MSc candidate at the School of Information Engineering, Nanchang Hangkong University, Nanchang, China. His current researches interests include opportunity network (919269210@qq.com).

Guirong Jiang born in 1996, and is a MSc candidate at the School of Software, Nanchang Hangkong University, Nanchang, China. His current researches interests include opportunity network (1132153564@qq.com).

Jiang Zhang born in 1992, and is a MSc candidate at the School of Information Engineering, Nanchang Hangkong University, Nanchang, China. His current researches interests include opportunity network (zhangjiangky@163.com).

Received: December 20, 2019; Accepted: May 02, 2020

Computer Science and Information Systems 18(3):1057–1076 https://doi.org/10.2298/CSIS200502018K

Dynamic Fractional Chaotic Biometric Isomorphic Elliptic Curve for Partial Image Encryption

Ahmed Kamal[1], Esam A. A. Hagras[2], H. A. El-Kamchochi[3]

[1] Engineering Dept., Air Defense College, Alexandria University,
Alexandria, Egypt
ahmed_kamal8030@yahoo.com
[2] Communications and Computers Department, Faculty of Engineering,
Delta University for Science and Technology,
Gamasa, Dakahlia, Egypt
esam.hagras@deltauniv.edu.eg
[3] Electrical Department, Faculty of Engineering, Alexandria University,
Alexandria, Egypt
helkamchouchi@hotmail.com

Abstract. In this paper, a Modular Fractional Chaotic Sine Map (MFC-SM) has been introduced to achieve high Lyapunov exponent values and completely chaotic behavior of the bifurcation diagram for high level security. The proposed MFC-SM is compared with the conventional non MFC-SM and it has an excellent chaotic analysis. In addition, the randomness test results indicate that the proposed MFC-SM shows better performance and satisfy all randomness tests. Due to the excellent chaotic properties and good randomization results for the proposed MFC-SM, it is used to be cooperated with the biometric digital identity to achieve dynamic chaotic biometric digital identity. Also, for real time image encryption, both Discrete Wavelet Transform (DWT)partial image encryption and Isomorphic Elliptic Curve (IEC)key exchange are used. In addition, the biometric digital identity is extracted from the user fingerprint image as fingerprint minutia data incorporated with the proposed MFC-SM and hence, a new Dynamic Fractional Chaotic Biometric Digital IdentityIEC (DFC-BID-IEC) has been introduced. Dynamic Fractional Chaotic Key Generator (DFC-KG) is used to control the key schedule for all encryption and decryption processing. The encryption process consists of the confusion and diffusion steps. In the confusion step, the 2D Arnold Cat Map (ACM) is used with secret parameters taken from DFC-KG. Also, the diffusion step is based on the dynamic chaotic self-invertible secret key matrix which can be generated from the proposed MFC-SM. The IEC key exchange secret parameters are generated based on Elliptic Curve Diffie–Hellman(ECDH) key exchange and the isomorphism parametre. Statistical analysis, differential analysis and key sensitivity tests are performed to estimate the security strengths of the proposed DFC-BID-IEC system. The experimental results show that the proposed algorithm is robust against common signal processing attacks and provides a high security level and high speed for image encryption application.

Keywords: Image encryption, Biometric identity, Elliptic curve cryptography, Chaotic Maps.

1. Introduction

In today’s digital world, usage of images arenotably increased across the network. It became an indispensable part of our life. Also,it has become a great source of information and contains personal data. Thus, strong security and protection must be ensured using cryptography. So, a large number of researchers have introduced numerous schemes for image encryption [1-3]. These encryption techniques are employed in two ways, namely Symmetric Encryption and Asymmetric Encryption [4]. In 1985, Neal Koblitz and Victor S. Miller [5-6] introduced a new public key cryptography EC which provides a high level of security and achieve computational efficiency in performance with smaller key size compared to other cryptographic technique [7]. Most of thetraditional ciphers are not efficient in image encryption because their slow speed, the large data volume and strong correlation among image pixels. So, chaotic cryptography has been attracting more attention of large number of researchers because of their high ergodicity and sensitivity to control parameter, initial conditions and non-linearity. Definitely, many chaos-based image encryption algorithms have been proposed for image encryption such as Chebyshev map, Logistic map [12], the ACM[8], Tent map [1], sine map [9] etc. However, these maps have some weaknesses, namely, non-uniform distribution, small key space and periodicity [10]. Some proposed recently hyperd maps can overcome these imperfections and enhance security [3]. Several paradigms have been used to extract cryptography key from biometric traits. The key based on the biometric features was applied earliest in online trading for the IBM transaction security system in 1989, by using signature pen and handwriting signal processor [11]. Many schemes for image encryption based on ECC and chaotic map are proposed. In [12],an image encryption scheme based on chaotic system and EC has been proposed usesECDHfor key exchange between sender and receiver in addition, logistic map is used to generate a chaotic sequence using initial condition from elliptic curve. In [13] an algorithm for image encryption uses ECC and modified hill cipher to secure the image data. In [8] Essam et. al. introduced a selective encryption algorithm use DWT and multi-map orbit hopping chaotic encryption, the multi-chaotic logistic maps generate a hopping pattern of random numbers used to encrypt the low-low sub-band decomposition only. In [14] a chaotic tent map used to encrypt a medical image extracted by DWT-DCT. In [15] Abd El-Latif et. al. proposed a hybrid image encryption scheme based on a cyclic EC and chaotic system. An image encryption algorithm based on a secure variant of Hill cipher and three one-dimensional (1D) chaotic maps suggested in [9], this algorithm aims to encrypt pixel-by- pixel all types of images with black background or with high correlation of adjacent pixels. In[16] present based on an ordered isomorphic EC for generating a large number of distinct, mutually uncorrelated, and cryptographically injective S-boxes. In [17] an image encryption algorithm based on the H-fractal and dynamic self-invertible matrix have been proposed.

This paper proposes an improved image encryption scheme uses IECfor initial key exchange between two parties. The generated IEC secret keys used to generate the initial conditions. Using the fractional modular chaotic map and biometric key based on IEC to build key schedule that is used as a parameter generator for the system. The image is scrambled using ACM and is encrypted using self invertible matrixto attain confusion and diffusion. The initial condition for the proposed MFC-SMare taken from the key schedule. the proposed MFC-SM is used to construct the self invertible matrix.

The rest of this paper is organized as following: Section 2 provides guidelines for Manuscript Preparation. Section 3 presents the proposedDFC-BID-IEC scheme. In Section 4 the simulation results and security analysis are introduced. Finally, conclusion and future work are given in Section 5.

2. Proposed Scheme Preparation

2.1. Novel Modular Fractional Chaotic Sin Map

Discrete fractional calculuswas introduced to efficiently incorporate and capture the memory effects in nonlinear discrete time systems [18]. Dynamical behaviors and applications of fractional difference models, on an arbitrary time scale, were investigated in the last decade where delta difference equation was utilized.Assume that a sequence $\rho(n)$ is given and the isolated time scale $\aleph_a$is represented in terms of real valued constant τ as $\{\tau, \tau+1, \tau+2, \ldots,\}$ such that $\rho\colon \aleph_\tau \to \mathbb{R}$. The difference operator is denoted by Δ, where $\Delta\rho(n) = \rho(n+1) - \rho(n)$ then some of the basic definitions related to discrete fractional calculus are summarized as follows:
For $\alpha > 0$, the fractional sum of order α is given by [18]

$$\Delta_{\boldsymbol{\tau}}^{-\boldsymbol{\alpha}}\boldsymbol{\rho(t)} = \frac{\mathbf{1}}{\boldsymbol{\Gamma(\alpha)}}\sum_{\boldsymbol{m=\tau}}^{\boldsymbol{t-\alpha}}\frac{\boldsymbol{\Gamma(t-m)}}{\boldsymbol{\Gamma(t-m-\alpha+1)}}\boldsymbol{\rho(m)}, \boldsymbol{t} \in \aleph_{\boldsymbol{\tau+\alpha}}. \tag{1}$$

the Caputo-like delta difference of order α is defined by [18]:

$${}^{C}\Delta_{\tau}^{\alpha}\rho(t) = \Delta_{\tau}^{-(n-\alpha)}\Delta^{n}\rho(t)$$

$$= \frac{1}{\Gamma(n-\alpha)}\sum_{m=\tau}^{t-(n-\alpha)}\frac{\Gamma(t-m)}{\Gamma(t-m-n+\alpha+1)}\Delta^{n}\rho \quad t \in \aleph_{\tau+n-\alpha},\ n = \lfloor\alpha\rfloor + 1 \tag{2}$$

the delta fractional difference equation of order α is represented by [18] and the equivalent discrete fractional integral is given by

$${}^{C}\Delta_{\tau}^{\alpha}\rho(t) = f(t+\alpha-1, \rho(t+\alpha-1)),$$

$$\rho(t) = \rho_0(t) + \frac{1}{\Gamma(\alpha)}\sum_{m=\tau+n-\alpha}^{t-\alpha}\frac{\Gamma(t-m)}{\Gamma(t-m-\alpha+1)}$$

$$\times f\big(m+\alpha-1, \rho(m+\alpha-1)\big),\ \ t \in \aleph_{\tau+n} \tag{3}$$

note that the initial iteration in this case is:

$$\rho_0(t) = \sum_{k=0}^{n-1}\frac{\Gamma(t-\tau+1)}{k!\,\Gamma(t-\tau-k+1)}\Delta^{k}\rho(\tau) \tag{4}$$

The *non-modular* fractional sine map with Caputo fractional order is introduced in [18], it leads to a highLyapunov exponent value, so we have to introduce some definitions about the fractional calculus. The *non-modular* fractional sine map is given by:

$$x(n) = x(0) + \frac{\mathrm{r}}{\Gamma(v)}\sum_{j=1}^{n}\frac{\Gamma(n-j+v)}{\Gamma(n-j+1)}sin(x(j-1) \tag{5}$$

the proposed new *modular* fractional sine map is given by:

$$x(n) = (x(0) + \frac{r}{\Gamma(v)}\sum_{j=1}^{n}\frac{\Gamma(n-j+v)}{\Gamma(n-j+1)}sin(x(j-1)))\ mod\ 1 \qquad (6)$$

where $'\mathrm{r}'$ is the control parameter of non-modular and MFC-SM and $v$is the difference order. Using more than one parameter of the sine map gives high Lyapunov exponent value [19], high chaotic range and a large key space. Fig. 1 shows the Lyapunov exponent and the bifurcation diagram of the Non modular fractional chaotic sine map. Also, Fig.2 shows both the Lyapunov exponent and the bifurcation diagram of the proposed modular fractional chaotic sine map. As shown in these figures, the proposed MFC-SMhas highly Lyapunov exponent values and completely chaotic behavior of the bifurcation diagram compared with the conventional Non modular fractional chaotic sine map.

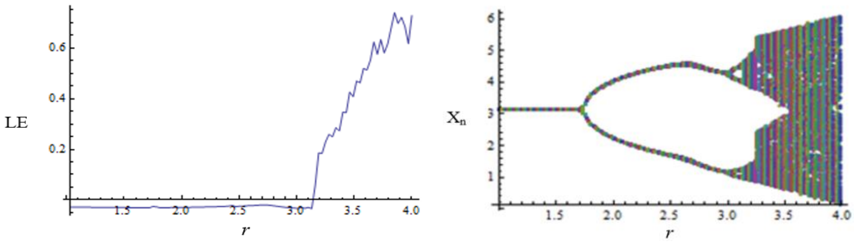


Fig. 1. Lyapunov exponents (LE) and Bifurcation diagram of the non-modular fractional order sine map.

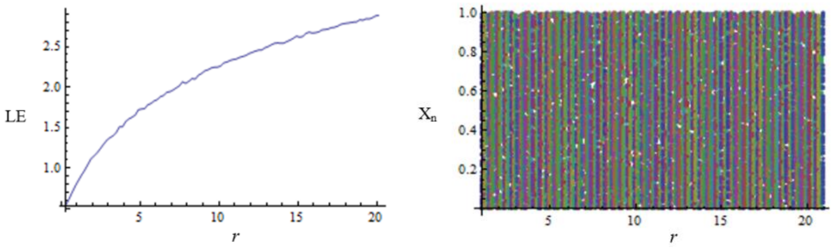


Fig. 2. Lyapunov exponents (LE) and Bifurcation diagram of the non-modular fractional order sine map.

The randomness of the proposed MFC-SM is tested by the NIST tests. These tests are defining if the generated sequence is random or not. The basic dependence within these tests is on the probability value (p-value). The p-value is compared by the significance level which is the threshold between rejection and non-rejection region. In NIST the significant level equal 0.01. For p-value less than 0.01 this means that the sequence is not random and reject and for p-value greater than 0.01 this means that the sequence is random and accepted. 10^6 bit binary sequence obtained from the proposed modular fractional sine map is tested by SP800-22 [3] and the results are given in Table 1.

Table 1. NIST Randomness Tests of the Proposed MFC-SMBINARY OUTPUT.

TEST	P-VALUE	RESULT
MONOBIT FREQUENCY	0.553273	PASSED
BLOCK FREQUENCY	0.538714	PASSED
RUNS	0.596352	PASSED
LONGEST-RUN-OF-ONES IN A BLOCK	0.692018	PASSED
BINARY MATRIX RANK	0.352617	PASSED
DISCRETE FOURIER TRANSFORM (SPECTRAL)	0.438291	PASSED
NON-OVERLAPPING TEMPLATE MATCHING	0.527163	PASSED
OVERLAPPING TEMPLATE MATCHING	0.592763	PASSED
MAURER'S UNIVERSAL STATISTICAL	0.421873	PASSED
LINEAR COMPLEXITY	0.537524	PASSED
SERIAL TEST	0.437163	PASSED
APPROXIMATE ENTROPY	0.418315	PASSED
CUMULATIVE SUMS	0.468232	PASSED
RANDOM EXCURSION	0.391823	PASSED
RANDOM EXCURSION VARIANT	0.538138	PASSED
CUMULATIVE SUMS TEST REVERSE	0.387263	PASSED
LEMPEL-ZIV COMPRESSION	0.498163	PASSED

2.2. Isomorphic Elliptic Curve

An EC over prime field $F_p$is defined with the cubic equation:

$$\boldsymbol{y^2 = x^3 + ax + b \bmod p} \tag{7}$$

where p is a large prime number and a, b satisfies the condition$4a^2 + 27b \neq 0$, each value of $a\ , b\ \epsilon\ p$ gives a different EC $EC_{p,a,b}$ where p, a and b are called the EC parameters. For two ECs $E_{p,a,b}$ and $E_{p,a',b'}$ over the field F_p are said to be isomorphic if and only if there exists an integer $i \in p \setminus \{0\}$. Such that, the EC parameter (a, b) will be $(a^{'}, b^{'})$ for the isomorphic elliptic curve, the value of the IEC parameters $(a^{'}, b^{'})$ will be computed using i as following:

$$\boldsymbol{a^{'} = ai^4 modp, b^{'} = bi^6 modp} \tag{8}$$

where i is called the isomorphism parameter between $E_{p,a,b}$ and $E_{p,a',b'}$. Thus, every point $(x, y) \in E_{p,a,b}$ will be$(x^{'}, y^{'}) \in E_{p,a',b'}$ and $(x^{'}, y^{'})$ will be computed as

$$\boldsymbol{x^{'} = xi^2 modp, y^{'} = yi^3 modp} \tag{9}$$

It is easy to observe that isomorphism is an equivalence relation on the family of all ECs over the field F_p. It is well-known that for prime p there exists a unique finite field F_p, up to the field isomorphism, with exactly $p$elements. There are $p^2 - p$ ECs over the field F_p. The number of ECs isomorphic to a given EC over $F_p$can be computed as the following:

For a prime $p > 3$ and $a, b \in [0, p - 1]$ are two integers. The number of ECs isomorphic to the EC $E_{p,a,b}$is

- $(p - 1)/6$ if $a = 0$ and F_p has a non-zero element of group order 6.

- $(p - 1)/4$ if $b = 0$ and F_p has a non-zero element of group order 4.
- $(p - 1)/2$ Otherwise.

The number of elements$\#E_{F_{p,a,b}}$in EC is equal to the number of points lying on EC over F_p. Hasse's Theorem gives the bounds of total number of points on EC [20]:

$$\boldsymbol{p + 1 - 2\sqrt{p} \le \#E_{F_{p,a,b}} \le p + 1 + 2\sqrt{p}} \tag{10}$$

The order of the EC is the total number of points lies on the EC along with the point at infinity $O(x = \infty;\ y = \infty)$ denoted by $\#E$. The smallest positive integer n for which nP is equal to point at infinity O $(nP = O)$ is called order of point P such that $n \le \#E$. Then, $P, 2P, \ldots, (n-1)P$ are distinct points on elliptic curve. For certain choice of a and b it is possible to choose a base point P of highest order $n = \#E$ [15].

For example, let $p = 37, a = -1, b = 6$ are the main EC parameters, $i = 3$ and $i = 5$ are two different isomorphism parameters. Such that, the IEC parameter for $i = 3$ computed as $a' = -1 * 3^4\ mod\ 37$ and $b' = 6 * 3^6\ mod\ 37$. thus,$(a' = 30, b' = 8)$ and for $i = 5$ computed as $a'' = -1 * 5^4\ mod\ 37$ and $b'' = 6 * 5^6\ mod\ 37$. Thus, $(a'' = 4,\ b'' = 29)$. Fig. 3 shows the difference between the points lays on the main EC and its tow isomorphic elliptic curves.

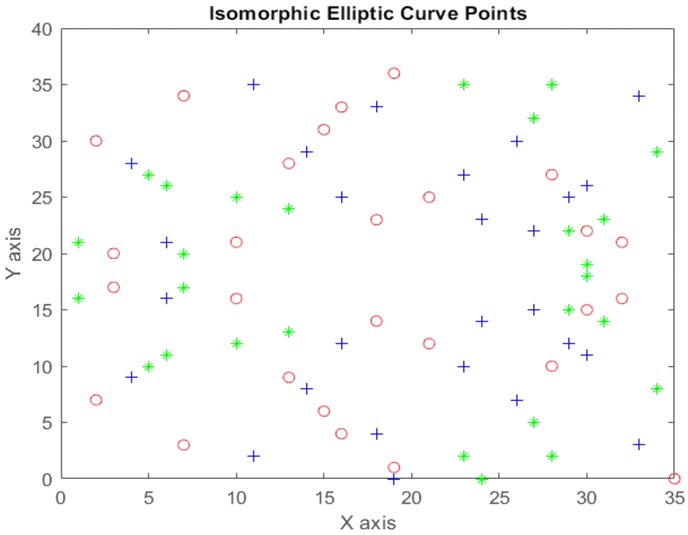


Fig. 3. The different points of the main EC represented by blue(+) and its isomorphic elliptic curves with isomorphic parameters $\boldsymbol{i = 3}$represented by red (O) and $\boldsymbol{i = 5}$ represented by green (*).

2.3. Elliptic Curve Diffie–Hellman Key Exchange

Let$G$ is a base point of an EC,P_A andP_B can be computed as

$$\boldsymbol{P_A = n_A.G,\ P_B = n_B.G} \tag{11}$$

where P_A ,P_B is the public keys of sender and receiver respectively and $n_{A,}n_B$is the privet keys. The shared key is computed as $n_A P_B$ and $n_B P_A$ by the sender and receiver respectively.

$$\boldsymbol{Sk = n_A P_B = n_A n_B G = n_B n_A G = n_B P_A} \tag{12}$$

2.4. Self Invertable Matrix

FirstlyHill used self-invertible matrices in his proposed encryption algorithm [17]. Hill cipher algorithm uses a matrix to convert the plain-text into cipher-text, and the key is the matrix itself. The Plaintext N is encrypted as:

$$\boldsymbol{C = K_{IM} \times N(modm)} \tag{13}$$

where C is the Cipher text block, K_{IM} is the Self-Invertible Matrix and m is the plain-text value range (For image encryption, m = 256). K_{IM} must satisfy the criteria [21] to be an invertible matrix and the gcd($det\ [K_{IM}]\ modm$, m) =1.

$$\boldsymbol{N = K_{IM} \times C(modm)} \tag{14}$$

$$\boldsymbol{K_{IM} \times K_{IM}{}^{-1} = I} \tag{15}$$

where I is the identity matrix. The receiver cannot decrypt the cipher message if K_{IM} is not invertible matrix. To create $K_{IM}$to be used for encryption and decryption.

$$\boldsymbol{K_{IM} = \begin{bmatrix} K_c & I - K_c \\ I + K_c & -K_c \end{bmatrix} mod\, m} \tag{16}$$

where K$_c$ is the Chaotic key matrix that iswanted to be self-invertible to be used for encryption and decryption.

2.5. Generation of Self-Invertible 4 × 4 Matrix

Let $K_{IM} = \begin{bmatrix} k_{11} k_{12} k_{13} k_{14} \\ k_{21} k_{22} k_{23} k_{24} \\ k_{31} k_{32} k_{33} k_{34} \\ k_{41} k_{42} k_{43} k_{44} \end{bmatrix}$be self-invertible matrixpartitioned as $\left[\begin{array}{c:c} K_{11} & K_{12} \\ \hdashline K_{21} & K_{22} \end{array}\right]$

where$K_{11} = \begin{bmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{bmatrix}$,$K_{12} = \begin{bmatrix} k_{13} & k_{14} \\ k_{23} & k_{24} \end{bmatrix}$,$K_{21} = \begin{bmatrix} k_{31} & k_{32} \\ k_{41} & k_{42} \end{bmatrix}$, $K_{22} = \begin{bmatrix} k_{33} & k_{34} \\ k_{43} & k_{44} \end{bmatrix}$

For example, a realexample for chaotic matrix $K_{4\times 4}$taken from the output of the proposed MFC-SM Used to construct the self-invertible matrix $K_{8\times 8}$ that used to encrypt and decrypt a real part of the scrambled Lena image $N_{8\times 8}$ as explained.

Let $K_{11} = \begin{bmatrix} 72 & 1 & 53 & 27 \\ 7 & 61 & 244 & 166 \\ 179 & 228 & 124 & 145 \\ 104 & 9 & 209 & 46 \end{bmatrix}$, $K_{22} = \begin{bmatrix} 184 & 255 & 203 & 229 \\ 249 & 195 & 12 & 90 \\ 77 & 28 & 132 & 111 \\ 152 & 247 & 47 & 210 \end{bmatrix}$

Take $K_{12} = I - K_{11}$ with $n = 1$.Then,

$$K_{12} = \begin{bmatrix} 185 & 255 & 203 & 229 \\ 249 & 196 & 12 & 166 \\ 77 & 28 & 133 & 111 \\ 152 & 247 & 47 & 211 \end{bmatrix}, K_{21} = \begin{bmatrix} 73 & 1 & 53 & 27 \\ 7 & 62 & 244 & 166 \\ 179 & 228 & 125 & 145 \\ 104 & 9 & 209 & 47 \end{bmatrix}$$

So, $$K_{IM} = \begin{bmatrix} 72 & 1 & 53 & 27 & 185 & 255 & 203 & 229 \\ 7 & 61 & 244 & 166 & 249 & 196 & 12 & 90 \\ 179 & 228 & 124 & 145 & 77 & 28 & 133 & 111 \\ 104 & 9 & 209 & 46 & 152 & 247 & 47 & 211 \\ 73 & 1 & 53 & 27 & 184 & 255 & 203 & 229 \\ 7 & 62 & 244 & 166 & 249 & 195 & 12 & 90 \\ 179 & 228 & 125 & 145 & 77 & 28 & 132 & 111 \\ 104 & 9 & 209 & 47 & 152 & 247 & 47 & 210 \end{bmatrix}$$

For, $$N = \begin{bmatrix} 194 & 194 & 72 & 71 & 134 & 129 & 146 & 144 \\ 195 & 194 & 71 & 71 & 134 & 128 & 146 & 144 \\ 61 & 62 & 158 & 156 & 154 & 149 & 88 & 85 \\ 62 & 62 & 157 & 157 & 154 & 149 & 87 & 85 \\ 124 & 124 & 123 & 122 & 139 & 134 & 113 & 110 \\ 126 & 126 & 120 & 117 & 137 & 137 & 108 & 110 \\ 156 & 156 & 51 & 46 & 73 & 75 & 90 & 87 \\ 154 & 154 & 44 & 44 & 71 & 72 & 87 & 89 \end{bmatrix},$$

By applying Eq.13,

$$C = \begin{bmatrix} 18 & 70 & 4 & 165 & 166 & 134 & 117 & 74 \\ 165 & 92 & 168 & 56 & 181 & 183 & 121 & 246 \\ 226 & 122 & 187 & 214 & 93 & 61 & 77 & 41 \\ 96 & 40 & 100 & 242 & 47 & 47 & 115 & 1 \\ 88 & 140 & 209 & 114 & 161 & 129 & 150 & 108 \\ 234 & 160 & 119 & 10 & 178 & 174 & 159 & 24 \\ 131 & 28 & 38 & 68 & 174 & 135 & 75 & 39 \\ 4 & 204 & 213 & 99 & 130 & 124 & 115 & 253 \end{bmatrix}$$

By applying Eq. 14,

$$N' = \begin{bmatrix} 194 & 194 & 72 & 71 & 134 & 129 & 146 & 144 \\ 195 & 194 & 71 & 71 & 134 & 128 & 146 & 144 \\ 61 & 62 & 158 & 156 & 154 & 149 & 88 & 85 \\ 62 & 62 & 157 & 157 & 154 & 149 & 87 & 85 \\ 124 & 124 & 123 & 122 & 139 & 134 & 113 & 110 \\ 126 & 126 & 120 & 117 & 137 & 137 & 108 & 110 \\ 156 & 156 & 51 & 46 & 73 & 75 & 90 & 87 \\ 154 & 154 & 44 & 44 & 71 & 72 & 87 & 89 \end{bmatrix}$$

It is distinct that there no relation between the original matrix Nvalues and the ciphered matrix C values$(N \neq C)$. vice versa the encrypted matrix N'istipically the same as the original matrix $(N = N')$. This prove the robustness of the proposed scheme.

2.6. Finger Print Biometric Identity Extraction:

Fingerprint is a biometric attribute that can be captured to extract digital data using several approaches, such as block based approach to generate the feature vector [18]. This feature vector is used to generate code word which can be of any arbitrary large size and random enough to use. The process subject to some steps, feature extraction, calculation of attributes of straight lines, obscuring straight lines attributes, biometric binary string generation. Firstly, extract Minutiae points(v_k), Core point(C_p) and Delta point (D_p) from fingerprint image. Calculate straight line attributes between the points in the set v_k. Let F is the fingerprint image, divide F to a number of small blocks each size $m \times m$ pixels, where $F = s \times q$ of all blocks. Calculate straight line attributes using all the blocks by computing all straight lines from one v_k of a block as a reference block to other v_k of all adjacent blocks, calculate length and angle of each straight line, length $(l$i) using Euclidean distance and angle (a_i) with reference to the x-axis. Let F_B represents a set of lengths and angles of straight lines for all blocks $F_B = \{(l_1, a_1), (l_2, a_2), \ldots., (l_{zb}, a_{zb})\}$ size ofF_B isz^b. Extract C_p and D_p from image$E$ by finding the block E_{lm} which contains the C_P, compute all straight lines from the C_P to all other v_k of surrounding neighborhood blocks, Let F_C denotes a set of lengths and angles of straight lines, where the size of the F_C is z^c. Similarly, the set of lengths and angles of lines represented as F_D with reference to D_p where the size of the F_D is z^d, set F_B,F_C and F_D into a single set R.

$$\boldsymbol{R = \{F_B \| F_C \| F_D\},\ z = z^b + z^c + z^d} \qquad (\mathbf{17})$$

For obscuring straight lines attributes, the extracted features XOR together and converted into a binary form, merge all bits in the feature setsto generate 256 bits.

2.7. Dynamic Fractional Chaotic Key Generation (DFC-KG)

In this section, we propose a new fast and efficient system for key generation using biometric identity XOR chaotic map sequence based on a hidden IEC. It consists of two parts; the first part serving for generation of initial Secret key which generated from EC using the parameters (p, a, b, G) as in Eqn. (7). Both the sender and receiver use their own private key to share the public as in Eqn. (11) using ECDH key exchange, thus, they generate the secret key Sk as in Eqn. (13). An isomorphism parameter i will be generated using the shared base point $G(x, y)$ as in Eqn. (18). It will be used as an isomorphism parameter as in Eqn. (8) to produce the IEC.Using $i$as in Eqn. (9) the public points $P_A\colon (x, y)$ and $P_B\colon (x, y)$ that lies on the EC will be $P_A^{'}\colon (x^{'}, y^{'})$ and $P_B^{'}\colon (x^{'}, y^{'})$ lies on the IEC. Similarly, $Sk\colon (x, y)$will be used to get the isomorphic secret key $Sk^{'}\colon (x^{'}, y^{'})$.

$$\boldsymbol{i = G(x \oplus y) \quad mod\, 8 + 3} \qquad (17)$$

Using point addition for the isomorphic public keys $P_A^{'}$and $P_B^{'}$ with $Sk^{'}$to get two different isomorphic secret keys $ISk_1^{'}$and $ISk_2^{'}$ as following

$$\boldsymbol{ISk_1^{'} = P_A^{'} + Sk^{'}} \qquad (18)$$

$$\boldsymbol{ISk_2'} = \boldsymbol{P_B'} + \boldsymbol{Sk'} \tag{19}$$

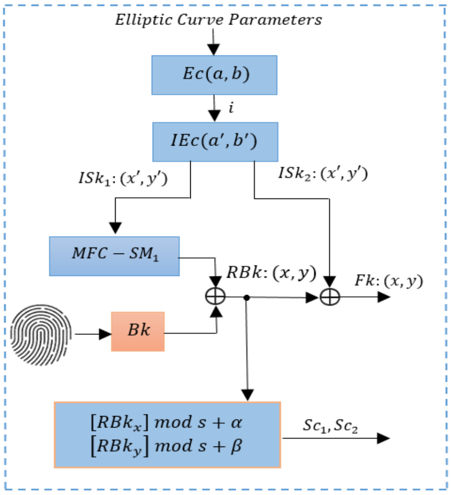


Fig.4. Proposed cryptosystem key generation.

The first $ISk_1'(x,y)$ will be used as initial condition for theMFC-SM$_1$as in Eqn. (21,22) to generate a Chaotic number $R:(x,y)$. This random number XOR with the saved 256 bit Biometric key $R:(x,y)$as in Eqn. (17) to provide randomness for the biometric key. So, the output sequence is a Random Biometric key $RBk:(x,y)$which XOR with $ISk_2':(x,y)$ to generate $Fk(x,y)$ as shown in Eqn. (24) to be used as initial condition for theMFC-SM$_2$as the same in Eqn. (21,22).Eqn. (23) shows the value of $\boldsymbol{v}$.

$$\boldsymbol{x_0} \;=\; \boldsymbol{ISk_1'}\,(\boldsymbol{x})\;/\;\boldsymbol{p} \tag{20}$$

$$\boldsymbol{r} \;=\; \boldsymbol{ISk_1'}(\boldsymbol{y})/(\boldsymbol{10}\;\times\;\boldsymbol{p}) \tag{21}$$

$$\boldsymbol{v} = \boldsymbol{x} + \boldsymbol{r} \qquad \boldsymbol{mod}\;\boldsymbol{1} \tag{23}$$

$$\boldsymbol{RBk}: (\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{Bk}: (\boldsymbol{x},\boldsymbol{y}) \oplus \boldsymbol{R}: (\boldsymbol{x},\boldsymbol{y}) \tag{22}$$

$$\boldsymbol{Fk}: (\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{RBk}: (\boldsymbol{x},\boldsymbol{y}) \oplus \boldsymbol{ISk_2'}(\boldsymbol{x},\boldsymbol{y}) \tag{23}$$

The second part of key generation, the parameters used for scrambling Sc_1, Sc_2 will be generated using the $RBk:(x,y)$ according to the Eqns. (26,27).

$$\boldsymbol{Sc_1} = \boldsymbol{RBk_x modS} + \;\boldsymbol{\alpha} \tag{24}$$

$$\boldsymbol{Sc_2} = \boldsymbol{RBk_y}\;\boldsymbol{mod}\;\boldsymbol{S} + \;\;\boldsymbol{\beta} \tag{25}$$

where S, α and β are integers used to eliminate the scrambling parameters. The proposed MFC-SM$_1$generates a chaotic sequence to get $K_c$which used to constructK_{IM}as in Eqn. (16). The $K_{IM}$is used for encryption and decryption. Table (2) shows the key schedule used in the whole system.

Table 2. Key Schedule

Symbol	Description	Symbol	Description
(a, b, p)	EC parameters	Sk'	Isomorphic secret key
$(a^{'}, b^{'}, p)$	IEC parameters	K_{IM}	InvertableKey Matrix
Sk	Initial secret key	n_A	Sender privet key
$ISk_1^{'}$	First isomorphic secret key	n_B	Reciver privet key
$ISk_2^{'}$	Second isomorphic secret key	P_A	Sender public key
Sc_1	Scrambling parameter (1)	P_B	Reciver public key
Sc_2	Scrambling parameter (2)	$P_A^{'}$	Sender isomorphic public key
i	Isomorphism parameter	$P_B^{'}$	Reciver isomorphic public key

2.8. Proposed DFC-BID-IEC Scheme

The proposed encryption scheme uses ECC to share the EC parameters between two parties using ECDH to provide authenticity and confidentiality. The hidden IEC used to provide secrecy to the ECparameters and the all keys. such that, inlarge the key space. The ACMused for image confusion according to the generated scrambling parameters. The proposed MFC-SM$_2$and K_{IM} to offer randomness and chaocity for image encryption. The system provides both system and user authentication, the proposed scheme goes as follows:

2.9. Partial Image Encryption

- DWT is applied to the image $N_{M\times M}$to generate vertical LH (CV), horizontal HL(CH), diagonal HH (CD) and approximation LL (CA) matrices [14].
- The approximation LL (CA) matrix only is scrambled using ACMas [8] with the scrambling parametersSc_1, Sc_2in Eqn. (26, 27) as it holds most of the image's information. Thus, save the computational time and cost.
- The IDWT is applied to reconstruct the scrambled image$N$[22, 23].
- The initial condition for the proposed MFC-SM$_2$ is derived using the Fk: (x, y) using Eqn. (21, 22) for generating a chaotic sequence to construct $K_{C_{H\times H}}$ where $H = M/2$.
- Constructing the K_{IM}using the $K_C$and the identity matrix I as in Eqn. (16). WhereK_{IM} dimention is $M \times M$.
- The cipher image C is computed as in Eqn. (13).

2.10. Partial Image Decryption

- Using the decrypt Fk' as the same in Eqn. (25) to be the initial condition for the proposed MFC-SM$_2$to generate a chaotic sequenceto be used for generating$K_{C_{N\times N}}^{'}$.
- Constructing the $K_{IM}^{'}$using the$K_C^{'}$and the identity matrix I as in Eqn. (16).
- The decipher image $N^{'}$is computed as in Eqn. (14).

- Applying The DWT for the decrypted image $N^{'}$ to generate vertical LH (CV'), horizontal HL(CH'), diagonal HH (CD') and approximation LL (CA') matrices
- The approximation LL (CA') matrix is descrambled using ACMwith parameters $Sc_1^{'}, Sc_2^{'}$wich are generated as in Eqn. (12, 13).
- The IDWT is applied to reconstruct the descrambled image$N^{'}$.

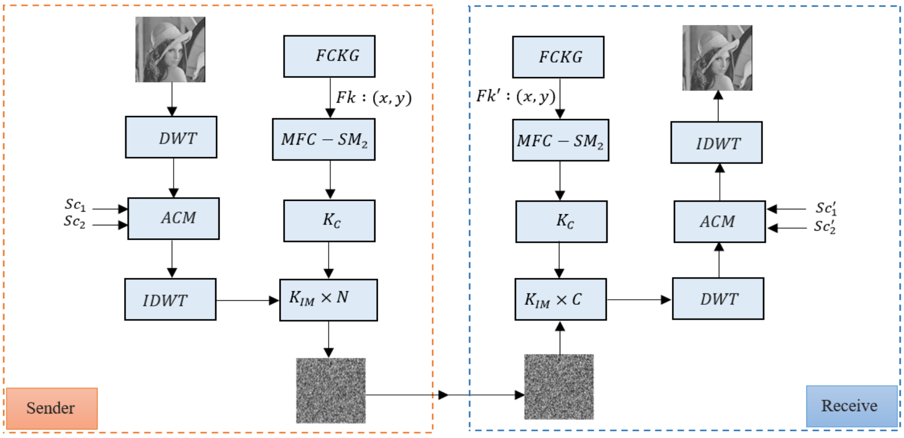


Fig. 5. Proposed partial image encryption and decryption diagram.

3. Simulation Results and Security Analysis

The laptop used is Intel(R) Core(TM) i7-4910MQ CPU@2.90GHz, 16GB RAM, Windows 10 (64-bit), MATLAB R2018b.Small parameters are used for simulation. The parameters chosen for EC $p = 113, a = -1, b = 17, G = (49,53)$. The proposed scheme will be applied on gray scale "Lena", "cameraman ", "boat", "pentagon" and "Barbara" images with size of 512×512. Fig.5 shows the result of image encryption. It is noted that the scheme converts the original images to encrypted image. The robustness of the encryption scheme is evaluated by measuring its resistance to several attacks such as known-plain text attack, cipher-text only attack, statistical attack, differential attack, and various brute-force attacks. Security analysis has been performed on the proposed scheme to be evaluated by discussing histogram, correlation coefficient, NPCR, UACI. The IEC parameters and the other output parameters of the DFC-KGthat are shown in the key schedule will be given as

Table 3. Simulation Parameters

Parameter	Value	Parameter	Value
i	7	Sk	(99, 76)
n_A	34	Sk'	(52, 78)
n_B	25	$ISk_1^{'}$	(11, 74)
P_A	(81, 27)	$ISk_2^{'}$	(97, 63)
P_B	(111, 24)	Sc_1, Sc_2	27, 92
$P_A^{'}$	(14, 108)	S, α, β	(40, 21, 60)
$P_B^{'}$	(15, 96)	Fk	(8, 62)

3.1. Histogram Analysis

Histogram plots the distribution of pixel intensities in an image. For a good encryption, the histogram of the encrypted image must be flat and uniform. So there is no information to reveal about the original image. The visual inspection for Fig. 6 shows that the histogram of the encrypted image is uniform and significantly different from the histogram of the original image. This is due to the applied confusion and diffusion in the proposed scheme. So it can defense against statistical attacks.

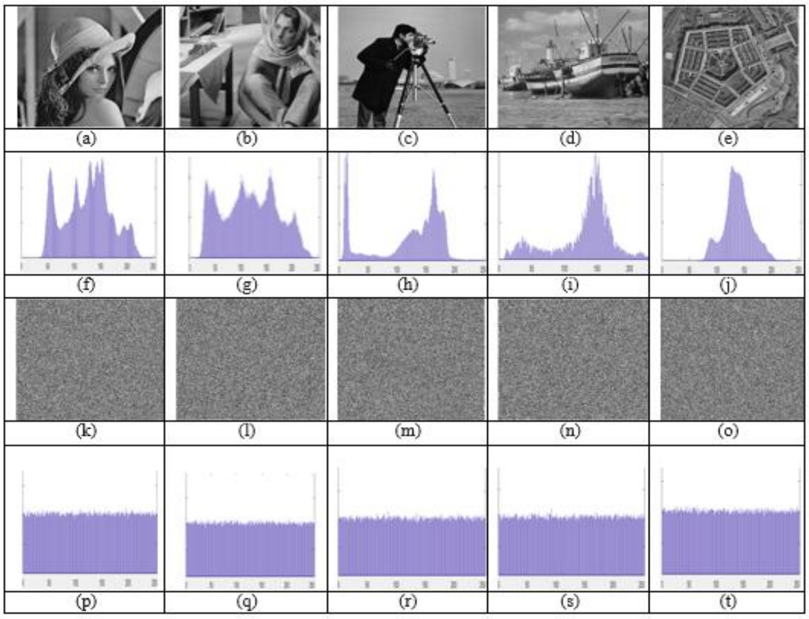


Fig. 6. Simulation results for images: a. Lena; b. Barbara; c. cameraman d. boat e. pentagon images; (f)-(j) Histogram of (a)-(e); (k)-(o) Cipher images of (a)-(e); (p)-(t) Histograms of (k)-(o).

3.2. Correlation Coefficient

In the original image each pixel is highly correlated with its adjacent pixels. A robust encryption scheme must reduce this correlation to the minimum possible value, so that it must be no correlation between adjacent pixels in vertical, horizontal and diagonal directions. Fig. 7 shows horizontal, vertical, and diagonal directions correlation of Fig.6 (a) "Lena" image and its cipher image in Fig. 6 (k). The correlation coefficient C_r between two adjacent pixels in an image is determined as in Eqn. (28), the C_r value should be almost equal to zero [15]. A comparison of the computed correlation coefficients for both the plain image and its corresponding cipher image is shown in Table (4). The comparison performed to grayscale tested images with the other recent schemes in [3, 28].

$$\boldsymbol{C_r} = \frac{\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{\sum_{i=1}^{N}(x_i - E(x))^2}\sqrt{\sum_{i=1}^{N}(yi - E(y))^2}} \quad (26)$$

Table 4. Correlation Coefficient of the Proposed Scheme for Gray Scale Tested Images.

Algorithm	Image	Horizontal		Vertical		Diagonal	
		Plain	Cipher	Plain	Cipher	Plain	Cipher
Ours	Lena	0.9741	0.0005	0.9862	0.0011	0.9619	0.0000
	Barbara	0.8954	0.0024	0.9589	0.0025	0.8830	-0.0004
	Peppers	0.9768	-0.0001	0.9792	-0.0018	0.9639	-0.0020
	Baboon	0.8665	-0.0007	0.7587	-0.0034	0.7262	0.0001
	House	0.9480	-0.0011	0.9577	-0.0025	0.9130	0.0014
Ref. [3]	Lena	0.9868	0.0019	0.9590	-0.0006	0.9717	-0.0014
	Barbara	0.9876	-0.00007	0.9704	-0.0022	0.9812	0.0007
	Peppers	0.9831	-0.0023	0.9658	-0.0013	0.9808	0.0012
	Baboon	0.754	-0.0004	0.7195	-0.0027	0.8635	0.0004
	House	0.9867	-0.0003	0.9713	0.0033	0.9841	-0.0017
Ref. [28]	Lena	0.9858	0.0019	0.9801	-0.0024	0.9669	-0.0011
	Barbara	0.9689	0.0024	0.8956	0.0031	0.8536	-0.0013
	Peppers	0.9807	-0.0028	0.9752	0.0039	0.9636	-.00024
	Baboon	0.7251	0.0024	0.8558	0.0011	0.6920	-0.0008
	House	0.8942	-0.0003	0.8936	0.0014	0.8401	0.0024

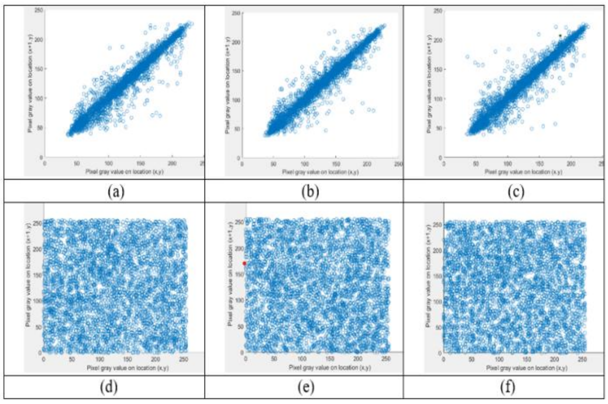


Fig. 7. Correlation of adjacent pixels in "Lena" image along (a) Plain image horizontal direction; (b) Plain image vertical direction; (c) Plain image diagonal direction; (d) Cipher image horizontal direction; (e) Cipher image vertical direction; (f) Cipher image diagonal direction.

It is clear that the correlation coefficients for the cipher images are near to zero, while for the original images are near to one. This indicates that the proposed scheme is highly resistant to statistical-based attacks.

3.3. Key Space Analysis

Key space is the set of all keys used in image encryption scheme. For efficient scheme it must be large enough to tackle the brute-force attack. It can be evaluated by measuring the key sensitivity and the number of keys.For 256-bit EC parameter used to be performed in the DFC-BID-IEC system. The key schedule shows allthe keysused in the

DFC-BID-IEC system. The total key space for the proposed encryption scheme is extremely large and it can be calculated as $2^{256} \times 2^{256} \times 2^{256} \times 2^{256} \times 2^{256} \times 2^{256} \times 2^{256} \times 2^{256} \times 2^{256} \times 2^{16} \times 2^{16} \times 2^{16} \times 2^{16} = 2^{2624}$, where, the first 256 bits are given from the EC $Sk,$ the second and third 256 bits are from the $P_A^{'}, P_B^{'}$, the fourth , fifth and sixth 256 bits given from $(Sk, ISk_1^{'}, ISk_2^{'})$, the seventh 256 bits are given from Bk, the eighth 256 bits are the MFC-SM$_1$ output, the ninth 256 bits are from RBk, the tenth 256 bits are for the Fk Also, the first 16 bits are the length of the initial values of the MFC-SM$_1$ and the second 16 bits are the length of the MFC-SM$_1$ control parameter similarly, the third and the forth is for MFC-SM$_2$. It is obvious that the total key space for the proposed encryption scheme is extremely large because of using the IEC and this is achieved without using EC point multiplication operation. A comparison for key space with recent schemes are shown in table (5).

Table 5. Key Space Analysis

	ours	Ref [29]	Ref[28]	Ref[3]
Key space size	2^{2624}	2^{512}	2^{564}	2^{772}

3.4. Key Sensitivity Analysis

A robust cryptographic scheme should have high sensitivity to all keys. A slight change in the key should provide a totally different cipher image. Also the recovery of the plain image will be impossible with slight change in the decryption key [25]. To test the key sensitivity in the proposed scheme Fig. 6 (a, b, c, d) "pentagon", " Lena ", " Barbara " and" cameraman "images is encrypted with the correct key as shown in Fig. 8 (e, f, g, h). The cipher images are decrypted with a correct key in Fig. 8 (i, j, k, l). Another key which is just one bit different from the original key used to encrypt the original images in Fig. 8 (l, m, n, o), Fig. 8 (p, r, s, t) shows the decrypted images which are encrypted using a wrong key which is just a bit different from the original key. These differences are huge, which proves that the proposed scheme has high sensitivity to the initial keys and so it has a strong defense against the brute-force and statistical attacks.

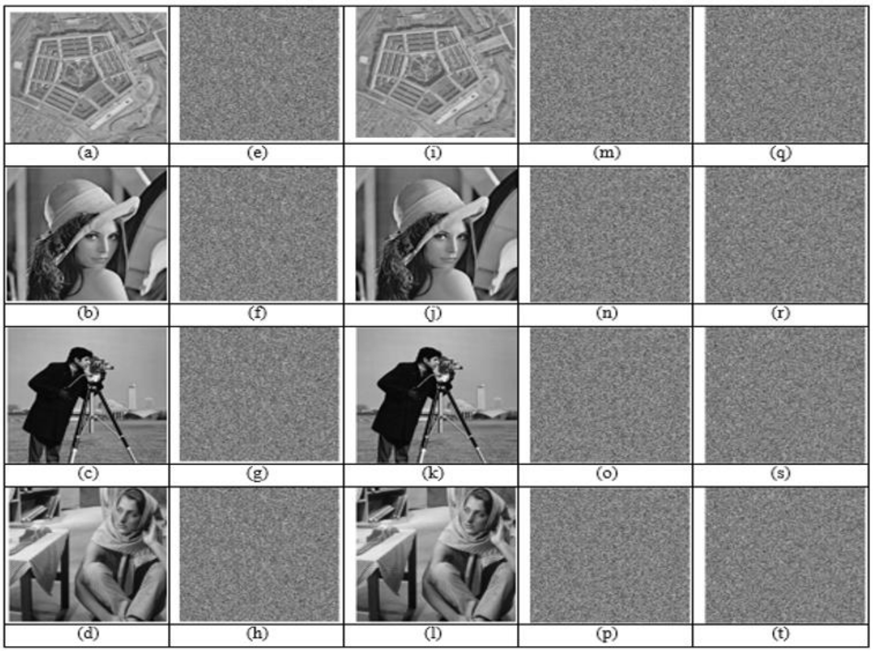


Fig. 8. The key sensitivity analysis. a. Pentagon; b. Lena; c. Barbara; d. cameraman plain images; (e)-(h) Encrypted image with original keys; (i)-(l) Decrypted images with correct key; (m)-(p) encrypted images with modified keys; (q)-(t) Decrypted images with wrong key respectively.

3.5. Differential Attack Analysis

A good diffusion performance is a measure of the strength of an image encryption scheme. It means a strong dependency of cipher image pixels on the plain image pixels. The differential attack resistance can be evaluated by comparing the differences of cipher images if the plain image one bit changed, it should provide a totally different cipher image. Number of pixels change rate (NPCR) and unified average changing intensity (UACI) are two quantitative measures used to ensure the security of an image encryption scheme against any differential attack [27].

$$\boldsymbol{NPCR} = \frac{\sum_{\mathrm{i,j}} \mathbf{D(i,j)}}{\boldsymbol{M \times N}} \times \mathbf{100}\% \tag{29}$$

$$\boldsymbol{UACI} = \frac{\mathbf{1}}{\boldsymbol{M \times N}} \frac{\sum_{i,j} |C_1(i,j) - C_2(i,j)|}{\mathbf{255}} \times \mathbf{100}\% \tag{27}$$

where M and N are the width and height size of the plain image respectively, $\boldsymbol{C_1(i,j)}$ and $\boldsymbol{C_2(i,j)}$ are the values of the pixels in the position $(\boldsymbol{i,j})$ of the two cipher images $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ before and after changing one bit of the plain image, $\mathbf{255}$ is the number of gray levels and $D(i,j)$ is given as:

$$D(i,j) = \begin{cases} 0, & for\, \mathrm{C_1\ (i,j) = C_2\ (i,j)} \\ 1, & for\, \mathrm{C_1\ (i,j) \neq C_2\ (i,j)} \end{cases} \tag{28}$$

The theoretical values of $NPCR = 99.61\%$ and $UACI = 33.46$. For better and more secure encryption scheme, the values of NPCR and UACI increase above or equal to these theoretical values. For the proposed scheme, the value of one pixel which randomly chosen is modified.Then the cipher image of the original and the modified image C_1 and C_2respectively, NPCR and UACI increase above or equal to these theoretical values. For the proposed scheme, the value of one pixel which randomly chosen is modified. Then the cipher image of the original and the modified image C_1 and C_2 respectively, NPCR and UACI are computed for the different images. The result tabulated in Table (6) and compared with Ref. [3, 28].

Table 6. NPCR AND UACI Comparison

Algorithm	Image	Lena	Baboon	Barbara	Peppers	House
Ours	NPCR (%)	99.63	99.61	99.60	99.61	99.61
	UACI (%)	33.48	33.55	33.51	33.42	33.51
Ref. [3]	NPCR (%)	99.62	99.61	99.60	99.61	99.61
	UACI (%)	33.48	33.46	33.44	33.55	33.52
Ref. [28]	NPCR (%)	99.61	99.61	99.57	99.61	99.62
	UACI (%)	33.46	33.49	33.42	33.48	33.50

3.6. Complexity Analysis

Low computation complexity and fast speed are properties of a good encryption scheme. EC point multiplicationis the most time consuming operation. The proposed scheme has a low number of EC point multiplication operation if it compared to many recent EC based image encryption schemes. A comparison with other recent schemes for encryption and decryption execution time for 256×256 and 512×512image sizes, the resultes are illustrated in Table (7). From Table (8), it is obvious that the proposed scheme has a time saving compared to other schemes. Also, it uses self-invertible matrix multiplication operation to reduce the computational time. So the proposed scheme is very efficient for real time image encryption.

Table 7. Encryption time (in sec.) comparison of the tested images

Image size	256×256	512×512
Ours	0.17685	0.2151
Ref. [3]	0.23	0.68
Ref. [29]	1.170844	4.73389
Ref. [30]	0.498021	0.938217
Ref. [31]	1.44	5.41

Table 8. Encryption time saving (%) of the proposed scheme

Image size	Time saving (%)	Ref. [29]	Ref. [30]	Ref[3]	Ref. [31]
256×256		84.89%	64.49%	23.1%	87.71%
512×512		95.45%	77.07%	68.36%	95.81%

4. Conclusion

In this paper, a new dynamic fractional chaotic biometric digital identity IEC mechanism has been proposed in order to achieve a robust partial image encryption scheme. The scheme consists of two parts of encryption. Firstly, the IEC Diffie-Hellman key exchange technique is used to solve the key distribution and management problem of symmetric key encryption. Secondly, the initial state, the biometric key and the proposed modular fraction chaotic sine map are used to build the key schedule. This condition allows the system to vary the keys every process to attain a good randomness and makes the scheme more resistant. Thus, overcome the chosen plaintext attacks. The keys generated from a combination between hidden cyclic isomorphic elliptic curve, biometric key and the proposed modular fraction chaotic sine map. The encryption and decryption process depend on scrambled plain image pixel values where scrambling is performed using Arnold's transformation. The proposed modular fraction chaotic sine map generates a chaotic sequence used to construct self-invertible matrix. From the security results, the proposed system is more efficient and has faster encryption and decryption time compared with other recent chaotic map EC based schemes. It has large key space, key-dependent pixel value replacement, low correlation and can resist statistical, differential and noise attacks. In the future work, and due to the proposed scheme advantages, it may be applied to multimedia such as audio and video with more performance improvement.

References

1. Y. Luo and M. Du, "A self-adapting image encryption algorithm based on spatiotemporal chaos and ergodic matrix," Chin. Phys. Rev. B, vol. 22, no. 8, pp. 316_324, 2013.
2. Y. Luo, L. Cao, S. Qiu, L. Hui, J. Harkin, and J. Liu, "A chaotic map control-based and the plain image-related cryptosystem," Nonlinear Dyn., vol. 83, no. 4, pp. 2293_2310, Mar. 2016.
3. Ro. Ismail, "Secure Image Transmission Using Chaotic Enhanced Elliptic Curve Cryptography," In: IEEE Access, vol. 7, no. 18576096,2019
4. G. J. Simmons, "Symmetric and asymmetric encryption," ACM Comput.Surv., vol. 11, no. 4, pp. 305_330, 1979.
5. M. Miller, "Uses of elliptic curves in cryptography". Advances in Cryptography Crypto '85.1986; 417-426.
6. N. Koblitiz, "Elliptic curve cryptosystems". Mathematics of computation. Vol. 48; No. 177; 1987; 203-208.
7. Ar. kumar,S.S.Tyagi, Man. Rana, Ne. Aggarwal, Pa. Bhadana, A Comparative Study of Public Key Cryptosystem based on ECC and RSA, Int. Journal on Com. Sci. and Eng., Vol. 3 No. 5 May 2011
8. Esam A. A. A. Hagras,"Selective Image Encryption Based on Multi-Level 2D-DWT and Multi-Map Chaotic System,"Int. Journal of Net. Security, vol. 9, no. 4, 2010.

9. M. Essaid, I. Akharraz, A. Saaidi and A. Mouhib, "Image encryption scheme based on a new secure variant of Hill cipher and 1D chaotic maps", Jou. of Inf. Sec. and Applications no.47 pp.173–187, 2019.
10. Arr. D., Alv. G, Fer. V., On the inadequacy of the logistic map for cryptographic applications., arXiv: 0805.4355, 2008.
11. D. G. Abraham, G. M. Dolan, G. P. Double and J. V. Stevens, "Transaction Security System", IBM Systems Journal, vol. 30, no. 2, pp. 206-229,2011.
12. Dol. Si. Laiphrakpam, Man. Si. Khumanthem, "A robust image encryption scheme based on chaotic system and elliptic curve over finite field"Springer Science Business Media, Multimed Tools Appl (2018) 77:8629–8652.
13. M. Bakr, M. A. Mok., A. Ta., "Modified Elliptic Curve Cryptography in Wireless Sensor Networks Security", 978-1-5386-9239-4/18, IEEE, 2018.
14. Y. Liu., J. Li., J. Liu, J.Che., J. Liu., L. Wang and X. Bai., "Robust Encrypted Watermarking for Medical Images Based on DWT-DCT and Tent Mapping in Encrypted Domain",X. Sun et al. (Eds.): ICAIS 2019, LNCS 11633, pp. 584–596, 2019.
15. Ah. A. Abd El-Latif, X. Niua, "A hybrid chaotic system and cyclic elliptic curve for image encryption", Int. J. Electron. Commun. (AEÜ) 67 (2013) 136– 143, 2013.
16. N. A. Azam, U. Hayat, I. Ullah, "An Injective S-Box Design Scheme over an Ordered Isomorphic Elliptic Curve and Its Characterization", Sec. and Com. Net.,Vol. 2018, A. ID 3421725, pp. 9, 2018.
17. X. Zhang, L.Wang, Y. Niu , G. Cui and S. Geng, " Image Encryption Algorithm Based on the H-Fractal and Dynamic Self-Invertible Matrix, Com. Int.andNeu. Vo. 2019, A. ID 9524080, pp. 12, 2019.
18. Guo-Cheng We, DumitruBaleana, Sheng-Da Zang, "Discrete chaos in fractional sine and standard maps," Physics letter A, 378, pp 484-487, 2014.
19. G. Jak. and K. P. Sub., "Discrete Lyapunov exponent and differential cryptanalysis," IEEE Trans. Circu. Syst. II, vol. 54, no. 6, pp. 499–501, Jun. 2007.
20. S. Turner, D. Brown, K. Yiu, R. Housley, and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information," RFC Editor RFC5480, 2009.
21. D. H. Ou, W. Sun, and B. Lin, "A novel image encryption scheme with the capability of checking integrity based on inverse matrix," Journal of Graphics, vol. 33, no. 2, pp. 89–92, 2012.
22. Zheng, P., Huang, J.: Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain. IEEE Trans. Image Process. 22, 2455–2468 (2013).
23. Y. a. Liu, J. Li, J. Liu, J. Cheng, J. Liu, L. Wang, and X. Bai, " Robust Encrypted Watermarking for Medical Images Based on DWT-DCT and Tent Mapping in Encrypted Domain", X. Sun et al. (Eds.): ICAIS 2019, LNCS 11633, pp. 584–596, 2019.
24. G. Panchal, D. Samanta, "A Novel Approach to Fingerprint Biometric-Based Cryptographic Key Generation and its Applications to Storage Security", Computers and Electrical Engineering 0 0 0 (2018) 1–18.
25. R. K. Kodali and Prof. N.V. Sarma, "ECC implementing using Koblitz's Encoding", Dep. of Ele. and Comm. Engineering, Nat. Ins. of Technology, Warangal.

26. T. Dhanashree, S. V. Rakesh and S. K. Premnath "An Approach for Security of Images over Elliptical Curve Cryptography and Digital Signature", Int. Jou. of Com. Applications, vol. 153, no. 11, 2016.
27. Y. Wu, J.P. Noonan, S. Agaian, " NPCR and UACI randomness tests for image encryption", J. Sel. Areas Telecommun. 4 (1) (2011) 31–38.
28. Y. Luo, X. Ouyang, J. Liu and L. CAO, " An Image Encryption Method Based on Elliptic Curve Elgamal Encryption and Chaotic Systems", vol. 7, ,IEEE Access ,2019.DOI 10.1109/ACCESS.2019.
29. L. D. Singh and Kh. M. Singh, "Image Encryption using Elliptic Curve Cryptography", Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015), Procedia Computer Science 54 (2015) 472 – 481, Elsevier.
30. S. Zhu, C. Zhu, and W. Wang, "A novel image compression-encryption scheme based on chaos and compression sensing," IEEE Access, vol. 6, pp. 67095_67107, 2018.
31. R. I. Abdelfatah, "A new fast double-chaotic based image encryption scheme", Multimedia Tools Appl., Springer Science+Business Media, LLC, part of Springer Nature 2019.

Ahmed Kamal, received the B.Sc. degree in Electrical Engineering from Alexandria University, Egypt in 2007, the M.Sc. degree in Electrical Engineering from Alexandria University, in 2021. He is currently a teaching assistant in communication science branch in the Air Defense College, Alexandria University. His current research interests include data protection in digital communication systems and developments of new encryption algorithms.

Esam A. A. Hagras received the B.Sc. degree in Electrical Engineering from Alexandria University, Egypt in 1994, the M.Sc. degree in Electrical Engineering from Mansoura University, Egypt, in 2001 and the Ph.D. degree in Electrical Engineering from Alexandria University, in 2008. He has been Head of Electronics & Communication Research Center, Armed Forces, Cairo, Egypt from 2010 to 2017. He is currently an Assistant Professor with Communications and Computer Department, Faculty of Engineering, Delta University for Science and Technology, Gamasa, Mansoura, Dakahlia, Egypt. His current research interests include data protection in digital communication systems and developments of new encryption algorithms.

H. A. El-Kamchochi Professor with Electrical Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt. His current research interests include data protection in digital communication systems and developments of new encryption algorithms. https://dblp.org/pid/12/1592.html

Received: May 02, 2020; Accepted: February 17, 2021.

Computer Science and Information Systems 18(3):1077–1100 https://doi.org/10.2298/CSIS200131034C

Time-aware Collective Spatial Keyword Query

Zijun Chen[1,2]* Tingting Zhao[1] and Wenyuan Liu[1,2]

[1] School of Information Science and Engineering, Yanshan University,
Qinhuangdao 066004, China
[2] The Key Laboratory for Computer Virtual Technology
and System Integration of Hebei Province,
Qinhuangdao 066004, China
zjchen@ysu.edu.cn, tingtingzhao@stumail.ysu.edu.cn, wyliu@ysu.edu.cn

Abstract. The collective spatial keyword query is a hot research topic in the database community in recent years, which considers both the positional relevance to the query location and textual relevance to the query keywords. However, in real life, the temporal information of object is not always valid. Based on this, we define a new query, namely time-aware collective spatial keyword query (TCoSKQ), which considers the positional relevance, textual relevance, and temporal relevance between objects and query at the same time. Two evaluation functions are defined to meet different needs of users, for each of which we propose an algorithm. Effective pruning strategies are proposed to improve query efficiency based on the two algorithms. Finally, the experimental results show that the proposed algorithms are efficient and scalable.

Keywords: Collection of objects, TR-tree, Valid time of the objects, Keyword query.

1. Introduction

As textual information on location-based geographic information has been paid more and more attention, spatial keyword query technology is proposed [8]. With the continuous development of spatial keyword query, in some practical applications, people begin to pay attention to the valid time of geo-spatial objects. For example, visitors want to plan a trip based on the opening time of geo-spatial objects, which is the time-aware spatial keyword query [5]. However, [5] only considers the case that a single object contains all the query keywords. As far as we know, there is no work to consider the valid time of objects based on the collective spatial keyword query. The collective spatial keyword query refers to finding a group of objects that together contain all of the query keywords and are near to the query location. Therefore, this paper proposes a new query, i.e., time-aware collective spatial keyword query (TCoSKQ). An example is illustrated in Fig. 1, where a user (at the location of q) plans to visit a "gym" from 7:00 to 8:30, and a "restaurant" from 9:00 to 10:00. Collective spatial keyword query may return $\{o_1, o_2\}$, while TCoSKQ may return $\{o_3, o_2\}$. The reason of the difference is that TCoSKQ considers the valid time of the objects. Although [6] is closely related to ours, they do not consider the case that there exists a query point. [21] considers the valid time of the objects too, but it aims to find a set of k objects ranked the highest according to a ranking function.

* Corresponding author

TCoSKQ considers the positional relevance, textual relevance and temporal relevance between the objects and the query at the same time. In order to solve the TCoSKQ problem, we define two evaluation functions to meet different needs of users, i.e., $score_1$ and $score_2$, which are suitable for the cases that the object containing the query keyword is close to and far from the query position, respectively. Then, TCoA1 algorithm and TCoA2 algorithm are designed for the two evaluation functions, respectively. Both algorithms use multiple Time-aware R-tree (TR-tree) [6] to index the valid time information and the spatial information of objects, that is, objects containing the same keyword are on the same TR-tree. TCoA1 algorithm finds feasible solutions by the distance dominators, which are searched from near to far, starting from the query location. Finally, the feasible solution with the largest $score_1$ is the final result. In order to improve the efficiency of the query, we propose effective pruning strategies for pruning the distance dominators that are unlikely to appear in the final result. TCoA2 algorithm searches for the center object from near to far from the query location, in which TCoA1* algorithm is called to find the feasible solution. TCoA1* algorithm is got by modifying TCoA1 algorithm. The feasible solution with the largest $score_2$ is the optimal solution, and effective pruning strategies are proposed to improve the query efficiency.

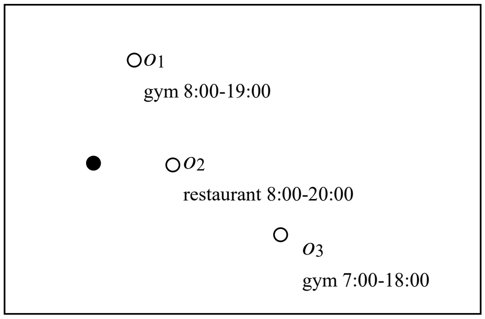


Fig. 1. Example of a TCoSKQ

To summarize, the main contributions of this paper are:

(1) We propose a new query, i.e., time-aware collective spatial keyword query (TCoSKQ).

(2) We propose two evaluation functions (i.e., $score_1$ and $score_2$), and propose effective pruning strategies and algorithms (i.e., TCoA1 and TCoA2) for these two functions to solve TCoSKQ.

(3) We conduct extensive experiments using the data sets to demonstrate the efficiency and scalability of our algorithms.

This paper introduces related work in Section 2. Section 3 gives problem definition. Section 4 gives the TR-tree indexing structure. Section 5 and 6 elaborates TCoA1 and TCoA2 algorithm, respectively. Section 7 gives the experimental results and analysis. Section 8 concludes the paper.

2. Related Work

In recent years, spatial keyword query has attracted much attention from spatial database community. Scholars have proposed effective techniques to deal with spatial keyword query. The early spatial keyword query is mainly for the case of retrieving a single object containing all query keywords and close to the query position. It is roughly divided into three types: Boolean kNN query [2], [18], ranked kNN query [14] and Boolean range query [19].

As people's needs increase, [1] observed that there may be situations where a single object cannot meet the needs of users, therefore, they first proposed collective spatial keyword query (CoSKQ). CoSKQ refers to retrieving a group of spatial web objects such that the group's keywords cover the query's keywords and such that objects are nearest to the query location and have the lowest inter-object distances. Based on two types of cost function, they study two instances of this problem, both of which are NP-complete. So they proposed effective approximate algorithms and exact algorithms to solve the two instances. Based on the shortcomings of the query in [1], [15] defined a new cost function describing the quality of set, proposed a new collective query processing method based on spatial keyword, and devised the corresponding approximate algorithm and exact algorithm.

With the intensive study of the collective spatial keyword query, some variants of the collective spatial keyword query have been proposed. [7] focused on specific spatial keyword set search in specific direction, and proposed a query algorithm based on grid index. [3] proposed an inherent-cost aware collective spatial keyword query, which takes into account the inherent cost of each object, and gave an exact algorithm and approximate algorithm that can solve the problem. Considering the importance of keyword level, [20] proposed a level-aware collective spatial keyword query, the corresponding cost function, the exact algorithm, and the approximate algorithm. Group-based collective keyword querying was proposed in [17], which considers the case of group of users. The query aims to find a region containing a set of objects that covers all the query keywords and these objects are close to the group of users and are close to each other. [9] and [22] have successively proposed methods for solving collective spatial keyword query on the road network. [11] considered the problem of scalable collective spatial keyword queries and proposed a distributed method to solve this problem effectively. In [4], a unified cost function and a unified method are proposed for the collective spatial keyword query problem to systematically solve the query problem.

With the in-depth study of spatial keyword queries, the valid time information of objects has attracted people's attention. [5] proposed a time-aware Boolean spatial keyword query (TABSKQ), which returns the k objects that satisfy users' spatio-temporal description and textual constraint, designed TA-tree index structure, and proposed algorithms to process TABSKQ efficiently. [21] proposed time-aware spatial keyword queries on road networks, which finds the k objects satisfying users' spatio-temporal description and textual constraint, and devised several effective algorithms using the TG index. [6] proposed a time-aware spatial keyword cover query (TSKCQ), devised a TR-tree for indexing the temporal information and the spatial information of objects, and proposed an exact algorithm to tackle TSKCQ.

As far as we know, the previous work on collective spatial keyword query does not consider the importance of the object's valid time information. Therefore, we propose a new query, that is, time-aware collective spatial keyword query, which considers the object's valid time based on the collective spatial keyword query.

3. Problem Definition

In spatial dataset, each object may be associated with one or multiple keywords. We convert the object with multiple keywords into multiple objects in the same location. For any object o with m ($m > 1$) keywords, we will create $m - 1$ other objects with the same location of o so that each of the m objects has only one different keyword. Let D be a set of objects. Each $o \in D$ is associated with a location denoted by $o.\lambda$, a keyword denoted by $o.k$ and a valid time denoted by $o.t$, where $o.t$ is in the form of (st, et) with st and et being the starting time stamp and the ending time stamp of o, respectively. Similar to [5] and [6], we integerize st and et of o, and take one hour as a time unit. As an example, any time stamp among [13:00, 14:00) can be converted to a time unit 13. (13:10, 18:00) can be converted to (13, 18). For simplicity, let et, $st \in [0, 24]$ and $st \leq et$. The input of a time-aware collective spatial keyword query (TCoSKQ) is $q = (\lambda, K, T)$, where $q.\lambda$ is the query location, $K = \{k_1, ..., k_m\}$, $T = \{t_1, ..., t_m\}$, $k_i (1 \leq i \leq m)$ is the ith query keyword, and t_i is a time interval specified in the query for k_i. For the query q, the object set $S = \{o_1, ..., o_m\} (S \subseteq D)$ containing all the keywords in K is called the feasible solution.

Definition 1. *(Time-aware collective spatial keyword query (TCoSKQ)) Given a spatial database D and a query $q = (\lambda, K, T)$, TCoSKQ returns an optimal solution S (which is also a feasible solution), such that $score(q, S) \geq score(q, S')$, where S' is any feasible solution.*

According to the two different needs of users, we give the definition of $score(q, S)$:

$$score(q, S) = \begin{cases} score_1(q, S), & \text{the first evaluation function} \\ score_2(q, S), & \text{the second evaluation function} \end{cases} \tag{1}$$

where $score_1$ and $score_2$ are evaluation functions that satisfy the user's first and second requirement, respectively.

(a) The $score_1$ is defined as:

$$score_1(q, S) = \alpha(1 - \max_{o \in S} \frac{dist(q, o)}{max\_dist}) + (1 - \alpha) \min_{o \in S, k_i \in K, o.k = k_i} \frac{|t_i \cap o.t|}{|t_i|} \tag{2}$$

where $dist(q, o)$ is the Euclidean distance between q and o, max_dist is the maximum distance between any two objects in the spatial database D, and $\alpha (0 < \alpha < 1)$ is a specified parameter. Let $\Gamma_o = \frac{|t_i \cap o.t|}{|t_i|}$, where the object o contains the keyword k_i in K.

(b) The $score_2$ is defined as:

$$score_2(q, S) = \max_{o \in S} score(q, S, o) \tag{3}$$

$$score(q, S, o) = \beta(1 - \frac{dist(q, o)}{max\_dist}) + (1 - \beta) score_1(q', S), o \in S \tag{4}$$

where $q' = (o.\lambda, q.K, q.T)$. According to (3), a function value is obtained by centering on each object in S, so that such an object is called the center object of S. We call o' the best center object if $o' = \arg\max_{o \in S} score(q, S, o)$.

For $score_1$ and $score_2$, the user could choose the function according to the query location. If the user is in the downtown area at present, $score_1$ may be chosen to find the result. But if the user is far away from downtown now, the user may have to choose $score_2$ to find the result.

The notations used in this work are summarized in Table 1.

Table 1. Symbols and Description

Notation	Description
D	a set of objects
$\mathit{\Gamma}_o$	the temporal overlap ratio of the object o
$score_1$	the evaluation function that satisfies the user's first requirement
$score_2$	the evaluation function that satisfies the user's second requirement
TR$_{k_i}$-tree	the time-aware R-tree for keyword k_i
$d(q, elem)$	the distance (or minimum distance) from query q to object (or node) $elem$
$C(q, r)$	a circle with q as the center and r as the radius
$NN(q, k)$	k-keyword nearest neighbor of q

4. TR-tree Index Structure

To process TCoSKQ, we use TR-tree [6] as the index structure of the algorithms, which is an extension of R-tree [10]. On the basis of R-tree, a new dimension is added for indexing valid time of objects.

As far as we know, the current collective spatial keyword queries use the index structure of a single tree, that is, a tree indexes objects in all geographic locations. A single tree structure suits the situation that most keywords are query keywords. However, in practice, users will only use a small fraction of keywords as query keywords. So, multiple trees are also used in this work, one for each keyword. The TR-tree for keyword k_i is denoted as TR$_{k_i}$-tree.

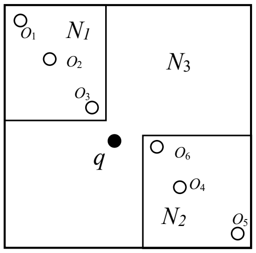
In the case of knowing the query keywords, we only need to find the TR-tree corresponding to the query keyword, which greatly reduces the number of objects to be considered in the query.

Next, we introduce the non-leaf nodes and leaf nodes of TR-tree in detail:

Non-leaf nodes of TR-tree contain entries of the form $N(ptrs, mbr, UT)$, where $ptrs$ is the address of a child node of N, mbr is the minimum bounding rectangle (MBR) of all rectangles in entries of the child node, and UT is a set of the time intervals that are the union of the valid times of the objects in N.

Leaf nodes of TR-tree contain entries of the form $o(id, l, t)$, where id refers to the object o, l represents the coordinates of o, and t is the valid time of o.

Fig. 2a gives the placement of objects containing the keyword "restaurant". Objects and valid time of objects are shown in Fig. 2b. The TR$_{k_1}$-tree for keyword "restaurant" is created, and the nodes and their corresponding valid time of the tree are shown in Fig. 2c.



(a) Object placement

Object	Valid time
o_1	(6:00, 10:00)
o_2	(7:00, 9:00)
o_3	(7:00, 11:00)
o_4	(13:00, 17:00)
o_5	(21:00, 24:00)
o_6	(9:00, 14:00)

(b) Objects and valid time of objects

Node	Valid time
N_1	(6:00, 11:00)
N_2	(9:00, 17:00), (21:00, 24:00)
N_3	(6:00, 17:00), (21:00, 24:00)

(c) Nodes and time intervals of nodes

Fig. 2. Example of a TR-tree

TR-tree inherits the important features of R-tree: the minimum distance from the query to the parent node is less than or equal to the minimum distance to the child node.

Example 1. In Fig. 2, we assume that the time interval specified in the query for the query keyword "restaurant" is (8:00, 19:00). Thus, we have $\Gamma_{o_1} = 2/11$, $\Gamma_{o_2} = 1/11$, $\Gamma_{o_3} = 3/11$, $\Gamma_{o_4} = 4/11$, $\Gamma_{o_5} = 0$, $\Gamma_{o_6} = 5/11$.

Definition 2. *(MinDist Distance [16]) In Euclidean space of dimension n, the minimum distance between a point q and MBR $N(s,u)$ is denoted by $MinDist(q,\ N(s,u))$, which is defined as follows:*

$$MinDist(q,N) = \sum_{i=1}^{n} |q_i - r_i|^2, r_i = \begin{cases} s_i, & q_i < s_i \\ u_i, & q_i > u_i \\ q_i, & \text{otherwise} \end{cases} \tag{5}$$

Given a query q,

$$Dist(q,elem) = \begin{cases} dist(q,elem), & elem \text{ is an object} \\ MinDist(q,elem), & elem \text{ is a node} \end{cases} \tag{6}$$

$$d(q,elem) = \frac{Dist(q,elem)}{max\_dist} \tag{7}$$

5. TCoA1 algorithm

We propose TCoA1 algorithm for the evaluation function $score_1(q, S)$ to solve the TCoSKQ problem. Before we introduce TCoA1 algorithm, we first introduce some notations.

Given a query $q = (\lambda, K, T)$, let S be a solution that satisfies the query, then

(1) The distance dominator of S is defined to be the object $o \in S$ that is most far away from q (i.e.,$o = \arg\max_{o \in S} dist(q, o)$).

(2) Let o be the distance dominator of S. A circle with q as the center and $d(q, o)$ as the radius is denoted by $C(q, r)$, where $r = d(q, o)$.

(3) Given a keyword k, for the objects containing k, the one who is closest to q is called the k-keyword nearest neighbor of q, denoted by $NN(q, k)$.

5.1. Pruning Strategies

The idea of TCoA1 algorithm is to find feasible solutions with the help of distance dominators. However, not every object can be distance dominators, so we will find the lower bound of distance to q for distance dominators by the following theorem.

Theorem 1. *Given a query $q = (\lambda, K, T)$, let FS be a feasible solution, let o be the distance dominator of FS, and let S =$\{NN(q, k_1), NN(q, k_2), ..., NN(q, k_m)\}$. Then, we have $d(q, o) \geq d_{LB}$, where $d_{LB} = \max_{o' \in S} dist(q, o')$.*

Proof. *Assume $d(q, o) < d_{LB}$. Let o_f be the distance dominator of S, i.e., $d_{LB} = d(q, o_f)$, and o_f contains the keyword k_f. Since $d(q, o) < d_{LB}$, we get $o_f \notin FS$, and there must be an object $o'_f \in FS$ containing k_f, such that $d(q, o'_f) \leq d(q, o)$. Therefore, we have $d(q, o'_f) < d_{LB}$. This conclusion contradicts that o_f is $NN(q, k_f)$, so it is true that $d(q, o) \geq d_{LB}$.* ■

If some distance dominators are too far from q, then the solutions obtained by such distance dominators would not be the optimal solution. Therefore, such distance dominators can be pruned by the following theorem.

Theorem 2. *Given a query $q = (\lambda, K, T)$, and the current optimal solution S. Let S' be any feasible solution, and let o be the distance dominator of S'. If $d(q, o) \geq d_{UB}$, where $d_{UB} = (1 - score_1(q, S))/\alpha$, then we have $score_1(q, S') \leq score_1(q, S)$.*

Proof. *Assume when o is the distance dominator of S' and $d(q, o) \geq d_{UB}$, there is $score_1(q, S') > score_1(q, S)$. Since $score_1(q, S') = \alpha(1-d(q, o))+(1-\alpha) \min_{o_j \in S'} \Gamma_{o_j} \leq \alpha(1 - d(q, o)) + (1 - \alpha)$, we have $\alpha(1 - d(q, o)) + (1 - \alpha) > score_1(q, S)$, i.e., $d(q, o) < (1-score_1(q, S))/\alpha$, that is, $d(q, o) < d_{UB}$, which contradicts the hypothesis.* ■

Fig. 3 shows the distance constraint when looking for distance dominators. Suppose o_1, o_2, o_3, o_4, o_5, and o_6 contain one query keyword in K respectively. Initially, we only need to consider the objects in the gray area (i.e., o_2, o_3, o_5) to be distance dominators. As the current optimal solution S changes, the upper bound d_{UB} will become smaller and smaller.

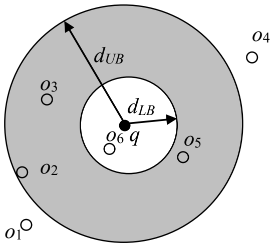


Fig. 3. Distance constraint for distance dominators

5.2. TCoA1 Algorithm Description

The steps of the algorithm are as follows:

Step 1: For each query keyword k_i in K, $NN(q, k_i)$ is found to form the current optimal solution S.

(1) For the m query keywords in K, the m TR-trees are found, respectively;

(2) For each TR$_{k_i}$-tree, a min heap $Minheap_i$ is created, with $d(q, elem)$ as the key, where $elem$ is the node or object of TR$_{k_i}$-tree;

(3) We create a max heap $maxHeap_i$ for each TR$_{k_i}$-tree, with $\varGamma_{o_i}$ as the key, where o_i represents the object accessed in the TR$_{k_i}$-tree;

(4) For each TR$_{k_i}$-tree, $NN(q, k_i)$ is found with the best-first strategy [12] to form the current optimal solution S, i.e., $S = \{NN(q, k_1)$, $NN(q, k_2)$,..., $NN(q, k_m)\}$.

The process of finding $NN(q, k_i)$ is as follows: The root node of TR$_{k_i}$-tree is kept in $Minheap_i$. Determine whether the top element of $Minheap_i$ is a node or an object. If it is a node, remove the node and store its children in $Minheap_i$. This process is repeated until the top element is an object, and the object is $NN(q, k_i)$.

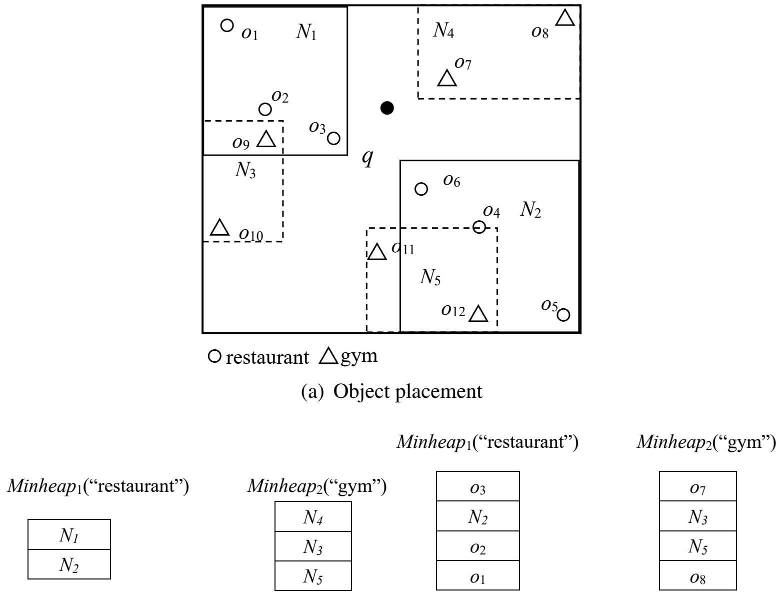
Example 2. Fig. 4 shows an example. Assume that the query keywords are "restaurant" and "gym". The initial state of $Minheap_1$ and $Minheap_2$ is shown in Fig. 4b. When S is found by using the best-first strategy for each TR-tree, the state of $Minheap_1$ and $Minheap_2$ is shown in Fig. 4c, we know that $NN(q, k_1) = o_3$, $NN(q, k_2) = o_7$, therefore, we have $S = \{o_3, o_7\}$.

Step 2: Using S, we can determine d_{LB}, which is used to identify the distance dominator. The object $o \in D$ will be put into the $maxHeap$ in ascending order $d(q, o)$. Once a distance dominator is put into the $maxHeap$, it will be used to find a feasible solution S'.

(1) Let $o_f = \arg\max_{o_f \in S} d(q, o_f)$, $d_{LB} = d(q, o_f)$;

(2) For each $o \in Minheap_i (1 \leq i \leq m)$, if $d(q, o) < d_{LB}$, then remove o from $Minheap_i$, and put it into $maxHeap_i$. All these objects will not be distance dominators according to Theorem 1.

(3) With best-first strategy, we look for a distance dominator $minobject$ from $Minheap_j$ $(1 \leq j \leq m)$, and put it into $maxHeap_j$ to find a new feasible solution S', which is formed by $minobject$ and the heads of all the max heaps except $maxHeap_j$. If $score_1(q, S)$ $< score_1(q, S')$, S is replaced by S'.



(a) Object placement

(b) Initial state of $Minheap_1$ and $Minheap_2$

(c) The changed state of $Minheap_1$ and $Minheap_2$

Fig. 4. An example of step 1

(4) According to Theorem 2, if $d(q, minobject) < (1 - score_1(q, S))/\alpha$, then we repeat (3), otherwise the algorithm terminates.

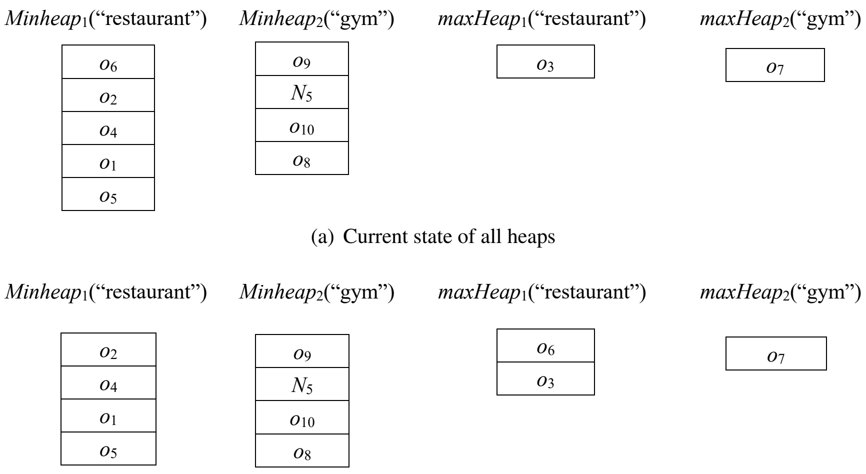
In (3), once a distance dominator $minobject$ is found, all the objects in $C(q, d(q, minobject))$ have been put into the max heaps. But we need not combine all the objects in the max heaps with $minobject$ to find the feasible solution with the largest $score_1$. Instead, we only get the feasible solution S', which is formed by $minobject$ and the heads of all the max heaps except the max heap of $maxHeap_j$. The reason is that $minobject$ is the distance dominator of S', that is, $minobject$ is the object furthest away from q in S'. Since $score_1(q, S') = \alpha(1 - d(q, minobject)) + (1 - \alpha) \min_{o \in S'} \Gamma_o$, we know that $\alpha(1 - d(q, minobject))$ is a fixed value when we compute $score_1(q, S')$. Therefore, we create corresponding max heap $maxHeap_i$ for each TR$_{k_i}$-tree, with Γ_{o_i} as the key, and just take the head of all the max heaps except $maxHeap_j$ to form S', which has the largest $score_1$ among all the combinations.

Example 3. From Example 2, it is known that $d_{LB} = d(q, o_7)$. Assume that the current state of $Minheap_1$, $Minheap_2$, $maxHeap_1$, and $maxHeap_2$ is shown in Fig. 5a. Next, we first compare $d(q, o_6)$ with $d(q, o_9)$. From Fig. 4a, we know that $d(q, o_6) < d(q, o_9)$. Then, o_6 is removed from $Minheap_1$, and Γ_{o_6} is calculated and stored in $maxHeap_1$. The state of all the heaps is shown in Fig. 5b. Since $d(q, o_6) > d_{LB}$, o_6 will be the distance dominator of the next feasible solution S', we get $S' = \{o_6, o_7\}$.

The pseudo-code of TCoA1 algorithm is presented in Algorithm 1. The m min heaps are initialized, and $Minheap_i$ is used to store objects or nodes in the TR$_{k_i}$-tree (line 1).

Algorithm 1: TCoA1(D, q)

```
Input: A spatial database $D$, a query $q = (\lambda, K, T)$, where $K = \{k_1, ..., k_m\}$,
       $T = \{t_1, ..., t_m\}$
Output: The result $S$
Initialize $m$ min heaps $Minheap_1, ..., Minheap_m$ with $d(q, elem)$ as key;
Initialize $m$ max heaps $maxHeap_1, ..., maxHeap_m$ with $\mathit{\Gamma}_{elem}$ as key;
$Lists = \emptyset$, $S = \emptyset$;
for each query keyword $k_i \in K$ do
    $Minheap_i \leftarrow$ the root in TR$_{k_i}$-tree;
    $Minheap_i =$ FindN($q$, $Minheap_i$, TR$_{ki}$-tree);
    $S \leftarrow Minheap_i.head$;
    $Lists \leftarrow Minheap_i$;
$d_{LB} = \max_{o \in S} d(q, o)$;
while $Lists \neq \emptyset$ do
    $Minheap_j = \arg \min_{Minheap_i \in Lists} d(q, Minheap_i.head)$;
    $minobject = Minheap_j.head$;
    Remove $Minheap_j.head$ from $Minheap_j$;
    $maxHeap_j \leftarrow minobject$;
    if $Minheap_j = \emptyset$ then
        Remove $Minheap_j$ from $Lists$;
    else
        $Minheap_j =$ FindN($q$, $Minheap_j$, TR$_{k_j}$-tree);
    if $d(q, minobject) < d_{LB}$ then
        continue;
    else
        if $d(q, minobject) < (1 - score_1(q, S))/\alpha$ then
            $S' = \emptyset$;
            for $(x = 1; x \leq m; x++)$ do
                if $x == j$ then
                    $S' \leftarrow minobject$;
                else
                    $S' \leftarrow maxHeap_x.head$;
            if $score_1(q, S') > score_1(q, S)$ then
                $S = S'$;
        else
            break;
return $S$;
```



(a) Current state of all heaps

(b) Changed state of all heaps

Fig. 5. An example of step 2

The m max heaps are initialized, and $maxHeap_i$ is used to store objects with keyword k_i (line 2). In lines 4-8, for each TR$_{k_i}$-tree corresponding to the query keyword k_i, Algorithm 2 is called to find $NN(q, k_i)$ (line 6). In line 7, $Minheap_i.head$ is $NN(q, k_i)$. After all the keyword nearest neighbors are found, the current optimal solution S is formed. In lines 10-32, for the heads of all the min heaps in $Lists$, the head $minobject$ with the smallest key is found and moved from $Minheap_j$ to $maxHeap_j$. After that, if $Minheap_j$ is not empty, the heap is processed until the head is an object. We process $minobject$ with d_{UB} and d_{LB}. In lines 22-30, for a distance dominator $minobject$, if $d_{LB} \leq d(q, minobject) < d_{UB} = (1 - score_1(q, S))/\alpha$, then a new feasible solution S' will be found. The current optimal solution will be updated by S' when $score_1(q, S') > score_1(q, S)$.

The pseudo-code of FindN algorithm for finding an object is presented in Algorithm 2. In lines 1-8, if the heap is not empty, then the heap will be traversed in the best-first strategy until the head of the heap is an object. In line 9, the changed heap is returned.

5.3. Time Complexity of TCoA1 Algorithm

We assume that the objects are uniformly distributed. According to [12], after finding the k nearest neighbors, the total cost is $O(klogk)$. In TCoA1 algorithm, there are m min heaps, which are used to find the nearest neighbor in the best-first strategy [12]. Each min heap is used separately. Each time only one of the min heap is chosen to find the nearest neighbor incrementally. And the found nearest neighbor will be put into the corresponding max heap. After the k nearest neighbors are put into a max heap, the total cost is $O(klogk)$. Once a distance dominator minobject from $Minheap_j$ is put into a max heap, a feasible solution is formed by minobject and the heads of all the max heaps except the max heap of $maxHeap_j$. The cost is $O(1)$. Let the maximum number of nearest neighbors found by min heap be k_a. So the time complexity of TCoA1 algorithm is $O(mk_alogk_a)$.

Algorithm 2: FindN(q, $heap$, TR$_{k_i}$-tree)

```
Input: a query $q$, a min heap $heap$ for keeping nodes or objects of TR$_{k_i}$-tree, TR$_{k_i}$-tree
       for the query keyword $k_i$
Output: $heap$
while $heap \neq \emptyset$ do
    $elem = heap.head$;
    if $elem$ is a node then
        Remove $elem$ from $heap$;
        for each child $e$ of $elem$ do
            $heap \leftarrow e$;
    else
        break;
return $heap$;
```

6. TCoA2 Algorithm

We propose TCoA2 algorithm for the evaluation function $score_2(q, S)$ to solve the TCoSKQ problem.

The idea of TCoA2 algorithm is to find the central object o in ascending order of $d(q, o)$, which is used to find a feasible solution. We propose the following theorem to prune the central object that cannot form the optimal solution.

The current optimal solution obtained during the search process is defined as follows: If S is the current optimal solution found by o, then for any result set S_i found by o_i so far, we have $score(q, S_i, o_i) \leq score(q, S, o)$.

Theorem 3. *Given a query $q = (\lambda, K, T)$, the current optimal solution S found by o. Let S' be any feasible solution, whose best center object is o'. If $d(q, o') \geq (1 - score(q, S, o))/\beta$, we have $score_2(q, S') \leq score(q, S, o)$.*

Proof. *Assume that if $d(q, o') \geq (1 - score(q, S, o))/\beta$, we have $score_2(q, S') > score(q, S, o)$. Since o' is the best center object of S', we have $score_2(q, S') = \beta(1 - d(q, o')) + (1 - \beta)score_1(q', S') \leq \beta(1 - d(q, o')) + (1 - \beta)$, where $q' = (o'.\lambda, K, T)$. Thus, $\beta(1 - d(q, o')) + (1 - \beta) > score(q, S, o)$, that is, $d(q, o') < (1 - score(q, S, o))/\beta$, which contradicts the hypothesis.* ■

It can be known from Theorem 3 that when the central object is searched from near to far, if the distance from the central object to the query reaches an upper bound, then it is not necessary to continue to search for the central object, and the search process can be terminated early.

The following example demonstrates that the center object found firstly may not be the best central object.

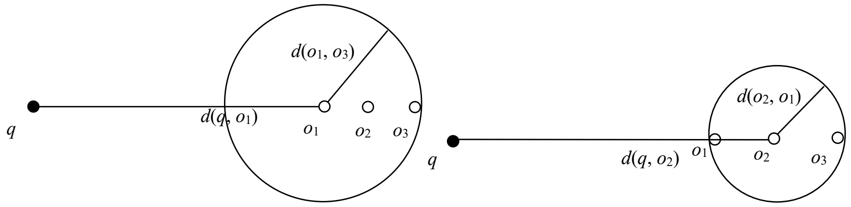
Example 4. Fig. 6 shows an example. Let $max\_dist = 100, dist(q, o_1) = 70, dist(o_1, o_3) = 20, dist(o_2, o_1) = 10, \alpha = 0.5, \beta = 0.3$, the feasible solution $S = \{o_1, o_2, o_3\}$, and $\min_{o_j \in S} \Gamma_{o_j}$=0.5.

(1) When S is found by the center object o_1, we have $score(q, S, o_1) = \beta(1 - d(q, o_1)) + (1 - \beta)score_1(q', S) = \beta(1 - d(q, o_1)) + (1 - \beta)(\alpha(1 - d(o_1, o_3)) + (1 - \alpha)\min\limits_{o_j \in S} \mathit{\Gamma}_{o_j}) = 0.3 * (1 - 70/100) + 0.7 * (0.5 * (1 - 20/100) + 0.5 * 0.5) = 0.545$.

(2) When S is found by the center object o_2, we have $score(q, S, o_2) = \beta(1 - d(q, o_2)) + (1 - \beta)score_1(q', S) = \beta(1 - d(q, o_2)) + (1 - \beta)(\alpha(1 - d(o_2, o_1)) + (1 - \alpha)\min\limits_{o_j \in S} \mathit{\Gamma}_{o_j}) = 0.3 * (1 - 80/100) + 0.7 * (0.5 * (1 - 10/100) + 0.5 * 0.5) = 0.55$.

(3) When S is found by the center object o_3, we have $score(q, S, o_3) = \beta(1 - d(q, o_3)) + (1 - \beta)score_1(q', S) = \beta(1 - d(q, o_3)) + (1 - \beta)(\alpha(1 - d(o_3, o_1)) + (1 - \alpha)\min\limits_{o_j \in S} \mathit{\Gamma}_{o_j}) = 0.3 * (1 - 90/100) + 0.7 * (0.5 * (1 - 20/100) + 0.5 * 0.5) = 0.485$.

According to the above result, it can be seen that only o_2 is the best central object of S.



(a) Find S by o_1

(b) Find S by o_2

Fig. 6. Find the best central object for S

Example 5. Consider Example 4 again. Assume that the current best solution S' is found by o' and $score(q, S', o') = 0.775$. By Theorem 3, we have $(1 - score(q, S', o'))/\beta = 0.75$. Since $d(q, o_1) = 0.7 < 0.75$, we should compute $score(q, S, o_1)$, which is 0.545. Because $0.545 < 0.775$, the current optimal solution is unchanged. Since $d(q, o_2) = 0.8 > 0.75$, the query is terminated according to Theorem 3. It is no longer necessary to calculate $score(q, S, o_2)$.

The main idea of TCoA2 algorithm is shown in Fig. 7.

The pseudo-code of TCoA2 algorithm is presented in Algorithm 3. The m min heaps are initialized, and $heap_i$ is used to store objects or nodes in the TR$_{k_i}$-tree (line 1). For each TR$_{k_i}$-tree, we store its root node in the corresponding heap (lines 3-5). In lines 6-23, for the head of each heap in $Lists$, the element $elem$ with the smallest key is found and removed. If $elem$ is a node, all children of $elem$ are stored in the corresponding heap. If $elem$ is an object, then it is a central object, which is used to find S'. In lines 16-21, if $d(q, elem) < (1 - Cscore))/\beta$, then it is possible to find the optimal solution by $elem$ according to Theorem 3. TCoA1* algorithm is called to find a feasible solution, and the final solution is updated.

TCoA1* algorithm is got by modifying TCoA1 algorithm as follows:

a. Different input. TCoA1* algorithm needs to know which object is to be the center object to find the result set;

```
Algorithm 3: TCoA2$(D, q)$
Input: A spatial database $D$, a query $q = (\lambda, K, T)$, where $K = \{k_1, ..., k_m\}$,
       $T = \{t_1, ..., t_m\}$
Output: The result $S$
Initialize $m$ min heaps $heap_1, ..., heap_m$ with $d(q, elem)$ as key;
$Lists = \emptyset$, $S = \emptyset$, $Cscore = 0$;
for each query keyword $k_i \in K$ do
    $heap_i \leftarrow$ the root in TR$_{k_i}$-tree;
    $Lists \leftarrow heap_i$;
while $Lists \neq \emptyset$ do
    $heap_j = \arg\min_{heap_i \in Lists} d(q, heap_i.head)$;
    $elem = heap_j.head$;
    Remove $heap_j.head$ from $heap_j$;
    if $elem$ is a node then
        for each child $e$ of $elem$ do
            $heap_j \leftarrow e$;
    else
        if $heap_j = \emptyset$ then
            Remove $heap_j$ from $Lists$;
        if $d(q, elem) < (1 - Cscore)/\beta$ then
            $q' = (elem.\lambda, K, T)$;
            $S'$ = TCoA1*$(D, q', elem)$;
            if $score(q, S', elem) > Cscore$ then
                $S = S'$;
                $Cscore = score(q, S', elem)$;
        else
            break;
return $S$;
```

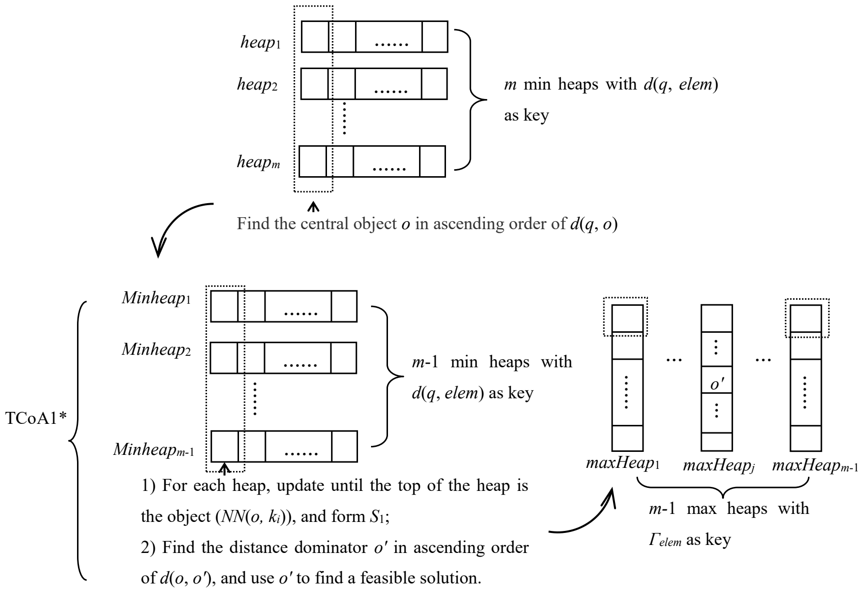

heap1
heap2
heapm
......
m min heaps with d(q, elem) as key
Find the central object o in ascending order of d(q, o)
TCoA1*
Minheap1
Minheap2
Minheapm-1
m-1 min heaps with d(q, elem) as key
1) For each heap, update until the top of the heap is the object (NN(o, ki)), and form S1;
2) Find the distance dominator o′ in ascending order of d(o, o′), and use o′ to find a feasible solution.
o′
maxHeap1
maxHeapj
maxHeapm-1
m-1 max heaps with Γelem as key

Fig. 7. The main idea of TCoA2 algorithm

b. For the input central object o, TCoA1* algorithm does not need to create a min heap and a max heap for the TR-tree corresponding to the keyword contained in o;

c. All result sets found by TCoA1* algorithm must contain the input central object.

To support the correctness of TCoA1* algorithm, we give the following Theorems by rewriting Theorem 1 and 2, respectively.

Theorem 4. *Given q'=$(o_i.\lambda, q.K, q.T)$, where o_i is a central object, let FS be a feasible solution containing o_i, let o be the distance dominator of FS, and let $S = \{NN(q', k_1)$,..., o_i, ..., $NN(q', k_m)\}$. Then, we have $d(q', o) \geq d_{LB}$, where $d_{LB} = \max\limits_{o' \in S} d(q', o')$.*

Proof. The proof is similar to that of Theorem 1. ■

Theorem 5. *Given q'=$(o_i.\lambda, q.K, q.T)$, where o_i is a central object, and the current optimal solution S containing o_i. Let S' be any feasible solution containing o_i, and let o be the distance dominator of S'. If $d(q', o) \geq d_{UB}$, where $d_{UB} = (1 - score_1(q', S))/\alpha$, then we have $score_1(q', S') \leq score_1(q', S)$.*

Proof. The proof is similar to that of Theorem 2. ■

For each found central object o, TCoA1* algorithm is called to find feasible solution S' containing o, where S' is the best solution with o being a central object according to Theorem 4 and 5.

Correctness analysis of TCoA2 algorithm: According to Definition 1, for any feasible solution S, it is necessary to calculate $score(q, S, o_i)$, where o_i is a best central object. But for S, we could not predict which object in S is the best central object, so we have to try each object o as a central object in ascending order of $d(q, o)$. For

TCoA2 algorithm, assume that the current optimal solution S_p is found by o_p, and the best central object of any feasible solution S is o. According to Theorem 3, if $d(q, o) \geq (1 - score(q, S_p, o_p))/\beta$, then we have $score_2(q, S) \leq score(q, S_p, o_p)$. Therefore, TCoA2 algorithm can return the correct result.

Time complexity of TCoA2 algorithm: We assume that the objects are uniformly distributed. In TCoA2 algorithm, there are m min heaps, which are used to find the nearest neighbor in the best-first strategy [12]. Once a nearest neighbor is found, it is used to call TCoA1* algorithm, whose time complexity is the same with that of TCoA1 algorithm. Let the maximum number of nearest neighbors found by min heap be k_b, and let the maximum number of nearest neighbors found by min heap in TCoA1* algorithm be k_a. So the time complexity of TCoA2 algorithm is $O(mk_b logk_b + mk_b mk_a logk_a)$.

7. Experiment and Result Analysis

All the algorithms are implemented in Java 1.7.0. All the experiments have been performed on a Windows 7 PC with an Intel(R) Core(TM) i5-2450M CPU and 4G RAM.

We use the datasets Oldenburg (https://www.cs.utah.edu /∼lifeifei/ SpatialDataset.htm) and GN (extracted from the Geographic Names Information System in USA, https://www.usgs.gov/), and generate randomly keyword information and valid time information for each object in the datasets. Oldenburg contains 6,105 objects and 59 different keywords. GN contains 2288631 objects and 1865 different keywords. We assign one keyword to each object using the Zipf distribution [13] in both datasets. The starting time stamp of objects in both datasets is set to follow the Gaussian distribution, and the ending time stamp of objects is computed by the starting time stamp plus the length of time interval which is in the range of [6, 12]. Each query location is randomly read from the dataset, and the query keywords and query time interval are randomly assigned to the query. We randomly produce 100 queries and report the average results.

7.1. TCoA1 algorithm and Baseline algorithm

The max heaps are used to obtain a feasible solution in step 2 of TCoA1 algorithm. In order to test the performance of those max heaps, we propose a baseline algorithm (Baseline) for comparison.

Baseline algorithm is got by modifying TCoA1 algorithm as follows:

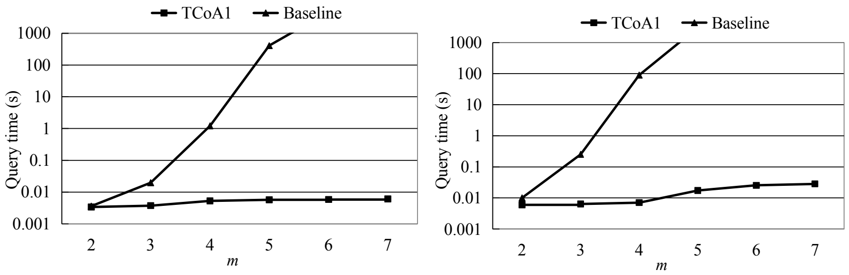
a. In step 1 (3) in section 5.2, the max heaps are replaced by lists. For each TR$_{k_i}$-tree, the list $list_i$ is created, which keeps objects that have been visited in the TR$_{k_i}$-tree.

b. In step 2 (3) in section 5.2, after finding the distance dominator $minobject$, we combine $minobject$ with all objects in each list except the list containing $minobject$ to get multiple feasible solutions. In each feasible solution, one object is taken from each list. We retain the solution S' with the largest $score_1$. Finally, the final result is updated with S'.

We should note that the pruning strategies for the distance dominators in TCoA1 algorithm are applicable in Baseline algorithm. Therefore, Baseline algorithm uses the pruning strategies too.

(1) Effect of the number of query keywords m

Fig. 8 shows the influence of m on the query time of TCoA1 algorithm and Baseline algorithm. We set $\alpha = 0.6$. According to Fig. 8, TCoA1 algorithm is faster than Baseline algorithm. It is because after finding the distance dominator $minobject$, TCoA1 algorithm directly obtains head of each heap except the heap containing $minobject$, and then combines with $minobject$ to get a feasible solution. However, Baseline algorithm needs to obtain all the objects in each list except the list containing $minobject$, and to combine with $minobject$ to get a large number of feasible solutions. Finally, we retain the feasible solution with the largest $score_1(q, S')$ to update the final result. The query time of both algorithms increases when m increases. This is because the total number of objects containing query keywords may increase when m increases. This may result in an increase in the total number of the distance dominators used. It can also be seen from Fig. 8 that TCoA1 algorithm are scalable.



(a) Query time (Oldenburg)

(b) Query time (GN)

Fig. 8. Query time versus m

(2) Effect of α

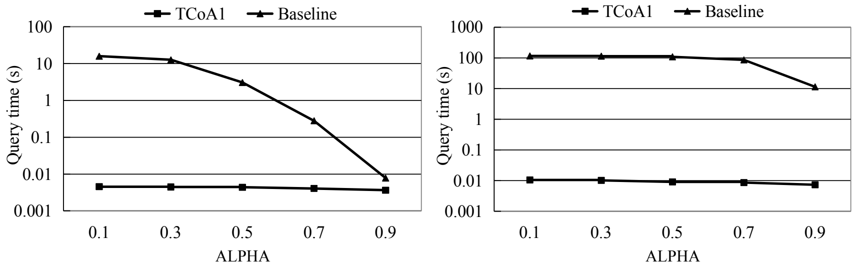
The α is a parameter used to adjust the distance and time. When α changes from 1 to 0, more weight is assigned to temporal relevance. When α is close to 1, TCoSKQ is more related to positional relevance.

Fig. 9 shows the influence of α on the query time of TCoA1 algorithm and Baseline algorithm. We set $m = 4$. According to Fig. 9, with α increasing, the query time of TCoA1 algorithm and Baseline algorithm decreases. This is because when α increases, the smaller $\max_{o \in S} d(q, o)$ is, the larger the $score_1(q, S)$ is. TCoSKQ prefer finding the result close to the query location. It is easier to find the result that satisfies the condition in a small range, so that few distance dominators are used.

7.2. TCoA1 algorithm and TCoA1NoPrune algorithm

In order to test the pruning effect on the total number of the distance dominators used (N_{du}), we remove a pruning strategy of TCoA1 algorithm (i.e., Theorem 2), then get TCoA1NoPrune algorithm. Next, we compare TCoA1 algorithm and TCoA1NoPrune algorithm on N_{du} and query time using the two datasets.

(1) Effect of m

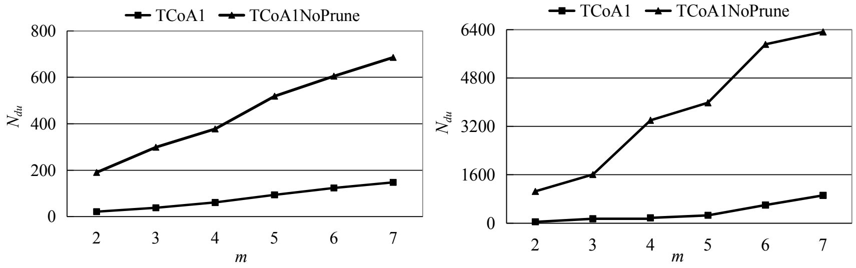


(a) Query time (Oldenburg) (b) Query time (GN)

Fig. 9. Query time versus α

Fig. 10 shows the influence of m on N_{du} in the query. We set $\alpha = 0.6$. According to Fig. 10, TCoA1 algorithm uses less total number of distance dominators than TCoA1NoPrune algorithm does. This is because TCoA1 algorithm uses pruning strategy to prune a large number of distance dominators. N_{du} in TCoA1NoPrune algorithm is the total number of objects containing the query keywords and the distance to q is not less than d_{LB}. With m increasing, N_{du} in TCoA1 algorithm and TCoA1NoPrune algorithm increases. This is because the total number of objects containing query keywords may increase when m increases.

Fig. 11 shows the influence of m on query time. We set $\alpha = 0.6$. According to Fig. 11, TCoA1 algorithm is faster than TCoA1NoPrune algorithm. As m increases, the query time of TCoA1 algorithm and TCoA1NoPrune algorithm increases. This is because the query time of the algorithms is related to N_{du}.

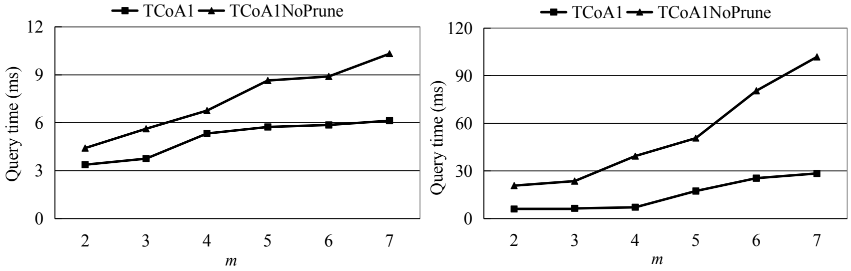


(a) N_{du} (Oldenburg) (b) N_{du} (GN)

Fig. 10. N_{du} versus m

(2) Effect of α

Fig. 12 shows the influence of α on N_{du} in the query. We set $m = 4$. According to Fig. 12, α has no impact on N_{du} in TCoA1NoPrune algorithm. This is because the distance dominator is not pruned in TCoA1NoPrune algorithm, and N_{du} is the total number of objects containing the query keywords and the distance to q is not less than d_{LB}.



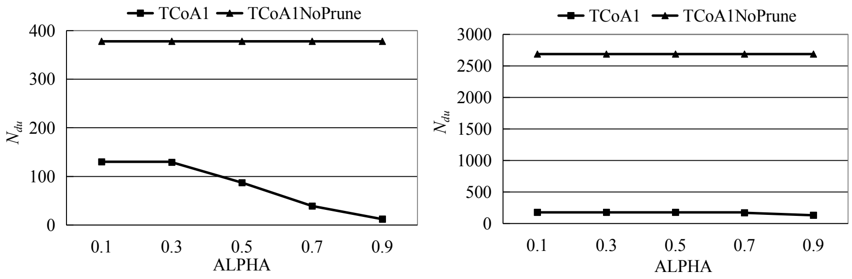
(a) Query time (Oldenburg)

(b) Query time (GN)

Fig. 11. Query time versus m

As α increases, N_{du} in TCoA1 algorithm decreases. This is because when α increases, TCoSKQ prefer positional relevance. It is easier to find the result that satisfies the condition in a small range, so that few distance dominators are used.

Fig. 13 shows the influence of α on query time. We set $m = 4$. As shown in Fig. 13, α has no impact on the query time of TCoA1NoPrune algorithm, and the query time in TCoA1 algorithm decreases with α increasing. This is because the query time of the algorithms is related to N_{du}.



(a) N_{du} (Oldenburg)

(b) N_{du} (GN)

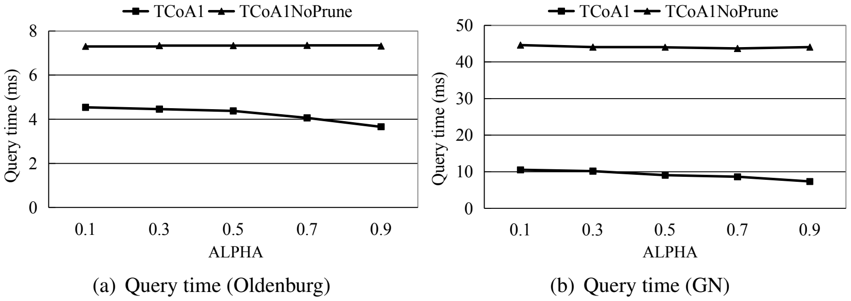
Fig. 12. N_{du} versus α

7.3. TCoA2 algorithm and TCoA2NoPrune algorithm

In order to test the pruning effect on the total number of center objects used (N_{co}), we remove the pruning strategy of TCoA2 algorithm (i.e., Theorem 3), then get TCoA2NoPrune algorithm. Next, we compare TCoA2 algorithm and TCoA2NoPrune algorithm on N_{co} and query time using the two datasets.

(1) Effect of m

Fig. 14 shows the influence of m on N_{co} in the query. We set $\alpha = 0.6$ and $\beta = 0.6$. According to Fig. 14, N_{co} in TCoA2 algorithm is less than TCoA2NoPrune algorithm.



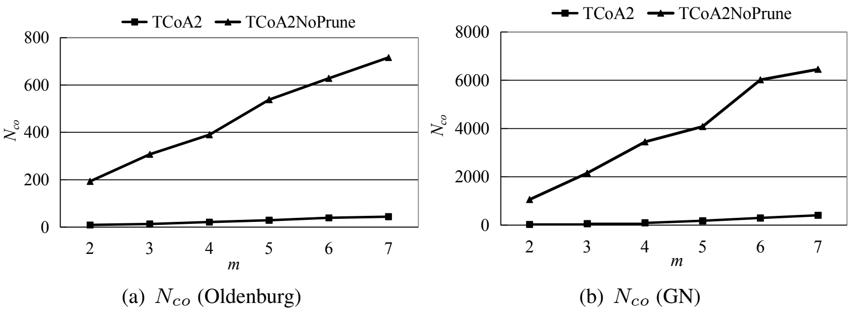
(a) Query time (Oldenburg)

(b) Query time (GN)

Fig. 13. Query time versus α

This is because TCoA2 algorithm uses pruning strategy to prune a large number of center objects. N_{co} in TCoA2NoPrune algorithm is the total number of objects containing the query keywords. When m increases, N_{co} in TCoA2 algorithm and TCoA2NoPrune algorithm increases. This is because the total number of objects containing query keywords may increase when m increases.

Fig. 15 shows the influence of m on query time. We set $\alpha = 0.6$ and $\beta = 0.6$. With m increasing, the query time of TCoA2 algorithm and TCoA2NoPrune algorithm increases, and TCoA2 algorithm is faster than TCoA2NoPrune algorithm. This is because the query time of the algorithms is related to N_{co}.



(a) N_{co} (Oldenburg)

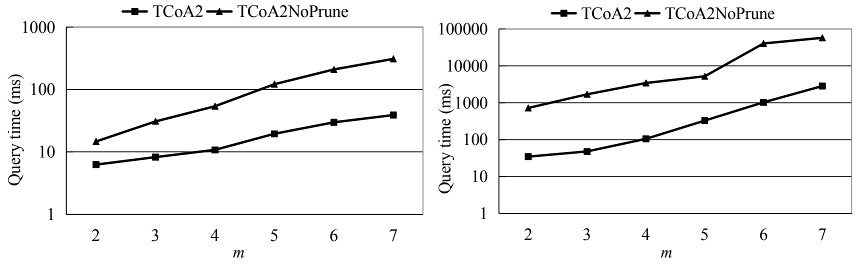
(b) N_{co} (GN)

Fig. 14. N_{co} versus m

(2) Effect of β

The β is a parameter used to adjust the distance and $score_1$. When β changes from 1 to 0, more weight is assigned to $score_1$. When β is close to 1, TCoSKQ is more related to positional relevance.

Fig. 16 shows the influence of β on N_{co} in the query. We set $m = 4$ and $\alpha = 0.6$. According to Fig. 16, β has no impact on N_{co} in TCoA2NoPrune algorithm. This is because there is no pruning strategy in TCoA2NoPrune algorithm, and N_{co} is the total number of objects containing the query keywords. As β increases, N_{co} in TCoA2 algo-

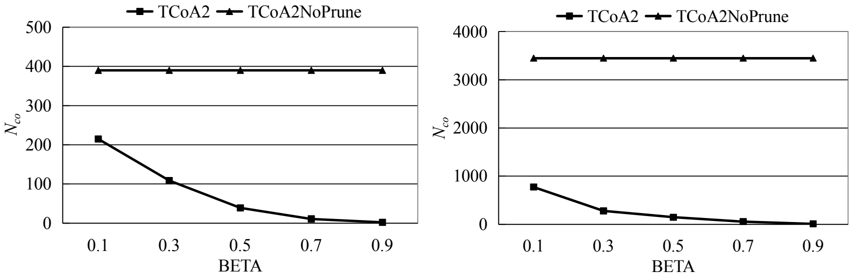


(a) Query time (Oldenburg)

(b) Query time (GN)

Fig. 15. Query time versus m

rithm decreases. This is because when β increases, TCoSKQ prefer positional relevance. It is easier to find the result that satisfies the condition in a small range, so that few center objects are used.



(a) N_{co} (Oldenburg)

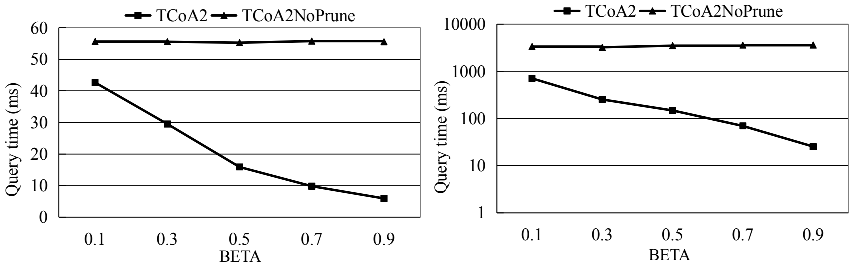
(b) N_{co} (GN)

Fig. 16. N_{co} versus β

Fig. 17 shows the influence of β on query time. We set $m = 4$ and $\alpha = 0.6$. As shown in Fig. 17, β has no impact on the query time of TCoA2NoPrune algorithm, and the query time in TCoA2 algorithm decreases with β increasing. This is because the query time of the algorithms is related to N_{co}.

8. Conclusions

This paper presents a new query, time-aware collective spatial keyword query (TCoSKQ). For different needs of users, we define two new evaluation functions, $score_1$ and $score_2$, and adopt the TR-tree index structure. For the evaluation function $score_1$, we propose pruning strategies for effectively pruning the number of the distance dominators used, and give TCoA1 algorithm for solving the query problem. For the evaluation function $score_2$, we propose effective pruning strategies to prune the number of the center objects used,



(a) Query time (Oldenburg)

(b) Query time (GN)

Fig. 17. Query time versus β

and give TCoA2 algorithm. Finally, the efficiency and scalability of the two algorithms are verified. In the future work, other evaluation functions could be proposed.

Acknowledgments. This work was supported by the Key Research and Development Program of Hebei Province, China (No. 18270307D).

References

1. Cao, X., Cong, G., Jensen, C.S., Ooi, B.C.: Collective spatial keyword querying. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. pp. 373–384 (2011)
2. Cary, A., Wolfson, O., Rishe, N.: Efficient and scalable method for processing top-k spatial boolean queries. In: Proceedings of the 22nd International Conference on Scientific and Statistical Database Management, SSDBM 2010, Heidelberg, Germany. pp. 87–95 (2010)
3. Chan, H.K.H., Long, C., Wong, R.C.W.: Inherent-cost aware collective spatial keyword queries. In: Proceedings of the 15th International Symposium on Advances in Spatial and Temporal Databases, SSTD 2017, Arlington, VA, USA. pp. 357–375 (2017)
4. Chan, H.K.H., Long, C., Wong, R.C.W.: On generalizing collective spatial keyword queries. IEEE Transactions on Knowledge and Data Engineering 30(9), 1712–1726 (2018)
5. Chen, G., Zhao, J., Gao, Y., Chen, L., Chen, R.: Time-aware boolean spatial keyword queries. IEEE Transactions on Knowledge and Data Engineering 29(11), 2601–2614 (2017)
6. Chen, Z., Zhao, T., Liu, W.: Time-aware spatial keyword cover query. Data & Knowledge Engineering 122, 81–100 (2019)
7. Chen, Z., Zhou, T., Liu, W.: Direction aware collective spatial keyword query. Journal of Chinese Computer Systems 35(5), 999–1004 (2014)
8. Felipe, I.D., Hristidis, V., Rishe, N.: Keyword search on spatial databases. In: Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, Cancún, Mexico. pp. 656–665 (2008)
9. Gao, Y., Zhao, J., Zheng, B., Chen, G.: Efficient collective spatial keyword query processing on road networks. IEEE Transactions on Intelligent Transportation Systems 17(2), 469–480 (2016)
10. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, Boston, Massachusetts, USA. pp. 47–57 (1984)

11. He, P., Xu, H., Zhao, X., Shen, Z.: Scalable collective spatial keyword query. In: 31st IEEE International Conference on Data Engineering Workshops, ICDE Workshops 2015, Seoul, South Korea. pp. 182–189. IEEE (2015)
12. Hjaltason, G.R., Samet, H.: Distance browsing in spatial databases. ACM Transactions on Database Systems 24(2), 265–318 (1999)
13. Joachims, T.: A statistical learning model of text classification for support vector machines. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2001, New Orleans, Louisiana, USA. pp. 128–136 (2001)
14. Li, Z., Lee, K.C., Zheng, B., Lee, W.C., Lee, D., Wang, X.: Ir-tree: An efficient index for geographic document search. IEEE Transactions on Knowledge and Data Engineering 23(4), 585–599 (2011)
15. Liu, W., Fu, Y., Chen, Z.: New collective query processing method based on spatial keyword. Journal of Chinese Computer Systems 34(8), 1831–1836 (2013)
16. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest neighbor queries. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, USA. pp. 71–79 (1995)
17. Su, S., Zhao, S., Cheng, X., Bi, R., Wang, J.: Group-based collective keyword querying in road networks. Information Processing Letters 118, 83–90 (2017)
18. Tao, Y., Sheng, C.: Fast nearest neighbor search with keywords. IEEE transactions on knowledge and data engineering 26(4), 878–888 (2014)
19. Vaid, S., Jones, C.B., Joho, H., Sanderson, M.: Spatio-textual indexing for geographical search on the web. In: Proceedings of the 9th International Symposium on Advances in Spatial and Temporal Databases, SSTD 2005, Angra dos Reis, Brazil. pp. 218–235 (2005)
20. Zhang, P., Lin, H., Yao, B., Lu, D.: Level-aware collective spatial keyword queries. Information Sciences 378, 194–214 (2017)
21. Zhao, J., Gao, Y., Chen, G., Chen, R.: Towards efficient framework for time-aware spatial keyword queries on road networks. ACM Transactions on Information Systems 36(3), 24:1–24:48 (2018)
22. Zhao, S., Cheng, X., Su, S., Shuang, K.: Popularity-aware collective keyword queries in road networks. GeoInformatica 21(3), 485–518 (2017)

Zijun Chen received the bachelor's degree from the Northeast Heavy Machinery Institute, China, the master's degree from Yanshan University, and the PhD degree from Fudan University in 2002, all in computer science. Since 1995, he has been with the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China, where he is currently a professor. His research interests include moving object databases and spatio-temporal databases.

Tingting Zhao received the bachelor's degree in computer science and technology from North China University of Science and Technology, China, in 2016. She received the master's degree in the School of Information Science and Engineering, Yanshan University, China, in 2019. Her research interest includes spatio-temporal databases.

Wenyuan Liu received the bachelor's and master's degrees from the Northeast Heavy Machinery Institute, China, and the PhD degree from the Harbin Institute of Technology in 2000, all in computer science. Since 1996, he has been with the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China, where he is currently a professor. His research interests include spatio-temporal databases, network sensing and mobile networks.

Received: January 31, 2020; Accepted: May 15, 2021.

Computer Science and Information Systems 18(3):1101–1138 https://doi.org/10.2298/CSIS200131014N

Conflict Resolution Using Relation Classification: High-Level Data Fusion in Data Integration

Zeinab Nakhaei[1], Ali Ahmadi[2], Arash Sharifi[1], and
Kambiz Badie[3]

[1] Department of Computer Engineering, Science and Research Branch,
Islamic Azad University, Tehran, Iran
{zeinab.nakhaei, a.sharifi}@srbiau.ac.ir
[2] Faculty of Computer Engineering, K. N. Toosi University of Technology,
Tehran, Iran
ahmadi@kntu.ac.ir
[3] Iran Telecommunication Research Center (ITRC),
Tehran, Iran
k_badie@itrc.ac.ir

Abstract. The aim of conflict resolution in data integration systems is to identify the true values from among different and conflicting claims about a single entity provided by different data sources. Most data fusion methods for resolving conflicts between entities are based on two estimated parameters: the truthfulness of data and the trustworthiness of sources. The relations between entities are however an additional source of information that can be used in conflict resolution. In this article, we seek to bridge the gap between two important broad areas, relation estimation and truth discovery, and to demonstrate that there is a natural synergistic relationship between machine learning and data fusion. Specifically, we use relational machine learning methods to estimate the relations between entities, and then use these relations to estimate the true value using some fusion functions. An evaluation of the results shows that our proposed approach outperforms existing conflict resolution techniques, especially where there are few reliable sources.

Keywords: conflict resolution, data fusion, relational machine learning, relation estimation, relation classification.

1. Introduction

The main challenge in a data integration system is *conflict resolution*. Conflicts occur at different levels, ranging from schema to value [1]. In this article, we deal mainly with the second of these. Conflict at the value level means that there are multiple sources describing the same real-world entity, providing different values for the same attribute of the entity. To resolve such conflicts, fusion techniques are used. In broad terms, data fusion (DF) is the process of combining multiple sources of data to achieve higher quality data than can be obtained from individual sources [2]. In data integration, the DF process is a combination of values that describe a similar entity from the real world, leading to one value which is closer to the real world. In this article, DF means the process and methods for achieving one accurate single value from multiple values.

In a conflict resolution problem, there are usually many claims about entities provided by a number of different sources. The basis for almost all current DF methods is voting based on two main assumptions: (1) that the claim provided by a reliable source is correct; and (2) that the source that provides the true value is reliable. Existing methods therefore attempt to estimate two parameters: the *truthfulness* of data and the *trustworthiness* of sources [3-7]. However, these methods prove inadequate in some cases. We analyze the causes of this in two respects: the number of reliable sources, and the long-tail phenomenon.

- The number of reliable sources: In some applications, there are few reliable sources, and incorrect information may be copied by multiple unreliable sources. For example, a website publishes fake news tendentiously, and this news is then republished in several weblogs and social networks.
- The long-tail phenomenon: This phenomenon occurs where information on entities is provided by very few sources, as is common in applications [8, 9]. In such cases, it is possible that there are some reliable sources, but these sources may not provide information about certain entities. As a result, the information about such entities is not sufficient to produce a correct value.

In both the above cases, further items of information are needed beyond the attributes of an entity and claims about these. As described in the survey carried out by Li et al. (2016) [10], most truth discovery methods assume that entities are independent, whereas in reality entities may have relations between them and may affect each other. For example, two people who are classmates at university are likely to have the same level of education. Our proposed approach seeks to exploit the additional information deriving from such relationships between entities, and thereby to achieve a higher level of data abstraction.

As discussed in an article by Snidaro et al. (2013) [11], evolving data sources require an entity of interest to be represented by a collection of distinct and complementary pieces of information at multiple levels of abstraction. At lower levels of abstraction, entities are described by low level data (such as information about data sources) and attributes. At higher levels, on the other hand, entities are described by their situation and relationship with respect to other entities: in other words, we are dealing with relations and patterns between entities.

In this article, we use the relations between entities in addition to the attributes of the latter. Drawing inferences about or predicting the relations between entities is one of the challenges in machine learning, as can be seen in problems like *link prediction* and *knowledge graph completion* [12,13]. The main challenge is how to devise a model that can reliably learn relations between new entities. Such models are often trained by supervised methods. These however require a large training dataset comprising both entities and the relations between them.

We need to mention that by relational data model we mean a set of relations in the form of triples (*subject*; *relation type*; *object*), where *subject* and *object* are an entity and *relation type* is a relationship between a *subject* and an *object*. A *relation schema* is a set of relation types. The triples in a relational data model are relation instances.

In order to model relational data in a conflict resolution problem, we need to deal with two basic challenges. First, there are no predefined relation types, and the data sets contain only the entities and values for their attributes. Second, while each row in the input data sets is related to only one entity and its attributes, there may be multiple differing values for each attribute claimed by different sources. To address the first problem, we define a relation schema based on the attributes involved. To deal with the second problem, we create a metadata set containing information about pairs of entities instead of looking at only one entity at a time. This enables us to predict the existence of a relationship between pairs of entities and find relation instances.

The key aim of our efforts is to ensure that all the relations are clearly and reliably defined. To achieve this, our study draws on a combination of two important and widely used areas: *truth discovery* and *relation estimation.* In summary, this article makes the following contributions to knowledge in this area:

1. We consider the problem of conflict resolution at a higher level of abstraction of data, and define new heuristics for using additional items of information about entities, namely the *relations* between them.
2. We define a relation schema based on the attributes of entities, and use this when there is no predefined relation between the entities in question.
3. We introduce a process for assessing the relation between two entities, employing a metadata set obtained from a primary small clean data set and our relation schema.
4. We bridge the gap between the two important broad areas of relation estimation and truth discovery, and demonstrate that there is a natural synergistic relationship between data integration and machine learning.
5. Finally, and most importantly, we demonstrate that using extra information in this way can improve the performance of fusion techniques, particularly in unreliable environments.

The rest of the article is organized as follows. In section 2, we review the existing literature in the two main areas of truth discovery and relation assessment. In section 3, we explain why we use relations to try to solve conflict resolution problem by illustrating how existing methods work that motivates our approach. In section 4, we define the problems surrounding conflict resolution. Section 5 describes our proposed approach in more detail, including the framework, algorithms and required formulations. Finally, the results of our experiments are analyzed in section 6.

2. Related Works

This article bridges the gap between two important areas: data fusion and relational machine learning. Our approach tries to use relational models to estimate relationships between entities, and then to apply this model in order to improve the performance of the data fusion method. In other words, our study is located at the intersection of these two areas. We therefore review in this section articles and existing methods in both areas, beginning with data fusion.

2.1. Data Fusion for Conflict Resolution

The first study that precisely defined the goals of data fusion for the purposes of conflict resolution was provided by Bleiholder and Neumann (2009) [14]. Their survey introduced the problem of data fusion in the larger context of data integration, where data fusion is the final step in a data integration process, schemata have been matched, and duplicate records identified. Data fusion involves merging these duplicate records into a single record, while at the same time resolving data conflicts.

There are two main kinds of data fusion that can be performed at data abstraction levels: low level data fusion and high-level data fusion.

Low level data fusion. All of the methods in the category of low-level data fusion estimate two parameters (the truthfulness of data and the trustworthiness of sources). These methods can be divided into three categories based on the model used to estimate these parameters, namely: iterative models, graphical models and optimizing models.

Iterative model: Early methods of data fusion attempted to estimate the correctness of claims and the reliability of sources, and to determine each of these iteratively. The first such method was truth finder by Yin et al. (2008) [4]. This uses Bayesian analysis, under which the correctness of each claim is calculated as the product of the degrees of reliability of its sources. Truth finder has gained considerable popularity, with a number of methods emerging based on its algorithm. These are reviewed and compared in an article by Li et al. (2012) [15].

Graphical model: There is also a substantial body of work on data fusion that uses the graphical model [3] in order to model the relationship between data correctness and source accuracy. In the proposed method of Zhao et al. (2012) [3], claims are modeled as random variables which depend on the truth of the facts they refer to as well as on the quality of their sources. With the actual claim data, it is then possible to go back and infer the facts most likely to be true and the quality of the relevant sources. More recently, SLiMFast was proposed by Rekadsinas et al. (2017) [16] as a discriminative model that also enables other features of data sources (such as, update date, number of citations) to be taken into account for fusion purposes; where there is sufficient labeled data, SLiMFast uses empirical risk minimization (ERM).

Optimization model: Finally, some further methods model the problem using an optimization framework, where truths and source reliability are defined as two sets of unknown variables like Meng et al. (2015) [6] and Yin et al. (2011) [17].

High level data fusion. Some research goes beyond the above and seeks to estimate additional parameters, including the *correlation between sources* [18] and the *relation between objects* [6, 7 and 17]. The latter relations may be temporal or spatial. These relations are partially addressed by Meng et al. (2015) [6]. However, this work is based on the key assumption that a correlation graph already exists, whereas in our approach the relations between entities are inferred by learning methods. Another study by Yin (2011) [17] features a semi-supervised approach that seeks to find true values with the help of ground truth data. Claims are connected to each other and thus form a graph. Both this work and another similar piece of work by Liu et al. (2018) [7] rely on the similarity of claims and consider this as the relationship between them. Ye et al. (2019) [9] meanwhile propose an algorithm called PatternFinder, that jointly and iteratively learns four variables, i.e.: the latent groups of entities that match to a particular regularity; the group-level representatives that indicate the true value for the attributes of each entity in each latent group; the attribute weights; and the source weights. They also propose an optimized grouping strategy to enhance the efficiency of this approach.

It is important to note that these methods focus only on the apparent characteristics of entities, and use a similarity function to draw inferences about relations. To address this limitation, our previous work (2019) [19] proposes a method for estimating relationships between entities based on clustering them in an embedding space instead of a feature space. In this approach, before clustering, the data points are mapped into an embedding space and are enhanced by creating more informative features. The true values are then determined by defining a confidence score based on the distance between the data and the centers of clusters in the embedding space. Our previous work differs from the work proposed in this article in two main respects. First, in our previous work, we assume that the entities in the same cluster are related, whereas in this article, we create a metadata set and use machine learning methods to infer some rules about the relationships between entities. Second, in order to resolve conflicts between related entities, in our previous article we use the distance of entities from the centroids of clusters. In this article, on the other hand, we define some fusion functions and use these to calculate the confidence score for each entity.

In summary, the methods based on relations between entities can be divided into two groups: those that are aware of relations between entities beforehand, and those that use similarities as relationships between entities. In our current approach, in contrast, there are no prior assumptions about relations, nor are there defined types of relations. Instead, the relations between entities are derived by mining some rules deduced by machine learning methods.

2.2. Relational Machine Learning

In this article, we use relational machine learning to derive relations between entities. Relational machine learning covers a number of methods for the statistical analysis of relational, or graph-structured, data. Nickel et al. (2016) [20] provide a review of how such statistical models can be trained on large knowledge graphs, and then used to predict new facts about the world (equivalent to predicting new edges on the graph). There are two main kinds of statistical relational models that try to predict new relations between entities. The first is based on latent feature models, such as the latent class model [21, 22], the distance model [23], and embedding nets [12, 24, 25 and 26]. The second type of model involves mining observable patterns in graphs.

We look first at three common types of latent feature models.

Latent class model: In this model, each entity is assumed to belong to an unobserved latent class, and a probability distribution describes the relationships between each pair of classes. Kemp et al. (2004) [21] define a generative model in which a particular relationship is obtained between a pair of entities such that their probability depends on the class of each entity. In their article, Airoldi et al. (2005) [22] propose a Bayesian model that uses a hierarchy of probabilistic assumptions about the way entities interact with one another in order to learn latent groups, their typical interaction patterns, and the degree of membership of entities to groups.

Distance Model: This model is based on the idea that entities are likely to be in a relationship if their latent representations are close in terms of distance. Hoff (2008) [23] proposes a model based on the idea of eigenvalue decomposition that represents the relationship between two nodes as the weighted inner-product of node-specific vectors of latent characteristics. Such a model is able to represent datasets with homophily patterns. Homophily provides an explanation for data patterns often seen in social networks, such as transitivity (“a friend of a friend is a friend”), balance (“the enemy of my friend is an enemy”), and the existence of cohesive subgroups of nodes. Note that we use homophily as one of the heuristics in our approach.

Embedding Nets: Recent studies have shown that neural-based representation learning methods are scalable, and are effective at encoding relational knowledge with low dimensional representations of both entities and relations, which means that they can be used to extract unknown relational facts. One of the early works in this area by Bordes et al. (2011) [26] proposes a model in which, for any given type of relation, there is a specific similarity measure that captures the relation in question between entities. This model has the architecture of a neural network. In order to embed entities effectively in this model, it is necessary to define a training objective that learns relationships. Bordes et al. (2013) [25] meanwhile introduce TransE, an energy-based model for learning low-dimensional embedding of entities. In TransE, relationships are represented as translations in the embedding space. Another work by Lin et al. (2015) [12] presents TransR, which embeds entities and relations in a distinct entity space and relation space, and learns to embed better via translations between projected entities.

The problem with all the above latent feature methods is the existence of a large number of entities and relations between them, whereas in problems of conflict resolution there are often no predetermined relation types.

The second main type of statistical relational model – based on mining observable patterns in graphs – seeks to address this problem. This method works on the observed variables of a knowledge graph, extracting rules via mining methods and then using these extracted rules to infer new links. This is the approach adopted in our study: we try to mine some rules for predicting relations. The challenge here is that, in conflict resolution problems, there is usually no training relational data set. We therefore need to create such a data set within our modeling framework. In practice, we use a small clean entity-attribute dataset in order to generate a sufficient metadata set. The proposed model can then mine observable patterns over the metadata set and predict the specific relations between unseen entities.

3. Motivation and Overview

Problems of conflict resolution generally involve dealing with often conflicting claims about an entity. The task of a conflict resolver (or truth finder) is to determine the correctness of each claim. Depending on the level of data abstraction, a conflict resolution problem may engage with several concepts, including entity (or object), attribute, data source, claim, and truth value. The main approach used in most current methods is based on estimating the reliability of data sources. As mentioned earlier, two heuristics are used in this approach: that the claim provided by a reliable source is likely to be correct; and that a source that provides true value will be reliable.

Let us look at an example that illustrates how existing methods work, and what motivates our approach.

Example 1: Suppose the entity about which there are claims from multiple sources is *a person*. Table 1 shows part of the dataset. A data integration system gathers values about the attributes of entities from several sources, that we shall call S_1 to S_3. Each claim vector $\boldsymbol{c}_i^j$ specifies person i described by six attributes – *name*, *workClass*, *education*, *age* and *outcome* – provided byS_j. In order to simplify notation, each claim vector is considered as an observation o. Because of the varying levels of reliability of sources, different values may be published for the attributes of a person. In Table 1, all incorrect values are marked in bold, with the correct values written in brackets after the incorrect ones.

Table 1. Part of dataset including entity, attribute and claim

Source	Observation	Name	WorkClass	Education	Age	Outcome
S_1	o_1	John	Private	Bachelors	32	<=50K
	o_2	Mary	Local-gov	Masters	41	>50K
S_2	o_3	John	Private	Bachelors	32	<=50K
	o_4	Bob	**Local-gov** (Private)	Bachelors	30	<=50K
S_3	o_5	Alice	Local-gov	**Bachelors** (Masters)	45	>50K
	o_6	Mary	**Private** (Local-gov)	Masters	41	>50K

In this example, we can see that two observationso_1 and o_3 describe a person named John, and similarly two observations o_2 and o_6 describe Mary. In contrast, only source S_2 provides data about Bob and only source S_3 provides data about Alice. Current methods work as follows. SinceS_1 and S_2 both provide the same information about John (o_1 ando_3), the trustworthiness of both sources is increased. This also increases the reliability of the information about Bob provided by source S_2 (observation o_4), even although this is the only source of information about Bob. However, in fact source S_2 is not reliable (the workClass information about Bob is incorrect). Similarly, in the absence of other information about Alice, observation o_5 is considered true information and the reliability of S_3 is somewhat increased. But in reality, source S_3 is an unreliable source (it provides two items of erroneous information).The upshot is that current methods will consider observationso_1,o_2,o_3,o_4 ando_5 all to be true; will fail to find the true values ofo_4 ando_5; and will moreover inaccurately estimate the reliability of source S_2 in particular.

The above example indicates clearly how current methods become less effective when they are faced with the long-tail phenomenon and have few reliable sources at the entity level. In such cases, more items of information are needed. Looking at the relations between two entities can provide additional information and so help to describe the entities more effectively than can be achieved at the entity level. When a relation is established between two entities, the value of an attribute belonging to one entity can identify, or at least help to identify, the value of the analogous attribute belonging to the other related entity. Take for example the relation *same age* between two persons: if we know the age of one person, we can determine the age of the other person. Similarly, the presence of a *classmate* relation between two persons can help us to identify the educational level of the two persons in question.

Relation-Based Conflict Resolver (RelBCR): Based on the above observations, we propose a relation-based method for conflict resolution. We investigate the use of further information that can be extracted from a higher level of data abstraction. Such additional information can be inferred by machine learning methods, in the form of a set of *rules* that describe the *relations* between entities. In this context, a relation is a triple that contains two entities and the type of relationship between them; a rule is a set of attributes and their values as an antecedent; and a triple (*subject*; *relation type*; *object*) as a consequent.

Example 2: Consider two relation types $r_1 =< same_{education} >$, $r_2 =< same_{workClass} >$ and the following two rules inferred about the existence of these relations between two entities e_1 ande_2.

Rule 1**:**workClass(e_1) = workClass(e_2), race(e_1) = race(e_2), outcome(e_1) = outcome(e_2) → ($e_1, same_{education}$,e_2).

Rule 2: education(e_1) = education(e_2), outcome(e_1) = outcome(e_2) → ($e_1, same_{workClass}$,e_2).

Applying Rule 1 to Table 1 above, the relation type$< same_{education} >$should exist between Alice and Mary. In other words, the education level of Alice should be equal to that of Mary. The value "Bachelors" for the attribute *education* of Alice should therefore be corrected to "Masters". Similarly, the value "Local-gov" should be corrected to "Private" for Bob based on the existence of relation type $< same_{workClass} >$ between Bob and John that is inferred based on Rule 2.

In sum, relations can be very informative and can help in estimating the correctness of values claimed about attributes. With proper and reliable rules, we can extract relations and which can then be used in the DF process.

However, discovering and using proper rules raises some challenges:

- In relational machine learning applications like link prediction or knowledge completion, relation types are predefined. For example, in social networks the relation type "friendship" is defined and entities are described by both attributes and relations. However, in a conflict resolution problem, the initial data is described only by attributes. This begs the questions: what are the relation types, and how can we define them?
- We use relations as additional information to increase the accuracy of the DF process. However, although there are a number of methods for estimating relations between entities used in applications like link prediction and ontology completion [12, 13, 24], in such problems there is usually a large training dataset of entities and relations. The challenge is how to draw inferences about relations when there is no training set containing entities and relations.
- After finding relations between entities, the issue is how to use these relations to estimate the correct values for each attribute.

In this article, we address these challenges. To meet the first challenge, we define a relation schema based on attributes. A relation schema is a set containing relation types in the form of $< same_{attribute} >$ and$< bigger_{attribute} >$. This means that it is always possible to define at least one relation type for each attribute. For example, for the attribute age, a relation type$< same_{age} >$ can be defined. Details related to the definition of relation types are given in section 4, Definition 1. For the second challenge (drawing inferences without a training set), we create metadata sets containing attributes of a pair of entities and one binary attribute which determines whether there is a relationship between the pair of entities. We next apply learning methods like classification and association rule mining across these metadata sets. Such classifiers or association rule miners serve as inference engines that can be used to identify appropriate rules about relations. We can then decide about new pairs of entities, and the relations between them, through these inferred rules. Finally, for the third challenge, we define some fusion functions and use these to select the correct values from among multiple values about entities. In summary, our approach uses *relations* as a new concept in conflict resolution problems at a high level of fusion.

4. Problem Definition

In this section, we first define concepts in the DF process for conflict resolution. We then define problems of particular interest in this article.

The problem of conflict resolution involves a range of general concepts. An *entity* is a real-world object of interest, like a person, book or film. An attribute is a feature of an entity that describes it at the entity level, such as the name, age, gender and race of a person. A *data source* is a resource that provides the values of attributes: for example, websites. These values may be correct or incorrect, and so are called *claims.*

Suppose there are N_s data sources providing claims about the attributes of entities. Let $O = \{e_1, \dots, e_{N_o}\}$be the set of all $N_o$entities and let $A = \{att_1, \dots, att_M\}$ be the set of all M attributes. Each attribute can be continuous or categorical. So let $AT = \{t_1, \dots, t_M\}$be the set of attribute types. For the j-th attribute type, t_j = 1 if att_j is categorical, and t_j = 2 if att_j is continuous. The claim about j-th attribute of the i-th entity provided by the k-th source is denoted as c_{ij}^k.All claims are collected in a $\{N_o \times M \times N_s\}$ third-order tensor$\boldsymbol{C} = \{c_{ij}^k\}$. We denote the k-th frontal slice of the tensor $\boldsymbol{C}$ by $\boldsymbol{C}_k$(which is a matrix of size $\{N_o \times M\}$), representing all of the claims provided by the k-th data source. The claims about the i-th entity provided by the k-th data source are a vector denoted by $\boldsymbol{c}_i^k = \{v_j\}_{j=1\dots M}$, where $v_j \in D_j$ is the value of the attribute subject to att_j in the domain attributeD_j. A relation is in the form of triple (*subject*; *relation type*; *object)*,where *subject* and *object* are an entity and *relation type*is a relationship between a *subject* and an *object*.

Definition 1 (relation schema): Given the set of attributes A and attribute type AT, we can define a *similarity relation type* according to both continuous and categorical attributes. For continuous attributes, a *comparative relation type* can also be defined. In this article, we use only one relation type for the categorical attribute att, $< same_{att} >$, and two relation types for the continuous attributes att,$< bigger_{att} >$ and $< same_{att} >$. Note that, as a general rule, the number of relation types can be more depending on the given data set and the application. A collection of these relation types forms the relation schema Rel_Sc. The number of defined relation types in Rel_Sc is$N_r = \sum_{j=1}^{M} t_j$,and the k-th relation type in Rel_Sc is denoted by r_k.

Example 3: Let us take the attribute set $A = \{age, education, race, outcome\}$ and $AT = \{2,1,1,2\}$. If for instance $att_1 = age$ and $t_1 = 2$, this attribute is a continuous attribute and two relation types$< same_{age} >$ and $< bigger_{age} >$ can be defined. The relation schema is $Rel_Sc = \{< same_{age} >, < bigger_{age} >, < same_{education} >, < same_{race} >, < same_{outcome} >, < bigger_{outcome} >\}$and total number of defined relation types is six.

Definition 2 (metadata set): Given a clean dataset, including entities $O = \{e_1, \dots, e_{N_o}\}$ and true values about each attribute, for each relation type r in the relation schema, a metadata set is created. Note that, each relation type is constructed based on an attribute. We indicate these attributes by k. Each row in the metadata set is a pair of entities (e_i, e_j) and the columns are all attributes of$e_i$ande_j except k. The last column is a binary attribute that indicates the existence of relation $(e_i; r; e_j)$. When we have clean data on N_o entities with M attributes, the number of instances in the metadata set is $N_o \times N_o$, and the number of columns is $2 \times (M-1)+1$.

Example 4: Let the set of entities be $O = \{John, Mary, Bob\}$ and the attribute set be $A = \{age, job, marital\}$. Given relation type $r =< same_{age} >$, a metadata MD set is created. The rows of MD be $(John, Mary)$, $(John, Bob)$,$(Mary, Bob)$ and the columns are lhd_job, rhd_job, $lhd\_marital$, $rhd\_marital$ and $same_{age}$. The term of lhd means left-hand entity and rhd means right-hand entity. For example, for the pair $(John, Mary)$, the values of columns are John's job, Mary's job, John's marital status, Mary's marital status and 1 if John and Mary are the same-aged and 0 otherwise.

In section 5.2, Example 7 illustrates the details of metadata creation process.

Definition 3 (Relation tensor): Given a set of entities O, and a set of relation types in relation schema Rel_Sc, all possible triples in $O \times O \times Rel\_Sc$ can be grouped naturally in a third-order tensor$\mathcal{RT} \in \{0,1\}^{N_o \times N_o \times N_r}$, whose entries are set such that

$$\mathcal{RT}_{ijk} = \begin{cases} 1 & , if\ relation\ type\ r_k\ \ exists\ between\ e_i, e_j \\ \quad 0 & , otherwise \end{cases}$$

The k-th frontal slice of tensor $\mathcal{RT}$denoted by$\boldsymbol{RT}_k$ is a matrix $\{N_o \times N_o\}$, that indicates the relation instances of the k-th relation type in Rel_Sc.

Example 5: Let the set of entities be $O = \{John, Mary, Bob, Alice\}$. Given the relation schema in Example 3, the third relation type $<same_{education}\ >$,

$$\boldsymbol{RT}_3 = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

$\boldsymbol{RT}_3$ shows that both John and Bob on the one hand, and Mary and Alice on the other hand, have the same education level.

Definition 4 (Confidence tensor): Let $\boldsymbol{C} = \{c_{ij}^k\}$ be the claims about the j-th attribute of the i-th entity provided by the k-th source. Confidence score of the claims are indexed by a third-order tensor$\mathcal{CT} \in [0,1]^{N_o \times M \times N_s}$,such that for the j-th attributeof the i-th entity$\sum_{k=1}^{N_s} \mathcal{CT}[i][j][k] = 1$. A higher confidence score indicates that the claim is closer to the real world and the probability of its correctness is high.

Example 6: In Table 1, confidence score for the value "Bachelors" about attribute *education* of Alice that is provided by S_3 must be few because it is an incorrect claim.

Problem definition: Given a collection of claims $\boldsymbol{C}$ about a set of entities O from N_s sources, we attempt to accurately infer the relation tensor $\mathcal{RT}$and confidence tensor $\mathcal{CT}$such that correct values have higher confidence score than other incorrect claims based on their relations.

5. Methodology

The main purpose of our proposed RelBCR is to improve the performance of DF using relations between entities. This approach contains two main parts: estimating relations by calculating $\mathcal{RT}$ and truth finding by calculating$\mathcal{CT}$.This section of the article first gives a broad perspective of our approach by introducing a framework in section 5.1.

Thereafter, in section 5.2, we talk in more detail about relation extraction, and explain the assumptions and requirements needed to infer relations. Finally, we discuss how to compute a confidence score based on the relations.

5.1. Framework

In conflict resolution problems where there is insufficient data at the entity level, we can gain additional information by drawing inferences from relations between entities, and using such additional information at a higher level to select true values. So, in RelBCR, there are two main parts. We have called the first part, drawing inferences about the existence of relations, the G-model. The second part, called the F-model, obtains related entities as an input and then calculates the accuracy of claims about entities.

Figure 1 contains illustrations of the F-model and G-model of our RelBCR.

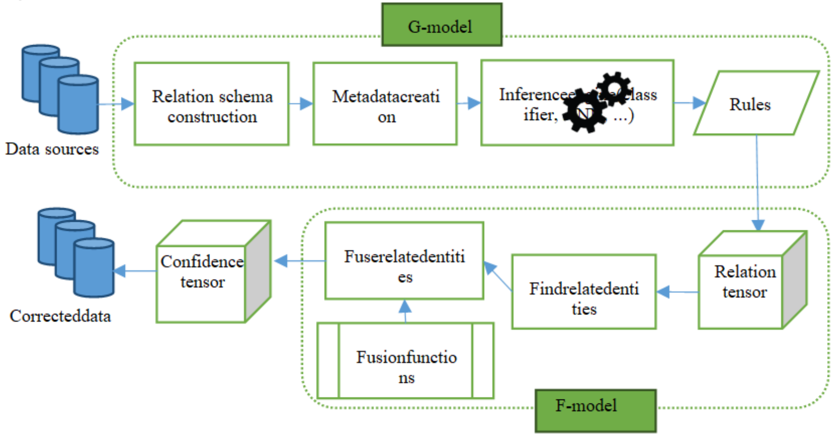


Fig. 1. Framework of proposed approach RelBCR

Data sources provide some claims about attributes of entities. All claims are collected to the tensor $\boldsymbol{C}$. The G-model contains three modules. Below, we explain the meaning of each module and the inputs and outputs involved.

Relation schema construction: According to Definition 1 given earlier, the task of this module is constructing relation types. The inputs for this module are a subset of attribute set A contains m attributes and related attribute type set AT, and the output is relation schema Rel_Sc. The relation schema is created as follows:

- For each attribute att create one relation in the form of$< same_{att} >$.
- For each continuous attribute att create one relation in the form of $< bigger_{att} >$.

The relation schema construction method is shown in Algorithm 1.

Algorithm 1: m is the number of attributes. The relation schema is defined as an array of the size of $\sum_{j=1}^{m} t_j$, where t_j indicates the type of attribute j. If the attribute is continuous, t_j is equal to two, indicating that two relation types must be added to the relation schema. If on the other hand the attribute is categorical, t_j is equal to one. For

ease of understanding relation types, these are added to the array Rel_Sc in the forms of $< bigger_i >$ and $< same_i >$, with subscript i for the$i$-th attribute.

Time complexity: In Algorithm 1, first the number of attributes is assigned to m. At most two relation types are added to the relation schema. So, the time complexity is $O(m)$.

Algorithm 1: Relation Schema Construction

```
Input: Attribute set $A = \{att_1, att_2, \ldots, att_m\}$ and attribute type set $AT = \{t_1, \ldots, t_m\}$.
Output:$Rel\_Sc$ ,the array in the size of$\sum_{j=1}^{m} t_j$.
$m \leftarrow$ length($A$)
$r \leftarrow 1$ // counter for relation types
for $i \leftarrow 1$ to $m$do{
  if$AT[i] = 2$ then{// attribute $i$ is continuous
    $Rel\_Sc[r] \leftarrow < bigger_i >$
    $r \leftarrow r + 1$
  }
  $Rel\_Sc[r] \leftarrow < same_i >$
  $r \leftarrow r + 1$
}
return$Rel\_Sc$
```

Metadata creation: To assess relations between entities, we create a metadata set as a training set, to be used as a training relation classifier. This training dataset needs to include sufficient negative and positive instances for each relation type. For this reason, we need a clean dataset, including entities and true values about each attribute. On the face of it, this may seem in contradiction with the main purpose of this article, which is to find true values for the attributes of entities. However, this dataset serves to provide *ground truth* data which, even on a very small scale, can greatly help us to create an appropriate metadata set (see section 6.6). In section 5.2, we explain the metadata creation process in more detail.

Inference engine: The input for this module is metadata, while the outputs are rules that can be used to indicate the existence of relations between two entities. The inference engine can be a learning method, such as classification, clustering or association rule mining. The types of rules about relations deduced by the inference engine depend on the learning method used. For example, if we use association rule mining, we will obtain association rules.

We can then decide about new entities, and the relations between them, through what we call a relation tensor $\mathcal{RT}$ – which is an output of the G-model and an input for the F-model. The F-model has two modules, described below.

Find related entities: Using $\mathcal{RT}$ for each entity and each relation type, related entities can be found. One entity can be in a relation with several entities. At the same time, there can be multiple relations for each entity. Because each relation is defined based on a specific attribute, this relation is used to compute the confidence score of claims related to that attribute.

Fuse related entities: The confidence scores for each claim for all of the entities are then calculated using the *fusion functions* set out in Table 4. The output of this module is confidence tensor $\mathcal{CT}$.

After calculating $\mathcal{CT}$, we select the value with the highest confidence score as the correct value.

5.2. RelBCRAlgorithm

Within the framework of our proposed RelBCR, there are two main phases for calculating $\mathcal{RT}$and $\mathcal{CT}$– the outputs of the G-model and F-model respectively. In this section, the processes and algorithms of these phases are explained and the time complexity in each phase is analyzed.

G-model: relation assessment phase. In this article, each relation type is denoted by $r =< same_{att}$ >or $< bigger_{att}$ >. The entities and relations between them are represented by a third-order tensor, with each entry showing the existence of a relation between two entities. Using tensor representation for relations makes it relatively easy to obtain additional information by tensor manipulation. For example, the k-th*frontal slice* of a tensor $\mathcal{RT}$ of size $N_o \times N_o \times N_r$ representing a relation is a matrix of size $N_o \times N_o$, which represents the existence of a k-th relation between the entities in question.

Figure 2 shows a schematic image of tensor $\mathcal{RT}$.

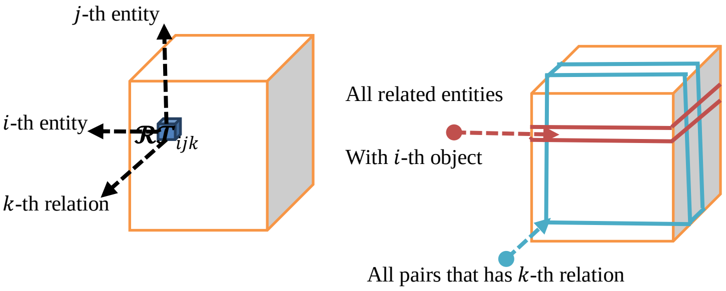


Fig. 2. Tensor representation of relations (left), frontal and horizontal slices of relation tensor (right)

Instances of a relation schema are gained by learning methods like classification. One such method, which we use, is known as triple classification. This seeks to judge whether a given triple (subject; relation type; object) is correct or not. We use metadata as a training set of classifiers, produced using a primary entity-attribute data set.

Example 7: Table 2 is part of an adult data set, in which each row is related to *one entity* that has several attributes like age, sex, education, and so on. This data set is considered as ground truth data. For each relation type in the relation schema, one metadata set is created. For a relation type in the form of <$same_{att}$> the entity orders is not important, but for a relation type in the form of <$bigger_{att}$> two orders $(e_1 e_2)$and $(e_2 e_1)$ are different from each other. Take for example the relation type $< same_{workClass}$ >. A metadata set is created in which each row is related to one pair of entities, and includes the attributes of both the subject (left-hand entity) and object (right-hand entity), except $workClass$. Thus, the class label becomes binary: if the attribute of $workClass$ is the same for both entities, the class label is 1; otherwise, it is 0. Table 3 shows part of the metadata set produced using Table 2.

We can later run all the classifiers or clustering methods over the new metadata set and thus predict the existence of relation type $< same_{workClass}$ >for new pairs of entities.

Table 2. Part of original entity-attribute data (adult data set) as ground truth

Entity	Sex	Education	Age	WorkClass
$\boldsymbol{e_1}$	Male	Bachelor	32	Private
$\boldsymbol{e_2}$	Male	Hs-grad	47	Private
$\boldsymbol{e_3}$	Female	Masters	35	Exec-managerial
$\boldsymbol{e_4}$	Male	Hs-grad	52	Private

Table 3. Example of metadata related to $< same_{workClass} >$ relation

Entity Pair	lhd_sex	lhd_edu	…	rhd_sex	rhd_edu	…	$\boldsymbol{same_{workClass}}$
$\boldsymbol{e_1 e_2}$	Male	Bachelor		Male	Hs-grad		1
$\boldsymbol{e_1 e_3}$	Male	Bachelor		Female	Masters		0
$\boldsymbol{e_1 e_4}$	Male	Bachelor		Male	Hs-grad		1
$\boldsymbol{e_2 e_1}$	Male	Hs-grad		Male	Bachelor		1
$\boldsymbol{e_2 e_3}$	Male	Hs-grad		Female	Masters		0
$\boldsymbol{e_2 e_4}$	Male	Hs-grad		Male	Hs-grad		1
$\boldsymbol{e_3 e_1}$	Female	Masters		Male	Bachelor		0
…							

The metadata creation module is summarized in Algorithm 2.

Algorithm 2: The number of entities in ground truth is n and the number of columns (attributes) is m. In the metadata set we consider all the pairs of an entity. So, if we have n^2 pairs of an entity, the row size of the metadata set is similarlyn^2. The columns of the metadata set contain attributes of both entities of each pair, except attribute k which represents the corresponding attribute of the relation; and one attribute as a class that indicates the existence of$k$-th relation type between these entities. The column size is therefore $(m-1)+(m-1)+1 = 2 \times m - 1$. The time complexity for metadata creation is thus $O(mn^2)$. Note that for the k-th relation type, we call the metadata creation algorithm by two inputs, the ground truth data set D and the index of attribute k.

Algorithm 2: Metadata Creation

```
Input: ground truth data set $D_{n \times m}$, index of attribute $k$ .
Output: metadata set $MD_{n1 \times n2}$
$n1 \leftarrow n \times n$ // number of rows in metadata set
$n2 \leftarrow 2 \times m - 1$ // number of columns in metadata set
$inx \leftarrow 0$
for$i \leftarrow$ 1 to $n$do{
  for $j \leftarrow$ 1 to $n$do{
    $inx \leftarrow inx + 1$
    $MD[inx][1..m-1] \leftarrow D[i][1..k-1, k+1..m]$
      //all attribute values of the first entity except $k$
    $MD[inx][m..2 \times m - 2] \leftarrow D[j][1..k-1, k+1..m]$
      //all attributes of the second entity except $k$
    if ($D[i][k] = D[j][k]$) then$MD[inx][2 \times m - 1] \leftarrow 1$
      else$MD[inx][2 \times m - 1] \leftarrow 0$
  }
}
return$MD$
```

After the creation of metadata for each relation type, these data sets are used as training sets for a classifier. Running the training classifier for each relation type allows us to infer models, which can then be used to construct the relation tensor $\boldsymbol{\mathcal{RT}}$. The k-th frontal slice of the relation tensor is populated for the k-th relation type and for each pair of entities, thus showing the relation instances for the k-th relation types. A complete illustrative example is provided at the end of this section.

F-model: fusion phase. In this section, we explain the process of calculating confidence tensor $\boldsymbol{\mathcal{CT}}$, and introduce fusion functions used to infer true values. The claims provided by multi-sources are also represented by a third-order tensor, with each entry showing the existence of a certain claim about the given attribute of an entity. The goal of the F-model is to estimate the truthfulness of each claim when it is indexed by the third-order tensor$\boldsymbol{\mathcal{CT}}$**.**

Let $\boldsymbol{C} = \left\{c_{ij}^k\right\}$be the set of all claims about the j-th attributeof the i-th entity provided by the k-th source. We calculate the confidence score of claims as follows. Let relation types about attribute $j$be in the set $Rel^j = \{r_n\}_{n=1,2}$. Note that, for some attributes this set contains only one relation. For such attributes the first sigma in equation (1) is eliminated. Using$\boldsymbol{\mathcal{RT}}$**,** entities related to the entity i are recognized and added to the set $RO^i = \{e_{i'}\}$, such that $\boldsymbol{\mathcal{RT}}_{ii'n} = 1$. Based on the claims about i' provided by the set of sources $DS = \{S_{k'}\}$, we calculate the confidence score using the following equations:

$$\boldsymbol{\mathcal{CT}}[i][j][k] = \frac{1}{Z}\sum_{n=1,2}\sum_{i'\in RO^i}\sum_{k'\in DS} F\left(c_{ij}^k, c_{i'j}^{k'}, r_n\right) \tag{1}$$

where F is a fusion function and Z is a normalization factor that is:

$$Z = \sum_k \sum_n \sum_{i'} \sum_{k'} F\left(c_{ij}^k, c_{i'j}^{k'}, r_n\right), \tag{2}$$

where k is the index of sources provided claims about entity i.

A variety of different fusion functions are possible, depending on the nature of relation type r_n. This function can be a similarity function if r_n is a relation in the form of $< same_{att} >$ or $< bigger_{att} >$, with function F equal to a simple subtraction function. We define three types of fusion function F, which are listed in Table 4.

Table 4. Fusion functions

Form of relation type r	*Attribute*	*Fusion function F*
$< same_{att} >$	categorical	$F\left(c_{ij}^k, c_{i'j}^{k'}, r_n\right) = sim(c_{ij}^k, c_{i'j}^{k'})$
$< same_{att} >$	continuous	$F\left(c_{ij}^k, c_{i'j}^{k'}, r_n\right) = \dfrac{1}{\lvert c_{ij}^k - c_{i'j}^{k'}\rvert + \varepsilon}$
$< bigger_{att} >$	continuous	$F\left(c_{ij}^k, c_{i'j}^{k'}, r_n\right) = c_{ij}^k - c_{i'j}^{k'}$

Two modules of the F-model, which identify related entities and fused related entities respectively, are shown in Algorithm 3 and Algorithm 4.

The inputs for Algorithm 3 come from relation tensor $\boldsymbol{\mathcal{RT}}$, entity i and relation r. The output is a vector RO a list of entities that are related to i-th entity based on r-th relation type.The time complexity of Algorithm 3 is thus $O(N_o)$.The aim of Algorithm 4 is to calculate the confidence score for each claim based on related entities. The inputs for this algorithm are relation tensor $\boldsymbol{\mathcal{RT}}$ and relation schema Rel_Sc. The output is confidence tensor $\boldsymbol{\mathcal{CT}}$.

Algorithm 3: find related entities

```
Input: relation tensor $RT_{N_o \times N_o \times N_r}$, entity $i$, relation type $r$.
Output: $RO$, a list of entities that are related to $i$ based on relation type $r$
for $j \leftarrow 1$ to $N_o$ do{
    if $RT[i][j][r] = 1$ then $RO$.add($j$)
}
return $RO$
```

Algorithm 4: Fuse related entities

```
Input: relation tensor $RT_{N_o \times N_o \times N_r}$, relation schema $Rel\_Sc$
Output: confidence tensor $CT_{N_o \times M \times N_s}$
1: for $i \leftarrow 1$ to $N_o$ do{
2:   for $j \leftarrow 1$ to $M$ do{
3:     $R \leftarrow$ the indices of relation types about attribute $j$
       //size of $R$ is 1 or 2 based on attribute type $j$
4:     for $k \leftarrow 1$ to $N_s$ do{
5:       $RO \leftarrow find\_related\_entities(RT, i, Rel\_Sc[R])$
         // $RO$ is the list of all related entities to entity $i$ based on the relation $Rel\_Sc[R]$
6:       DS $\leftarrow find\_sources(RO)$
         // a procedure that finds the list of all sources providing value about
          entities in $RO$
7:       calculate $CT[i][j][k]$ according to Eq. (1)
8:     }
9:   }
10: }
11: return $CT$
```

Algorithm 4: In this algorithm, for each entity i, each attribute j, and each data source k, $CT[i][j][k]$ is calculated. First, in line 3 the indices of relation types about attribute$j$are stored in array R. Then in line 5, according to the relation types in R, all entities that are related to entity i are stored in the array RO using Algorithm 3 (Find related entities). Next, in line 6, all the sources producing value about entities in RO stored in the array DS. Finally, $CT[i][j][k]$ is calculated using Equation (1).

Time complexity: There are three loops in Algorithm 4. The time complexity is $O(N_r {N_o}^2 M {N_s}^2)$, where N_r is the number of relation types, N_o is the number of entities, M is the number of attributes and N_s is the number of data sources. Because of $N_o \gg M, N_r, N_s$, Algorithm 4 is a quadratic-time algorithm with respect to N_o, which is validated experimentally in section 6.6.

5.3. Illustrative Example

In this section, an illustrative example is provided for the whole approach step by step, from data claims with conflicts to the resulting fused data.

Example 8: Suppose the entity about which there are claims from multiple sources is *a person.* Attribute set is $A = \{$*age, work-class, sex, education-level, outcome*$\}$. Attribute type set is $AT = \{2, 1, 1, 1, 1\}$. Table 5 below shows clean data set for the persons (entities) that is the ground truth.

Table 5. A sample of the ground truth for the persons

ID	Age	Work-Class	Sex	Education-Level	Outcome
1	32	State-gov	Female	2	<=50k
2	35	State-gov	Male	2	<=50k
3	37	State-gov	Female	2	<=50k
4	46	Self-employed	Male	1	<=50k
5	29	Private	Male	3	>50k
6	53	Private	Male	1	>50k
7	45	Self-employed	Male	3	>50k
8	48	Self-employed	Male	3	>50k
9	35	State-gov	Female	2	<=50k
10	58	Private	Male	3	>50k

For one of the relation type in the relation schema <$same_{age}$ >, the metadata set is created using Algorithm 2. This metadata set has 100 rows and 9 attributesinclude: *lhd_work-class, lhd_sex, lhd_education-level, lhd_outcome, rhd_work-class, rhd_sex, rhd_education-level, rhd_outcome,* and*sameAge*. To calculate the values of attribute *sameAge*, we first discretize the values of attribute *age* into four categories ([20-30), [30-40), [40-50) and [50-60)). If the categories of attribute age for the left-hand entity and the right-hand entity are the same, the value of attribute *sameAge* is equal to 1, otherwise it is 0. Table 6 shows part of the metadata set.

The next step is inferring rules about the existence of relation type $< same_{age} >$ between the entities. We use CAR by Thabtah et al. (2005) [27] as a classifier. The following rules in Table 7 are extracted by this classifier using metadata set as a training set.

Table 8 shows some claims about seven persons with *personID* from 1 to 7. These claims are provided by three sources S_1, S_2 and S_3. Note that although there are some conflicts in the values of only one attribute in this example, but generally the rest of the attributes can also have conflicts in their values. All incorrect values are marked in bold, with the correct values written in brackets after the incorrect ones.

Based on Table 8, the first vertical slice of claim tensor $\boldsymbol{C}$represents all claims about the value of attribute age provided by all sources.

$$\boldsymbol{C}_{\text{age}} = \begin{bmatrix} \mathbf{45} & \mathbf{35} & \times \\ \mathbf{36} & \times & \mathbf{26} \\ \times & \mathbf{32} & \times \\ \times & \mathbf{30} & \times \\ \mathbf{28} & \times & \times \\ \mathbf{57} & \times & \mathbf{47} \\ \times & \times & \mathbf{48} \end{bmatrix}$$

Then, one frontal slice of tensor $\boldsymbol{\mathcal{RT}}$ that is related to relation type <$same_{age}$ > is constructed using the rules extracted by the classifier in Table 7.

Table 6. Part of metadata set related to relation type $< \boldsymbol{same_{age}} >$

Entity pair	lhd _work-class	lhd _sex	lhd _educ.-level	lhd _outcome	rhd _work-class	rhd _sex	rhd _educ.-level	rhd _outcome	SameAge

1,2	State-gov	Female	2	<=50k	State-gov	Male	2	<=50k	1
1,3	State-gov	Female	2	<=50k	State-gov	Female	2	<=50k	1
1,4	State-gov	Female	2	<=50k	Self-employed	Male	1	<=50k	0
…									
2,3	State-gov	Male	2	<=50k	State-gov	Female	2	<=50k	1
2,4	State-gov	Male	2	<=50k	Self-employed	Male	1	<=50k	0
…									
7,8	Self-employed	Male	3	>50k	Self-employed	Male	3	>50k	1
7,9	Self-employed	Male	3	>50k	State-gov	Female	2	<=50k	0
7,10	Self-employed	Male	3	>50k	Private	Male	3	>50k	0
…									

Table 7. Extracted rules related to relation type $< same_{age} >$

Rule 1: *(lhd_work-class = State-gov)Λ(rhd_sex= Male)=>sameAge=0*
Rule 2: *(rhd_work-class=State-gov)Λ(lhd_sex= Male)=>sameAge=0*
Rule 3: *(rhd_work-class= State-gov)Λ(lhd_work-class= State-gov)=>sameAge=1*
Rule 4: *(lhd_work-class= Private)Λ(rhd_outcome=<50k)=>sameAge=0*
Rule 5: *(rhd_work-class= Private)Λ(lhd_outcome =<50k)=>sameAge=0*
Rule 6: *(rhd_work-class = Self-employed)Λ(lhd_work-class = Self-employed)=>sameAge=1*
Rule 7: *(rhd_work-class = Private)Λ(lhd_work-class = Self-employed)=>sameAge=0*
Rule 8:*(rhd_work-class = Private)Λ(lhd_work-class = Private)=>sameAge=0*

$$\boldsymbol{RT}_{<same\ _{age}>} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \end{bmatrix}$$

Using Algorithm 3, the related entities of each person are found: persons that their *personID* are 1, 2 and 3 are related to each other and persons 4, 6 and 7 are related to each other too.

Table 8. Data claims provided by the sources

Source	PersonID	Age	Work-Class	Sex	Education-Level	Outcome
S_1	1	**45** (35)	State-gov	Female	2	<=50k
	2	36	State-gov	Female	2	<=50k
	5	28	Private	Male	3	>50k

	6	**57** (47)	Self-employed	Male	1	>50k
S_2	1	35	State-gov	Female	3	<=50k
	3	32	State-gov	Female	2	<=50k
	4	**30** (45)	Self-employed	Female	2	<=50k
S_3	2	**26** (36)	State-gov	Female	2	<=50k
	6	47	Self-employed	Male	1	>50k
	7	48	Self-employed	Male	3	>50k

Final step is fusing the values of attribute *age* of entities using algorithm 4. We now calculate the confidence score of two claims about attribute *age* of person 1. The list of related entities to entity 1 is $RO^1 = \{2,3\}$; the set of sources providing claims about $i' \in RO^1$ is$DS = \{S_1, S_2, S_3\}$. All values provided by the sources in DS about the entities in RO^1 is 36, 32 and 26. In this example, we use relation type r =<$same_{age}$ >, therefore we apply second fusion function in Table 4. So, for claim $c^1_{1,1} = 45$ and according to equation (1):

$$\mathcal{CT}[1][1][1] = \frac{1}{Z}\sum_{i'=2,3}\sum_{k'=1,2,3} F\left(c^1_{1,1}, c^{k'}_{i',1}, r\right)$$

$$= \frac{1}{Z}\left(\frac{1}{|45-36|} + \frac{1}{|45-32|} + \frac{1}{|45-26|}\right)$$

$$= \frac{1}{Z}\,0.24$$

In the same way, for $c^2_{1,1} = 35$, its confidence score is $\mathcal{CT}[1][1][2] = \frac{1}{Z}(\frac{1}{|35-36|} + \frac{1}{|35-32|} + \frac{1}{|35-26|}) = \frac{1}{Z}1.4$; we calculate normalization parameter Z using equation (2):

$$Z = 0.24 + 1.4 = 1.64$$

Finally, the confidence score of $c^1_{1,1}$ is $\mathcal{CT}[1][1][1] = \frac{0.24}{1.64}$ =0.15, and $c^2_{1,1}$ is $\mathcal{CT}[1][1][2] = \frac{1.4}{1.64}$ =0.85. As a result, the claim provided by S_2 is selected as a correct value. As for person 2, there are two claims, and the value 36 is selected as a correct value, because the probabilities of the correctness for these two claims are 0.8 for $c^1_{2,1}$and 0.2 for $c^3_{2,1}$. The first vertical slice of confidence tensor $\mathcal{CT}$represents all confidence scores of claims about attribute age provided by all sources.

$$\boldsymbol{CT}_{age} = \begin{bmatrix} \mathbf{0.15} & \mathbf{0.85} & \times \\ \mathbf{0.8} & \times & \mathbf{0.2} \\ \times & \mathbf{1} & \times \\ \times & \mathbf{1} & \times \\ \mathbf{1} & \times & \times \\ \mathbf{0.12} & \times & \mathbf{0.88} \\ \times & \times & \mathbf{1} \end{bmatrix}$$

About person 5 there is no related entity so the single value provided by S_1 is considered as the correct value. Although one value is provided about person 4 by S_2 but there are two entities that are related to it, persons 6 and 7. Because of this, there is a tradeoff between the selection of this single value and the average value of the related entities. It depends on the amount of confidence score for this single value and also the precision of the rules. We can decide to select one of these values by defining a threshold.

6. Experiments

In this section, we use two real data sets to evaluate our proposed approach. The aim of our experiments is to answer these questions:

Q1: Can classifiers be used directly to predict the value of entity attributes instead of having to infer the existence of a relationship between entities?

Q2: To what extent can the classifiers make correct and accurate predictions of the relationships in the final output of data fusion?

Q3: How accurate is high-level data fusion in terms of the number of reliable sources, compared to low-level fusion?

1.1. Experimental Setup

Data sets

To demonstrate the advantages of our proposed approach, especially in an environment involving few reliable data sources, we conducted experiments on real data sets generated from UCI machine learning data sets. The basic assumption about the data is that the entities must have one or more relationship(s) between them. In other words, there are several relations between entities in the dataset.We chose the **UCI Adult** (http://archive.ics.uci.edu/ml/datasets/Adult) and **UCI Bank** (http://archive.ics.uci.edu/ml/datasets/Bank+Marketing) datasets because the entities are related to each other by attributes. These two datasets contain raw data. We therefore perform some preprocessing on these in order to clean them up, as follows:

- Deletion of attributes: If the percentage of entities that have an unknown value or the same value for a specific attribute is more than 80%, then this attribute is deleted.For example, in the Bank dataset, 36,959 entities – more than 80% of the entities – have an unknown value. The attributes related to these were therefore deleted from the dataset. In total, in the Bank dataset eight out of 17 attributes, and in the Adult dataset four out of 15 attributes, were deleted by this process.
- Remove instances: If an entity has an unknown value for one or more attributes, this entity is removed from the dataset. For example, in the Adult dataset, 1836 entities have an unknown value related to the "*workClass*" attribute, so these entities are removed. As a result of this process, the number of entities in the Adult dataset was reduced to 30,162, and in the Bank dataset to 43,193.
- Discretization: Since attributes selected for classification must be a categorical attribute, we discretize any continuous attributes. Discretizing these attributes

makes them into ordinal categorical attributes, on which it is then possible to build comparative relationships.

After these three steps to preprocess the initial rough dataset, the resulting dataset is regarded as the ground truths. To answer Q1, we use these datasets as inputs for the classifier in order to train the model to predict the value of attributes. We then create a metadata set as a training set for the classifier to train the model to infer the existence of relationships between entities. Based on attributes selected to create noisy data sources, we then construct a *relation schema*. Table 9 shows the statistics of these datasets, while Table 10 displays the relation schema for each dataset.

Metadata creation: To create metadata sets for each relation type, we implement Algorithm 2 in section 5.2. The ground truth we use to create metadata sets contains 1000 entities, so the Adult and Bank datasets each have 1,000,000 rows of metasets. The number of columns in the Adult dataset is $10 \times 2 + 1 = 21$ and in the Bank dataset is $8 \times 2 + 1 = 17$. (Note that, in section 6.5, we carry out an experiment to investigate the effect of the number of entities used to build the metadata set on the accuracy of our approach.)

In section 6.2, we report the results of our experiments designed to predict the value of attributes vs. the existence of a relation, and hence to answer Q1 (*Can classifiers be used directly to predict the value of entity attributes instead of having to infer the existence of a relationship between entities?*). In the remainder of this section, we explain how to build data sources.

Table 9. Statistics of data sets

	Adult	**Bank**
#entities	30162	43193
#attributes	11	9
#relation types	5	5
#data sources	10	10
#claims provided by data sources	126679	216032

Creating data sources: We generate a data set consisting of multiple conflicting sources, by injecting different levels of noise into the ground truths as the inputs to our approach and baseline methods. We select four attributes in each dataset (The attributes *age*, *education*, *workClass* and *occupation* were selected in the Adult dataset; and the attributes *age*, *job*, *marital* and *education* in the Bank dataset), whose values are then randomly flipped to generate facts that deviate from the ground truths. A parameter α is used to control the degree of reliability of each source. To put it another way, α stands for the percentage of noisy data. In this way, we can generate datasets which contain 10 sources with various degrees of reliability (α= 50, 55, 60, …, 95).

Table 10. Relation schema

Adult	Bank
$\{<same_{age}>,<bigger_{age}>,$ $<same_{education}>,<same_{workClass}>,$ $<same_{occupation}>\}$	$\{<same_{age}>,<bigger_{age}>,$ $<same_{education}>,<same_{job}>,$ $<same_{marital}>\}$

Algorithm Implementation

In section 5, we explained the modules of RelBCR and presented the algorithms related to each module as Algorithm 1 to 4. In this section, we describe in more details about some of the implementation-related issues.

Algorithm 1 is for relation schema construction. In this algorithm, based on the type of each attribute, one or two relation types are created. The type of attributes must be specified as an input to the problem and it is a part of the problem knowledge. We use comma-separated values (CSV) files for input datasets. Therefore, when reading data from the input file, we can specify the type of each column (attribute). In addition, in Algorithm 1 for ease of understanding we show relation types in the form of $< same_{att} >$ and $< bigger_{att} >$. In reality, we use a cell array that is an array of length M (number of attributes) such that each element can be an array of length one or two. The first element of this array represents relation type $< same_{att} >$, and the second of this, if any, represents another relation type $< bigger_{att} >$.

The next point is about metadata sets. Although input data set contains different types of attributes, but after discretization in the preprocessing step, all attributes become categorical and ordinal attributes. So, metadata sets contain only categorical attributes.

Performance Measure

Our proposed framework consists of two main parts. The first part contains the classifier methods for predicting the relations between entities (the G-model), while the second part comprises the fusion functions for finding the truth between conflicting values (F-model). We need to evaluate the performance of the methods used for both parts. Two measures, *precision* and *recall*, are used to evaluate the G-model, while *accuracy* is used to evaluate the F-model:

- **Precision** (or confidence) denotes the proportion of predicted positive instances that are correct real positives.
$$precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{3}$$
- **Recall** (or sensitivity) is the proportion of real positive instances that are correctly predicted positive.
$$recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{4}$$
- The **F-measure** is the harmonic mean of precision and recall.
$$F - measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$
- We use **accuracy** as performance measure which is computed as the percentage of the output of the F-model that is the same as the ground truths.

$$accuracy = \frac{1}{n_g}\sum_{i=1}^{n_g} 1\{g_i = f_i\} \tag{6}$$

where n_g is the number of entities in the ground truth, g_i is the entity in the ground truth set and f_i is the fusion output.

Environment

All the experiments in this study were conducted on a workstation with 8GB RAM, an Intel® Core™ i5-4300U CPU @ 1.90GHz 2.50 GHz, and Windows 10 pro. All the algorithms, including those from earlier methods, were implemented in Matlab R2017a. Weka 3.8 data mining tools were used to preprocess the datasets and infer classifiers.

6.1. Relation Estimation

In this section, in order to answer Q1, we need to carry out some experiments to classify each attribute against the relationtype related to this attribute. Two classification methods, a decision tree and classification based on association rules (CAR), are used for this purpose. The aim of this experiment is to investigate the impact of using additional information on the performance of fusion techniques. Such additional information can be extracted either from the original dataset or from the metadata set.

In the first stage, we try to train a classifier that can predict the values of the attributes of entities. These attributes are considered as a *class*. For example, the attribute *occupation* in the Adult data set is a categorical attribute that has 14 values, such as *exec-managerial* and *handlers-cleaners*. Our classifier must therefore be trained to predict 14 classes of *occupation*.

In the second stage, we use relation types as a *class* and train classifiers accordingly. However, relation types are not pre-defined and there is no training set available that contains entities and relations between them (as is also true of applications in relational machine learning such as knowledge graph completion). A metadata set is therefore constructed as described in section 5.2. Each relation type in the relation schema is regarded as a class. This is known as triple classification: it aims to judge whether a given relation instance (*entity1; relation type; entity2*) is correct or not. This is a binary classification task, as explored by Lin et al. (2015) [12] and Socher et al. (2013) [28] in order to evaluate a link prediction task.

We use a decision tree (C4.5) and CAR by Thabtah et al. (2005) [27] as classifiers. C4.5 is a popular algorithm for constructing decision trees. These two classifiers are used to infer the classification models of attributes in our two datasets, the Adult data and the Bank data. Also, we use these classifiers to train models over metadata sets that are considered as training set for each relation type.

For the Adult data set, we use separately the attributes of *age, education, workClass* and *occupation* as classes, and then evaluate the C4.5 classifier. There are a total of 30,162 instances in the Adult dataset, of which 60% are considered as a training set and the rest as a test set. Next, the relation types $< same_{age} >$, $< bigger_{age} >$, $< same_{education} >$,$< same_{workClass} >$and$< same_{occupation} >$ are considered as classes. We construct metadata such that each row includes the attributes of two entities and the relation between them. 1000 entities are used to construct each metadata set, with the same number of positive and negative instances.

For the Bank data, the attributes of *age*, *job*, *marital* and *education* are used as classes, and then to evaluate the C4.5 classifier. There is a total of 43,193 instances in the Bank data set, of which 60% are considered as a training set and the rest as a test set. Next, the relation types $< same_{age} >$, $< bigger_{age} >$, $< same_{job} >$,$< same_{marital} >$ and $< same_{educatio\ n} >$ are considered as classes. Again, we construct metadata such that each row includes the attributes of two entities and the relation between them. 1000 entities are used to construct each metadata set, with the same number of positive and negative instances.

Figure 3 shows the evaluation results of this classifier. The x-axis in the figure indicates the attributes and relation types as a class, while the y-axis presents the percentage of instances classified correctly in all classes by the classifier C4.5. These results show that, when we use the classifier to predict attribute values, the number of instances classified into the correct classes is lower than the number of correctly classified instances when the classes are the existence of relationships. When a specific attribute is considered as a class, the instances are entities and the classes are the different values of this attribute. For example, if the attribute *marital* in the Bank dataset is considered as a class, the instances will be classified into three categories of *married*, *single* and *divorced*. The percentage of all true positives is 67.1%.On the other hand, when one relation type is considered as a class, the instances are pairs of entities and the classes are *yes* or *no*, depending on whether the given relationship exists between a particular pair of entities. For example, the percentage of all true positives in the classification of $< same_{marital} >$is 98.3%.

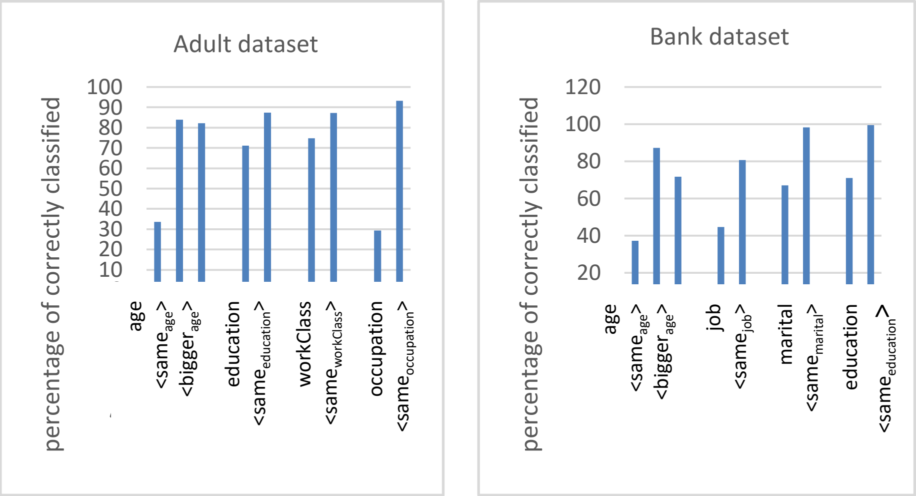


Fig. 3. Percentage of correctly classified instances using methods of attribute as a class and relation as a class

Tables 11 and 12 show the evaluation results of classification through both the original dataset and the metadata set.

Table 11 reports the classification results of two attributes, *occupation* and *age*. *Occupation* is a categorical attribute with 14 values, such as *Exec-managerial*, *Handlers-cleaners* and so on. The classifier therefore classifies the data into 14 classes. The precision and recall of each class of *occupation* are shown on the left-hand side of Table 11. As examples, the class *Other-service* was classified with a precision of 0.279 and a recall of 0.284, while *Armed-Forces* was not classified. The precision and recall of the classification of relation type $< same_{occupation} >$ were 0.928 and 0.955 respectively. The continuous attribute *age* is first divided into 10 categories. The data is then classified into 10 classes. The precision and recall of this classification are reported on the right-hand side of Table 11. Two relation types of the attribute *age* are defined, $< same_{age} >$ and $< bigger_{age} >$. Table 12 contains the results of classifying the attributes *workClass* and *education*, and their related relation types and Table 13 is related to the attributes and relation types of Bank dataset.

Table 11. Comparison of classifier performance measures related to attributes *occupation* and *age* (Adult dataset)

Occupation				Age			
Class	P	R	F	**Class**	P	R	F
Exec-managerial	0.283	0.301	0.292	**cat1 (-inf-24.3]**	0.566	0.733	0.639
Handlers-cleaners	0.129	0.119	0.124	**cat2 (24.3-31.6]**	0.312	0.305	0.308
Prof-specialty	0.47	0.525	0.496	**cat3 (31.6-38.9]**	0.258	0.257	0.257
Other-service	0.279	0.284	0.281	**cat4 (38.9-46.2]**	0.287	0.441	0.348
Adm-clerical	0.313	0.41	0.355	**cat5 (46.2-53.5]**	0.218	0.098	0.136
Sales	0.201	0.167	0.183	**cat6 (53.5-60.8]**	0.188	0.068	0.1
Transport-moving	0.164	0.106	0.129	**cat7 (60.8-68.1]**	0.264	0.187	0.219
Farming-fishing	0.278	0.195	0.229	**cat8 (68.1-75.4]**	0.125	0.019	0.034
Machine-op-inspct	0.185	0.131	0.154	**cat9 (75.4-82.7]**	0.059	0.009	0.015
Tech-support	0.125	0.047	0.069	**cat10 (82.7-inf)**	0	0	0
Craft-repair	0.292	0.368	0.326	$< \boldsymbol{same_{age}} >$ **- yes**	0.835	0.86	0.848
Protective-serv	0.378	0.244	0.297	$< \boldsymbol{same_{age}} >$ **- no**	0.843	0.816	0.829
Armed-Forces	?	?	?	$< \boldsymbol{bigger_{age}} >$**-yes**	0.796	0.793	0.794
Priv-house-serv	0.279	0.119	0.167	$< \boldsymbol{bigger_{age}} >$ **- no**	0.841	0.843	0.842
$< \boldsymbol{same_{occupation}} >$- yes	0.928	0.955	0.941				
$< \boldsymbol{same_{occupation}} >$- no	0.943	0.91	0.927				

Table 12. Comparison of classifier performance measures related to attributes work class and education (Adult dataset)

WorkClass				Education			
Class	P	R	F	**Class**	P	R	F

State-gov	0.213	0.039	0.066	**(-inf-4.75]**	0.2	0.001	0.002
Private	0.771	0.97	0.859	**(4.75-8.5]**	0.067	0	0.001
Federal-gov	0.279	0.013	0.024	**(8.5-12.25]**	0.715	0.904	0.798
Local-gov	0.452	0.19	0.268	**(12.25-inf)**	0.703	0.586	0.639
Self-emp-inc	0.303	0.028	0.051	$<\boldsymbol{same}_{\boldsymbol{education}}>$ - **yes**	0.894	0.873	0.884
Never-worked	?	?	?	$<\boldsymbol{same}_{\boldsymbol{education}}>$ - **no**	0.849	0.874	0.862
$<\boldsymbol{same}_{\boldsymbol{workClass}}>$- yes	0.866	0.857	0.861				
$<\boldsymbol{same}_{\boldsymbol{workClass}}>$ - yes	0.878	0.886	0.882				

Table 13. Comparison of classifier performance measures related to attributes of *age*, *job*, *marital* and *education* (Bank dataset)

Age				**Job**			
Class	P	R	F	**Class**	P	R	F
cat1 (-inf-25.7]	0.478	0.18	0.262	**management**	0.585	0.843	0.691
cat2 (25.7-33.4]	0.459	0.569	0.508	**technician**	0.347	0.276	0.307
cat3 (33.4-41.1]	0.354	0.471	0.404	**enterpreter**	0	0	0
cat4 (41.1-48.8]	0.258	0.157	0.196	**blue-colar**	0.438	0.736	0.549
cat5 (48.8-56.5]	0.279	0.203	0.235	**retired**	0.509	0.553	0.53
cat6 (56.5-64.2]	0.417	0.269	0.327	**admin**	0.236	0.164	0.194
cat7 (64.2-71.9]	0.273	0.186	0.221	**services**	0.178	0.078	0.109
cat8 (71.9-79.6]	0.25	0.195	0.219	**self-employed**	0.063	0.002	0.004
cat9 (79.6-87.3]	0.167	0.056	0.083	**unemployed**	0.141	0.014	0.026
cat10 (87.3-inf)	0	0	0	**housemaid**	0.303	0.03	0.055
$< \boldsymbol{same_{age}} >$ **- yes**	0.889	0.912	0.9	**student**	0.501	0.35	0.412
$< \boldsymbol{same_{age}} >$ **- no**	0.842	0.803	0.822	$< \boldsymbol{same_{job}} >$ **- yes**	0.817	0.776	0.796
$< \boldsymbol{bigger_{age}} >$ **- yes**	0.693	0.636	0.663	$< \boldsymbol{same_{job}} >$ **- no**	0.799	0.837	0.817
$< \boldsymbol{bigger_{age}} >$ **- no**	0.734	0.781	0.757				
Education				**Marital**			
Class	P	R	F	**Class**	P	R	F
tertiary	0.788	0.672	0.725	**married**	0.677	0.891	0.769
secondry	0.706	0.847	0.77	**single**	0.648	0.48	0.551
primary	0.536	0.325	0.404	**divorced**	0.333	0.001	0.002
$< \boldsymbol{same_{education}} >$**- yes**	0.995	0.996	0.996	$< \boldsymbol{same_{marital}} >$ **- yes**	0.978	0.998	0.988
$< \boldsymbol{same_{education}} >$ **- no**	0.992	0.989	0.991	$< \boldsymbol{same_{marital}} >$ **- no**	0.995	0.95	0.972

As shown earlier in Figure 3, there are a lot fewer accurate values for attributes than for relations. In addition, for some classes the classifier is unable to construct models; the accuracy of these classes is therefore unknown. That said, while the precision and recall of the classifier are not high for certain relation types like $< same_{occupation} >$, they are of an acceptable level for the F-model, as we demonstrate in the next section.

Let us now look at classification based on association rules (CAR). CAR is a method for extending an efficient frequent pattern mining method, for FP-growth, for constructing a class distribution-associated FP-tree, and for mining large databases efficiently. Moreover, it applies a CR-tree structure to store and retrieve mined association rules efficiently, and prunes rules effectively based on confidence, correlation and database coverage. In effect, this classifier selects the most effective rule(s) from among all the rules mined for classification. Below, we show that using relation classification is more efficient than attribute classification. To demonstrate this, we look at some rules that CAR infers for predictions about attributes. The first three rules are inferred, using the Adult dataset, to predict some values of the *occupation* attribute. We then look at two rules inferred using the metadata set for the relation type $< same_{occupation} >$. These rules show that, for some values of attributes, no rule can be inferred; whereas, for relation types, rules with a high degree of accuracy can be obtained. Finally, we discuss the reasons for these results.

Some of the rules used by CAR for predictions about the attribute *occupation= Prof-specialty, other services* and *Adm-clerical*, and the related level of accuracy, are as follows:

workclass= Local-gov, age=5,hours-per-week=3, education-num=4, outcome= <=50K ==>occupation= Prof-specialty acc:(0.99).

workclass= Local-gov, race= Black, education-num=2, sex= Female, ==>occupation= Other-service acc:(0.97).

workclass= Federal-gov, race= Black, hours-per-week=3, age=3, ==>occupation= Adm-clerical acc:(0.95).

Note that there are no rules for some values of *occupation*.

We next apply CAR to the metadata. For the relation <$same_{occupation}$>,this produces rules such the following:

lhd_workClass= Federal-gov, lhd_education-num='(-inf-2.5]', rhd_age='(3.5-5.5]', lhd_age='(-inf-3.5]', rhd_education-num='(-inf-3.5]' ==>$same_{occupation}$*=1*acc:(0.99).

lhd_workClass= Local-gov, lhd_age='(5.5-6.5]', rhd_outcome =>50K, lhd_education-num='(2.5-3.5]' ==>$same_{occupation}$*=1* acc:(0.99).

The method of relations as a class does not suffer from the problem of lack of rules, because it is a binary classification and so, using the metadata set, relations can be deduced for every pair of entities.

Now we can answer Q1: Can classifiers be used directly to predict the value of entity attributes instead of having to infer the existence of a relationship between entities? The answer is *No*. There are a number of reasons why relations are better than attributes as a class.

1- The method of *relations as a class* deals with binary class prediction rather than the multi-class prediction used in the *attributes as a class* method.Multi-values make the accurate prediction of values difficult, and require extensive training data. In binary classes, in contrast, learning is both faster and more accurate.

2- In multi-classification the classifier may be unable to infer some classes. For example, for the attribute class of *occupation* there are no rules for the values *Armed-Forces* and *Priv-house-serv.* In binary classification, on the other hand, the value of attributes is not important; all that matters is whether the attributes have the same value or not.

3- In our proposed approach, all we need to understand is the relationship between one entity and another entity, not the exact values of the attribute of a given entity.

4- In high level fusion, we have a range of relations and methods, such as embedding nets [12, 24] and MLN [29] that can be applied to extract relations. These are used in our G-model – which is explained further below.

6.2. Effect of G-Model

For the G-model in our framework, we use classifiers trained by metadata, with each class being a relation. In this section, we examine the impact of the performance of the G-model on the accuracy of the F-model in answering Q2 (To what extent can the classifiers make correct and accurate predictions of the relationships in the final output of data fusion?). We use simulated models with different levels of precision and recall, and then evaluate the accuracy of the proposed fusion method. For each value of precision and recall, we repeat the experiment five times, with the average accuracy over these five repetitions shown in Figure 4. In this experiment we use the Adult data set. There are 10 data sources, of which only one is reliable. The accuracy of voting is 0.65.

As expected, the higher precision and recall of the G-model increases the accuracy of the fusion. Figure 4 shows that,for some values of precision and recall, the accuracy of our framework is higher than that of voting (the values that are above of the horizontal dash line). For example, in the G-model precision is 0.8 and recall is 0.7, which leads to the accuracy of the F-model is 0.91. For some values of precision and recall, the performance of this model is the same as (or lower than) voting. We consider such values as a *fail point* of our model. Table 14 reports the accuracy of the F-model for different amounts of precision and recall of the G-model. The empty cells indicate where the precision and recall of the G-model do not lead to appropriate results in the F-model – and so are fail points. For example, Precision =0.8 and recall = 0.4 is a fail point.

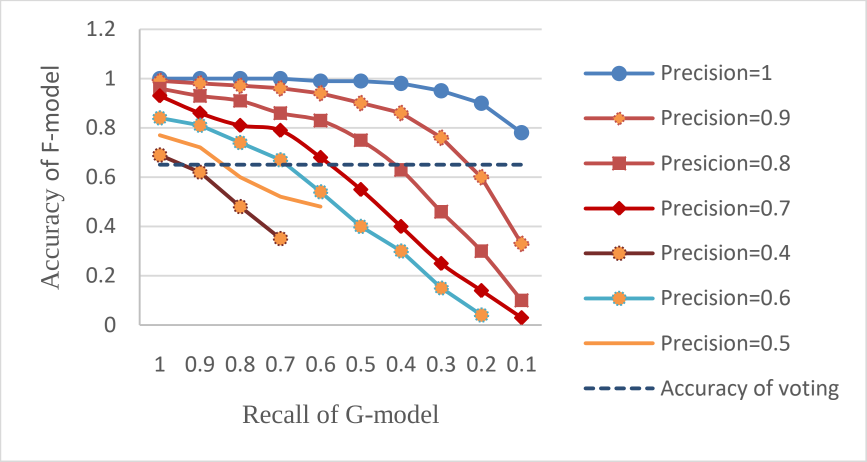


Fig. 4. Effect of G-model on performance of fusion

As can be seen from these tables, precision is more important than recall: with high precision, the framework is robust against low recall. For example, when precision is 0.7, the accuracy is better than voting in order to recall values of more than 0.6. This means that it is very important that our G-model does not mistake wrong relations for true ones. When our G-model is pessimistic about the existence of relations, its precision increases. By using multi-relations for each attribute, we can increase the precision of the G-model.

Table 14. Accuracy of F-model for different amounts of precision and recall

Precision→	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
Recall ↓							
1	0.69	0.77	0.84	0.93	0.96	0.99	1
0.9	×	0.72	0.81	0.86	0.93	0.98	1
0.8	×	×	0.74	0.81	0.91	0.97	1
0.7	×	×	0.67	0.79	0.86	0.96	1
0.6	×	×	×	0.68	0.83	0.94	0.99
0.5	×	×	×	×	0.75	0.90	0.99
0.4	×	×	×	×	×	0.86	0.98
0.3	×	×	×	×	×	0.76	0.95
0.2	×	×	×	×	×	×	0.90
0.1	×	×	×	×	×	×	0.78

6.3. High-Level vs Low-Level Fusion

In order to answer Q3 (*How accurate is high-level data fusion in terms of the number of reliable sources, compared to low-level fusion?*), we now compare our framework with low level fusion techniques including voting, Hub [15] and truth finder [4]. These low-level fusion methods, which we examined in our earlier work [30], contrast with RelBCR, which is a high-level fusion method. In this experiment, we fix the total number of sources as 10, and set the parameter α as the constant number 50%, which corresponds to an unreliable source. We then evaluate the performance of methods with different numbers of reliable sources. The precision of the G-model when used as a decision tree is 0.84 and its recall is 0.8.Figure 5 illustrates each method’s accuracy on the dataset for different numbers of reliable sources.

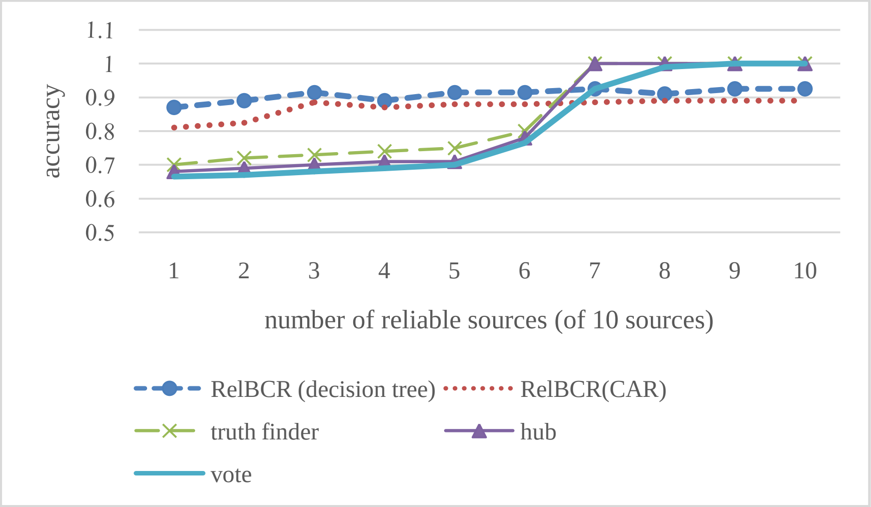


Fig. 5. Performance with respect to number of reliable sources

The following observations can be drawn from our results. First,our proposed approach outperforms existing conflict resolution techniques when there are few sources, because of its use of additional information about relations between entities. Second, when more than 50% of the sources are reliable, the performance of other existing voting models is slightly better than our approach. In general, it is easier to detect truths when we have a larger number of reliable sources, especially when we estimate the reliability of sources. In this experiment, the precision of the G-model – when the model is used as a decision tree – is 0.84 and its recall is 0.8, while the accuracy of the F-model is higher at 0.925. In section 6.3 we show that, if the precision of the G-model is increased, the performance of our overall approach can also be improved. Theoretically, therefore, by using a stronger inference engine we can obtain a higher level of accuracy. At the same time, the advantage of knowing the reliability of sources can be used to increase the accuracy of the F-model. We plan to examine this further in future research. Third and finally,using the G-model to estimate relations between entities and applying it to conflict resolution gives us more scope to estimate the reliability of sources, unlike with low level fusion methods.

6.4. Cost Analysis

As explained earlier, our approach uses additional information to improve the performance of the fusion process. Inevitably, the process of extracting and using such information makes the model more complicated. There are two main procedures in this approach: training the relation classifier in the G-model, and calculating the truthfulness of each claim in F-model. In this section, we discuss the overall costs associated with our approach.

Time Complexity

Here we test the computational complexity of calculating the truthfulness of each claim in the F-model (Algorithm 4, explained earlier). Figure 6 validates our time complexity analysis, confirming that the F-model is a quadratic-time algorithm with respect to the number of entities.

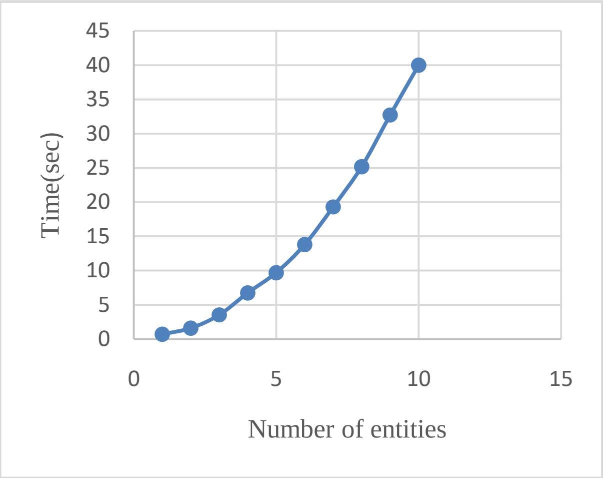


Fig. 6. Running time of confidence score calculation with respect to number of entities

Necessity of Clean Training Dataset

As explained in section 5.2, Figure 7 (left-hand side) shows the precision of the G-model, using various proportions of clean entities for metadata creation. We train the classifier for two relations, $< same_{workclass} >$ and $< bigger_{age} >$. As can be seen, the level of precision improves over iterations most of the time. Figure 7 (right-hand side) shows the accuracy of the F-model based on the G-model trained by various proportions of clean data.

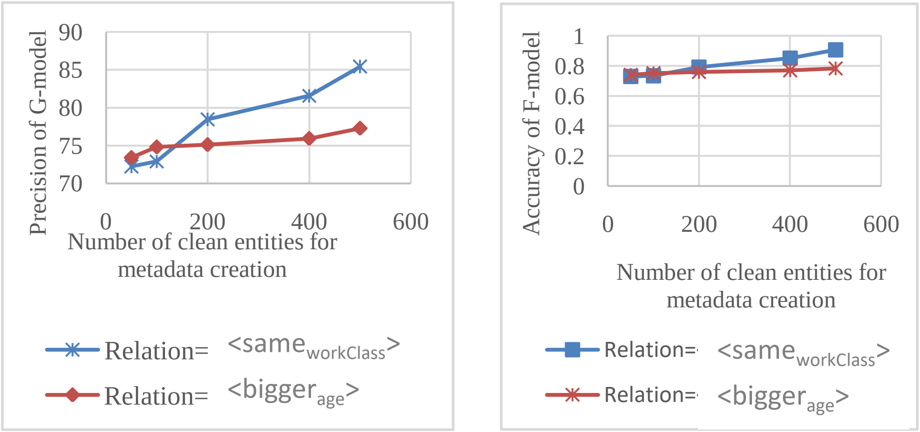


Fig. 7. Changes in precision of G-model (left) and accuracy of F-model (right) with respect to number of clean entities used in metadata creation

Figure 7 clearly illustrates that the precision of the G-model can be improved by increasing the proportion of clean data. However, for some relations like $< bigger_{age} >$, this improvement is very slow; more informative clean data is thus needed to train the classifier of this relation. It seems that using incremental learning to train the classifier, along with the fusing data procedure, can compensate for the lack of sufficient clean data. Hence, after training the classifier with a small clean dataset, we fuse the data and thereby obtain more clean data which can be used to retrain the classifier.

Performance with Respect to the Number of Related Entities

As explained in section 5.2, the truthfulness score of each claim about a specific entity depends on the claims about other entities that are related to it. Figure 8 shows the accuracy of the F-model with respect to the average of entities related to one entity. For this experiment, we use the G-model with a precision of between 0.8 and 0.95.

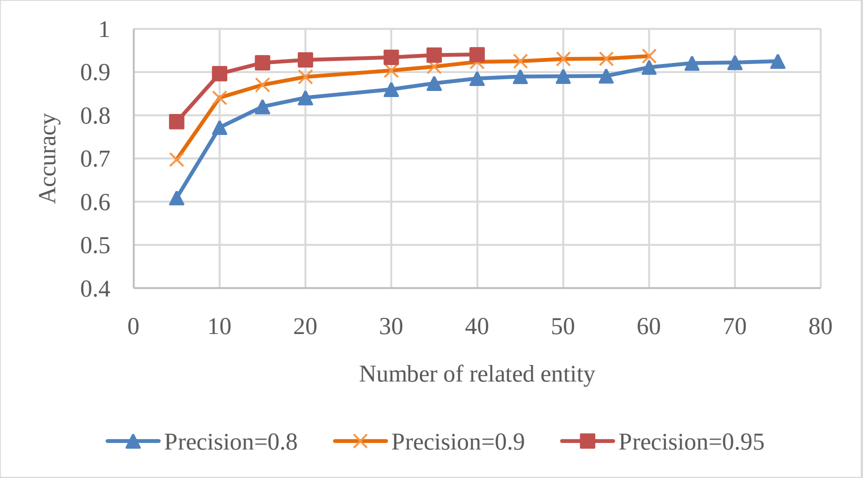


Fig. 8. Accuracy of F-model with respect to number of assessed related entities

This experiment showsthat, when the number of entities related to each entity increases, more evidence is collected for a given claim, and the accuracy consequently increases.This means that, if there are few relevant entities in the dataset for a specific entity, calculating the truthfulness score for this entity becomes very difficult. Figure 9 shows the number of related entities for each category of entities, with the value of the attribute *age* discretized to 10 intervals. As can be seen, in the Adult dataset, the number of related entities for cat2 of the relation $< same_{age} >$ is very small, and it is therefore necessary to obtain additional information from other relations.

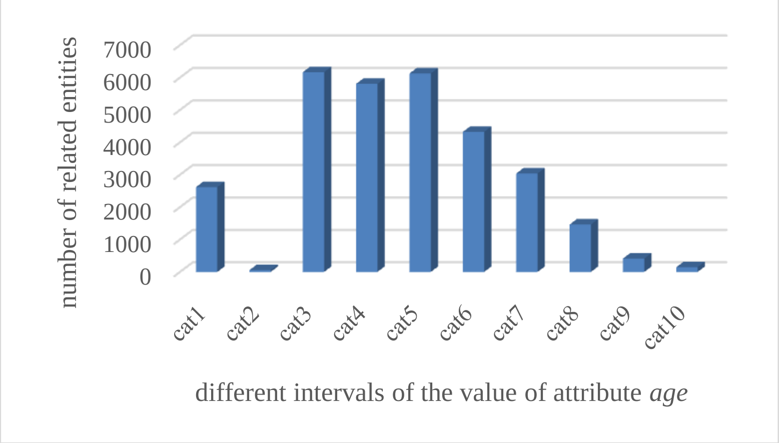


Fig. 9. Number of related entities for each entity with value of attribute *age* discretized into cat1 to cat10 in Adult dataset

7. Conclusion and Future Work

This article proposes a new approach for conflict resolution based on relations between entities, called RelBCR (Relation Based Conflict Resolution). In order to resolve conflicts between entities, fusion methods are used to try to estimate the reliability of data sourcesand then find the true values from among multiple conflicting values. Virtually all previous methods attempt to estimate two parameters: the *truthfulness* of data and the *trustworthiness* of sources. These methods prove however inadequate in some cases: in particular, when there are few reliable sources, and when all data sources don't provide claims about all entities. Using additional information, as our approach does, proves very effective. While previous studies perform only at the entity level or consider the similarity of entities as a relation between entities, our RelBCR approach uses machine learning methods to draw inferences about relations. It consists of two main parts: The G-model and the F-model.

In the G-model, when there are no predefined relations, a usable relation schema is first constructed. Next, a metadata set is created that contains the attributes of two entities and the specific relations between them, instead of only the attributes of one entity. Furthermore, in this phase there is a classifier that learns relations using the metadata. The output of the G-model is a relation tensor. In the F-model, based on defined fusion functions, the true value of the left-hand entity in the relation tensor is estimated.

The results of our experiments contain three major findings. First, relation types can be obtained using datasets that contain only entities and attributes. Second, in order to estimate correct values, using classifiers to infer the existence or absence of relations is more effective than predicting attribute values. Finally, the accuracy of the output data can be improved over other current solutions by using additional information inferred by learning methods. In order to apply this method successfully, there are a few requirements. First, we need to create a metadata set with sufficient positive and negative instances. To achieve this, a clean training data set is necessary. Second, there must be relevant entities for each entity in the dataset. We should also point out to some disadvantages of our approach. First, if the classifier has low negative predictive value (i.e. there is no relationship between the entities but the classifier predicts relationship between them), then the accuracy of our approach will decrease. Because the wrong values in the process of fusion replace the correct values based on the wrong related entities. Second, when almost all data sources provide wrong values about all attributes, the precision of G-model decreases and then the accuracy of our approach decreases.

As regards suggestions for future work, we believe that RelBCR could be strengthened in the following directions:

- **Adaptive entity selection for metadata creation**: We created metadata as a training set for classifiers in order to learn models for relation prediction. If we have 100 entities in the primary data set, we will have 100*100 = 10000 entity pairs in the metadata set. In other words, we have to deal with a very large amount of data. A system for adaptive entity selection that produces a smaller but still informative metadata set could be a useful enhancement.
- **Using latent feature model for relation estimation**:In this article we use classification methods to predict relations, and applied observable patterns to

extract relations. In other words, we used the attributes of pair entities to estimate relations. We can also use some methods that explain relations via latent features of entities (Embedding networks [12, 24] and RBM [31] are examples of such methods).

- **Using weighted multi-relations in the F-model:** To increase the precision of the classifier in relation prediction, we used multi-relations all of which have the same effect. Using a more varied and flexible approach to select the degree of contribution of each relation in truth discovery could produce better results.

References

1. Dong, X. L., Naumann, F. Data fusion: resolving data conflicts for integration. Proceedings of the VLDB Endowment 2, no. 2, 1654-1655. (2009)
2. Foo, P. H., Ng, G. W. High-level information fusion: An overview.J. Adv. Inf. Fusion 8, no. 1, 33-72. (2013)
3. Zhao, B.,Rubinstein,B. IP., Gemmell, J., Han, J.A bayesian approach to discovering truth from conflicting sources for data integration. Proceedings of the VLDB Endowment 5, no. 6, 550-561. (2012)
4. Yin, X., Han, J., Philip, S. Y. Truth discovery with multiple conflicting information providers on the web. IEEE Transactions on Knowledge and Data Engineering 20, no. 6,796-808. (2008)
5. Li, Q., Li, Y., Gao, J., Zhao, B., Fan, W., Han, J. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In Proceedings of the 2014 ACM SIGMOD international conference on Management of data, 1187-1198. (2014)
6. Meng, C., Jiang, W., Li, Y., Gao, J., Su, L., Ding, H., Cheng, Y. Truth discovery on crowd sensing of correlated entities. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems,169-182. (2015)
7. Liu, W., Liu, J., Wei, B., Duan, H., Hu, W. A new truth discovery method for resolving object conflicts over Linked Data with scale-free property. Knowledge and Information Systems, 1-31. (2018)
8. Li, Q., Li, Y., Gao, J., Su, L., Zhao, B., Demirbas, M., Fan, W., Han, J. A confidence-aware approach for truth discovery on long-tail data. Proceedings of the VLDB Endowment 8, no. 4, 425-436. (2014)
9. Ye, C., Wang, H., Ma, T., Gao, J., Zhang, H., Li, J.PatternFinder: Pattern discovery for truth discovery. Knowledge-Based Systems 176,97-109. (2019)
10. Li, Y., Gao, J., Meng, C., Li, Q., Su, L., Zhao, B., Fan, W., Han, J. A survey on truth discovery. ACM Sigkdd Explorations Newsletter 17, no. 2, 1-16. (2016)
11. Snidaro, L., Visentini, I., Llinas, J., Foresti, G. L. Context in fusion: some considerations in a JDL perspective. In Information Fusion (FUSION), 2013 16th International Conference on, IEEE, 115-120. (2013)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In AAAI, vol. 15, 2181-2187. (2015)
13. Bordes, A., Glorot, X., Weston, J., Bengio, Y. A semantic matching energy function for learning with multi-relational data. Machine Learning 94, no. 2, 233-259. (2014)
14. Bleiholder, J., Naumann, F.Data fusion.ACM Computing Surveys (CSUR) 41, no. 1, 1.(2009)
15. Li, X., Dong, X. L., Lyons, K., Meng, W., Srivastava, D. Truth finding on the deep web: Is the problem solved? In Proceedings of the VLDB Endowment, vol. 6, no. 2, VLDB Endowment, pp. 97-108. (2012)

16. Rekatsinas, T., Joglekar, M., Garcia-Molina, H., Parameswaran, A.,Ré, C. Slimfast: Guaranteed results for data fusion and source reliability. In Proceedings of the 2017 ACM International Conference on Management of Data,1399-1414. (2017)
17. Yin, X., Tan, W.Semi-supervised truth discovery. In Proceedings of the 20th international conference on World wide web, ACM, 217-226. (2011)
18. Pochampally, R., Das Sarma, A., Dong, X.L., Meliou, A., Srivastava, D. Fusing data with correlations.In Proceedings of the 2014 ACM SIGMOD international conference on Management of data, ACM, 433-444. (2014)
19. Nakhaei, Z., Ahmadi, A. Unsupervised Deep Learning for Conflict Resolution in Big Data Analysis. In International Congress on High-Performance Computing and Big Data Analysis, Springer, Cham, pp. 41-52. (2019)
20. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.A review of relational machine learning for knowledge graphs. Proceedings of the IEEE 104, no. 1, 11-33. (2016)
21. Kemp, C., Griffiths, T.L., Tenenbaum, J.B.Discovering Latent Classes in Relational Data.(2004)
22. Airoldi, E., Blei, D., Xing, E., Fienberg, S.A latent mixed membership model for relational data. In Proceedings of the 3rd international workshop on Link discovery, ACM, 82-89. (2005)
23. Hoff, P. Modeling homophily and stochastic equivalence in symmetric relational data. In Advances in neural information processing systems, 657-664. (2008)
24. Yang, B., Yih, W.T., He, X., Gao, J., Deng, L. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575. (2014)
25. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.Translating embeddings for modeling multi-relational data.In Advances in neural information processing systems, 2787-2795. (2013)
26. Bordes, A., Weston, J., Collobert, R., Bengio, Y.Learning Structured Embeddings of Knowledge Bases. In AAAI, vol. 6, no. 1, p. 6. (2011)
27. Thabtah, F., Cowling, P.,Peng, Y. MCAR: multi-class classification based on association rule. In Computer Systems and Applications, The 3rd ACS/IEEE International Conference, p. 33. (2005)
28. Socher, R., Chen, D., Manning, C.D., Ng, A.Reasoning with neural tensor networks for knowledge base completion.In Advances in neural information processing systems, 926-934.(2013)
29. Richardson, M., Domingos, P. Markov logic networks.Machine learning 62, no. 1-2, 107-136. (2006)
30. Nakhaei, Z., Ahmadi, A. Toward high-level data fusion for conflict resolution. In Machine Learning and Cybernetics (ICMLC), 2017 International Conference on, vol. 1, IEEE, 91-97. (2017)
31. Ge, L., Gao, J., Li, X., Zhang, A. Multi-source deep learning for information trustworthiness estimation." In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 766-774. (2013)

Zeinab Nakhaei received the BS degree in software computer engineering from the AmirKabir University of Technology in 2006 and the MS degree in Artificial Intelligence and Robotics from the Iran University of Science andTechnology in 2010. She is currently pursuing the Ph.D. degree with Science and Research Branch ofIslamic Azad University, Tehran, Iran; she is also a lecturer in the Electrical and ComputerEngineering, Islamic Azad University, Tehran, Iran. Her research interests include data integration and data fusion.

Ali Ahmadireceived the Ph.D. in Computer & System Sciences, Majority of Image Processing and Neuralnetworks, University of Osaka Prefecture, Osaka, Japan, March 2004, the MS in Computer & SystemSciences, Majority of Image Processing and Neural Networks, University of Osaka Prefecture, Osaka,Japan, March 2001, and BS in Electrical Engineering, Amirkabir University of Technology, Tehran, Iran,in Sep. 1990. He is currently an associate professor in the Computer Engineering Department, K. N. Toosi University of Technology, Tehran, Iran. His research interests include Semantic data mining andInformation fusion and Interactive learning models.

Arash Sharifi received the Ph.D. in Artificial Intelligence from the Science and Research Branch ofIslamic Azad University, Tehran, Iran, in 2011.He is currently an assistant professor in the Computer and Electronics Department, Science and Research Branch ofIslamic Azad University, Tehran, Iran. His research interests include machine learning, deep learning and data science.

Kambiz Badie received all his degrees from Tokyo Institute of Technology, Japan, majoring in pattern recognition. He has been actively involved in doing research in a variety of issues, such as machine learning, cognitive modeling, knowledge processing & creation in general, and analogical knowledge processing. At present, he is a member of scientific board of IT Research Faculty (Full Professor) at ICT Research Institute, an adjunct professor at Faculty of Engineering Science in the University of Tehran, and in the meantime, the editor-in-chief of International Journal of Information & Communication Technology Research (IJICTR) being published periodically by ICT Research Institute and also an invited member of Iranian Academy of Science.

Received: January 31, 2020; Accepted: April 02, 2021.

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд

004

COMPUTER Science and Information Systems : the International journal / Editor-in-Chief Mirjana Ivanović. – Vol. 18, No 3 (2021) - . – Novi Sad (Trg D. Obradovića 3): ComSIS Consortium, 2021 - (Belgrade : Sigra star). –30 cm

Polugodišnje. – Tekst na engleskom jeziku

ISSN 1820-0214 (Print) 2406-1018 (Online) = Computer Science and Information Systems
COBISS.SR-ID 112261644

Cover design: V. Štavljanin
Printed by: Sigra star, Belgrade