J Inf Process Syst, Vol.20, No.4, pp.458~476, August 2024
https://doi.org/10.3745/JIPS.01.0106
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Network Intrusion Detection Using Transformer and BiGRU-DNN in Edge Computing

Huijuan Sun*

Abstract

To address the issue of class imbalance in network traffic data, which affects the network intrusion detection performance, a combined framework using transformers is proposed. First, Tomek Links, SMOTE, and WGAN are used to preprocess the data to solve the class-imbalance problem. Second, the transformer is used to encode traffic data to extract the correlation between network traffic. Finally, a hybrid deep learning network model combining a bidirectional gated current unit and deep neural network is proposed, which is used to extract long-dependence features. A DNN is used to extract deep level features, and softmax is used to complete classification. Experiments were conducted on the NSLKDD, UNSWNB15, and CICIDS2017 datasets, and the detection accuracy rates of the proposed model were 99.72%, 84.86%, and 99.89% on three datasets, respectively. Compared with other relatively new deep-learning network models, it effectively improved the intrusion detection performance, thereby improving the communication security of network data.

Keywords

Bi-directional Gated Recurrent Unit, Class Imbalance, Deep Neural Network, Edge Computing, Network Intrusion Detection, Transformer-Encoder

1. Introduction

At present, network communication technology and Internet of Things (IoT) are developing rapidly, and a variety of intelligent devices have emerged one after another, followed by a sharp increase in the amount of data [1]. Distributed cloud computing is used to process large amounts of data. This method mainly transmits all data collected by various intelligent devices to the cloud for deep processing. Its characteristic is that tens of thousands of data points can be processed in a short time. However, with the rapid development of data, it is difficult to meet the development requirements of Internet communication by relying solely on data processing methods based on cloud computing. Therefore, data processing from the cloud to the edge has gradually become popular, and the processing mode of edge computing has come into being.

The edge computing mode can provide service computing at the data source side without transmitting massive data to the cloud, which can not only improve the user experience but also reduce the pressure on cloud services [2]. The edge terminal device can complete computing tasks that need to be completed

Manuscript received March 28, 2023; first revision May 30, 2023; accepted June 25, 2023.
* **Corresponding Author:** Huijuan Sun (ourbigdata@126.com)
College of Computer and Information Technology, Henan Finance University, Zhengzhou, China (ourbigdata@126.com)
Current affiliation for author, Huijuan Sun, is School of Computer and artificial intelligence, Henan Finance University, Zhengzhou, Henan, China

on an edge or cloud server. Distributing distributed training tasks of the model to the edge terminal devices can bring many advantages. On the one hand, deep learning (DL) technology requires massive data as the basis. The quantity of data directly determines the final performance of the DL model. Model training is directly deployed to the data source, avoiding the transmission of original data and reducing the transmission cost of equipment and cloud storage cost. On the other hand, the massive data in the edge terminal device involve user privacy (such as personal photos, track information, and APP usage records) being transmitted to the cloud for centralized training, which will cause serious data security problems.

Owing to their special network structure, edge network nodes are vulnerable to illegal intrusion threats; thus, certain security measures must be taken to deal with illegal intrusion [3]. Abnormal network traffic detection is an effective security defense strategy. By deploying an appropriate detection system in the edge network node, it can actively detect abnormal intrusion data in the network connection data passing through the edge network node and send an alarm in time to ensure the safe operation of the edge computing network system. Therefore, research on abnormal detection models and algorithms suitable for edge network nodes to detect illegal activity has great significance.

Owing to the limited computing power of edge devices, large-scale data training tasks cannot be performed. A DL model trained offline can be embedded in an edge device to detect abnormal traffic [4]. When abnormal results are detected, the edge device can record logs or send detection results to the cloud. Without considering the data transmission delay of the edge device, this study focused on improving detection performance. Thus, a combined framework was designed based on the transformer architecture, which consists of four layers: data preprocessing, data encoding, deep feature extraction, and abnormal traffic classification.

Section 2 describes the relevant work in network traffic intrusion detection, summarizes the problems of these methods, and elaborates on the study motivation. Section 3 introduces the overall system architecture and designed composite network model. In Section 4, the reliability of the designed model is verified through well-designed and rich experiments, and the advantages of the proposed model are verified by comparison with several newer models. In Section 5, the experimental implications are summarized, along with the shortcomings of the study and prospects for the next step of the work.

2. Related Works

With the successful application of machine learning (ML) and DL in other fields, network security has begun to use ML and DL models to achieve intelligent detection and improve network intrusion detection performance.

2.1 ML-based Methods

Traditional network security defense methods primarily include firewalls, antivirus software, and network security hardware products that detect intrusion traffic or virus programs through pattern matching. However, these methods have poor defense against new attack behaviors. To this end, the industry has introduced ML algorithms to improve the performance of various types of attacks by learning intrusion-traffic features.

For example, Hassan et al. [5] proposed a random forest-based method that has certain advantages in

the multi-classification problem of unbalanced data. Salo et al. [6] proposed a network intrusion detection model that integrates information gain and ML. These methods have certain advantages; however, they do not significantly improve the detection rates of various attacks.

Zhang et al. [7] comprehensively considered the three dimensions of time, space, and content in data and proposed a multi-dimensional feature fusion and overlay integration mechanism. The classification accuracy was effectively improved by integrating multiple decision trees.

ML-based methods can analyze the surface features of data and achieve autonomous detection through feature learning. However, ML methods have the following three limitations [8]: overreliance on robust features, high-dimensional feature processing capability, and weak ability to extract dynamic features.

2.2 DL-based Methods

Compared with ML-based methods, DL-based methods can mine deeper data features, improve detection efficiency, reduce false positives, and help identify potential security threats in computer network systems [9].

To protect IoT systems from ransomware attacks, Al-Hawawreh and Sitnikova [10] proposed a detection model based on stacked variational self-coding. Liu et al. [11] proposed a bidirectional generative adversarial network (BiGAN)-based method for industrial control systems. However, these methods typically yield unsatisfactory results when addressing high-dimensional data.

To better process high-dimensional data, Li et al. [12] utilized DL autoencoders combined with coefficient penalties and reconstruction losses in the encoding layer to extract high-dimensional data features and then used extreme learning machines to quickly and effectively classify the extracted features. The authors of [13] used a preprocessing method combining dimensionality reduction technology and feature engineering to generate meaningful features and proposed two DL-based detection methods, achieving good detection results. In [14], principal component analysis (PCA) was used to simplify data characteristics based on data dimension and time series characteristics, and a stacked gated current unit (GRU) detection model based on transfer learning was used to conduct intrusion detection on the simplified characteristics. Although this method can achieve good results in simple network traffic attack detection, it cannot solve complex malicious botnet problems. For the complex malicious botnet problem, Liaqat et al. [15] designed a composite framework integrating convolutional neural network, deep neural network, and long short-term memory (CNN-DNN-LSTM), which can detect complex malicious botnets in a timely and effective manner in a medical IoT environment. However, this method does not consider the limitations of general convolution.

Saharkhizan et al. [16] proposed using LSTM to learn the dependency relationships between temporal data, using the LSTM set as a detector and combining the output of the detector into a decision tree for effective classification. However, it is not easy to solve the trade-off between "curse of dimensionality," accuracy, and other important performance indicators. To address this issue, Mushtaq et al. [17] proposed a hybrid framework (BiLSTM-AE) that combined a bidirectional LSTM (BiLSTM) and depth automatic encoder (AE). The best features were obtained using AE and then classified into normal and abnormal samples using BiLSTM. Although this method considers spatiotemporal and deep features, it ignores the correlations between network traffic. To address this issue, Li et al. [18] designed a new detection model with a three-layer packet flow. This model can simultaneously consider the spatiotemporal feature correlation both in and between network flows, thereby improving the network traffic classification

performance.

Although these methods have achieved high detection rates, most of them do not attach great importance to the problem of dataset imbalance [19]. Fernando and Tsokos [20] proposed an intrusion detection model using a dynamic weighted loss function, and Liu et al. [21] proposed an intrusion detection model using difficult set sampling. These two models have certain effects on alleviating the class imbalance problem, but they cannot be used to solve the problem of minority class data generation. Zhang and Liu [22] combined DL and statistical concepts to solve the problem of few samples and proposed a fusion model based on an improved conditional variational automatic encoder (ICVAE) and boundary line synthesis few oversampling technology (BSM) for the IoT, called ICVAE-BSM, which has achieved significant results in solving class imbalance problems. However, the fusion model failed to effectively extract the interdependence and long-dependency features between network traffic, which affected the overall performance of the model to some extent.

2.3 Study Motivation

Based on the above methods, it can be concluded that most existing network intrusion detection methods have the following problems:

1) Trade-off between "curse of dimensionality," accuracy, and other important performance indicators;
2) Correlation problem between network traffic;
3) Long dependency feature extraction problem;
4) Deep feature extraction problem;
5) Class imbalance problem.

Based on these issues, a new detection model using a transformer is proposed. Firstly, Tomek Links [23], the SMOTE algorithm [24], and the Wasserstein Generative Adversarial Network (WGAN) [25] are used to preprocess the data to solve the class imbalance problem. Secondly, a transformer [26] is used to encode the data to extract the correlation between network traffic. Finally, a network model that integrates a bidirectional gated current unit (BiGRU) [27] and DNN [28] is proposed, which can avoid the "curse of dimensionality" and simultaneously mine local and global features of the data. Long-dependent temporal features are extracted using BiGRU, while deep-level features are extracted by DNN.

3. Abnormal Detection Model using TEBGD Network

3.1 Framework of TEBGD

The proposed framework is shown in Fig. 1.

The detection method is mainly divided into two stages. The first stage involves checking the user consistency, mainly by checking the identities of known users. The historical behavior data of all legal users are trained into a multi-classifier, which is then used for the supervised classification and identification of users, including normal and abnormal legal users. The second stage is anomaly detection, which is primarily used to detect whether a single legitimate user is abnormal. The edge server is responsible for monitoring the abnormal behavior of the users.

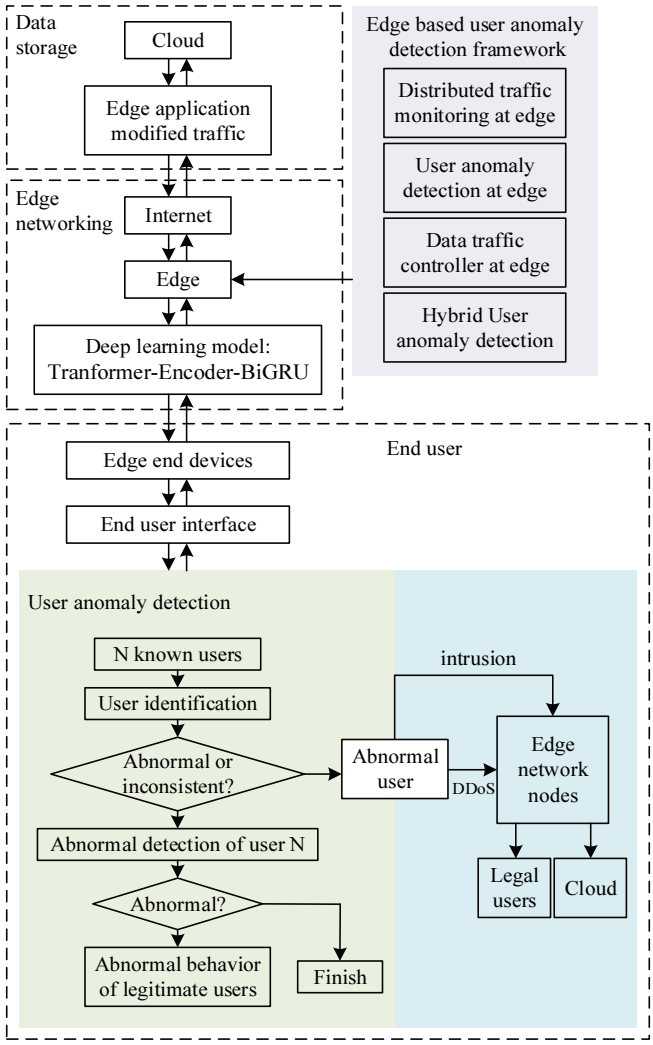


Fig. 1. Framework of abnormal detection model using TEBGD.

3.2 Transformer-Encoder-BiGRU-DNN (TEBGD) Model Architecture

The traditional intrusion detection algorithm can only detect attacks at the current time but cannot do anything about an attack with a long duration, resulting in a forgetting phenomenon in the iterative learning process. The Transformer-Encoder and BiGRU have solved the aforementioned problems and achieved satisfactory results in time-series processing. However, after a more in-depth analysis of the two, it was found that their respective disadvantages are obvious. The core of the Transformer-Encoder is dilated causal convolution, which is simple in structure. It offers the characteristics of convolution kernel sharing, low memory consumption, high computing speed, and ease of stacking. However, due to the unidirectional structure, the extraction of information is not sufficiently comprehensive; although the diffusion rate has been doubled and the receptive field has been expanded, it is still limited, which is quite different from LSTM and GRU. BiGRU makes full use of memory and is capable of processing long time-series data. However, its structure is complex and calculation time is long.

The combination of the two can complement their advantages and disadvantages, and the feature extraction is more comprehensive. Higher accuracy can be achieved at the cost of less time to obtain more optimized results. Therefore, this paper proposes a Transformer-Encoder and BiGRU to process time-series information.

The detection process mainly includes three stages: network traffic preprocessing, feature extraction using the fusion framework of the transformer and BiGRU-DNN, and network traffic classification. The overall architecture is shown in Fig. 2.

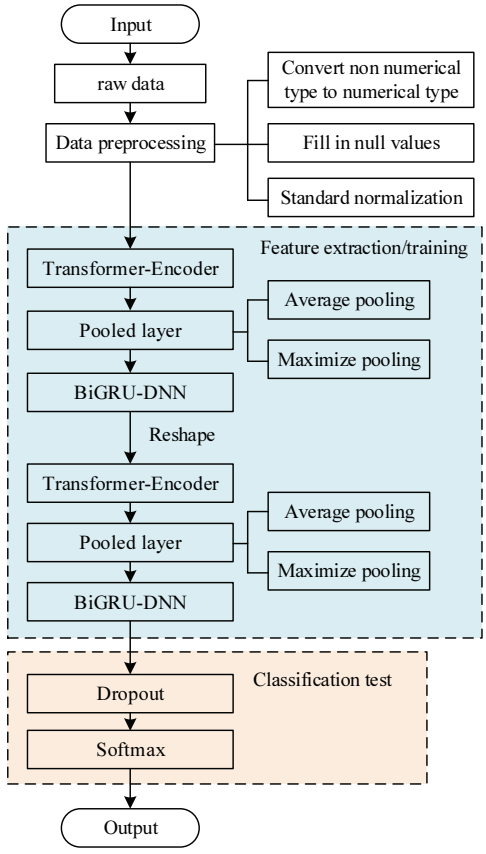


Fig. 2. Architecture of TEBGD deep learning model.

3.3 Data Preprocessing

3.3.1 Imbalanced data processing

First, the raw data (X, Y) are subjected to undersampling using the Tomek Links algorithm to eliminate noisy and boundary overlapping samples. The majority classes are then downsampled using the SMOTE algorithm, performing preliminary upsampling on the minority class data to generate data (X_{os}, Y), where $X_{s,l}$ represents the real training data labeled l generated after the initial upsampling of SMOTE. Next, the random noise data X_r pass through the generator to generate forged data X_{fake}. $X_{s,l}$ and X_{fake} are used to iteratively train the generator for each class. Finally, the trained model is used to generate minority class data (X_g, Y). The detailed process of data balancing is shown in Fig. 3.

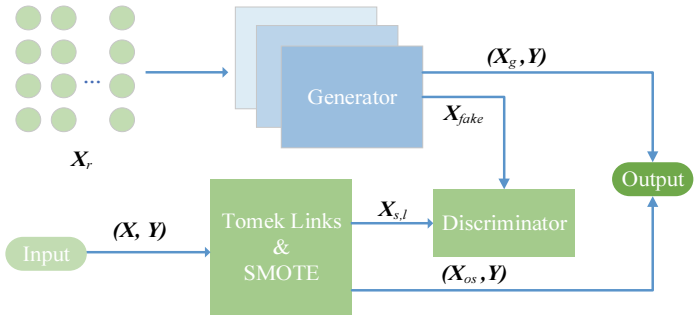


Fig. 3. Data balancing processing.

By creating fresh minority class samples between minority class samples, SMOTE provides data balance. This algorithm is prone to blurring the lines between majority and minority classes, making classification more difficult, and it is unable to solve the problem of data distribution marginalization in imbalanced datasets. The distribution of minority data cannot be fully learned by WGAN if the initial amount of data is too small. To overcome the aforementioned problems, WGAN is used to fully learn the distribution of minority class data based on the data provided by Tomek Links and SMOTE, thereby enhancing the quality of the generated data. The loss function is calculated as follows:

$$Loss(c) = \frac{1}{m}\sum_{i=1}^{m} f_w(x_i) - \frac{1}{m}\sum_{i=1}^{m} f_w\big(g_\theta(z_i)\big), \tag{1}$$

$$Loss(g) = \frac{1}{m}\sum_{i=1}^{m} f_w\big(g_\theta(z_i)\big), \tag{2}$$

where $Loss(c)$ and $Loss(g)$ represent the loss functions of the discriminator and generator in WGAN, respectively; g_θ represents a generator in WGAN; f_w represents the discriminator in WGAN; x represents real data; z represents random noise data; and m represents the size of a batch. When training the WGAN network, Adam can cause instability during model training. Therefore, RMSProp is used as the optimizer for WGAN network training.

3.3.2 MLP encoding

The data used in the experiment were discrete; therefore, one-hot encoding is used, inserting initial features, training it as a part of the whole, and then performing standard normalization on all data.

Similar to the word embedding layer in natural language processing, a multi-layer perceptron (MLP) [29] is used to encode each feature data, amplify and map features to different subspaces, extract richer features and achieve the input dimensions required by the model, and dynamically adjust MLP parameters during training.

3.4 Transformer-Encoder Model

The original model structure of the transformer includes two parts: encoding and decoding. Owing to the specific requirements of the detection tasks and fixed length of each data in the dataset, this model only uses the encoding part of the transformer and fine-tunes some of its parameters. The attention

mechanism uses dot-product attention; that is, there are three inputs: query, keys, and values. Dot product attention can be used for parallel operations to reduce training time:

$$Att(A,B,C) = softmax\left(\frac{AB^{\mathrm{T}}}{\sqrt{w_b}}\right)C, \tag{3}$$

where A, B, and C represent the three matrices of query, key, and value, respectively, and w represents the dimensions of key. To enrich the extracted features, the structure of multi-head attention is used. The multi-head attention is defined as

$$\begin{cases} A_i = AQ_i^A, B_i = BQ_i^B, C_i = CQ_i^C \\ H_i = Att\ (A_i, B_i, C_i) \\ MH(A,B,C) = Concat\ (H_1, H_2, \dots, H_i, \dots, H_n) \end{cases}, \tag{4}$$

where $i = 1,2,3, \dots, n$, Q represents the weight, H represents head, and MH represents multi-head. The feed-forward neural network part is a perceptron with only one hidden layer, and its input and output dimensions are the same. The total number of neurons in the hidden layer is set to be twice that of the input layer because of the limited nonlinear mapping ability of the single hidden layer network and the trade-off between computational cost and mapping ability.

As is apparent from Formula (5), the activation function is a Gaussian error linear unit (GELU):

$$G(x) = \frac{1}{2}x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right)\right). \tag{5}$$

Fig. 4 depicts the overall structural layout of the Transformer-Encoder module. This section employs a residual connection to stop gradient disappearance.

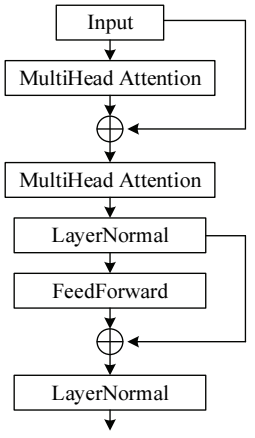


Fig. 4. Structure of Transformer-Encoder module.

3.5 BiGRU-DNN

The processing of BiGRU is shown in Fig. 5.

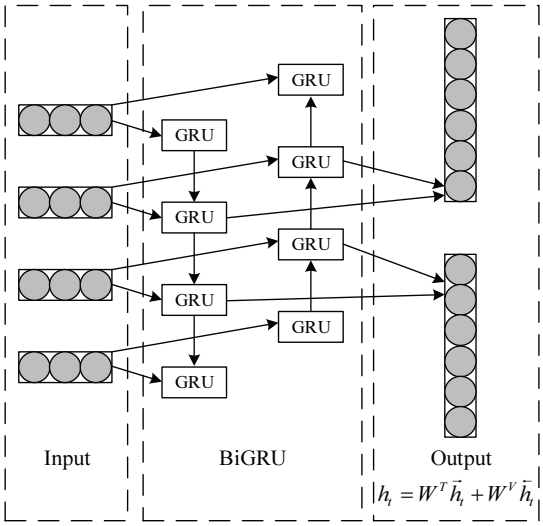


Fig. 5. Process of BiGRU model.

The processing of the BiGRU model is represented by the following formula:

$$\begin{cases} \vec{h}_t = GRU(x_t, \vec{h}_{t-1}) \\ \overleftarrow{h}_t = GRU(x_t, \overleftarrow{h}_{t-1}) \\ h_t = Q_F \vec{h}_t + Q_B \overleftarrow{h}_t \end{cases} \tag{6}$$

where Q_F and Q_B are the weighted factors of the BiGRU output layer.

The DNN in the proposed model has two hidden layers: ReLU and Dropout, where the Dropout is set to 0.5. The DNN structure is shown in Table 1, where b_s represents the batch size and * represents settings based on actual needs. If N classification tasks are performed, it is set to N. The general formula for the DNN calculation is as follows:

$$f(x) = \sigma(WX + b) \tag{7}$$

Table 1. Structure of DNN

Layer type	Output dimension
Input	(b_s, 128)
Linear-1	(b_s, 64)
ReLU-2	(b_s, 64)
Dropout-3	(b_s, 64)
Linear-4	(b_s, 32)
ReLU-5	(b_s, 32)
Dropout-6	(b_s, 32)
Linear-7	(b_s, *)

3.6 Classifier Design

After the model training is completed, the trained model is used to classify the test set and obtain the prediction type. A k-fold cross-validation method is used to test the model and ensure credibility of the

test results. The softmax function is used to calculate the probability of the classification results and compare them with the original labels. The calculation of softmax is as follows:

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}, j = 1,2, \dots, K \tag{8}$$

In addition, for the multi-classification problem of abnormal traffic, a multi-classification calculation formula for softmax was designed as follows:

$$h_\theta\left(x^{(i)}\right) = \begin{bmatrix} P\left(y^{(i)} = 0 \mid x^{(i)}; \theta\right) \\ P\left(y^{(i)} = 1 \mid x^{(i)}; \theta\right) \\ \vdots \\ P\left(y^{(i)} = k-1 \mid x^{(i)}; \theta\right) \end{bmatrix} = \frac{1}{\sum_{j=0}^{k=1} e^{\theta_j^T x(i)}} \begin{bmatrix} e^{\theta_0^T x(i)} \\ e^{\theta_1^T x(i)} \\ \vdots \\ e^{\theta_{k-1} T_1 x(i)} \end{bmatrix}, \tag{9}$$

where $h_\theta(x^{(i)}), \theta_0, \theta_1, \dots, \theta_{k-1}$ is the parameter to be determined and $\frac{1}{\sum_{j=0}^{k=1} e^{\theta_j^T x(i)}}$ is the normalization factor of the function.

4. Experiments

4.1 Environment

The abnormal detection system was developed using PC devices. It mainly uses Python language and its related library to realize data collection, data preprocessing, behavior detection, and analysis functions. The MySQL database management software is used to realize data storage function. The Django framework is used to build a front-end interface based on the http protocol, which can output behavior detection results.

Table 2. Hardware environment

Name	Configuration
OS	Windows 10 1903
CPU	AMD Ryzen 3700x
GPU	NVIDIA RTX 2070 Super 8 GB
Hard disk	1 TB
Memory	8 GB
Programming language	Python 3.5
Virtual machine OS	Ubuntu 16.04 LTS

Table 3. Software configuration environment

Name	**Configuration**
IDE	Visual Studio Code
Virtual machine software	VMare Workstation
Command line	Windows Terminal
Database management	MySQL 5.5
Development language	Python 3.7
Python dependency library	TensorFlow

The hardware configuration of the abnormal detection system is described in Table 2. The software environment of the abnormal detection system is described in Table 3.

4.2 Datasets

Three datasets, namely NSLKDD [30], UNSWNB15 [31], and CICIDS2017 [32], were used for the experimental evaluations. Based the NSLKDD dataset, the redundant and repetitive data in the training and test sets were eliminated so that the setting of the dataset was more reasonable and to obtain a more accurate detection rate. The UNSWNB15 dataset, released in 2015, overcomes the limitations of the NSLKDD dataset to a certain extent. The CICIDS2017 dataset was derived from network data collected by the Canadian Institute of Network Security from July 3 to 7, 2017, including benign and the latest common attacks in the field of network intrusion, filling the gap in that there are no network-based attacks in the UNSWNB15 dataset.

These three datasets are primarily used for multi-category attack prediction. In the NSLKDD dataset, there are five types: normal traffic, denial of service attack, port attack, empowerment attack, and remote user attack. The UNSWNB15 dataset contains ten types: normal, DoS, exploits, generic, conservation, words, shellcode, analysis, backlight, and fuzzer. In the CICIDS2017 dataset, there are 15 types in total, among which there are nine traffic types: BENIGN, DoS, portscan, DDoS, pattor, bot, webattack, infiltration, and heartbeat, obtained after merging those with similar nature of abnormal attacks.

4.3 Evaluation Indicators

In general, the performance evaluation of abnormal detection methods includes five evaluation criteria: accuracy, precision, recall, F1 value, and false alarm rate (FAR). These evaluation criteria are defined using four functions: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TP is the number of samples that correctly predict attack samples as attack categories, TN represents the number of normal samples predicted as normal categories, FP represents the number of samples wrongly predicted as attack categories, and FN represents the number of samples wrongly predicted as normal categories. These functions can be obtained from the confusion matrix C. The elements C_{ij} of confusion matrix C represent the number of samples that belong to category i predicted as category j. The formulas for accuracy, precision, recall, F1 value, and false alarm rate are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{2 * Precision * Recall}{TP + FN} \tag{12}$$

$$F1 = \frac{TP}{Precision + Recall} \tag{13}$$

$$False\ alarm\ rate = \frac{FP}{TN + FP} \tag{14}$$

4.4 Model Performance Training

The proposed model was tested for multiple indicators, and the change trends of accuracy, precision, recall, F1 value, and false alarm rate with epochs on the NSLKDD, UNSWNB15, and CICIDS2017 datasets are shown in Fig. 6.

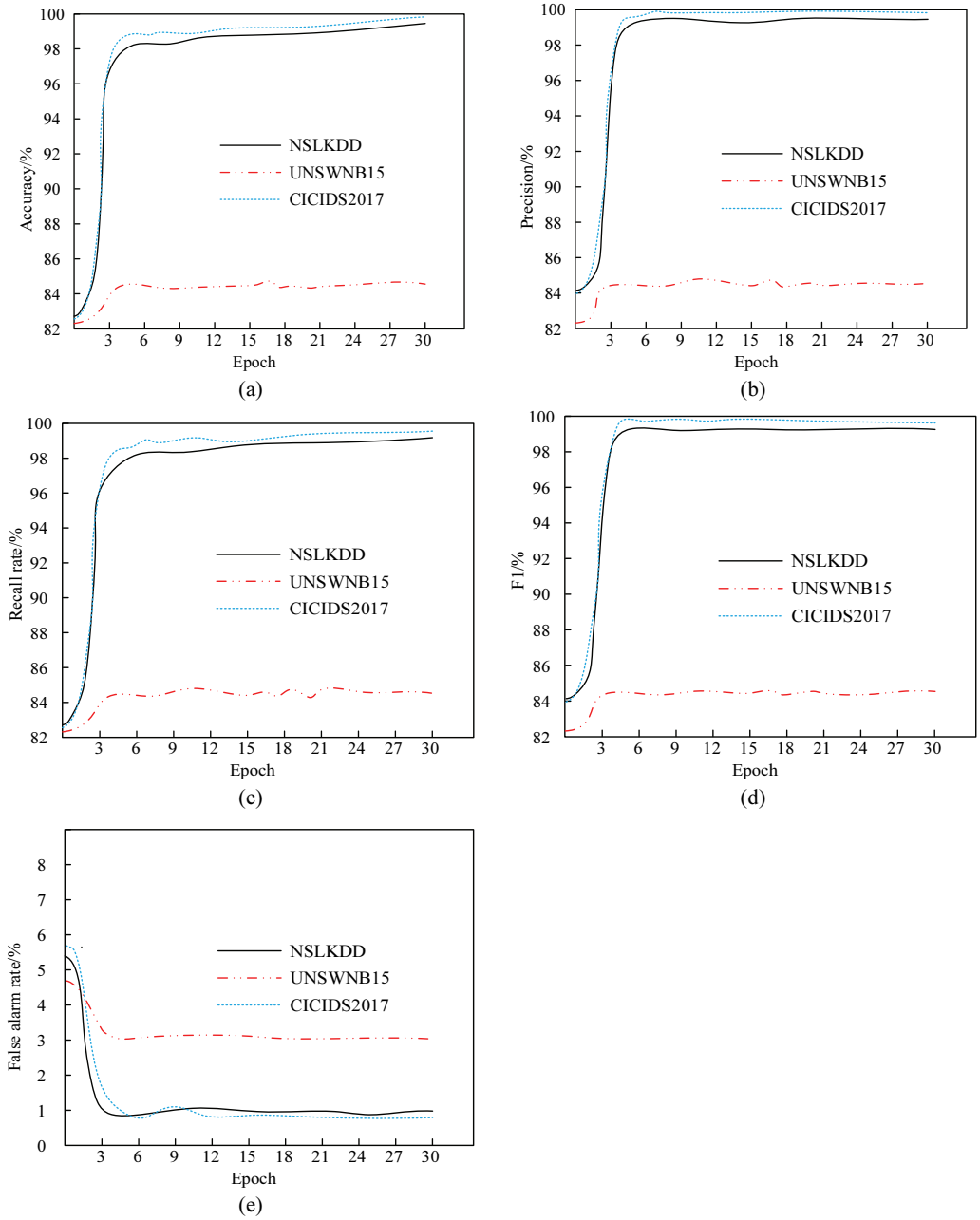


Fig. 6. Experimental results of different evaluation indicators: (a) accuracy vs. epochs, (b) precision vs. epochs, (c) recall vs. epochs, (d) F1 value vs. epochs, and (e) false alarm rate vs. epochs.

The time consumed by the TEBGD-based abnormal detection method when using different datasets varied with the amount of data, as shown in Fig. 7.

As shown in Fig. 6, as the number of epochs increases, the detection performance of the proposed model shows an upward trend when using the three different datasets. Ultimately, convergence can be achieved on all three datasets, with accuracy rates as high as 98.96%, 83.23%, and 99.78%, respectively. The false alarm rates were 0.35%, 3.98%, and 0.28%, respectively, all of which are below 4%.

As shown in Fig. 7, when the number of samples is above 2.5 × 105, it only takes approximately 3 seconds to detect the user's abnormal behavior in the data. Therefore, the proposed abnormal detection method has superior detection performance and detection efficiency.

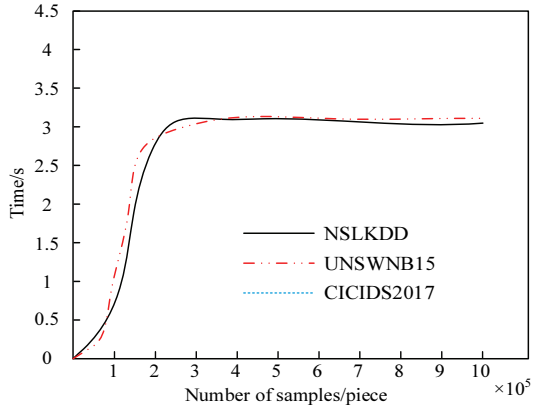


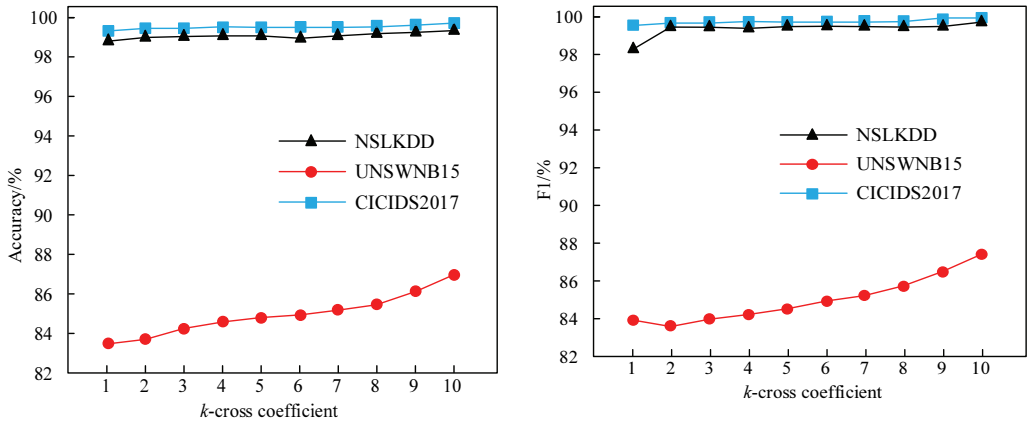
Fig. 7. Relationship between model time consumption and sample number.

4.5 Model Performance Analysis Experiment

4.5.1 Using k-fold cross-validation analysis

The k-fold cross-validation method was used to verify the intrusion detection performance of the proposed model. The accuracy and F1 values obtained as a function of k are shown in Fig. 8.

As shown in Fig. 8, after testing with k values ranging from 2 to 10, the accuracy and F1 values of the NSLKDD, UNSWNB15, and CICIDS2017 datasets all increased with increasing k values. This is because, as the k-value increases, the number of partitions in the dataset increases, and the data used as the training set will also increase. The more data used in training, the higher the final evaluation indicators, such as test accuracy, will be. The optimal accuracy of multi-classification in the NSLKDD dataset was 99.72% when k=10, and the optimal F1 value was 99.52% when k = 10. The optimal accuracy and F1 value for multi-classification in the UNSWNB dataset were also obtained at k = 10, with values of 84.86% and 84.12%, respectively. Similarly, the optimal multi-classification performance in the CICIDS2017 dataset was achieved at k = 10, with an accuracy and F1 value of 99.89% and 99.28%, respectively. When k = 10, the proposed model achieved good multi-classification performance, and as the number of folds increased, the number of samples for each attack or normal type also increased. Therefore, the model will be able to better classify them.



(a) (b)

Fig. 8. Detection performance changes on the three datasets corresponding to the k-crossover coefficient: (a) accuracy and (b) F1 value.

4.5.2 Performance impact of different encoding methods

Because DL can only process numerical data, it is necessary to convert irregular string content in the original dataset into numerical data. Currently, two common numerical methods exist: single-hot encoding and label encoding. To demonstrate the reliability of unique encoding in the proposed model, three datasets were preprocessed using different encoding methods. The detection results obtained after inputting the processed encoded data into the proposed DL network model are listed in Table 4.

Table 4. Comparison of detection accuracy (%) corresponding to one-hot encoding and tag encoding

Dataset	Encoding method	
	One-hot	Tag
NSL-KDD	99.72±0.10	98.96±0.16
UNSW-NB15	84.86±0.15	83.23±0.19
CIC-IDS2017	99.89±0.05	99.78±0.10

From Table 4, using the unique hot encoding method to digitize the string type features in the dataset has a slightly higher detection accuracy than using the label encoding method. This is because label encoding converts features into continuous numerical values, that is, numbering discontinuous features, which leads to size relationships between features and produces partial ordering, which has a certain impact on the classification performance. However, unique encoding can transform the discrete features of the original data into Euclidean space through a series of feature transformations, keeping the distances between features consistent, solving the aforementioned problems, and improving detection accuracy. Therefore, all the experiments with the proposed model used a single-hot encoding method to preprocess the data.

4.5.3 Performance influence of different pooling methods

In the pooling stage of the proposed model, average and max pooling are fused to improve the feature extraction capabilities. Three different pooling methods were used in the experiments to verify the superiority of the fusion method used. The detection results on the NSLKDD, UNSWNB15, and

CICIDS2017 datasets are listed in Table 5.

Table 5. Impact of three different pooling methods on detection results (%)

Dataset	Pooling method		
	Average	Maximum	Fusion
NSL-KDD	99.02±0.09	99.26±0.12	99.72±0.12
UNSW-NB15	83.28±0.17	83.87±0.15	84.86±0.15
CIC-IDS2017	99.49±0.05	99.61±0.06	99.89±0.05

From Table 5, compared with the pooling scheme alone, the fusion method can achieve a higher detection accuracy. The averaging method can extract features with global significance, whereas the maximum method can extract local features with certain significance. By integrating global and local pooling, essential features can be extracted more accurately, thereby maximizing the feature extraction capability of the proposed model.

4.6 Comparison of Proposed and Several Other Advanced Models

The proposed model was compared with CNN-DNN-LSTM [15], BiLSTM-AE [17], and ICVAE-BSM [22] under the conditions of the three datasets, and the results are listed in Table 6 and Fig. 9.

Table 6. Results of different methods using three datasets (%)

Dataset	Method	Accuracy	Precision	Recall	F1 value	FAR
NSLKDD	Proposed method	99.72	99.68	99.80	99.65	0.35
	CNN-DNN-LSTM [15]	97.63	97.59	97.70	97.56	0.43
	BiLSTM-AE [17]	96.53	96.49	96.61	96.46	0.52
	ICVAE-BSM [22]	98.92	98.88	99.00	98.85	0.40
UNSWNB15	Proposed method	84.86	84.73	85.21	84.88	3.98
	CNN-DNN-LSTM [15]	83.08	82.95	83.42	83.10	4.21
	BiLSTM-AE [17]	82.14	82.02	82.48	82.16	5.53
	ICVAE-BSM [22]	84.18	84.05	84.53	84.20	4.15
CICIDS2017	Proposed method	99.89	99.92	99.86	99.91	0.28
	CNN-DNN-LSTM [15]	95.30	95.32	95.27	95.31	0.34
	BiLSTM-AE [17]	90.91	90.94	90.88	90.93	0.53
	ICVAE-BSM [22]	86.73	86.76	86.70	86.75	3.72

Based on the above results, several comparative models achieved satisfactory results on the three datasets. An analysis of the reasons for this shows that, due to the strong nonlinear fitting ability of neural networks, they can map any complex nonlinear relationship, and the feature extraction ability of several comparative models is strong, resulting in high accuracy.

Compared with the other models, the proposed model achieved the best results. Analyzing the reasons for this, both the CNN-DNN-LSTM and BiLSTM-AE models achieved good results. Compared with CNN-DNN-LSTM, the BiLSTM-AE model performed better, mainly owing to the powerful feature filtering ability of AE. Moreover, BiLSTM, through the stacking of two layers of LSTM, breaks away from the limitation that the model can only predict the output of the next time based on the temporal

information of the previous time and can better combine context for output. However, the results of these two models are inferior to those of ICVAE-BSM, mainly because ICVAE has a strong feature encoding ability and alleviates the problem of data class imbalance through BSM before feature extraction. The problem of data class imbalance can significantly affect the performance of a model. The proposed model not only considers the problem of data class imbalance but also introduces a transformer with strong encoding ability. In addition, BiGRU can effectively extract the temporal features of data like BiLSTM, and DNN can be used to extract deep-level features. In other words, the proposed model simultaneously possesses the advantages of the CNN-DNN-LSTM, BiLSTM-AE, and ICVAE-BSM models; therefore, the proposed model can achieve the best results.

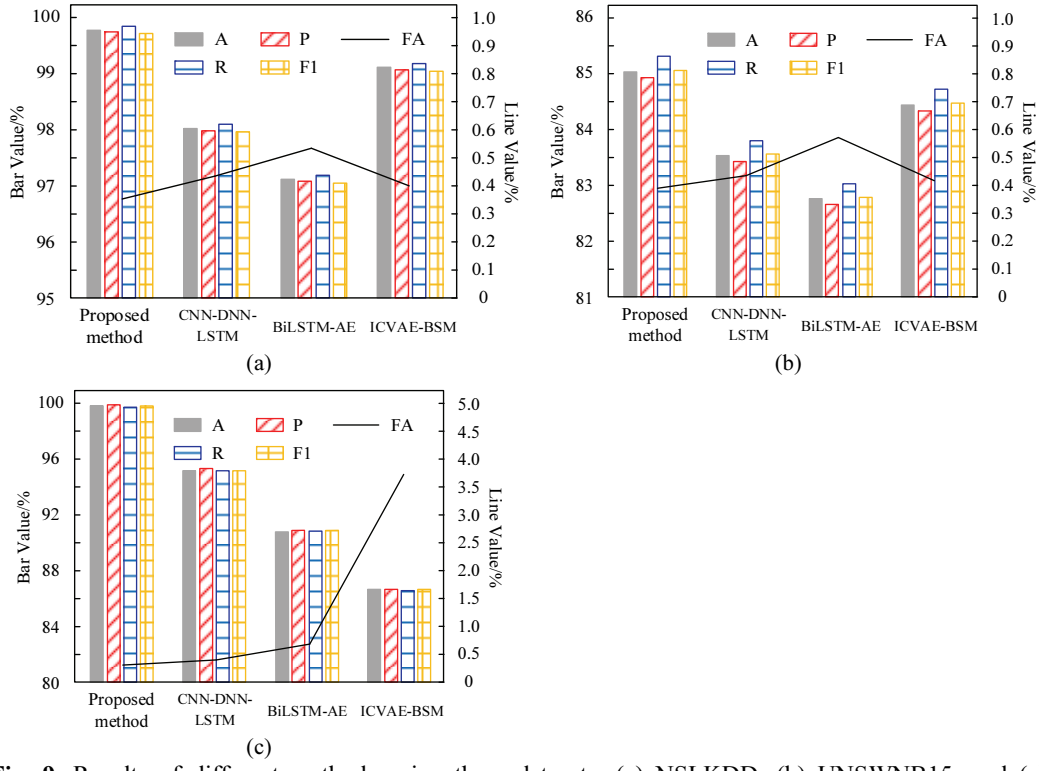


Fig. 9. Results of different methods using three datasets: (a) NSLKDD, (b) UNSWNB15, and (c) CICIDS2017.

5. Conclusion

A combined framework integrating the transformer and neural network models was proposed to alleviate data imbalance factors in network traffic data that affect network intrusion detection performance. Experiments were conducted on the NSLKDD, UNSWNB15, and CICIDS2017 datasets, with detection accuracy rates of up to 99.72%, 84.86%, and 99.89%, respectively. Compared to other relatively new DL network models, the proposed model effectively improved the detection results, thereby improving the communication security of network data. Through an analysis of the experimental results, the following conclusions were drawn:

1) Integrating Tomek Links, SMOTE, and WGAN can effectively solve class imbalance problems.
2) Using the transformer to encode data can effectively extract the correlation between network traffic, making it more conducive for the model to identify abnormal traffic.
3) Integrating BiGRU and DNN can effectively extract long-dependent temporal features of data, as well as deep-level features, thereby enhancing the global and local feature extraction capabilities of the model.
4) Although the proposed model achieved superior intrusion detection results, the model is more complex and needs to be trained in advance before it can be deployed to an edge device.

The proposed model can be improved to a lightweight model without affecting the performance of model detection for better application to real edge computing scenarios. In addition, the proposed model will be deployed in many real edge computing scenarios, and joint training will be conducted using big data from real network traffic in actual application scenarios, which will help improve the generalization ability of the model.

Conflict of Interest

The author declare that they have no competing interests.

Funding

None.

References

[1] Y. Shen, K. Zheng, and C. Wu, "A hybrid PSO-BPSO based kernel extreme learning machine model for intrusion detection," *Journal of Information Processing Systems*, vol. 18, no. 1, pp. 146-158, 2022. https://doi.org/10.3745/JIPS.03.0174

[2] R. Amin, M. Hussain, M. Alhameed, S. M. Raza, F. Jeribi, and A. Tahir, "Edge-computing with graph computation: a novel mechanism to handle network intrusion and address spoofing in SDN," *Computers, Materials & Continua*, vol. 65, no. 3, pp. 1869-1890, 2020. https://doi.org/10.32604/cmc.2020.011758

[3] L. Nie, Y. Wu, X. Wang, L. Guo, G. Wang, X. Gao, and S. Li, "Intrusion detection for secure social internet of things based on collaborative edge computing: a generative adversarial network-based approach," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 134-145, 2022. https://doi.org/10.1109/TCSS.2021.3063538

[4] H. Bangui and B. Buhnova, "Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms," *Computers and Electrical Engineering*, vol. 100, article no. 107901, 2022. https://doi.org/10.1016/j.compeleceng.2022.107901

[5] I. H. Hassan, M. Abdullahi, M. M. Aliyu, S. A. Yusuf, and A. Abdulrahim, "An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection," *Intelligent Systems with Applications*, vol. 16, article no. 200114, 2022. https://doi.org/10.1016/j.iswa.2022.200114

[6] F. Salo, A. B. Nassif, and A. Essex, "Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection," *Computer Networks*, vol. 148, pp. 164-175, 2019. https://doi.org/10.1016/j.comnet.2018.11.010

[7] H. Zhang, J. L. Li, X. M. Liu, and C. Dong, "Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection," *Future Generation Computer Systems*, vol. 122, pp. 130-143, 2021. https://doi.org/10.1016/j.future.2021.03.024

[8] C. Zhang, D. Jia, L. Wang, W. Wang, F. Liu, and A. Yang, "Comparative research on network intrusion detection methods based on machine learning," *Computers & Security*, vol. 121, article no. 102861, 2022. https://doi.org/10.1016/j.cose.2022.102861

[9] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, article no. 102031, 2020. https://doi.org/10.1016/j.simpat.2019.102031

[10] M. Al-Hawawreh and E. Sitnikova, "Industrial Internet of Things based ransomware detection using stacked variational neural network," in *Proceedings of the 3rd International Conference on Big Data and Internet of Things*, Melbourne, Australia, 2019, pp. 126-130. https://doi.org/10.1145/3361758.3361763

[11] H. Liu, Z. Zhou, and M. Zhang, "Application of optimized bidirectional generative adversarial network in ICS intrusion detection," in *Proceedings of 2020 Chinese Control and Decision Conference (CCDC)*, Hefei, China, 2020, pp. 3009-3014. https://doi.org/10.1109/CCDC49329.2020.9164558

[12] Y. Z. Li, Y. Li, and S. Zhang, "Intrusion detection algorithm based on deep learning for industrial control networks," in *Proceedings of the 2019 The 2nd International Conference on Robotics, Control and Automation Engineering*, Lanzhou, China, 2019, pp. 40-44. https://doi.org/10.1145/3372047.3372092

[13] I. Al-Turaiki and N. Altwaijry, "A convolutional neural network for improved anomaly-based network intrusion detection," *Big Data*, vol. 9, no. 3, pp. 233-252, 2021. https://doi.org/10.1089/big.2020.0263

[14] N. B. Singh, M. M. Singh, A. Sarkar, and J. K. Mandal, "A novel wide & deep transfer learning stacked GRU framework for network intrusion detection," *Journal of Information Security and Applications*, vol. 61, article no. 102899, 2021. https://doi.org/10.1016/j.jisa.2021.102899

[15] S. Liaqat, A. Akhunzada, F. S. Shaikh, A. Giannetsos, and M. A. Jan, "SDN orchestration to combat evolving cyber threats in Internet of Medical Things (IoMT)," *Computer Communications*, vol. 160, pp. 697-705, 2020. https://doi.org/10.1016/j.comcom.2020.07.006

[16] M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K. K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting IoT cyber attacks using network traffic," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8852-8859, 2020. https://doi.org/10.1109/JIOT.2020.2996425

[17] E. Mushtaq, A. Zameer, M. Umer, and A. A. Abbasi, "A two-stage intrusion detection system with auto-encoder and LSTMs," *Applied Soft Computing*, vol. 121, article no. 108768, 2022. https://doi.org/10.1016/j.asoc.2022.108768

[18] Y. Li, T. Qin, Y. Huang, J. Lan, Z. Liang, and T. Geng, "HDFEF: a hierarchical and dynamic feature extraction framework for intrusion detection systems," *Computers & Security*, vol. 121, article no. 102842, 2022. https://doi.org/10.1016/j.cose.2022.102842

[19] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, article no. 27, 2019. https://doi.org/10.1186/s40537-019-0192-5

[20] K. R. M. Fernando and C. P. Tsokos, "Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 7, pp. 2940-2951, 2022. https://doi.org/10.1109/TNNLS.2020.3047335

[21] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion detection of imbalanced network traffic based on machine learning and deep learning," *IEEE Access*, vol. 9, pp. 7550-7563, 2020. https://doi.org/10.1109/ACCESS.2020.3048198

[22] Y. Zhang and Q. Liu, "On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples," *Future Generation Computer Systems*, vol. 133, pp. 213-227, 2022. https://doi.org/10.1016/j.future.2022.03.007

[23] S. Al and M. Dener, "STL-HDL: a new hybrid network intrusion detection system for imbalanced dataset on big data environment," *Computers & Security*, vol. 110, article no. 102435, 2021. https://doi.org/10.1016/j.cose.2021.102435

[24] L. Tian and Y. Lu, "An intrusion detection model based on SMOTE and convolutional neural network ensemble," *Journal of Physics: Conference Series*, vol. 1828, no. 1, article no. 012024, 2021. https://doi.org/10.1088/1742-6596/1828/1/012024

[25] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017 [Online]. Available: https://arxiv.org/abs/1701.07875.

[26] J. Wang, N. Chen, J. Yu, Y. Jin, and Y. Li, "An efficient intrusion detection model combined bidirectional gated recurrent units with attention mechanism," in *Proceedings of 2020 7th International Conference on Behavioural and Social Computing (BESC)*, Bournemouth, UK, 2020, pp. 1-6. https://doi.org/10.1109/BESC51023.2020.9348310

[27] W. Yang, S. Wang, and M. Johnstone, "A comparative study of ML-ELM and DNN for intrusion detection," in *Proceedings of the 2021 Australasian Computer Science Week Multiconference*, Dunedin, New Zealand, 2021, pp. 1-7. https://doi.org/10.1145/3437378.3437390

[28] S. Hanzawa, T. Sakata, K. Kajigaya, R. Takemura, and T. Kawahara, "A large-scale and low-power CAM architecture featuring a one-hot-spot block code for IP-address lookup in a network router," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 853-861, 2005. https://doi.org/10.1109/JSSC.2005.845554

[29] F. Amato, N. Mazzocca, F. Moscato, and E. Vivenzio, "Multilayer perceptron: an intelligent model for classification and intrusion detection," in *Proceedings of 2017 31st International conference on advanced information networking and applications workshops (WAINA)*, Taipei, Taiwan, 2017, pp. 686-691. https://doi.org/10.1109/WAINA.2017.134

[30] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT," *Procedia Computer Science*, vol. 167, pp. 1561-1573, 2020. https://doi.org/10.1016/j.procs.2020.03.367

[31] G. Kocher and G. Kumar, "Analysis of machine learning algorithms with feature selection for intrusion detection using UNSW-NB15 dataset," 2021 [Online]. Available: https://doi.org/10.2139/ssrn.3784406.

[32] L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Inter-dataset generalization strength of supervised machine learning methods for intrusion detection," *Journal of Information Security and Applications*, vol. 54, article no. 102564, 2020. https://doi.org/10.1016/j.jisa.2020.102564

Huijuan Sun **https://orcid.org/0000-0002-5039-8585**

She graduated from Henan University with a Master of Science in 2008 and she is now an associate professor and works in Henan Finance University, whose research interests include optimization algorithms and big data analysis.