Computer Science and Information Systems 11(2):601–622 DOI: 10.2298/CSIS130415035A

Flow-Based Anomaly Intrusion Detection System Using Two Neural Network Stages

Yousef Abuadlla[1], Goran Kvascev[2], Slavko Gajin[3], and Zoran Jovanović[3]

[1] School of Electrical Engineering, University of Belgrade,
Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia
abouadlla@gmail.com
[2] School of Electrical Engineering, University of Belgrade,
Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia
kvascev@etf.bg.ac.rs
[3] School of Electrical Engineering, University of Belgrade,
Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia
{slavko.gajin, zoran}@rcub.bg.ac.rs

Abstract. Computer systems and networks suffer due to rapid increase of attacks, and in order to keep them safe from malicious activities or policy violations, there is need for effective security monitoring systems, such as Intrusion Detection Systems (IDS). Many researchers concentrate their efforts on this area using different approaches to build reliable intrusion detection systems. Flow-based intrusion detection systems are one of these approaches that rely on aggregated flow statistics of network traffic. Their main advantages are host independence and usability on high speed networks, since the metrics may be collected by network device hardware or standalone probes. In this paper, an intrusion detection system using two neural network stages based on flow-data is proposed for detecting and classifying attacks in network traffic. The first stage detects significant changes in the traffic that could be a potential attack, while the second stage defines if there is a known attack and in that case classifies the type of attack. The first stage is crucial for selecting time windows where attacks, known or unknown, are more probable. Two different neural network structures have been used, multilayer and radial basis function networks, with the objective to compare performance, memory consumption and the time required for network training. The experimental results demonstrate that the designed models are promising in terms of accuracy and computational time, with low probability of false alarms.

Keywords: Intrusion Detection system, Anomaly detection system, Neural Network, NetFlow.

1. Introduction

With the rapid growth of the Internet and due to increase in number of attacks, computer security has become a crucial issue for computer systems. Intrusion Detection

Systems (IDS) technology is an effective approach in dealing with the problems of network security.

Intrusion Detection Systems (IDS) have become increasingly important in recent years to reveal the growing number of attacks. They need to be able to adapt to the rise in the amount of traffic as well as the increase in line speed [1]. However, researchers assess the payload-based IDSs processing capability to lie between 100 Mbps and 200 Mbps when commodity hardware is used [2, 3], and close to 1 Gbps when dedicated hardware is employed [4, 5]. Well-known systems like Snort [6] and Bro [7] exhibit high resource consumption when confronted with the overwhelming amount of data found in high-speed networks [8]. In addition, the spread of encrypted protocols poses a new challenge to payload-based systems. In addition to that, the constant increase in network traffic and the fast introduction of high speed (tens of Gbps) network equipment [9] make it hard to preserve traditional packet based intrusion detection systems. Such systems rely on deep packet inspection, which does not scale well.

Having this in mind, flow-based approaches seem to be a promising candidate for research in the area of IDS. A flow is defined as a unidirectional stream of packets that share common characteristics, such as source and destination addresses, ports and protocol type. Additionally a flow includes aggregated information about the number of packets and bytes belonging to the stream, as well as its duration. Flows data are often used for network monitoring, allowing us to obtain a real time overview of the network traffic. Common tools for this purpose are Nfsen [10] and Flowscan [11], while the *de facto* standard technology in this field is Cisco NetFlow, particularly its versions 5 and 9 [12], [13]. This technology is now becoming a formal standard through the work of the IETF IPFIX working group [14].

Network flows are monitored by specialized accounting modules usually placed in network routers. These modules are responsible for calculating flow statistics and exporting these statistics (flow-data) to external collectors. Flow-based IDSs analyze these flows data to detect anomaly and alarm possible attacks. Compared to traditional IDSs based on deep packet inspection, flow-based IDSs have to handle a considerably lower amount of data. Another important motivation for using flow-data in our research lays in the fact that flow-data is easily collected from network routers (several network nodes or just one central router) using standard protocols (such as Cisco NetFlow, Juniper Jflow, IETF IPFIX), without need to install additional software and collect data from every single computer on the network.

Therefore we have developed our approach by focusing on network flows. Approaches that rely on aggregated traffic metrics, such as *flow-based* approaches, show better scalability and therefore seem to be more promising.

In general, the techniques for Intrusion Detection (ID) fall into two major categories depending on the modeling methods used: misuse detection and anomaly detection. Misuse detection compares the usage patterns with known techniques of compromising computer security. Although misuse detection is effective against known intrusion types, it cannot detect new attacks that were not predefined. Anomaly detection, on the other hand, approaches the problem by attempting to find deviations from the established patterns of usage. Anomaly detection may be able to detect new attacks. However, it may also have a significant number of false alarms because the normal behavior varies widely and obtaining complete description of normal behaviors is often difficult. Architecturally, an intrusion detection system can be categorized into three types: host based IDS, network based IDS and hybrid IDS [15], [16]. A host based intrusion detection system uses the audit trails of the operating system as a primary data source. Network based intrusion detection systems, on the other hand, use network traffic information as their main data source. Hybrid intrusion detection systems use both methods [17].

In recent years, Neural Networks (NN) have been successfully used in the context of network intrusion detection. They have been extensively used in discriminating normal behavior from abnormal behavior in a variety of contexts. Neural networks have become a very useful technique to reduce information overload and improve decision making by extracting and refining useful information through a process of searching for patterns from the extensive collected data. Classification is a very common neural network task. In classification (section 3), we need to examine the features of newly presented objects and try to assign it to one of the predefined sets of classes. Supervised learning methods are applied to solve classification problems. Multilayer Feedforward NN (MLNN), and radial basis function (RBF) are representative supervised learning methods that can be applied to classification problems.

In this paper, flow-based anomaly IDS is implemented using two neural network stages. In many previous studies [18],[8],[19] the implemented system is a neural network based on DARPA [20] or KDD [21] dataset with the capability of detecting normal or abnormal traffic. In our study the existence of attacks and the attack type is classified by using extracted data from the labeled DARPA dataset. The restriction for the extracted data was that it should be limited to the type of data that can also be extracted from the router NetFlow data. This labeled data in the following text will be named labeled NetFlow dataset or simply NetFlow dataset, and the training procedure for neural networks is based on it.

This paper is organized as follows, section 2 present an overview of some of the previous works, section 3 provides a brief introduction about classification methods, section 4 explains the proposed system and feature selection, section 5 evaluates the proposed system, and section 6 discusses the experimental results followed by conclusions and future work plans.

2. Previous Work

Due to the increase in network speed, flow-based techniques attracted the interest of researchers, especially in analysis of high-speed networks. Day to day increase in network usage and load, have clearly pointed out that *scalability* is a growing problem. In this context, flow based solutions to monitor and, moreover, to detect intrusions help to solve the problem. They achieve, indeed, *data and processing time reduction*, opening the way to high-speed detection on large infrastructures. Gao and Chen [22] designed and developed a flow-based intrusion detection system. Karasaridis et al. [23], Shahrestani et al. [24] and Livadas et al. [25] proposed a concept for the detection of botnets in network flows. Sperotto et al. [1] provided a comprehensive survey on current research in the domain of flow-based network intrusion detection. A sound evaluation of a neural network based IDS requires high-quality training and testing datasets. The de facto standard is still the DARPA dataset created by Lippmann et al. [26]. Despite its severe weaknesses and the critique published by McHugh [25], it is still used. The KDD-99 [21] dataset can be regarded as another popular dataset. All these datasets were prepared for deep packet inspection and preprocessing was necessary to prepare flows. Sperotto et al. [28] created the first labelled flow based dataset intended for evaluating and training flow based network intrusion detection systems.

Several Neural Network approaches were implemented for Intrusion Detection systems based on NetFlow and DARPA [20] dataset. Muna Mhammad T. Jawhar [29] used Neural Networks and Fuzzy C-Mean (FCM) clustering algorithms. Rodrigo Braga [30] used OpenFlow and the SOM unsupervised neural network. Vallipuram and Robert [31] used back-propagation Neural Networks having all features of KDD (Knowledge Discovery in Databases) data [21]. Tie and Li [32] used the back propagation (BP) network with Genetic Algorithms (GAs) to enhance BP, for selected attacks and some features of the KDD dataset as input. Mukkamala, Andrew, and Ajith [33] used Back Propagation Neural Network with many types of learning algorithm. Jimmy and Heidar [34] used Neural Network for classification of unknown attacks. Dima, Roman and Leon [35] used MLP and Radial Based Function (RBF) Neural Network for classification of five types of attacks. Iftikhar, Sami and Sajjad [36] used Resilient Back propagation algorithm for detecting network intrusion attacks in a precise way by using the power of RPROP (Resilient Backpropagation) learning algorithm.

3. Detection and Classification Methods

Neural Networks (NNs) have recently attracted more attention compared to other techniques due to their strong discrimination and generalization abilities, when utilized for classification purposes [37], especially in the case of large amounts of data. An increasing amount of research in the last few years has investigated the application of Neural Networks to intrusion detection. If properly designed and implemented, Neural Networks have the potential to address many of the problems encountered by rule-based approaches. Neural Networks were specifically proposed to learn the typical characteristics of system's users and identify statistically significant variations from their established behavior. In order to apply this approach to Intrusion Detection, several steps have been taken:

Learning of Neural network: introducing data representing attacks and normal network flow to the Neural Networks to adjust the coefficients of these Networks automatically during the training phase. In other words, it will be necessary to collect data representing normal and abnormal behavior to train the Neural Networks.

Testing: after training has been accomplished, a certain number of performance tests with real network traffic and attacks have been conducted [38]. Instead of processing sequentially, Neural Network based models simultaneously explore several hypotheses through the use of several computational interconnected elements (neurons); this parallel processing may imply time savings in malicious traffic analysis [39]. In our study and based on prior research, two different neural network methods have been used for our intrusion detection system: Multilayer Feedforward neural network (MLFF), and Radial Basis Function Network (RBFN), [40]. NetFlow data has been used for training of these neural networks, while in many previous studies [18], [8], [19]; the implemented system is a neural network with the use of other features from the DARPA dataset that don't exist in NetFlow data [20].

3.1. Multilayer Feedforward NN

A multilayer feedforward NN is a structure that maps sets of input data onto a set of appropriate outputs. An MLFF consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one as shown in Figure 1. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLFF utilizes a supervised learning technique called backpropagation for training the network [41], [42] and has the ability to classify data that is not linearly separable [43].

The training algorithm rule repetitively calculates an *error function* for each input and backpropagates the error from one layer to the previous one. The weights for a particular node (w_{ij}) are adjusted in direct proportion to the error in the units to which it is connected.

The error function for pattern p is defined as to be proportional to the square of the difference of desired output d_{pj} (j-th node) and actual output y_{pj} for all nodes in output layer ($j=1,\ldots,n$). The backpropagation algorithm implements weight changes that follow the path of steepest descent on a surface in weight space. The height of any point on this surface is equal to the error measure E_p. This can be presented by showing that the derivative of the error measure with respect to the fact that each weight is proportional to the weight change dictated by the delta rule, with a negative constant of proportionality, i.e.,

(1)

The most used training algorithm is back propagation algorithm gradient descent (GDA) with the disadvantage of slow training. In other hand Levenberg-Marquardt [44], [45] is one of the accurate algorithms and faster than GDA, but consumes more memory space.

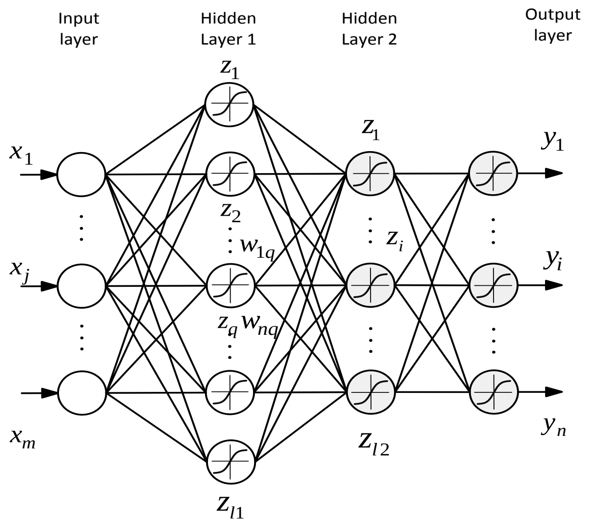

Fig. 1. Multilayer Feedforward NN, *m* – inputs (x_j), *l1,l2* – neurons in hidden layers (z_q), *n* – outputs (y_i), with different activation functions *f* for hidden and output layer

3.2. Radial Basis Function Network

The radial basis function network (RBFN) [46],[40] has the architecture of the instar-outstar model (Figure 2) and uses the hybrid unsupervised and supervised learning scheme, unsupervised learning in the input layer and supervised learning in the output layer.

The purpose of the RBFN is to pave the input space with overlapping receptive fields. For an input vector $\boldsymbol{x}$ lying somewhere in the input space, the receptive fields with centres close to it will be appreciably activated. The output of the RBFN is then the weighted sum of the activations of these receptive fields. The RBFN is designed to perform input-output mapping trained by examples, pairs of inputs and outputs ($\boldsymbol{x}$, $\boldsymbol{y}$). The hidden nodes in the RBFN have normalized Gaussian activation function.

$$z_q = g_q\left(\boldsymbol{x}\right) = \frac{R_q\left(\boldsymbol{x}\right)}{\sum_k R_k\left(\boldsymbol{x}\right)} = \frac{\exp\left(-\frac{\left|\boldsymbol{x}-\boldsymbol{m}_q\right|^2}{2\sigma_q^2}\right)}{\sum_k \exp\left(-\frac{\left|\boldsymbol{x}-\boldsymbol{m}_k\right|^2}{2\sigma_k^2}\right)} \tag{2}$$

where $\boldsymbol{x}$ is the input vector and z_q output of hidden layer. m_q and σ_q are the mean (an *m*-dimensional vector) and variance of the *q*-th Gaussian function in hidden layer.

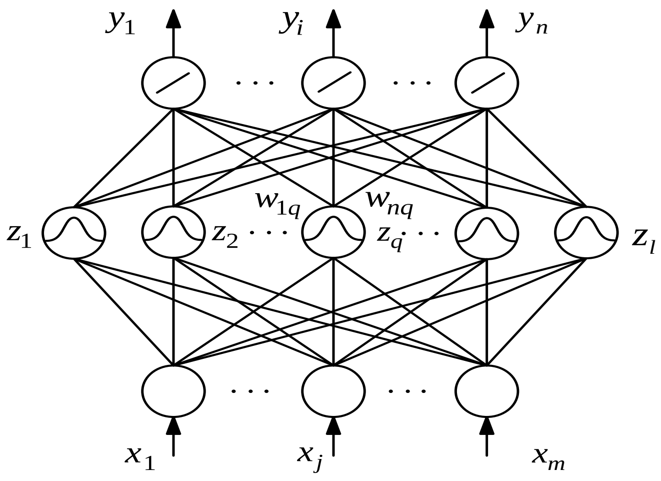

Fig. 2. Radial basis function network structure, with *m* inputs, *l* nodes in hidden layer, and *n* outputs

The output of the RBFN is simply the weighted sum of the hidden node output:

$$y_i = f_i\left(\sum_{q=1}^{l} w_{iq} z_q + \theta_i\right) \tag{3}$$

where $f_i(.)$ is the output activation function, generally linear function, and θ_i is the threshold value.

The weights in the output layer can be updated simply by using the delta learning rule (supervised learning). The unsupervised part of the learning involves the determination of the receptive field centres m_q and widths σ_q, $q = 1, 2... l$. The proper centres m_q can be found by unsupervised learning rules such as the vector quantization approach, competitive learning rules, or simply the Kohonen learning rule. Another learning rule for the RBFN with node-growing capability is based on the orthogonal least squares learning algorithm [47]. This procedure chooses the centres of radial basis functions one by one in a rational way until an adequate network has been constructed, or maximal number of nodes is reached.

The RBFN offers a viable alternative to the two-layer neural network in many applications of signal processing, decision making algorithms, pattern recognition, control, and function approximation. It has been shown that the RBFN can fit an arbitrary function with just one hidden layer [48], but they cannot quite achieve the accuracy of the back-propagation network. Although, RBFN can be trained several orders of magnitude faster than the back-propagation network, and this is a very important advantage in real or semi real time applications.

4. The Proposed NN Based Two Stages System

Our proposed system for intrusion detection and classification (Figure 3) consists of the following five main modules:

- Flow collector module.
- Feature preparation module.
- Anomaly detection module (NN stage one).
- Detection and classification module (NN stage two).
- Alert module.

NetFlow enabled routers are considered as external devices which permanently monitor network traffic, account statistics, and export flow-data to our system according to Cisco NetFlow [12], [13] or similar protocols.

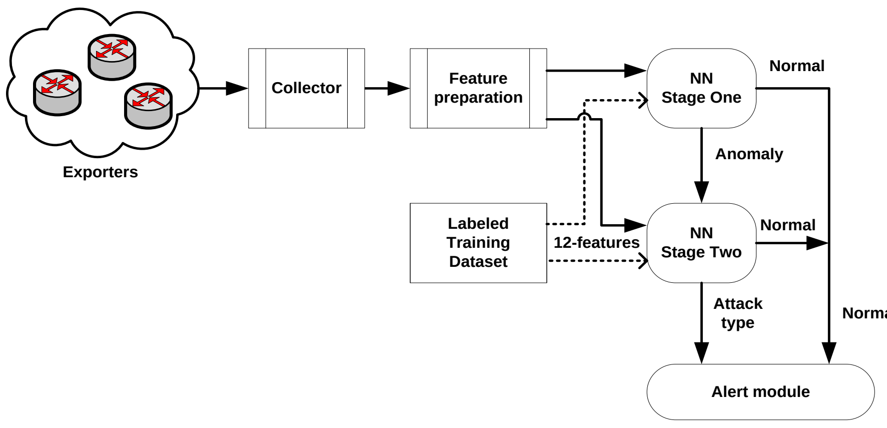

Fig. 3. Implemented two stages NN based system

4.1. Flow Collector Module

The operation of this module is to collect flow-data exported from one or several exporters. The received data need to be recognized by protocol and version (for instance NetFlow version 5 or 9, J-flow or IPFIX) and transform into an internal format. These data are constantly being sent to the Feature preparation module.

4.2. Feature Preparation Module

The Feature preparation module receives and processes flow-data sent from the Flow collection module. The main function is to prepare the features that are important for anomaly intrusion detection and classification modules. The features are combined in two groups: 7-*tuples and 12-tuples* that are passed to stage one and stage two detection modules respectively. Section 4.2.1 gives a more detailed explanation of the stage one features and section 4.2.2 demonstrate the added features for stage two modules. The selected features for both stages are suitable only for the selected attack in our study, and many other attacks have deviations of these features. Preprocessing must be done on all selected features before passing them to the detection modules; this phase involves normalizing all features by mapping all the different values for each feature to [0, 1] range.

4.3. Stage one Features

In stage one there are features that have most influence on the decision whether a traffic flow is normal or abnormal (an attack). It can also have the role to warn the network administrators that there is unusual traffic. A more detailed description of the vector of features is as following:

Average Flow Size: it provides a useful hint for anomalous events, such as port scan, and it is typically very small in order to increase the efficiency of attacks.

Average Packet Size: another factor is the size of each packet in the flow; low average size can be a sign of anomaly. For example, in TCP flooding attacks, packets of 120 bytes are typically sent.

Average Packet Number: one of the main features of DoS attacks is the source IP spoofing, which makes the task of tracing the attacker’s true source very difficult. A side effect is the generation of flows with a small number of packets, i.e. about 3 packets per flow. This differs from normal traffic that usually involves a higher number of packets per flow.

Number of different flows to the same Destination IP: This feature counts the number of flows to the same destination IP address. A high number of flows could mean a flood attack or a port scan attack.

Number of flows to different Destination Ports: also it has influence on detecting attacks. An abnormally large number of different destination ports means that the system is probably under attack (port scan attack).

Land: this feature is responsible for checking whether there is a land attack in the network or not.(i.e.SrcIP=DestIP,SrcPort=DestPort)

SYN - SYN/ACK: this feature was used by many researchers [49] to detect DoS Attack, by comparing the numbers of SYN and SYN/ACK packets that a host receives and returns respectively. Under normal conditions, the two numbers should be balanced since every SYN packet is answered by a SYN/ACK packet. Consequently, a high number of unanswered SYN packed is an indication of ongoing SYN flood.

4.4. Stage Two features

In addition to the stage one feature there are five additional features that can significantly improve both the detection rate and the classification of the type of attack. Warnings and alerts to network administrators are created from this stage. The classification done by the system can be verified by inspecting complete corresponding flows when new types of attacks appear or when the network administrator wants to verify the classification done by stage 2. The additional five features are:

Number of flows from the same source IP: attacker can send for example ICMP ping packets to every possible address within a subset and wait to see which machine respond.

Number of flows from different source IP: IP spoofing is widely used by attackers to attack the networks. A high number of different IP addresses to the same destination address within a short period of time is a strong sign for attack (DoS/ DDoS attack).

Number of flows to the same Destination Port: in some cases the attacker sends GET request to some ports only (ex. Port 80) to crash the server.

Number of flows from different source Port: As IP spoofing is generated by DDoS attack; ports can also be changed during an attack at random.

Protocol type (TCP, UDP, and ICMP): knowing the protocol type in combination to the all previous features can help to determine the type of attack.

All added features have an important role in improving detection and classification of attacks. In order to have a full picture of what's going on in the network, a full history of the source address, used ports, and protocol type should be considered in many cases.

4.5. Anomaly Detection Module (Stage one NN1)

Anomalies in our system are defined as unusual activities in the network. The purpose of this module is to find out such activities using a small number of features extracted from NetFlow raw data. For the neural network that used in stage one the algorithm below is a simplified general description of the detection process.

```
Algorithm: Anomaly detection module

Loop
Read 7-tuple inputs from the feature
preparation module.
Feed parameters to the NN1
If the data is "normal", then
Assign "01" to the output of the NN1 as normal
traffic
      Else
Assign "10" to the output of NN as anomaly
activity.
Call stage two Procedure.
End Loop
```

The number of input nodes of the NN1 corresponds to the number of the selected features of the NetFlow dataset for the first stage (7 Features). The implemented NN1 includes one input layer, one hidden layer and an output layer of 2 nodes (01 as normal traffic, and 10 as anomaly traffic). The number of nodes in the hidden layers has been determined based on the back propagation (BP) computation process and the process of trial and error.

4.6. Attack Detection and Classification Module (Stage two NN2)

In our work, neural networks were used for attack classification. It was crucial to successfully train the network with reliable data gained from classified attacks and then to test the trained network. The result from the network is classified into one of five possible categories. Table 1 maps these categories to the actual outputs from Neural Network module NN2.

Table 1. Neural Network Classified Categories

No	Category	NN2 outputs
1	DoS/ DDoS Attack	10000
2	Port Scan Attack	01000
3	Land Attack	00100
4	Other/unknown Attack	00010
5	Normal	00001

The number of input nodes to the NN2 corresponds to the number of the selected features from NetFlow dataset for the second stage NN2 (12 Features). The implemented neural network includes one input layer, one hidden layer and an output layer of 5 nodes (Table 1 contains the descriptions of the outputs). The numbers of nodes in the hidden layers has been determined based on the back propagation (BP) computation process and the process of trial and error. Table 2 describes the detection and classification procedure.

Table 2. Detection and Classification Procedure

```
Stage two Procedure

Begin
Read corresponding 12/tuple inputs for NN2
If the data is "normal", then
Assign "00001" to the output of NN2
Else
Assign appropriate attack category to the NN2 outputs according to Table 1.
End
```

4.7. Alert Module

This is the final stage of the proposed system. This stage involves identifying the events that occurred. It also presents status of the observed network to the administrator and creates alarms when appropriate.

5. Experimental Results

The experiments were performed in MATLAB, using neural network toolbox which implements several training algorithms including Resilient Backpropagation, Radial Basis Function net, and Levenberg-Marquardt.

Considering the fact that the previous works commonly use DARPA dataset as a trusted labeled dataset for intrusion detection research, we built our NetFlow dataset as a subset of DARPA dataset. Since DARPA dataset is in form of TCP dump data, therefore we created flows from the raw DARPA dataset using a modified version of softflowd [50]. In our dataset, a flow closely follows the NetFlow v5 definition and has the following form:

F= {IPsrc,IPdst,Psrc,Pdst,Pckts,Octs,Flags,Protcl,Tstart,Tend}

It represents the unidirectional communication from the source IP addresses IPsrc and port number Psrc to the destination IP address IPdst and port number Pdst, using protocol type Protcl. The Pckts and Octs give the total number of packets and octets transferred during this communication. The field Flags is related to the TCP header flags which are computed as a binary OR of TCP flags in all packets of the flow. The start and end time of the flow are given by Tstart and Tend respectively, in millisecond resolution. The extracted flows are labeled according to the log file of DARPA dataset and used to prepare all selected features, which used to train NN1 and NN2.

In the experimental stages we have used different number of iterations and hidden layers to determine the level of training. This test has been done to find out when the neural network was trained properly to detect attacks. This test has also provided the background for choosing the number of hidden layers and iterations for the training of the neural network for the last experiments.

The experiments show that Levenberg-Marquardt is the best training algorithm because it takes less time, low number of epochs and has good performance and high accuracy. The Detection Rate (DR) and False Positive rate (FP) have been calculated for different scenarios. The considered scenarios in our experiments are as follows:

1. Anomaly Detection Phase: This case includes detection of attacks by deciding whether flows are normal or abnormal. The training and testing have been performed with seven selected features. One input layer, one hidden layer, and one output layer have been used in NN1 module.

The experiments have three phases: a training phase, a validation phase and a testing phase. All experiments in this stage were done with 96852 records of attack traffic, and 48556 of normal traffic. 21816 records were used for testing the neural network and it contains 14527 records of attack traffic, and 7289 records of normal traffic. Table 3 and Figure 4 show the performance of anomaly detection module. Detection rate and false positive rate has been calculated according to the following formulas:

$$Detection\ Rate = \frac{number\ of\ detected\ attacks}{total\ number\ of\ attacks} \times 100[\%] \quad (4)$$

$$False\ positive\ Rate = \frac{number\ of\ normal\ classifed\ as\ attacks}{total\ number\ of normal\ traffic} \times 100[\%] \quad (5)$$

Table 3. The Results of Anomaly Detection phase

Training Algorithm / Parameters	Resilient Backpropagation Test 1	Levenberg-Marquardt Test2	Radial Basis Function Net Test 3
Training dataset	101806		
Validation Data	21816		
Testing set	21816		
Hidden Layer	50	50	20
Number of detected attacks(14527)	13468	13684	13234
Number of detected traffic as normal(7289)	6757	6866	6640
Detection Rate	92.7%	94.2%	91.1%
False positive Rate	3.6%	3.4%	5.1%

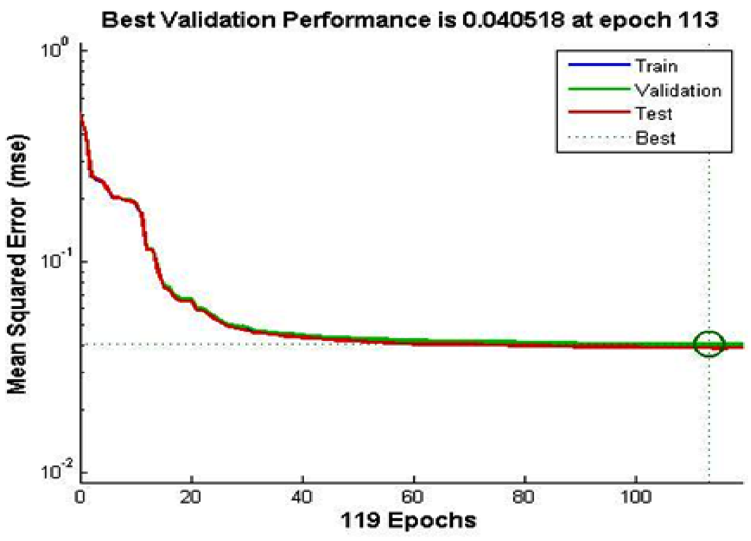

Fig. 4. Performance of the anomaly detection module (stage 1)

2. Detection and Classification Phase: This scenario includes detection of packets and their classification as normal with a larger number of inputs, or one of the four main attack types (DoS/DDoS, Port Scan, Land attack, or other/unknown attack). 12 selected features were used for training and testing the neural network. One input layer, one hidden layer, and output layer. All experiments in this stage have been done with 96852 records of attack traffic, and 48556 of normal traffic. 21816 records were used for testing the neural network and it contains 14527 records of attack traffic, and 7289 records of normal traffic. Table 4 and Figure 5 show the performance of the detection and classification module. Detection rate and false positive rate was also calculated. During the testing phase, the classification rate of each attack types was calculated according to the following formula:

$$Classification\ Rate = \frac{number\ classifed\ \ attacks}{total\ number\ of attack\ type} \times 100[\%] \tag{6}$$

The best result of the classification module during the test phase is shown in table 5.

Table 4. Results of detection and classification module

Training Algorithm / Parameters	Resilient Backpropagation Test 1	Levenberg-Marquardt Test2	Radial Basis Function Net Test 3
Training dataset	96852		
Validation Data	21816		
Testing set	21816		
Hidden Layer	80	80	40
Number of detected attacks(14527)	14439	14443	13858
Number of detected traffic as normal(7289)	7145	7246	6953
Detection Rate	99.4%	99.42 %	95.4%
False positive Rate	0.3%	0.32%	2.6%

Table 5. Classification results of Stage 2

Attack name	Total number of attacks	Number of classified attacks	Classification rate
DoS	4490	4490	100%
Port scan	9929	9919	99.9%
Land	85	85	100%
Unknown	23	18	78%

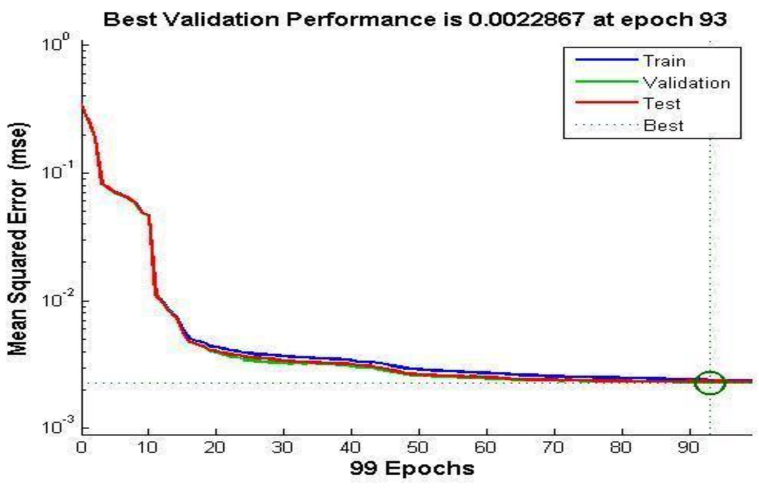

Fig. 5. Training performance of the detection and classification module (stage 2)

6. Discussion of Results

Two stages of Anomaly detection systems using neural networks and based on NetFlow dataset have been proposed and tested. Three different training algorithms (Resilient Backpropagation, Radial Basis Function net, and Levenberg-Marquardt) were used for training of both neural network stages. Anomaly detection stage (NN1) was trained until the best validation performance 0.0405 was met at epoch 113 as shown in Figure 4. The results in Table 3 show that the detection rate is 94.2% with false positive of 5.8%. Results from detection and classification stage (NN2), show significantly larger improvement of prediction accuracy than the Anomaly detection phase. Figure 5 shows that, the best validation performance 0.0022 was met at epoch 93.Table 4 shows that the detection rate is relatively high at 99.42% for MLP, and 95.4% for RBF detection algorithm. The false alarms were as low as 0.58% in MLP neural network and 4.6% in RBF neural network. Table 5 shows that, 100% of DoS attack, 99.9% of port scan attack, 100% of land attack, and 78% of unknown attack were detected and classified correctly by using stage two neural networks. The analysis of both stages results shows that, MLP with Levenberg-Marquardt is found to be fast compared to Resilient Backpropagation, has low memory consumption compared to Radial Basis Function, and has a lower false alarm rates.

6.1. Comparison of Results

In this section, we compare our results with the other researcher's results available in the literature. Vallipuram and Robert [31] used backpropagation neural network based on KDD'99 dataset, the detection rate was 86% with high false alarm rate at 14%.

Mukkamalaa [51] used backpropagation neural networks with the use of DARPA dataset, and the detection rate was 97.04% and false alarm rate of 2.06%. Dima, Roman, and Leon [35] used both MLP and RBF neural network with KDD'99 as a dataset, their results was 93.2% for RBF, and 92.2% for MLP with 7.2% as false alarm. Muna Mohammad [29] used MLP AND Fuzzy-clustering algorithm with the use of DARPA dataset, the detection rate was 99.9% and low false alarm rate 0.1%. Rodrigo Braga [30] used unsupervised neural network with flow dataset and the results for detection rate was 99.11% with false alarm rate of 0.99%. Govindarajan and Chandrasekaran [53] used neural based hybrid classification methods and they used flow dataset, their results were 96.67% for abnormal traffic, and 96.54% for normal traffic. Prasanta, Bhattacharyya, Borah and Jugal [54] used both supervised and unsupervised neural network with the use of flow dataset and KDD'99 dataset, the detection rate were 99.1% for flow dataset and 92.26% for KDD'99 dataset with false rate of 0.9%.

In our research with two neural network stages based on extracted NetFlow dataset, we have achieved the detection rate at 99.4% for MLP, and 94.6% for RBF neural network with low false alarm rate at 0.6%. Figure 6 shows that our proposed system is greatly competitive and performs significantly better Detection Rate (DR). From Table 6, we observe and conclude that our system with two neural network stages based on flow dataset and the use of a small number of extracted features can effectively and efficiently detect and classify both known and unknown attacks. The obtained false alarm rate is low compared to other methods that use different techniques and different datasets.

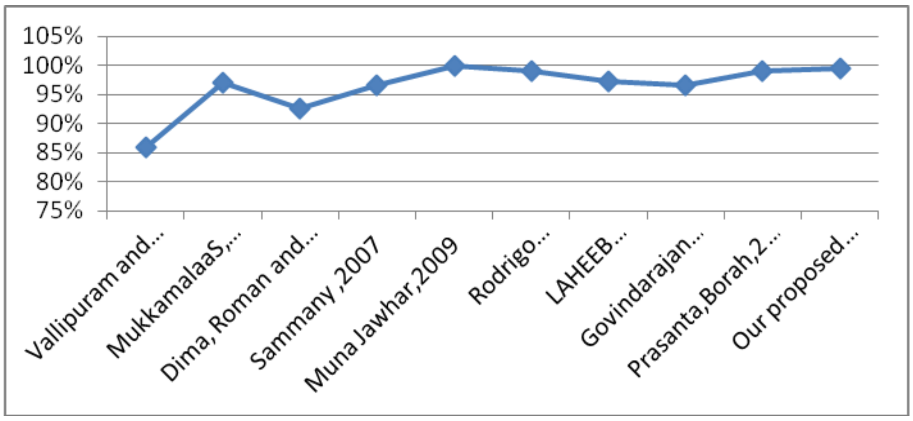

Fig. 6. Detection Rate on Different Datasets for IDSs

Table 6. Comparison of Intrusion Detection Systems Using NN.

Research	NN type	Dataset used	Detection Rate (%)	False Alarm Rate
Vallipuram and Robert,2004	Backpropagation	KDD-99	86% for normal traffic	14%
[51] Mukkamalaa S., 2005	Backpropagation	DARPA	97.04%	2.06%
Dima, Roman and Leon,2006	MLP and RBF	KDD-99	93.2% using RBF and 92.2% using MLP	8.8%
[52] Sammany M,2007	2 hidden layers MLP	DARPA	96.65%	3.35%
Muna Mhammad T. Jawhar,2009	MLP and Fuzzy C-Mean (FCM) clustering algorithms	DARPA	99.9%	0.1%
Rodrigo Braga,2010	SOM	Open flow dataset	99.11%	0.99%
Laheeb Mohamad Ibrahim,2010	Distributed Time-Delay Neural Network	KDD-99	97.24%	2.76%
[53] Govindarajan , Chandrasekaran,2011	hybrid classification methods	Flow data set	96.67% for abnormal traffic, and 96.54% for normal traffic	3.33%
[54] Prasanta Gogoi, Bhattacharyya,Borah and Jugal Kalita,2013	Supervised and unsupervised neural network	Packet Level and Flow Level dataset, KDD-99	99.1% for packet/flow level data, and 92.26% for KDD	0.9%
Our proposed IDSs	Two stage neural network	NetFlow dataset	94.2% for stage one NN, and 99.4% for stage two NN	0.6%

7. Conclusion and Future Work

In this paper we presented flow based intrusion detection and classification method using two neural networks for separate tasks. One neural network detects traffic anomalies that can be attacks and the other one classifies attacks if they exist. This system can easily be extended, configured, and/or modified by replacing some features or adding new features for new types of attacks.

The training of the NNs modules requires a very large amount of NetFlow data with known types of attacks and considerable time to ensure that the results from the NNs are accurate. The changes in patterns of usage of the network should not be undetected, but at the same time, these changes are isolated to NN1. Appearance of new patterns of attack affects only classification in NN2, which is the main reason to have two stage neural networks instead of one. Consequently, the events that require retraining for the two networks are completely independent. Experiments with different NNs were crucial to define the NN which yields the best classification and training speed results for both NN stages.

The experimental results of the proposed method prove that the use of NetFlow dataset and extracting only features that significantly contribute to intrusion detection gives promising results. The obtained detection rate (94.2% for anomaly detection at stage one, and 99.4% for classification at stage two) is remarkably good compared to other approaches, which use larger training sets [20]. These results are comparable to the best researches that are based on a similar approach using the same type of training dataset.(Table 6).

The multilayer Feedforward neural network has a better classification ability compared to RBFN, but memory and time consumption is 3-5 times greater. Otherwise, RBFN has a simple architecture and hybrid learning algorithm which leads to less time/memory consumption and it is better for working in real-time and for retraining with new data.

Our future research will be directed towards developing a more accurate model that can be used in real-time for detecting and classifying anomaly with minimum features and less time for training.

References

1. A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller.” Anoverview of ip flow-based intrusion detection”. IEEE Communications Surveys &Tutorials, 12(3):343–356, (2010).
2. H. Lai, S. Cai, H. Huang, J. Xie, and H. Li. A parallel intrusion detection system for high-speed networks. In Proc. of the 2nd Int. Conf. Applied Cryptography and Network Security, pages 439–451,(May 2004).
3. M. Gao, K. Zhang, and J. Lu. Efficient packet matching for gigabit network intrusion detection using TCAMs. In Proc. of 20th Int. Conf. on Advanced Information Networking and Applications (AINA’06), pages 249–254, (2006).

4. W. de Bruijn, A. Slowinska, K. van Reeuwijk, T. Hruby, L. Xu, and H. Bos. Safe- Card: A Gigabit IPS on the Network Card. In Proc. of the 9th Int. Symp. on Recent Advances in Intrusion Detection (RAID '06), pages 311–330,(2006).
5. G. Vasiliadis, S. Antonatos, M. Polychronakis, E. P. Markatos, and S.Ioannidis.Gnort: High Performance Network Intrusion Detection Using Graphics Processors.In Proc. of the 11th Int. Symp. on Recent Advances in Intrusion Detection, pages 116–134, (2008).
6. M. Roesch. Snort, intrusion detection system. http://www.snort.org, Sept. 2010.
7. V. Paxson. Bro: a system for detecting network intruders in real-time. Computer Networks, 31(23–24):2435–2463, (1999).
8. J. Ryan, M. Lin, and R. Miikkulainen, "Intrusion Detection with Neural Networks," *AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAI Workshop,* Providence, RI, pp. 72-79, (1997).
9. Internet2 NetFlow: Weekly Reports. NetFlow.internet2.edu/weekly, (April 2008).
10. Net flow sensor: nfsen.sourceforge.net
11. D. Plonka. Flowscan. Available at: www.caida.org/tools/utilities/flowscan/, (April 2008).
12. B. Claise. Cisco Systems NetFlow Services Export Version 9. Request for Comments: 3954, (October 2004).
13. Cisco IOS NetFlow Configuration Guide. Available at: www.cisco.com, (April 2008).
14. IP Flow Information Export Working Group. Available at: www.ietf.org/html.charters/ipfix-charter.html, (April 2008).
15. J., Muna. M. and Mehrotra M., "*Intrusion Detection System : A design perspective"*, 2rd International Conference On Data Management, IMT Ghaziabad, India. (2009).
16. M. Panda, and M. Patra, "*Building an efficient network intrusion detection model using Self Organizing Maps*", proceeding of world academy of science, engineering and technology, Vol. 38.(2009).
17. M. Khattab Ali, W. Venus, and M. Suleiman Al Rababaa, "*The Affect of Fuzzification on Neural Networks Intrusion Detection System"*, IEEE computer society,(2009).
18. James Cannady, "Artificial neural networks for misuse detection," Proceedings of the 1998 National Information Systems Security Conference, Arlington, VA,(1998).
19. Srinivas Mukkamala, "Intrusion detection using neural networks and support vector machine," *Proceedings of the 2002 IEEE International* Honolulu, HI, (2002)
20. DARPA1998. Available at: http://www.ll.mit.edu/IST/ideval/docs/1998.
21. KDDCup1999. Available at: http://kdd.ics.uci.edu/databases.
22. Y. Gao, Z. Li, and Y. Chen, "A DoS Resilient Flow-level Intrusion Detection Approach for High-speed Networks," in *Proc. of the 26th IEEE International* Conference on Distributed Computing Systems, Washington, USA, p.39,(2006).
23. A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale botnet detection and characterization," in Proc. of the first conference on Hot Topics in Understanding *Botnets (HotBots '07)*, Berkeley, CA, USA, p.7,(2007).
24. A. Shahrestani, M. Feily, R. Ahmad, and S. Ramadass, "Architecture for Applying Data Mining and Visualization on Network Flow for Botnet Traffic Detection," in Proc. of the International Conference on Computer Technology and Development, Washington, DC, USA, pp. 33 – 37,(2009).
25. C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Using Machine Learning Techniques to Identify Botnet Traffic," in *2nd IEEE LCN Workshop on Network Security*, pp. 967 – 974,(2006).
26. R. P. Lippmann *et al.*, "Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation," in *Proc. of the DARPA Information* Survivability Conference and Exposition, pp.12 – 26,(2000).
27. J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999

DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262 – 294, (2000).

28. A. Sperotto, R. Sadre, F. Vliet, and A. Pras, "A Labeled Data Set for Flow-Based Intrusion Detection," in Proc. of the 9th IEEE International Workshop on IP *Operations and Management*, Berlin, Heidelberg, pp. 39 – 50, (2009).
29. Muna Mhammad T. Jawhar," Design Network Intrusion Detection System using hybrid Fuzzy- Neural Network", International Journal of Computer Science and Security, Volume (4),(2009)
30. Rodrigo Braga," Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow", 35th Annual IEEE Conference on Local Computer Networks LCN 2010, Denver, Colorado 6.(2010)
31. M. Vallipuram and B. Robert, *"An Intelligent Intrusion Detection System based on Neural Network"*, IADIS International Conference Applied Computing.(2004).
32. T. Zhou and LI Yang, *"The Research of Intrusion Detection Based on Genetic Neural Network"*, Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, IEEE.(2008).
33. S. Mukkamala, H. Andrew Sung, and A. Abraham, *"Intrusion detection using an ensemble of intelligent paradigms"*, Journal of Network and Computer Applications 28. pp167–2,(2005).
34. S. Jimmy and A. Heidar, "*Network Intrusion Detection System using Neural Networks",* IEEE computer society.Vol.05,pp 242-246,(2008).
35. D. Novikov, V. Roman Yampolskiy, and L. Reznik, *"Anomaly Detection Based Intrusion Detection"*, IEEE Third International Conference on Communication, Networking & Broadcasting. pp 420-425,(2006).
36. I. Ahmad, S. Ullah Swati and S. Mohsin, "*Intrusions Detection Mechanism by Resilient Back Propagation (RPROP)"*, European Journal of Scientific Research ISSN 1450-216X Vol.17 No.4, pp.523-531. (2007).
37. M. Al-Subaie, "*The power of sequential learning in anomaly intrusion detection"*, degree master thesis, Queen University, Canada. (2006).
38. D. Novikov, V. Roman Yampolskiy, and L. Reznik, *"Artificial Intelligence Approaches for Intrusion Detection"*, IEEE Long Island Systems Applications and Technology Conference. pp 1-8, (2006).
39. S. Lília de Sá, C. Adriana Ferrari dos Santos, S. Demisio da Silva, and A. Montes, *"A Neural Network Application for Attack Detection in Computer Networks",* Instituto Nacional de Pesquisas Espaciais – INPE, BRAZIL.(2004).
40. Chin-Teng Lin, C. S. George Lee: Neural fuzzy systems: a neuro-fuzzy synergism to intelligent, Prentice-Hall, Inc. Upper Saddle River, (1996).
41. Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, (1961).
42. Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: MIT Press, (1986).
43. Cybenko, G. "Approximation by super positions of a sigmoidal function", *Mathematics of Control, Signals, and Systems*, 2(4), 303–314,(1989).
44. Hagan, M.T., H.B. Demuth, and M.H. Beale, Neural Network Design, Boston, MA: PWS Publishing, (1996).
45. Hagan, M.T., and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," IEEE Transactions on Neural Networks, Vol. 5, 6, pp. 989–993, (1994).
46. Moody, J., and Darken C.: "Fast learning in networks of locally-tuned processing units", Neural Comput. MIT Press Cambridge, MA, USA, 1:281-294. (1989).

47. Chen, C. H., ed. Neural Networks in Pattern Recognition and Their Applications. River Edge, NJ: World Scientific, (1991)
48. E. J. Hartman, J. D. Keeler, and J. M. Kowalski. Layered neural networks with Gaussian hidden units as universal approximations. Neural Computation, 2:210–215, (1990)
49. H. Wang, D. Zhang, and K. G. Shin, .SYN-dog: Sniffing SYN Flooding Sources, In Proc. of 22nd International Conference on Distributed Computing Systems, Vienna, Austria, (2002).
50. Softflowd. Available at:: https://code.google.com/p/softflowd/.
51. Mukkamala, S.; and Sung, A.H. Feature selection for intrusion detection using neural networks and support vector machines. *Transportation Research Record*, 1822, 33-39. (2003).
52. Sammany, M.; Sharawi, M.; El-Beltagy, M.; and Saroit, I. Artificial neural networks architecture for intrusion detection systems and classification of attacks. *Accepted for publication in the 5th international conference INFO2007*, Cairo University.(2007).
53. M.Govindarajan, and RM.Chandrasekaran," Intrusion detection using neural based hybrid classification methods", computer networks journal. Volume 55, issue 8, pp.1662-1671, (2011).
54. Prasanta Gogoi, D.K. Bhattacharyya, B. Borah, and Jugal K. Kalita "MLH-IDS: A Multi-Level Hybrid Intrusion Detection Method", the Computer Journal Advance Access published May 12, (2013).

Yousef Abuadlla Received his BSc degree in Computer Engineering from University of Tripoli, Libya in 1993, and his MSc degree in Computer Engineering from University of Belgrade, Serbia in 2003. Currently he is a PhD candidate in Computer Engineering at the School of of Electrical Engineering, University of Belgrade. His research interests include network security, intrusion detection system and data mining.

Goran Kvascev was born in 1975. He received the Engineer, Magister and Ph.D. degrees from School of Electrical Engineering, University of Belgrade, Serbia, in 2000, 2005, and 2012, respectively. Since 2000, he has been with the Control Department, School of Electrical Engineering, where he was a research and teaching assistant, and became an Assistant Professor in 2013. His main areas of research interest are real-time and industrial process control, neural networks, robust system identification, thermal power plant control.

Slavko Gajin received dipl. Eng., MS and PhD degrees from University of Belgrade, School of Electrical Engineering, Serbia, in 1993, 1999, and 2007, respectively. He is a director of Belgrade University Computer Center, where he started working as a network engineer since he received bachelor's degree. He is a professor at the Department of Computer Engineering and Computer Science at the School of Electrical Engineering, University of Belgrade. His current research interests include the modelling and performance analysis of communication systems, routing algorithms in NoC and interconnection networks, computer network management, monitoring and performances.

Zoran Jovanović received his dipl. eng., MS and PhD degrees from University of Belgrade, School of Electrical Engineering in 1978, 1982, and 1988, respectively. He is the Director of the National Research and Education Network of Serbia (AMRES) and professor at the Department of Computer Engineering and Computer Science at School of Electrical Engineering, University of Belgrade. His current research interests include parallel computing, networking, concurrent and distributed programming.

Received: April 15, 2013; Accepted: May 25, 2014