Computer Science and Information Systems 20(4):1519–1540 https://doi.org/10.2298/CSIS230418058W

Intrusion Detection Model of Internet of Things Based on Deep Learning ⋆

Yan Wang, Dezhi Han, and Mingming Cui

College of Information Engineering
Shanghai Maritime University , China
202130310093@stu.shmtu.edu.cn
dzhan@shmtu.edu.cn
mmcui@stu.shmtu.edu.cn

Abstract. The proliferation of Internet of Things (IoTs) technology is being seriously impeded by insecure networks and data. An effective intrusion detection model is essential for safeguarding the network and data security of IoTs. In this paper, a hybrid parallel intrusion detection model based on deep learning (DL) called HPIDM features a three-layer parallel neural network structure. Combining stacked Long short-term memory (LSTM) neural networks with convolutional neural network (CNN) and SK Net self-attentive mechanism in the model allows HPIDM to learn temporal and spatial features of traffic data effectively. HPIDM fuses the acquired temporal and spatial feature data and then feeds it into the CosMargin classifier for classification detection to reduce the impact of data imbalance on the performance of the Intrusion Detection System (IDS). Finally, HPIDM was experimentally compared with classical intrusion detection models and the two comparative models designed in this paper, and the experimental results show that HPIDM achieves 99.87% accuracy on the ISCX-IDS 2012 dataset and 99.94% accuracy on the CICIDS 2017 dataset. In addition, it outperforms other comparable models in terms of recall, precision, false alarm rate (FAR), and F1_score, showing its feasibility and superiority.

Keywords: intrusion detection, deep learning (DL), Long short-term memory (LSTM), convolutional neural network (CNN), SK Net self-attentive mechanism.

1. Introduction

With the rapid development of wireless sensor networks (WSN), 5G communication technology, big data processing technology, and artificial intelligence technology, the IoTs have been widely used and opened a new era of the Internet of Everythings [27]. According to a white paper released by Cisco, global mobile data traffic has surged by 17 times over the last five years, with nearly 650 million new mobile devices added.

In the era of the IoTs, everything is interoperable, which also means that cyber-attacks can easily invade the real world. Data shows that in the past 10 years, cyber attacks have evolved from individual hackers to organized cyber armies, and the areas of attack are becoming larger and larger, from Internet computers and information networks to military

⋆ The Top-notch Innovative Talent Training Program for Graduate students of Shanghai Maritime University under Grant 2021YBR008

and civilian critical information infrastructures. The use of standards and specifications for the IoTs has become a crucial factor in the development of the industry, the premise of which is to ensure the security of the network and data. An IDS is software or hardware that detects malicious activity on a specific computer or network [37], [14]. IDS reacts to detected intrusions in real-time and alerts administrators and is used to secure the network.

As network attacks become more sophisticated and efficient, traditional intrusion detection methods based on machine learning (ML) are insufficient in detecting and preventing such attacks. As a result, new network attack defense methods must be explored. Intrusion detection technology based on DL has garnered significant attention from both academic and business communities, providing a novel idea for the network security research of the IoTs [46]. DL-based IDS identifies suspicious network activity, prevents hackers from gaining access, and notifies users. They usually have well-known labels and common attack formats. This helps protect against risks such as data breaches. By analyzing traffic more accurately, reducing the number of false alarms, and assisting security teams in distinguishing malicious from legitimate network activity, DL, CNN, and recurrent neural networks (RNNs) can be used to develop smarter IDS [25]. The primary contributions of our paper are as follows.

(1) A DL-based hybrid parallel intrusion detection model (HPIDM) is proposed. The three-layer parallel neural network structure of HPIDM is composed of stacked LSTM and CNN as well as the SK Net attention mechanism, which enables HPIDM to learn the Spatial and temporal features of traffic data effectively. Not only can HPIDM automatically and fully learn the spatial and temporal features of traffic data, but it can also effectively address the issue of data imbalance through multiple-feature fusion.

(2) Based on the HPIDM, two comparison versions are proposed. Comparison model 1 is to change the Fully Convolutional Network(FCN) module of the first layer to a conventional CNN model on HPIDM to verify the effectiveness of the FCN module, and comparison model 2 is to change the combination of the CNN and the stacked LSTM module of the second layer to a conventional CNN module to verify the effectiveness of the stacked LSTM.

(3) The results of the ablation experiments showed that the experimental accuracy of the HPIDM on the ISCX 2012 dataset was 99.87%, which was 0.06%, 0.05%, 0.11%, 0.13%, and 0.12% higher than the TPCNN, TPCNN-C, CROSS_CNN, CROSS_CNN_LSTM, and HPM models respectively, and 0.13% and 0.12% higher than the comparison models model1 and model2 by 0.13% and 0.02% respectively. The experimental accuracy on the CIC-IDS 2017 dataset was 99.94%, which was 0.03%, 0.02%, 0.02%, 0.03%, and 0.04% higher than the TPCNN, TPCNN-C, CROSS_CNN, CROSS_CNN_LSTM, and HPM models, respectively. Moreover, the HPIDM outperforms its counterparts in terms of accuracy, recall, precision, FAR, F1 score, and other related metrics. This validates its feasibility and superiority, as well as the effectiveness of the FCN module and the stacked LSTM module in the HPIDM.

Based on the abbreviations in this document, important symbols are explained in this section using Table 1. The remainder of this paper is structured as follows. In Second 2, the DL-based approach, intrusion detection models, and Back Propagation (BP) neural networks are briefly introduced. Section 3 provides a detailed description of the dataset, data preprocessing algorithms, and the proposed intrusion detection model. The experimental environment and parameters are first presented, and then ablation experiments on both the ISCX-IDS2012 and CICIDS2017 datasets are conducted in Section 4. Finally, the full paper is summarized, and future work prospects are in Section 5.

Table 1. Explanation of abbreviations.

Abbreviations	Explanation
IoTs	Internet of Things
DL	deep learning
LSTM	Long short-term memory
CNN	convolutional neural network
SK Net	Selective Kernel Networks
IDS	Intrusion Detection System
FAR	false alarm rate
WSN	wireless sensor networks
ML	machine learning
RNNs	recurrent neural networks
FCN	Fully Convolutional Network
RFF	radio frequency fingerprinting
FPN	Feature pyramid network

2. Related Work

Early IDS used a single-layer architecture that could only detect misuse or anomaly attacks. To accurately identify misuse and anomaly attacks, Zhang et al. [43] propose an adaptive serial hierarchical attack identification system (SHIDS) that can automatically train a new classifier and adaptively modify its structure after the new classifier is trained. However, the adaptive learning capability is limited and is not able to learn the features of malicious traffic autonomously. Hall et al. [24] propose a new approach to integrate radio frequency fingerprinting (RFF) technology into a wireless IDS. This approach can effectively control the unauthorized use of network resources by media but is superior to the aging of transceivers and other reasons, which can affect the classification success rate and is relatively homogeneous in terms of scalability.

With the development of machine learning techniques, intrusion detection is gradually shifting towards machine learning-based methods. These methods automatically identify new attacks by learning patterns of attack behavior from large amounts of network data and are thus better able to respond to unknown attack methods [2], [22]. Dina et al. [6] propose a comprehensive summary of machine learning-based intrusion detection methods proposed in the literature over the past decade: artificial neural networks, association rules, fuzzy association rules, Bayesian networks, clustering, decision trees, integrated learning, evolutionary computation, hidden Markov models, inductive learning, etc. Sarnovsky et al. [30] propose a hierarchical IDS based on a primitive symmetric combination of machine learning methods and knowledge-based methods to support the detection of the severity of existing types and novel network attacks.

But with the rise of emerging technologies such as cloud computing and the IoTs, intrusion detection is also facing new challenges. For example, virtualization technologies in cloud computing environments may result in traditional intrusion detection methods being unable to accurately distinguish traffic between virtual machines; and the large number of devices in the IoTs may pose a large data volume and complexity challenges for intrusion detection [4], [28]. Conventional machine learning methods can no longer meet cybersecurity needs, and DL networks with end-to-end features can solve new types of malicious traffic feature extraction problems. Researchers are therefore applying DL to the field of intrusion detection to improve the accuracy and real-time performance of intrusion detection, as well as to better adapt to the changing network environment. Tao et al. [33] propose a deep reinforcement learning approach to detect malicious attacks in aerial computing networks of UAVs. Fatani et al. [7] propose an advanced feature extraction and selection method for an IoTs IDS based on DL and Aquila optimizer. Cai et al. [3] propose a hybrid parallel DL model for efficient intrusion detection based on metric learning, which improves the detection accuracy of malicious traffic.

Although DL-based intrusion detection techniques are currently the main techniques for network traffic intrusion, a major drawback is that they are highly dependent on feature design and have a high FAR, which does not perform well in real-world applications [35], [17]. Researchers have made several efforts to improve the detection and classification performance of malicious traffic, however, they have neglected the accuracy of malicious sample classification. To this end, this paper abstracts the CNN underlying intrusion traffic data into high-level features, extracts sample features autonomously, interweaves stacked LSTM and multi-scale convolutional operations into the neural network, automatically learns the spatial and temporal features of the traffic data adequately through multiple feature fusions, and optimizes the network parameters to converge the model through a stochastic gradient descent algorithm, and finally performs a sample test to detect the network's intrusion behavior. Simulation results show that the method proposed in this paper has high detection accuracy and true positive rate, as well as a low FAR.

3. Models and Methods

This section introduces the design of a hybrid parallel neural network model, HPIDM, which leverages DL techniques to improve the performance of IoTs anomaly traffic detection.

3.1. Data Pre-processing

In this study, the ICSX 2012 and CIC-IDS 2017 datasets are utilized that include both header and payload information and are considered more novel than the KDD99 dataset [34], [13]. Before conducting experiments, data preprocessing is performed to reduce the interference of noise, missing values, and inconsistent data. The preprocessing steps in this study comprise traffic segmentation, traffic cleaning, image generation, and IDX conversion [36], [20].

(1) Flow cut-off

First, the continuous pcap traffic is divided into discrete traffic units based on quintuple information to extract information from each data file. The discrete traffic data file is created by considering every 5th packet in the data stream as a whole traffic cell. If the number of packets is less than 5, the forward padding method is used. Since packet lengths are variable, the first 96 bytes of each packet are used to represent it. All malicious traffic is then stored in a CSV file by iterating over the packets [38], [26].

(2) Flow cleaning

The traffic cleaning process involves the replacement of MAC addresses at the data link layer and IP addresses at the IP layer with new randomly generated addresses [5]. This strategy is employed to eliminate the influence of these addresses on the identification results. Specifically, the IP and MAC addresses of each flow are replaced with random numbers, ensuring that the addresses are consistent within each flow after the replacement. Once this process is completed, file cleaning is performed.

(3) Image generation

The preprocessed files are normalized to a fixed length in bytes. If the file is longer than the designated number of bytes, it is truncated, and if it is shorter, it is padded with 0x00 at the end. The normalized file is then converted into a binary grayscale image, where each byte represents a grayscale pixel value. Specifically, the value 0x00 corresponds to black, and 0xff corresponds to white. This conversion allows for visual analysis of the data in an easily interpretable format. The resulting image is saved in PNG format.

(4) IDX conversion

To train the CNN and LSTM networks in the experiments, the collected data must be transformed into the appropriate format. For the CNN, the images must be converted into IDX format files. And for the LSTM network, the input format is flexible, but the maximum input length is limited to prevent excessively long inference times [45].

3.2. Model Design

As shown in Fig. 1, HPIDM consists mainly of a three-layer parallel convolutional neural network, which is used to extract temporal and spatial features of the data by interspersing stacked LSTM and SK Net self-attentive mechanism structures in the convolutional neural network, and achieve accurate classification of small sample datasets through feature fusion, and good experimental results were obtained on the test set. Furthermore, the HPIDM leverages feature fusion technology to enhance the learning performance of traffic data features and effectively address data imbalance issues. The model achieves good detection rates on the CIC-IDS 2017 and ISCX 2012 datasets.

The HPIDM utilizes a three-layered neural network. The first layer implements the Fully Convolutional Network (FCN) to capture more detailed traffic features. To avoid losing the temporal features of traffic data, a combined network structure of CNN and LSTM is used in the second layer to learn the temporal features and improve the accuracy of the predicted values. Lastly, the third layer integrates the convolutional layer and pooling layer (Max pool) with the SK attention mechanism to enhance the model's performance.

(1) Top branch

In HPIDM, the upper branch employs a FCN to extract more precise traffic features. The FCN pioneers the use of convolutional neural networks for semantic segmentation, enabling it to process input images of any size. Unlike traditional CNN, the FCN incorporates a fully convolutional layer, which grants it the flexibility to handle images of varying dimensions. By utilizing a deconvolutional layer for upsampling, the FCN can generate segmentation results that match the input image's size. Furthermore, it dispenses

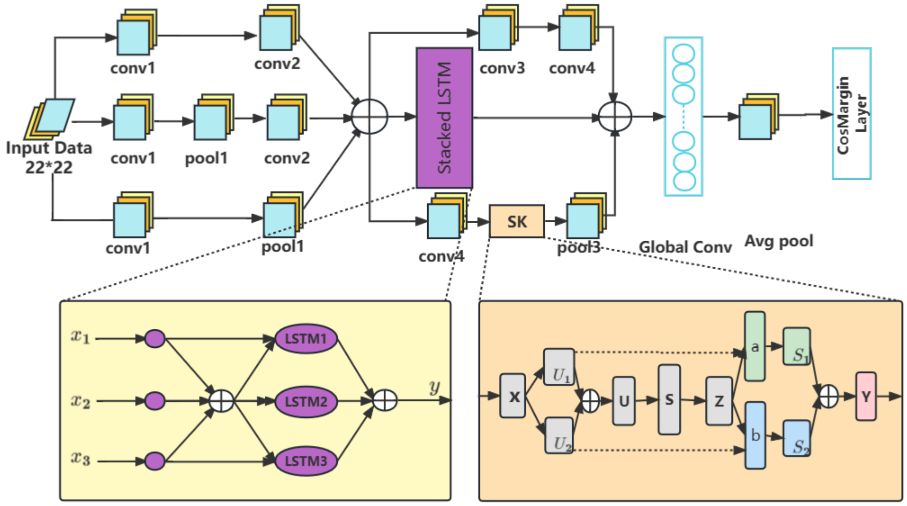

Fig. 1. Local details of the HPIDM

with pooling layers, thereby reducing the model's parameters and computational requirements. As a result, the efficiency of the model is significantly improved.

The input data is convolved four times. The kernel size is 3, the padding is 1, and the stride size is 1 in the first and third convolutions. While the second and fourth convolutional layers have a kernel size of 3, padding of 1, and stride size of 2.

From Eq. (1), it can be seen that $n_{out} = n_{in}$in convolution layers 1 and 3, i.e., the output size is equal. And $n_{out} = \frac{n_{in}}{2}$in convolution layers 2 and 4, i.e., the output is 1/2 of the input.

$$n_{out} = \frac{n_{in} - kernel + 2padding}{stride} + 1 \tag{1}$$

Where $kernel$ is the number of convolution kernels, padding is the filling value, stride is the sliding step size.

(2) Intermediate layer branching

The convolutional layer of the CNN model enables local perception within each feature of the data, followed by higher-level synthesis operations to obtain global information. The pooling layer serves to reduce feature dimensions, compress data and parameters, decrease overfitting, and enhance the fault tolerance rate of the model, thereby ensuring adequate feature learning [29] [12]. Consequently, the lower branch utilizes a combination of convolution and pooling to eliminate redundant information, expand the perception field, and reduce dimensionality and parameter numbers. Considering that the temporal features of the traffic data would be lost if only the traffic features learned using the convolutional network were used, an LSTM structure was added to this layer to learn the temporal features of the traffic.

The HPIDM utilizes the heap LSTM network to capture time sequence features of traffic data. The core idea of LSTM is gated logic. LSTM is made up of memory blocks rather than neurons. Through a storage unit and three control gates, it can allow the model to selectively process data and develop memories of pertinent historical information over extended time intervals.

The LSTM model comprises three gates, the forgetting gate, the input gate, and the output gate. The forgetting gate utilizes the sigmoid function to regulate the extent of memory retention from the previous time, as shown in Eq. (2), where f_t is between 0 and 1.

$$f_t = \sigma(U_f \times X_t + W_f \times h_{t-1} + b_f), \tag{2}$$

Where σ represents the sigmoid function, U and W are the weights of variables, X_t is the input variables, h is the input variables, and b is the intercept term.

The input gate first employs activation and excitation functions to filter and store input variables, then produces new vectors, and finally updates cell states based on the old cell states and the new variables, as depicted in Eqs. (3) to (5).

$$i_t = \sigma(U_i \times X_t + W_i \times h_{t-1} + b_i) \tag{3}$$

$$\tilde{c_t} = tanh(U_c \times X_t + W_c \times h_{t-1} + b_c)) \tag{4}$$

$$c_t = f_t c_{t-1} + i_t \tilde{c_t} \tag{5}$$

where i_t takes the value of 0 or 1, $\tilde{c_t}$ is the saved input variable, $tanh$ is the tangent excitation function, c_{t-1} is the old cell state value, c_t is the new cell state value, and f_t is the degree of forgetting.

The output gate determines the output variables according to the activation function and processes the data using the excitation function, as shown in Eqs. (6) to (7).

$$o_t = \sigma(U_o \times X_t + W_o \times h_{t-1} + b_o) \tag{6}$$

$$h_t = o_t \times tanh(c_t) \tag{7}$$

where o_t is the input gate activation function and h is the output variable.

CNN is used to extract spatial features, which are subsequently forwarded to the LSTM module for time series feature learning. The resulting time series features are combined with the spatial features learned in the first and third layers for feature fusion. The fused features are fed to each layer to facilitate further learning.

(3) Bottom branch

The bottom branch utilizes traditional CNN in conjunction with the SK Net self-attentive mechanism. While extracting data features alone, CNN may fail to fully reflect the influence of high-frequency features. In recent years, multi-scale geometric analysis theory has introduced a novel approach to image edge detection. Non-subsampling Shearlet multi-scale decomposition is the feature of multi-scale, multi-directionality, translation invariance, and anisotropy, and has high operational efficiency and unrestricted decomposition methods [15], [9]. To enable different images to learn convolution kernels of varying importance, the SK attention mechanism is incorporated into the local path, allowing it

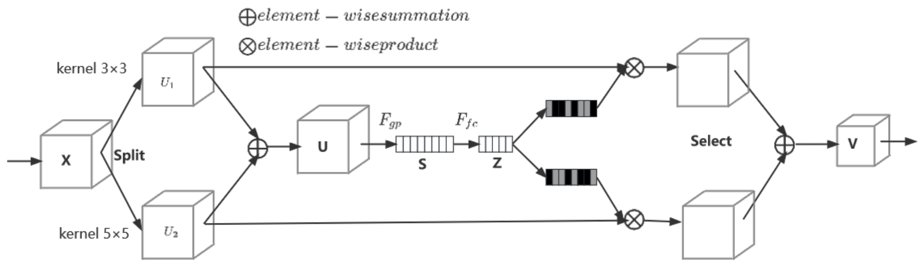

Fig. 2. SK Net model diagram

to select convolution kernels of different sizes depending on the target scale and produce differing effects [31], [41]. The specific model diagram is presented in Fig. 2.

SK Net can be divided into three phases, splitting, fusion, and selection. In the splitting stage, the original feature map is passed through two parallel convolution kernels of size 3×3 and 5×5 filters for parallel convolution operations, and the convolution results of different scales U_1 and U_2 are fused with features. In the fusion stage, the part of each convolutional kernel weight is calculated and the feature maps of the two parts are summed by the element, as shown in Eq. (8).

$$U = U_1 + U_2 \tag{8}$$

The generated U is globally averaged pooled, S is obtained by the F_{gp} function, and the feature map dimension changes from [C×H×W] to [1×1×C], as shown in Eq. (9).

$$S = F_{gp}(U) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} U(i,j) \tag{9}$$

Full concatenation is used to generate compact features z. δ is the RELU activation function, $\mathcal{B}$ denotes batch normalization (BN), the dimension of z is the number of convolution kernels, the dimension of W_s is $d \times C$, d represents the feature dimension after full concatenation, L has a value of 32 in the text, and r is the compression factor, as shown in Eqs. (10) and (11).

$$z = F_{fc}(S) = \delta(\mathcal{B}(W_s)) \tag{10}$$

$$d = max(C/r, L) \tag{11}$$

After the first two stages, the weight information of different scale spaces is obtained. $Select$ is the process of the new feature map obtained after the calculation of the convolution kernel with different weights. If it is two convolution kernels, then $a_c + b_c = 1$, the dimension of Z will be $d * 1$, the dimension of A will be $C * d$, B will be $C * d$, then the dimension of $a = A * Z$ will be $1 * C$. A_c and B_c are the cth row data of A and B, and a_c is the b_c element of a. This gives the weights of each convolution kernel, respectively, as shown in Eq. (12).

$$a_c = \frac{e^{A_c z}}{e^{A_c z} + e^{B_c z}}, b_c = \frac{e^{B_c z}}{e^{A_c z} + e^{B_c z}} \tag{12}$$

Applying the weights to the feature map, where $V = [V_1, V_2, ..., V_c]$, the dimension of V_c is H*W and the final feature map V is obtained by the attention weights on each kernel. The final output is presented in Eq. (13).

$$V_c = a_c \times U_{1c} + b_c \times U_{2c}, a_c + b_c = 1 \tag{13}$$

Due to the limited perceptual field of convolutional operations, feature extraction of images using a single convolutional kernel of fixed scale size has certain limitations On the contrary, multi-scale feature extraction can get more comprehensive features.

(4) Feature cross-fertilization

The feature fusion method can make comprehensive use of multiple image features to achieve the complementary advantages of multiple features and obtain the robustness and accuracy of recognition results [23]. Feature pyramid network (FPN) is mainly proposed for the multi-scale features of targets in images. It is used to extract features of different scales for classification in the field of target recognition [8], [42]. Based on the high resolution of low-level features and the semantic information of high-level features, the prediction effect is achieved by fusing these features of different layers [21]. FPN upsamples the deep layer information and sums the shallow layer information element by element, thus constructing a feature pyramid structure.[10]

The main methods of feature fusion are early fusion, late fusion, feature non-fusion, etc. In this experiment, concat and add in the early fusion method is used. add is the increase of information under the features describing the image, but the dimension of the image itself does not increase, only the amount of information under each dimension increases [44]. And concat is the merging of the number of channels, which means that the number of features (the number of channels) describing the image itself increases, while the amount of information under each feature does not increase [11].

In this paper, the multi-scale feature fusion of FPN is applied to the HPIDM. The output features of different layers are cascaded several times to obtain the fused feature matrix, which enables the model to fully learn the spatial and temporal features of the traffic data.

First, after the first down sampling of the three-layer neural network, the output feature maps are fused using channel cascading, which does not change the size of the feature maps, but only the multiplicity of channels [32].

Since the convolution kernel of each output channel is independent, only the output of a single channel is concerned. Suppose the two input channels are $X_1, X_2, ..., X_c$ and $Y_1, Y_2, ..., Y_c$.

Then the single output channel of concat is presented in Eq.(14), where * denotes convolution.

$$Z_{concat} = \sum_{i=1}^{c} X_i * K_i + \sum_{i=1}^{c} Y_i * K_{i+c} \tag{14}$$

The fused feature maps are fed simultaneously into a three-layer neural network, with the first layer first passing through a 3*3 sliding convolution window and then downsampling to reduce the size of the feature maps. The second layer is passed through a stacked LSTM module. The third layer is downsampled after passing through a sliding convolution window and an SK attention mechanism.

The three output feature mappings are subjected to an add fusion operation and then outputted after a global convolution operation and a global average pooling layer [40].

The single output channel of add is Eq.(15).

$$Z_{add} = \sum_{i=1}^{c} (X_i + Y_i) * K_i = \sum_{i=1}^{c} X_i * K_i + \sum_{i=1}^{c} Y_i * K_i \tag{15}$$

The output data is fed into a fully connected layer and a CosMargin layer to classify multiclass imbalanced malicious traffic. A batch processing normalization layer is shelved after each convolutional layer to speed up the convergence of the network model.

4. Experiment and Result Analysis

This section presents the experimental environment and parameter settings. The systematic evaluation of the experiment employs primarily five evaluation metrics. The validity of the model is verified on the ISCX-IDS 2012 ID and CICIDS2017 datasets. The control group comprises classical models such as TPCNN, TPCNN_C, CROSS_CNN, CROSS_CNN_LSTM, and HPM.

4.1. Experimental Environment and Parameter Settings

Table 2. Experiment environment.

Equipment	Example
OS	Windows 10 Professional Edition
CPU	Intel(R)Core(TM)i7-8700CPU@3.20GHz3.19GHz
GPU	RTX 2060
RAM	8G
Compiler environment	Python 3.8

The experimental environment is shown in Table 2. The proposed model is verified using three features, namely data header, payload, and data header with payload. 256-dimensional features are extracted from each data stream and then scaled to 16*16 grayscale images for network training. In the experiments, Adam is employed as an accelerated convergence method, and the optimizer is set to 0.0005 to prevent overfitting, with a fixed momentum factor of 0.9. The learning rate is set to 0.001 for the first eight phases for better speed. In the next three phases, the learning rate is reduced to 0.0001, and the learning rate is set to 0.00001 with a batch size of 256 in the last two phases. No additional data enhancement is used during the testing and training phases to effectively validate the proposed model.

4.2. Selection of Datasets

Compared to other datasets, the CIC-IDS 2017 and ISCX 2012 datasets use the original traffic and contain various types of attacks. Furthermore, they are relatively new and have good robustness and stability.

The CIC-IDS2017 dataset is generated in a simulated environment and spans over five days, incorporating both benign and common attacks to emulate real-world data. The dataset is fully labeled for various types of traffic and consists of source data (PCAP) and network traffic analysis results (CSV files) based on timestamps, source and target IP addresses, source and target ports, protocols, and attack flow. The CIC-IDS2017 dataset divides the acquired network traffic data into a total of 12 categories, there is an unbalanced number of different attacks, which are distributed among the pcap network traffic. Table 3 shows the collection date of the dataset and its corresponding data volume.[39][16]

Table 3. Category distribution of the CIC-IDS2017 dataset.

Data	Description	Data volume size
Monday, July 3, 2017	Normal flow	11G
Tuesday, July 4, 2017	Normal traffic + malicious traffic	11G
Wednesday, July 5, 2017	Normal traffic + malicious traffic	13G
Thursday, July 6, 2017	Normal traffic + malicious traffic	7.8G
Friday, July 7, 2017	Normal traffic + malicious traffic	8.3G

Unlike the KDD99 dataset, the content of the ISCX2012 dataset is newer and its data sample size is larger. The dataset is created using a dynamic approach that encompasses both malicious and non-malicious network behaviors.

The anomaly distribution in the CIC-IDS 2017 and the ISCX 2012 datasets are shown in Figs. 3 and 4, respectively.

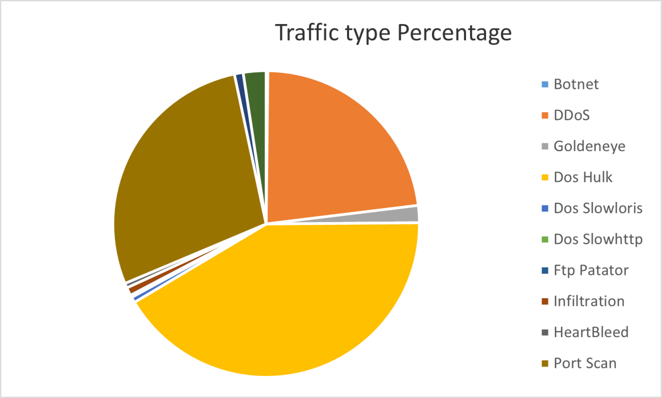

Fig. 3. Anomaly distribution in the CIC-IDS 2017 dataset

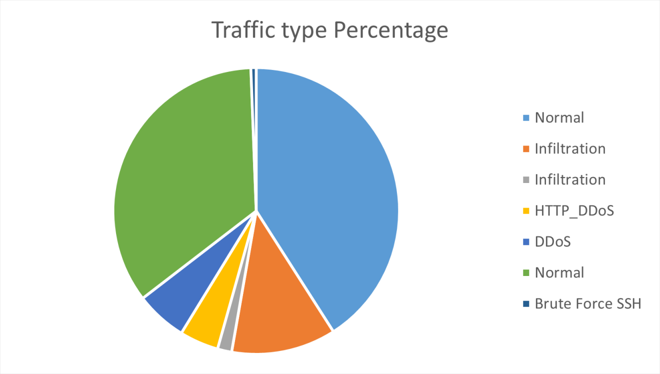

Fig. 4. Anomaly distribution in the ISCX 2012 dataset

The CICIDS2017 dataset is divided into experiments, of which 80% is used as the training set and the remaining 20% as the test set. To distribute each category equally between the training and test sets, it is necessary to divide each type in a ratio of 4:1.

4.3. Five Indicators for experimental evaluation

This paper focuses on the evaluation of intrusion detection from five metrics, accuracy, recall, precision, FAR, and F1_score. The classification of indicators is shown in Table 4.

Table 4. Classification of indicators.

	Relevant	**Not Relevant**
Retrieved	True Positives (TP)	False Positives (FP)
Not Retrieved	False Negatives (FN)	True Negatives (TN)

TP refers to the number of accurately identified positive samples, TN refers to the number of accurately identified negative samples, FP represents the number of falsely identified positive samples, and FN represents the number of falsely identified negative samples.

(1) Accuracy

a metric that measures the ratio of correctly classified samples by the classifier to the total number of samples in a given test data set. It indicates the system's ability to accurately identify intrusions from various behaviors. A detection system with a low accuracy may mistake legitimate activities for intrusions and produce false alarms, which is called false alarm phenomena.[19]It is defined as in Eq. (16).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

(2) Recall

It is the ratio of the number of positive samples that are correctly identified to the total number of all positive samples that should be retrieved, and it is defined as Eq. (17).

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

(3) Precision

It is also called accuracy rate which is the ratio of the number of positive samples correctly retrieved to the number of positive samples retrieved, which is defined in Eq. (18).

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

(4) FAR

It is also known as the false positive rate which is defined as the ratio of incorrectly predicted attack samples to all normal samples and is defined in Eq. (19).

$$FAR = \frac{FP}{FP + TN} \tag{19}$$

(5) F1_score

an evaluation metric that reflects both precision and recall. It is defined as the harmonic mean of precision and recall, as shown in Eq. (20). The F1 score can provide a balanced evaluation of the model's performance by considering both the true and false positives.

$$F1\_score = 2 * (\frac{Precision * Recall}{Precision + Recall}) \tag{20}$$

In the experiments, the positive categories are considered to be the accurately detected categories, while the negatively detected categories are considered negative. The quality of the model is assessed using the five evaluation metrics mentioned earlier. A higher value of accuracy, precision, recall, and F1_score indicates better model performance, while a lower value of the FAR indicates better performance.[18][1]Accuracy is a general measure of a model's classification effectiveness, while Precision, Recall, and F1_score are more focused on assessing the model's effectiveness in detecting different categories.

4.4. Ablation Experiment and Result

To further verify the performance of abnormal network traffic detection, the HPIDM is compared with the conventional network models like TPCNN, TPCN_C, CROSS_CNN, CROSS_CNN_LSTM, and HPM using the datasets ISCX-IDS 2012 and CICIDS2017. Fig. 5 shows the results of the comparison of the recognition accuracy of the models on the ISCX-IDS 2012 dataset. Fig. 6 presents the comparison between the HPIDM and other models in terms of precision, recall rate, F1%score, and FAR. As seen in Fig. 5, the HPIDM has the highest overall detection accuracy on the ISCX-IDS 2012 dataset, outperforming the classical model by 0.06%, 0.05%, 0.11%, 0.13%, 0.12%, respectively, outperforming the two comparison models by 0.13%, 0.02%. From Fig. 6 (A), it can be seen that the detection accuracy of the HPIDM is higher and smoother, which indicates that the proposed method in this paper effectively improves the problem of low detection rate due to data imbalance. As shown in Fig. 6 (B), the recall rate of the HPIDM is higher than that of other comparable models, indicating that it has a stronger positive sample identification ability. As shown in Fig. 6 (C), compared with other models, the F1_score of the HPIDM is higher, which means that the HPIDM is more robust. It can be seen from Fig.6 (D) that the FAR of the HPIDM is lower and smoother. The experimental results show that the HPIDM is significantly better than other classical network traffic anomaly detection models.

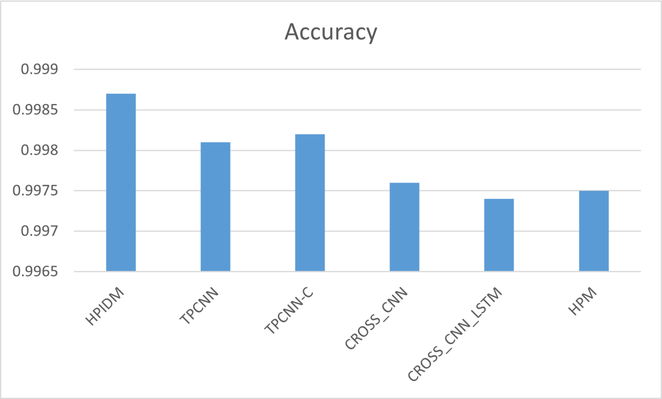

Fig. 5. Comparison of the recognition accuracy of each model on the ISCX-IDS 2012 dataset

Moreover, a comparison of the HPIDM with other classical models in terms of time consumption is shown in Figs. 7and 8. It is evident that although the HPIDM is not the fastest in training time, its testing time is much less than the TPCNN and TPCNN_C models. Additionally, while the training and testing time of the CROSS_CNN and CROSS_CNN_LSTM models are shorter than the HPIDM, they exhibit inferior performance regarding precision, recall, and other relevant aspects. In summary, the HPIDM has high training accuracy and feasibility despite its relatively longer training time.

Additional experiments are conducted in the CICIDS2017 dataset to further validate the efficacy of the HPIDM. The experimental results are presented in Tables 5 to 9. As shown in Table 9, all classifiers exhibit a classification accuracy of over 99%. Notably, the HPIDM displays the highest classification accuracy compared to all other models. Moreover, Table 5 to9 indicate that the HPIDM surpasses other models in terms of accuracy, recall, and F1_score on the CICIDS2017 dataset, thus reinforcing its effectiveness.

Furthermore, the training duration of distinct classifiers is also evaluated on the CICIDS2017 dataset, as depicted in Fig. 9. Tables 5 to 8 and Fig. 9 reveal that the HPIDM outperforms other models in terms of accuracy and training time. The experimental outcomes obtained from the CICIDS2017 dataset provide evidence of the HPIDM's feasibility.

Additionally, two upgraded versions of the HPIDM are introduced and the FCN structure of the first layer and the stacked LSTM module of the second layer are taken as variables for comparative experiments to validate the fusion of feature information. The

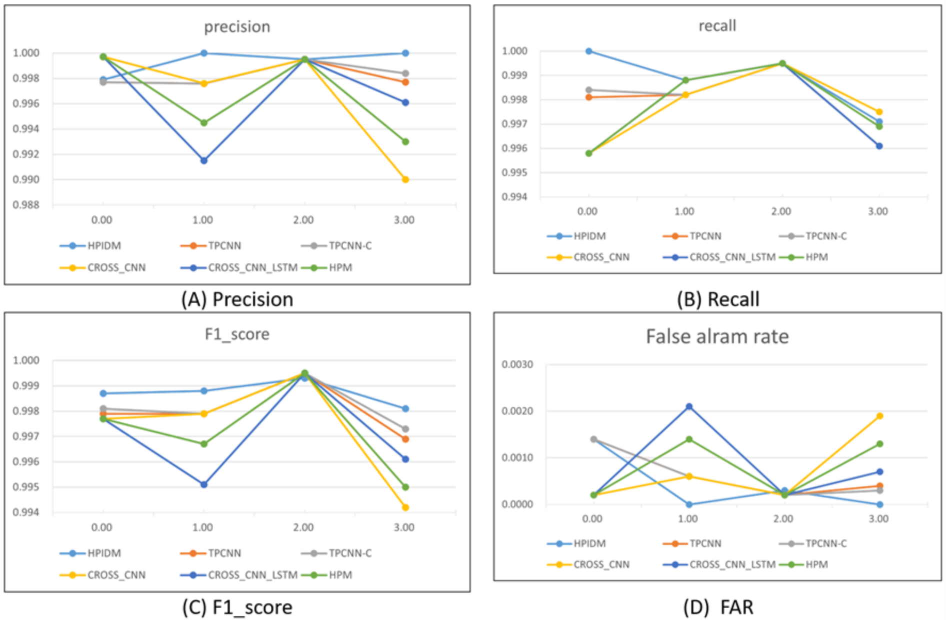

Fig. 6. Comparison of precision, recall, F1 score, and FAR for each model on the ISCX-IDS 2012 dataset

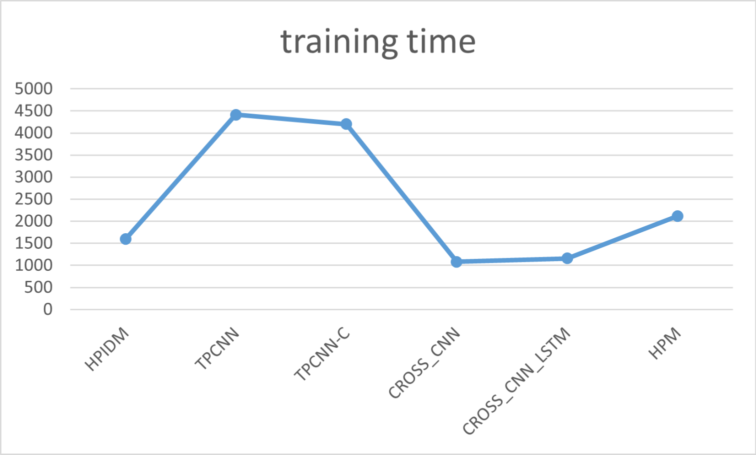

Fig. 7. Comparison of training time for each model on ISCX-IDS 2012 data

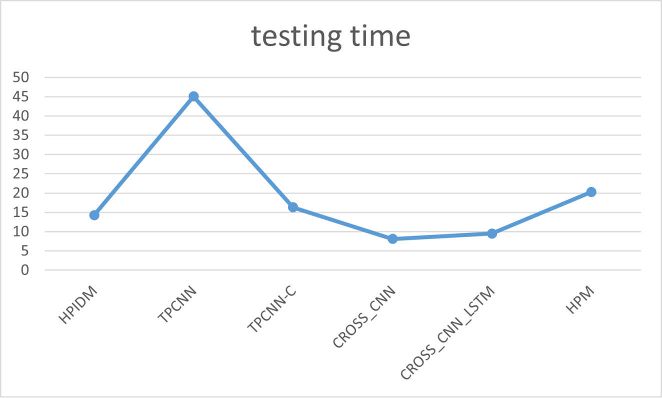

Fig. 8. Comparison of test times for each model on ISCX-IDS 2012 data

Table 5. Precision comparison

Label	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
HPIDM	1.0000	1.0000	0.9928	0.9987	0.9911	0.9919	1.0000	1.0000	1.0000	0.9999	1.0000	0.9991
TPCNN	1.0000	1.0000	0.9848	0.9992	0.9800	0.9967	0.9997	1.0000	1.0000	1.0000	0.9998	0.9967
TPCNN_C	1.0000	1.0000	0.9955	0.9990	0.9817	0.9938	1.0000	1.0000	1.0000	0.9999	1.0000	0.9972
CROSS_CNN	1.0000	1.0000	0.9956	0.9992	0.9860	0.9924	0.9997	1.0000	1.0000	1.0000	0.9997	0.9976
CROSS_CNN_LSTM	1.0000	1.0000	0.9906	0.9987	0.9889	0.9910	0.9998	1.0000	1.0000	1.0000	1.0000	0.9986
HPM	1.0000	1.0000	0.9913	0.9986	0.9889	0.9910	1.0000	1.0000	1.0000	0.9999	0.9998	0.9967

Table 6. Recall comparison

Label	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
HPIDM	1.0000	1.0000	0.9706	0.9998	0.9801	0.9934	0.9997	1.0000	1.0000	1.0000	0.9998	0.9995
TPCNN	1.0000	1.0000	0.9796	0.9996	0.9750	0.9891	0.9995	1.0000	1.0000	1.0000	0.9998	1.0000
TPCNN_C	1.0000	1.0000	0.9754	0.9999	0.9867	0.9862	0.9999	1.0000	1.0000	0.9999	0.9998	0.9995
CROSS_CNN	1.0000	1.0000	0.9803	0.9998	0.9860	0.9929	0.9994	1.0000	1.0000	1.0000	0.9998	0.9991
CROSS_CNN_LSTM	1.0000	1.0000	0.9698	0.9997	0.9838	0.9929	0.9995	1.0000	1.0000	1.0000	0.9998	0.9991
HPM	1.0000	1.0000	0.9681	0.9997	0.9809	0.9924	0.9995	0.9995	1.0000	1.0000	0.9996	0.9986

Table 7. F1_score comparison

Label	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
HPIDM	1.0000	1.0000	0.9815	0.9993	0.9856	0.9927	0.9999	1.0000	1.0000	0.9999	0.9999	0.9991
TPCNN	1.0000	1.0000	0.9822	0.9994	0.9775	0.9929	0.9996	1.0000	1.0000	1.0000	0.9998	0.9983
TPCNN_C	1.0000	1.0000	0.9854	0.9994	0.9842	0.9900	0.9997	1.0000	1.0000	0.9999	0.9999	0.9983
CROSS_CNN	1.0000	1.0000	0.9879	0.9995	0.9860	0.9926	0.9996	1.0000	1.0000	0.9998	0.9998	0.9991
CROSS_CNN_LSTM	1.0000	1.0000	0.9801	0.9992	0.9863	0.9919	0.9997	1.0000	1.0000	1.0000	0.9999	0.9988
HPM	1.0000	1.0000	0.9796	0.9992	0.9848	0.9917	0.9997	0.9997	1.0000	0.9999	0.9997	0.9976

Table 8. FAR comparison

Label	0	1	2	3	4	5	6	7	8	9	10	11
HPIDM	0.0	0.0	0.0001	0.0009	0.0001	0.0	0.0	0.0	0.0	0.0	0.0	0.0
TPCNN	0.0	0.0	0.0003	0.0006	0.0001	0.0001	0.0	0.0	0.0	0.0	0.0	0.0
TPCNN_C	0.0	0.0	0.0001	0.0007	0.0001	0.0001	0.0	0.0	0.0	0.0	0.0	0.0
CROSS_CNN	0.0	0.0	0.0001	0.0006	0.0001	0.0001	0.0	0.0	0.0	0.0	0.0	0.0
CROSS_CNN_LSTM	0.0	0.0	0.0002	0.0009	0.0001	0.0001	0.0	0.0	0.0	0.0	0.0	0.0
HPM	0.0	0.0	0.0002	0.0009	0.0001	0.0001	0.0	0.0	0.0	0.0	0.0	0.0

Table 9. Comparison of the recognition accuracy of each model on the CICIDS2017 dataset

Label	HPIDM	TPCNN	TPCNN_C	CROSS_CNN	CROSS_CNN_LSTM	HPM
Accuracy	0.9994	0.9991	0.9992	0.9992	0.9991	0.99900

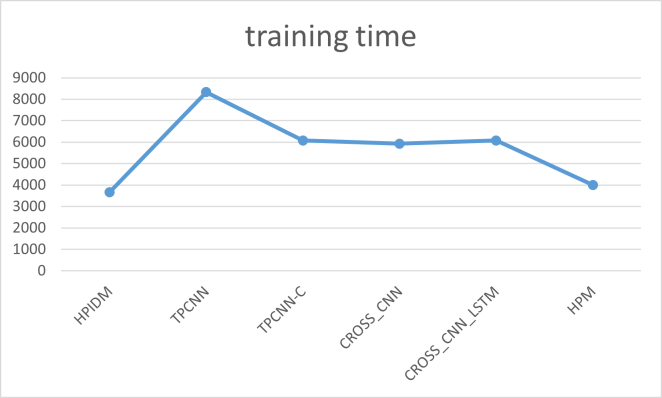

Fig. 9. Comparison of training time for each model on the CICIDS2017 dataset

network model structures for comparison tests 1 and 2 are illustrated in Figs. 10and 11. The experiments remain consistent except for the FCN structure of the first layer and the stacked LSTM module of the second layer.

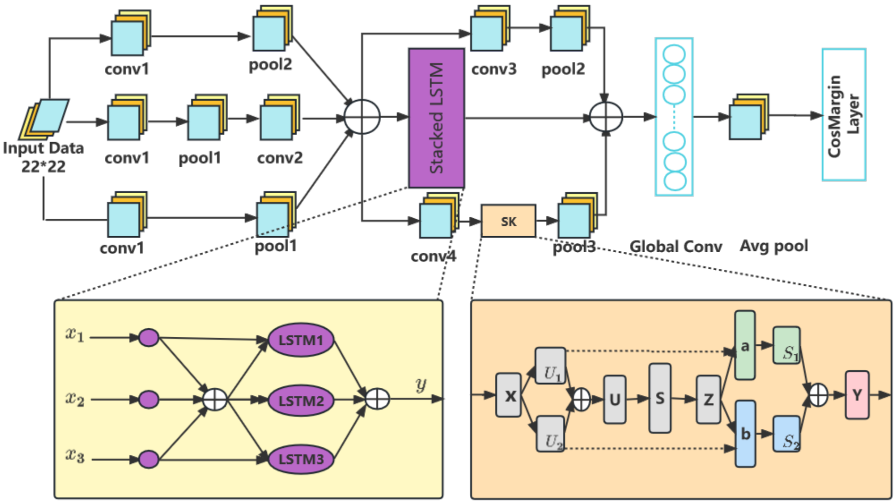

Fig. 10. Model-1 structure

Table 10 shows the accuracy, recall, precision, FAR, F1_score of HPIDM, model-1, and model-2 on the ISCX 2012 dataset. The detection accuracy of the HPIDM on the ISCX 2012 dataset remains the highest, 0.13% higher than that of the comparison model 1 and 0.02% higher than that of the comparison model 2, and F1_score and Recall are optimal, thus demonstrating the effectiveness of the FCN structure and stacked LSTM module in the HPIDM, which indicates that the structure can better learn the features of the traffic data and is more effective in detecting abnormal network traffic.

Table 10. Comparison of experimental results between the HPIDM and the improved model on the ISCX-IDS 2012 dataset

Classifier	Accuracy	Precision	Recall	F1_score	FAR	Training time
HPIDM	0.9987	0.9988	0.9988	0.9987	0.0004	1593.5
Model-1	0.9974	0.9984	0.9979	0.9981	0.0006	1901.8
Model-2	0.9986	0.9987	0.9984	0.9985	0.0005	1104.08

5. Conclusion

This paper introduces a novel hybrid parallel intrusion detection model (HPIDM) based on deep learning. Ablation experiments conducted demonstrate the superior performance

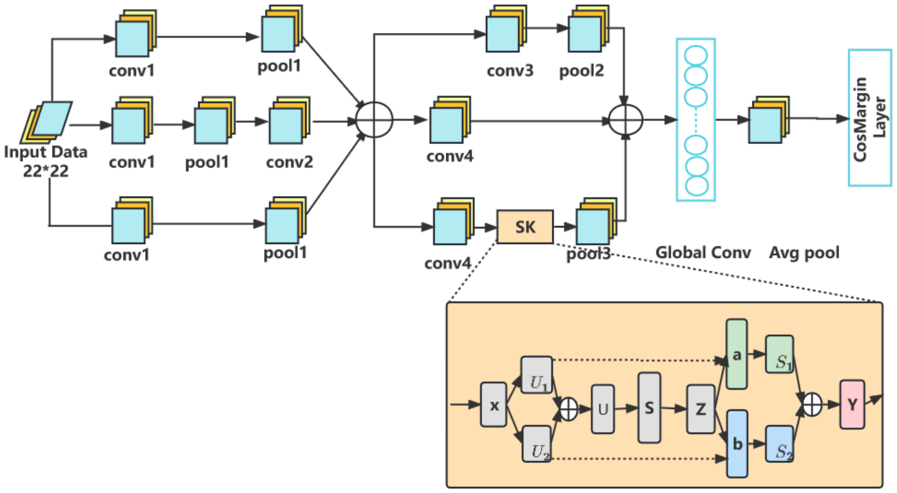

Fig. 11. model-2 structure

of HPIDM compared to other models. On the ISCX-IDS 2012 and CICIDS 2017 datasets, HPIDM achieves remarkable accuracy rates of 99.87$\%$ and 99.94$\%$, respectively, surpassing the comparison models. Furthermore, HPIDM exhibits superior recall, accuracy, FAR, and F1_score compared to existing classical models. The success of HPIDM can be attributed to its three-layer parallel neural network structure, which combines a stacked long and short-term memory (LSTM) neural network with a convolutional neural network (CNN), along with the SK Net self-attentive mechanism. This unique combination enables HPIDM to efficiently learn both the temporal and spatial features of traffic data. The experiments comparing the improved models Model1 and Model2 with the HPIDM model also show the soundness of the design of our model in the FCN module and the stacked LSTM module. HPIDM fuses the acquired temporal and spatial feature data several times and then feeds it into the CosMargin classifier for classification detection to reduce the impact of data imbalance on Intrusion Detection System (IDS) performance, resulting in a model with strong robustness and positive sample recognition rate.

HPIDM shows better results in terms of detection accuracy and other evaluation criteria. This model effectively identifies malicious activities and network attacks on specific computers or networks, providing timely alerts to administrators for network security. Its versatility extends to various applications, such as smart homes where it detects unauthorized access to home networks, safeguarding residents' privacy. In industrial control systems, the model detects cyber attacks on critical infrastructure, preventing potential damage. Additionally, it contributes to healthcare systems by detecting and preventing unauthorized access to sensitive patient data.

However, it is important to note that the dataset used in this study comprises a large number of labeled samples, covering various anomalous network traffic classes. In real-world network environments, a substantial amount of data remains unlabeled, and the anomalous traffic classes are unknown. Given the impracticality of labeling all data, future research will focus on exploring semi-supervised intrusion detection methods. The aim is to design effective models and methods that can be trained using limited labeled data and abundant unlabeled data. One potential approach involves leveraging Generative Adversarial Networks (GANs) to learn the underlying distribution of network traffic, enabling anomaly detection by comparing real and generated data. This research direction aims to further enhance the accuracy of anomalous network traffic detection, thereby ensuring the network and data security of IoT systems.

References

1. Akshay Kumaar, M., Samiayya, D., Vincent, P.M.D.R., Srinivasan, K., Chang, C.Y., Ganesh, H.: A Hybrid Framework for Intrusion Detection in Healthcare Systems Using Deep Learning. Frontiers in Public Health 9 (2022)
2. Cai, S., Han, D., Li, D.: A Feedback Semi-Supervised Learning With Meta-Gradient for Intrusion Detection. IEEE Systems Journal (2022)
3. Cai, S., Han, D., Yin, X., Li, D., Chang, C.C.: A Hybrid parallel deep learning model for efficient intrusion detection based on metric learning (2022)
4. Chen, P., Han, D.: Effective wind speed estimation study of the wind turbine based on deep learning. Energy 247 (2022)
5. Cui, Z., Chen, W., Chen, Y.: Multi-Scale Convolutional Neural Networks for Time Series Classification (2016)
6. Dina, A.S., Manivannan, D.: Intrusion detection based on Machine Learning techniques in computer networks (2021)
7. Fatani, A., Dahou, A., Al-qaness, M.A.A., Lu, S., Abd Elaziz, M.A.: Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System (2021)
8. Fu, Y., Cao, L., Guo, G., Huang, T.S.: Multiple Feature Fusion by Subspace Learning
9. Gao, N., Han, D., Weng, T.H., Xia, B., Li, D., Castiglione, A., Li, K.C.: Modeling and analysis of port supply chain system based on Fabric blockchain. Computers & Industrial Engineering 172 (2022)
10. Golz, M., Sommer, D., Chen, M., Trutschel, U., Mandic, D.: Feature Fusion for the Detection of Microsleep Events. The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology 49(2) (2007)
11. Golz, M., Sommer, D., Chen, M., Trutschel, U., Mandic, D.: Feature Fusion for the Detection of Microsleep Events. The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology 49(2) (2007)
12. Guo, Z., Han, D.: Sparse co-attention visual question answering networks based on thresholds. Applied Intelligence 53(1) (2023)
13. Han, D., Zhu, Y., Li, D., Liang, W., Souri, A., Li, K.C.: A Blockchain-Based Auditable Access Control System for Private Data in Service-Centric IoT Environments. IEEE Transactions on Industrial Informatics 18(5) (2022)
14. Hassan, M.M., Gumaei, A., Alsanad, A., Alrubaian, M., Fortino, G.: A hybrid deep learning model for efficient intrusion detection in big data environment. Information Sciences 513 (2020)
15. Hu, H., Pang, L., Shi, Z.: Image matting in the perception granular deep learning. Knowledge-Based Systems 102 (2016)
16. Jiang, K., Wang, W., Wang, A., Wu, H.: Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network. IEEE Access 8 (2020)
17. Lakshminarayana, D.H., Philips, J., Tabrizi, N.: A Survey of Intrusion Detection Techniques. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, Boca Raton, FL, USA (2019)

18. Li, D., Han, D., Weng, T.H., Zheng, Z., Li, H., Liu, H., Castiglione, A., Li, K.C.: Blockchain for federated learning toward secure distributed machine learning systems: a systemic survey. Soft Computing 26(9) (2022)
19. Li, D., Han, D., Xia, B., Weng, T.H., Castiglione, A., Li, K.C.: Fabric-GC: A Blockchain-based Gantt chart system for cross-organizational project management. Computer Science and Information Systems 19(3) (2022)
20. Li, H., Han, D., Tang, M.: A Privacy-Preserving Storage Scheme for Logistics Data With Assistance of Blockchain. IEEE Internet of Things Journal 9(6) (2022)
21. Li, J., Han, D., Wu, Z., Wang, J., Li, K.C., Castiglione, A.: A novel system for medical equipment supply chain traceability based on alliance chain and attribute and role access control. Future Generation Computer Systems 142 (2023)
22. Li, M., Han, D., Yin, X., Liu, H., Li, D.: Design and Implementation of an Anomaly Network Traffic Detection Model Integrating Temporal and Spatial Features. Security and Communication Networks 2021 (2021)
23. Li, Z., Zhou, F.: FSSD: Feature Fusion Single Shot Multibox Detector (2018)
24. Liang, C., Shanmugam, B., Azam, S., Karim, A., Islam, A., Zamani, M., Kavianpour, S., Idris, N.B.: Intrusion Detection System for the Internet of Things Based on Blockchain and Multi-Agent Systems. Electronics 9(7) (2020)
25. Liu, H., Han, D., Cui, M., Li, K.C., Souri, A., Shojafar, M.: IdenMultiSig: Identity-Based Decentralized Multi-Signature in Internet of Things. IEEE Transactions on Computational Social Systems (2023)
26. Ma, W., Zhang, Y., Guo, J., Yu, Q.: Few-Shot Abnormal Network Traffic Detection Based on Multi-scale Deep-CapsNet and Adversarial Reconstruction. International Journal of Computational Intelligence Systems 14(1) (2021)
27. Maheswari, M., A. Karthika, R.: A Novel Hybrid Deep Learning Framework for Intrusion Detection Systems in WSN-IoT Networks. Intelligent Automation & Soft Computing 33(1) (2022)
28. Marir, N., Wang, H., Feng, G., Li, B., Jia, M.: Distributed Abnormal Behavior Detection Approach Based on Deep Belief Network and Ensemble SVM Using Spark. IEEE Access 6 (2018)
29. Meliboev, A., Alikhanov, J., Kim, W.: Performance Evaluation of Deep Learning Based Network Intrusion Detection System across Multiple Balanced and Imbalanced Datasets. Electronics 11(4), 515 (2022)
30. Sarnovsky, M., Paralic, J.: Hierarchical Intrusion Detection Using Machine Learning and Knowledge Model (2020)
31. Shen, X., Han, D., Guo, Z., Chen, C., Hua, J., Luo, G.: Local self-attention in transformer for visual question answering. Applied Intelligence (2022)
32. Sun, Q.S., Zeng, S.G., Liu, Y., Heng, P.A., Xia, D.S.: A new method of feature fusion and its application in image recognition. Pattern Recognition 38(12) (2005)
33. Tao, J., Han, T., Li, R.: Deep-Reinforcement-Learning-Based Intrusion Detection in Aerial Computing Networks (2021)
34. Tian, Q., Han, D., Li, K.C., Liu, X., Duan, L., Castiglione, A.: An intrusion detection approach based on improved deep belief network. Applied Intelligence 50(10) (2020)
35. Wang, Z., Han, D., Li, M., Liu, H., Cui, M.: The abnormal traffic detection scheme based on PCA and SSH. Connection Science 34(1) (2022)
36. Wei, G., Wang, Z.: Adoption and realization of deep learning in network traffic anomaly detection device design. Soft Computing 25(2) (2021)
37. Xia, B., Han, D., Yin, X., Na, G.: RICNN: A ResNet&Inception convolutional neural network for intrusion detection of abnormal traffic. Computer Science and Information Systems 19(1) (2022)
38. Xiao, Y., Xing, C., Zhang, T., Zhao, Z.: An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks. IEEE Access 7 (2019)

39. Yang, H., Wang, F.: Wireless Network Intrusion Detection Based on Improved Convolutional Neural Network. IEEE Access 7 (2019)
40. Yang, J., Yang, J.y., Zhang, D., Lu, J.f.: Feature fusion: parallel strategy vs. serial strategy. Pattern Recognition 36(6) (2003)
41. Yang, Y., Xia, X., Lo, D., Grundy, J.: A Survey on Deep Learning for Software Engineering (2020)
42. Zhang, C., Costa-Perez, X., Patras, P.: Adversarial Attacks Against Deep Learning-Based Network Intrusion Detection Systems and Defense Mechanisms. IEEE/ACM Transactions on Networking 30(3) (2022)
43. Zhang, C., Jiang, J., Kamel, M.: Intrusion detection using hierarchical neural networks (2005)
44. Zhang, Z., Zhang, X., Peng, C., Xue, X., Sun, J.: ExFuse: Enhancing Feature Fusion for Semantic Segmentation. In: Computer Vision – ECCV 2018, vol. 11214. Springer International Publishing, Cham (2018)
45. Zhou, X., Hu, Y., Liang, W., Ma, J., Jin, Q.: Variational LSTM Enhanced Anomaly Detection for Industrial Big Data. IEEE Transactions on Industrial Informatics 17(5) (2021)
46. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: DEEP AUTOENCODING GAUSSIAN MIXTURE MODEL FOR UNSUPERVISED ANOMALY DETECTION (2018)

Yan Wang is a second-year graduate student majoring in Computer Science and Technology at Shanghai Maritime University, China. Her research interests include deep learning, cyber security, and the Internet of Things.

Dezhi Han is a professor at the School of Information Engineering, Shanghai Maritime University, a senior member of the Chinese Computer Society, a member of the Special Committee on Computer Information Storage, an expert in communication review for several authoritative journals, and an expert in communication review for the National Natural Science Foundation of China. His research interests include cloud computing and cloud storage security technology, IOT network security technology, intelligent image processing and pattern recognition, blockchain theory and application technology, massive network data information storage, retrieval and analysis technology, machine learning, and intelligent information processing.

Mingming Cui received a B.S. degree in Computer Science and Technology from the Anhui University of Finance and Economics, China. She is currently pursuing a Ph.D. degree from Shanghai Maritime University, China, and is a Visiting Ph.D. student at the Nanyang Technological University, Singapore. Her research interests include cryptology, blockchain, data privacy protection, network security, VANETS security, and the Internet of Things.

Received: April 18, 2023; Accepted: July 25, 2023.