IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 June 2023, accepted 26 June 2023, date of publication 28 June 2023, date of current version 6 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3290613

RESEARCH ARTICLE

A Hierarchical Intrusion Detection Model Combining Multiple Deep Learning Models With Attention Mechanism

HONGSHENG XU[1,2], LIBO SUN[1,3], GANGLONG FAN[1,2], WANXING LI[1,2], AND GUOFANG KUANG[4]

[1]College of Electronic Commerce, Luoyang Normal University, Luoyang, Henan 471934, China
[2]Henan Key Laboratory for Big Data Processing and Analytics of Electronic Commerce, Luoyang Normal University, Luoyang, Henan 471934, China
[3]Binghamton University, State University of New York, Binghamton, NY 13902, USA
[4]School of Information Technology, Luoyang Normal University, Luoyang, Henan 471934, China

Corresponding author: Hongsheng Xu (xhsls@lynu.edu.cn)

This work was supported in part by the National Natural Science Funds of China under Grant 61272015, in part by the 2022 Henan Province Key Research and Development and Promotion Projects (Science and Technology) under Grant 222102320342, and in part by the 2023 Henan Province Key Research and Development and Promotion Projects (Science and Technology) under Grant 232102320016.

ABSTRACT In order to ensure the security of computer systems and networks, it is very important to design and implement intrusion detection systems that can detect and mitigate network attacks and threats. Deep learning has great advantages in processing complex, high-dimensional and large-scale traffic data. Therefore, intrusion detection system based on deep learning method has better detection effect. Through the analysis of the research status, this paper finds that there are some problems in the existing intrusion detection system. To solve the problems of low detection accuracy, structure to be optimized and high false positive rate, this paper presents a hierarchical intrusion detection model which combines multiple deep learning models with attention mechanism. The advantages of this hierarchical model include: Firstly, the SCDAE model is used to extract the features of traffic data and reduce noise; Secondly, CNN is used to extract spatial features of network traffic data from the spatial dimension; Thirdly, BiLSTM is able to fully consider the relationship between the front and back features, so that the temporal features of network traffic data can be mined; Fourthly, a Self-Attention mechanism is added to weight the output of each time step to sum up and retain the important information in it. Thus, a CNN-BiLSTM-Attention model is constructed; Finally, the Softmax classifier is used to obtain the classification results. To verify the effectiveness of the proposed model, four time-sensitive and representative datasets are selected for experiments and five classical detection models are compared in this paper. The experimental results show that the classification accuracy of the proposed model reaches 93.26 % and the false positive rate reaches 7.53%.

INDEX TERMS Deep learning, intrusion detection system (IDS), stacked convolutional denoising autoencoders (SCDAE), convolution neural network (CNN), bi-directional long short-term memory (BiLSTM), attention mechanism.

I. INTRODUCTION

With the rapid development of the network, network security incidents have occurred frequently in recent years, and network security is facing a huge challenge. In order to solve the problem of network security, researchers have proposed intrusion detection technology. Intrusion detection technology is an active means of network security defense. This technology identifies attacks by analyzing the data flow characteristics of systems and networks, and then takes appropriate security measures to stop the attacks, thereby securing the network [1].

Traditional intrusion detection technology is facing many challenges, such as the complexity of network data, the diversity of intrusion methods. Intrusion detection technology

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino.

based on deep learning can extract multi-layer, abstract and high-quality features from data by collecting information of some key nodes in the network. The model ensures a high accuracy rate for multiple classifications of intrusion detection data while reducing the false positive rate. To a certain extent, it is a good solution to the problems of current intrusion detection technology.

Firstly, this paper analyzes the related literature on intrusion detection methods in detail. This paper uses the unsupervised Generic Adversary Networks model to solve the problem of data imbalance. Then the random forest classifier is used to detect the performance of the model, and the detection effect is better than other data set balancing methods [2]. However, the accuracy of this GAN model is low. The author uses the improved KNN algorithm to obtain high detection rate by using part of the data [3]. Staudemeyer first applied a combination of Long Short-Term Memory and Recurrent Neural Network to network intrusion detection [4], and the experimental results proved that the method is well suited for classifying high-frequency attacks. However, the detection structure of these two methods is relatively single and cannot detect complex networks.

The paper proposes the use of stacked sparse autoencoders to extract high-level feature representations of intrusion behavior information. The original classification features are introduced into the stacked sparse self-encoder, and automatic learning of deep sparse features is achieved for the first time [5]. However, the detection ability of this method is insufficient. The authors proposed a network intrusion detection method that integrates CNN and BiLSTM. Compared with using CNN and Bi LSTM network alone, this method has high accuracy and low false positive rate. However, the model parameters of this method are too many, which makes the model easy to fall into local optimum and cannot consider the global situation [6].

This article proposes a wireless intrusion detection system classifier based on deep long short-term memory network [7]. Using NSL-KDD dataset, DLSTM-IDS is compared with the existing classical methods. The experimental results show that the performance of DLSTM-IDS is better than the existing methods. The limitation of this method is the problem of low training efficiency. The paper presents an intrusion detection method based on a lightweight dynamic autoencoder network. This method realizes efficient feature extraction through lightweight structure design, which greatly reduces the calculation cost and model size [8]. The limitation of this method is that it does not consider the relationship between features comprehensively, resulting in a high false positive rate of the model.

Vipin et al. used K-Means algorithm for intrusion detection, which was validated on the NSL-KDD dataset to improve the accuracy of intrusion detection [9]. However, the accuracy of this method is low. Karatas et al. compared the performance of different ML algorithms by using the latest underlying CIC-IDS2018 dataset [10]. However, the test dataset adopted in this experiment is not recent. The authors propose a model called HAST-IDS that combines CNN and LSTM networks to learn directly from the original web stream files, automatically learning traffic features without manual feature engineering techniques [11]. The limitations of this model are insufficient detection ability and single structure. Sheraz et al. [12] developed anomaly detection models based on different depth neural network structures. These depth models are trained and evaluated on the NSL-KDD dataset. The limitation of this method is low training efficiency and poor learning ability. This paper proposes an intrusion detection method based on attention mechanism and LSTM network. This method uses the advantages of attention mechanism to solve the problem that key attributes cannot be concerned in intrusion detection [13]. The limitation of this method is that the accuracy and precision rate are low.

Through the analysis of existing intrusion detection methods, it is found that the existing intrusion detection models have low training efficiency, insufficient detection capability, single structure and low accuracy rate. In addition, the existing models do not fully consider the relationship between features, resulting in a high false positive rate. To solve these problems, this paper proposes a hierarchical intrusion detection model that combines multiple deep learning models and self attention mechanism.

The main functions of this model are as follows: The data is scanned byte by byte based on the SCDAE model to extract the features of the traffic data and perform the noise reduction process; We use CNN to mine the spatial features of network traffic data. BiLSTM can preserve the contextual information of the data for a long time and thus extract time series features. In the process of feature extraction, the fusion model needs to consider not only the spatial features of data, but also the correlation features of data in time series. By combining CNN with BiLSTM, the advantages of both can be exploited to extract the full range of information from the data, which can be used to better improve the classification of intrusion detection. Finally, the Self-Attention mechanism is introduced on the basis of CNN-BiLSTM model. In weight calculation, it is less dependent on external data, and better at capturing the internal correlation between features, so that some important features are focused on during model training, and the classification accuracy of the model is improved. Finally, the Softmax function is used for data classification.

To verify the effectiveness of this hierarchical intrusion detection model, this paper selects the latest CIC-DDoS2019, CIC-IDS2017, CIC-IDS2018 and the unbalanced NSL-KDD dataset for experiments, and compares classical models such as KNN, RF, CNN, BiLSTM and CNN BiLSTM. The experimental results show that the accuracy, precision and F1-score of the proposed model are higher than other models, while the false positive rate is relatively low, thus proving the superiority of the proposed model in this paper.

II. RELATED TECHNICAL ANALYSIS

A. AUTOENCODERS

Autoencoders belongs to unsupervised learning, and their main function is to reduce the dimension of data and feature extraction [8]. The feature of automatic encoder is that the input and output content information can be the same after training. The network structure is composed of input layer, hidden layer and output layer. The input layer is used to input the original data, the middle hidden layer is used to learn the data features, and the output layer is used to output the reconstruction of the input data [14].

Assuming the input sample x, the encoding function f of the encoder is obtained as follows Eq.

$$h = f(x) \tag{1}$$

The output data r with the same dimension as the original input data is obtained by calculating the encoded vector h through the decoder's decoding function g as follows Eq.

$$r = g(h) = g(f(x)) \tag{2}$$

The implementation in the encoder and decoder is a nonlinear mapping, and the implementation scheme is represented by the following two equations.

$$f(x) = s_f(W_x + b) \tag{3}$$

$$g(x) = s_g(W_x^T + d) \tag{4}$$

In the above equation, s_f represents the activation function on the encoder and s_g represents the activation function on the decoder. W^T represents the weight matrix between layers, b and d represent the bias vectors.

The purpose of the Autoencoders network is to make the decoded r as similar as possible to the pre-encoding x. The decoded samples are compared with the real original samples and the reconstruction error can be calculated. The error between them can be expressed by the loss function L(x,r), as shown in the following equation.

$$L(x, r) = \|\bar{x}\|^2 r \tag{5}$$

In the above equation, x represents the sample and L(x,r) denotes the loss function.

B. CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural network(CNN) is a deep neural network that contains convolutional operations. Convolutional neural network has the ability of representation learning and can learn the spatial hierarchy of input information [15]. The features learned by CNN are translation invariant and can be used for supervised and semi-supervised learning. A convolutional neural network consists of input layer, convolutional layer, pooling layer, fully connected layer and output layer.

The most important structure of convolutional neural network is the convolutional layer, which is also known as the feature extraction layer [16]. The input to the convolutional layer comes from the previous layer, which is input and pooling layer. The principle of convolution layer is to conduct convolution operation between the input of the previous layer and the convolution core of the current layer. Finally, with the corresponding bias, the output is obtained by using the activation function. The formula is as follows.

$$x_j^l = f\left(\sum_{i \in p_j} x_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{6}$$

In the above equation, x_i^{l-1} is the characteristic value of the ith window of the l-1th layer, $*$ represents the convolution operation. b_j^l represents the bias value on layer l, k_{ij}^l represents the bias on layer l, and $f(\cdot)$ is the activation function.

The pooling layer is usually located in the middle of two convolutional layers [17]. The pooling layer reduces the number of network model parameters by gradually reducing the spatial size of the data, making the model training process require fewer resources. The pooling layer ensures that the model parameters are reduced and the complexity of the network structure is reduced without losing any important information. This improves the generalization ability of the network and also reduces the risk of overfitting. The pooling process is shown in the following equation.

$$h_j^l = subsampling(x_j^{l-1}) + b_j^l \tag{7}$$

In the above equation, h_j^l represents the net activation of channel j of pooling layer l. This value is obtained by downsampling the output feature map from the previous layer and adding a bias, $subsampling(*)$ denotes the pooling function.

In the fully connected layer, each neuron is connected to all the outputs of the previous layer. The final classification role is usually achieved at the end of the CNN. After the input data is convolved and pooled, the output feature vectors go through a fully connected layer and are classified by softmax function to output the prediction results. The output calculation formula of the full connected layer is as follows.

$$x_l = f(h_l) \tag{8}$$

$$h_l = \omega_l x_{l-1} + b_l \tag{9}$$

In the above equation, $f(\cdot)$ is the activation function of fully connected layer and h_l denotes the net activation of the fully connected layer. x_{l-1} represents the output feature map of the previous layer, b_l represents the bias of the fully connected layer, and ω_l is the weight of the layer.

C. LONG SHORT-TERM MEMORY NETWORK

Long short-term memory network is one of the recurrent neural networks (RNN). RNN has always had a latency problem, and LSTM was designed for this problem [18]. By analyzing the structure of the LSTM principle, we can see that the LSTM has three more controllers than the conventional RNN, which are the forgetting gate, the input gate and the output gate [19].

(1) The forgetting gate

The forgetting gate determines what information is discarded. The tth neural unit that is currently in the sequence

gets the output f_t based on the previous implied state h and the current input x. The formula is as follows.

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) \tag{10}$$

In the above equation, f_t represents the output of the forgetting gate, σ represents the sigmoid activation function, and w_f represents the weight of the forgetting gate. h_{t-1} represents the implied state of the previous cell, x_t represents the current input, and b_f represents the deviation value of the forgotten gate.

(2)The input gate

The input gate determines which new information is allowed to be added to the cell. The operation requires two steps: First, the sigmoid layer of the input gate layer determines the information to be updated, and tanh generates the update content vector $\tilde{C}_t$; Then the two parts are combined to perform one update of the cell's state. The formula is shown below.

$$\begin{cases} i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \\ \tilde{C}_t = tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right) \\ C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \end{cases} \tag{11}$$

In the above equation, W_i represents the weight matrix of the input gate, b_i represents the bias term of the input gate, W_c represents the weight matrix of the cell state, and b_c represents the bias term of the cell state gate.

(3) The output gate

The role of the output gate is to decide which values are output by the current neural unit. The implied state h_t of the cell is calculated based on the output O_t of the output gate. The formula for the output gate is shown below.

$$\begin{cases} O_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \\ h_t = O_t * tanh\left(C_t\right) \end{cases} \tag{12}$$

In the formula, O_t represents the output of the output gate, W_o represents the weight of the output gate, b_o represents the deviation value of the output gate, and h_t represents the implied state of the current neural unit.

D. ATTENTION MECHANISM

Attention mechanism is the attention model simulating the human brain. At a certain point in time, people's attention will always focus on a certain focus of the object they see, while ignoring other parts [20].

The main purpose of attention mechanism is to select more important information for current task objectives from numerous information and give more attention. Less attention is given to other components to achieve a segregated allocation of resources and thus reduce the impact of non-critical factors. It improves the classification performance of the model by learning the importance of different elements and merging them according to their importance [21]. The attention mechanism in the neural network can obtain the attention probability distribution by weighting and coding the input data, and finally obtain the specific output.

The key to implementing the attention mechanism is to calculate the weighting of the raw data and find the focused target data based on the weights given. The calculation of weights in the attention mechanism is a dynamic updating process and is not obtained by predetermination. The computational process of the attention mechanism is similar to that of an autoencoder in a neural network and consists of two processes: encoding and decoding.

The calculation of attention mechanism is divided into three main steps: The first step is to calculate the similarity between q value and k value; The second step is to normalize the calculated similarity as the weight coefficient of each value; The third step is a weighted summation of the value values.

To calculate the Attention value, we first need to calculate the attention distribution. Let i be the location where the input information is selected, the key-value pair (K,V)=[(k_1,v_1),(k_2,v_2),….(k_N,v_N)] represents the input N information, and Q represents an element in the input target. Attention mechanism will combine context semantics and tags to calculate a group of attention scores s_i. The size of s_i is directly related to the noticeable degree of the word in the text [22]. The higher of this value indicates the stronger the attention the word receives in the context. The attention score is calculated by the following formula, where F is the attention scoring function.

$$s_i = F(Q, k_i) \tag{13}$$

After calculating the attention score s_i, the weight of attention α_i is calculated by the following formula. The probability vector consisting of α_i is called the attention distribution.

$$\alpha_i = soft\ \max(s_i) = \frac{\exp(s_i)}{\sum_{j=1}^{N} \exp(s_j)} \tag{14}$$

The input information is summarized by weighted average. Using the computed attention distribution α_i, the Attention value is obtained by weighting the sum of V with the following formula.

$$Attention((K, V), Q) = \sum_{i=1}^{N} \alpha_i \upsilon_i \tag{15}$$

Attention mechanism can select the input of neural network through structured feature representation, which can reduce the dimension of high-dimensional input data and computational complexity. Adding attention mechanism to a neural network can help the network find useful information related to the input information in the current data output. Through the weight expression of each element in the neural network, the data feature information with high weight value can be learned, so the redundancy can be reduced to improve the effectiveness of network output data, and higher quality features can be extracted.

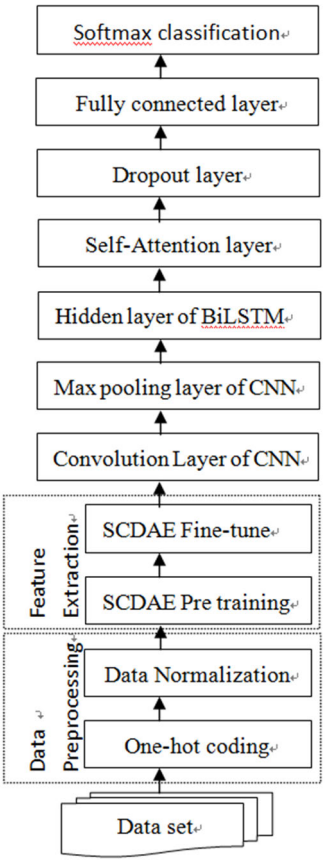

FIGURE 1. The hierarchical intrusion detection model combining multiple deep learning models with attention mechanism.

III. CONSTRUCTION OF A HIERARCHICAL INTRUSION DETECTION MODEL BASED ON MULTIPLE DEEP LEARNING MODELS

In this paper, we first perform data preprocessing on the original network traffic data, and then input the preprocessed data into a hierarchical intrusion detection model [23]. The SCDAE model is used for noise reduction and feature extraction. Spatial features are extracted based on CNN model, and then temporal features are mined using BiLSTM model. After the hierarchical network based on CNN-BiLSTM has extracted the features of network traffic data, the self attention mechanism is introduced to automatically calculate the weight of each feature. The calculated results are input to the classification module of the full connected layer. Finally, the Softmax classifier is used to obtain the classification probability of each stream. The index with the highest probability is the classification result based on SCDAE-CNN-BiLSTM-Attention model on the data stream. The intrusion detection model proposed in this paper is shown below.

A. FEATURE EXTRACTION AND NOISE REDUCTION BASED ON SCDAE MODEL

Denoising Autoencoder (DAE) is one of the variants of autoencoder, which is an autoencoder that improves the robustness of encoding by adding noise [24]. It has the ability to scan the data byte by byte to find encoded features. The most important feature of DAE is the ability to encode and decode the polluted or destroyed raw data, and then restore the real original data to some certain extent. Therefore, compared with the ordinary autoencoders, DAE has good noise reduction ability.

The input to the DAE is the output of the previous layer of noise reduction self-encoder, which results in a high-level feature representation of the input data. DAE can be used to compress the high-dimensional traffic data to obtain new feature samples instead of the original data during intrusion model detection. Therefore, DAE can effectively compress and reduce feature dimensions while preserving the original data. DAE can reduce the learning and calculation of models and improve the speed of intrusion detection.

As the structure of the autoencoder, the deep neural network can help the autoencoder extract more abstract features of the original data. We often use a layer-by-layer stacking approach to train deep autoencoders [5]. Stacked autoencoders generally use Layer-Wise Training to learn network parameters.

SCDAE is Stacked Convolutional Denoising Autoencoders. In this paper, the SCDAE model is used to scan the data byte by byte to extract the features of the traffic data and reduce the noise. The purpose of this model is to reduce the impact of loss or damage of the original information of traffic data on traffic classification detection results.

For a vector x, we first get a corrupted vector $\hat{x}$ by randomly setting the values of some dimensions of x to 0 according to a scale U. The corrupted vector $\hat{x}$ is then input to the autoencoder to obtain the encoding z. The original lossless input x is reconstructed using code z.

Its training process can be divided into encoding and decoding. The encoding process extracts features from the original input data. If the input vector of the SCDAE model is $x = (x_1, x_2, \ldots x_n), x_i \in [0, 1]$, the value $h = (h_1, h_2, \ldots h_m)$ of the hidden layer can be obtained after the following formula processing.

$$h = f(x) = s(W_1 x + p) \tag{16}$$

The decoding process is to reconstruct the input data based on the learned features to obtain the output $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots \tilde{x}_n)$. This is shown in the following equation.

$$\tilde{x} = g(h) = s(W_2 h + q) \tag{17}$$

In these formulas, f, s and g represent the encoding function, activation function and decoding function, respectively. W_1 represents the weight matrix from the input layer to the hidden layer and W_2 represents the weight matrix from the hidden layer to the output layer. Both p and q represent bias vectors.

Assuming there are N training samples, the average activation value of the jth neuron in the hidden layer is shown in

the following formula.

$$\hat{\rho}_j = \frac{1}{N}\sum_{n=1}^{N} z_j^{(n)} \tag{18}$$

In the above equation, $\hat{\rho}_j$ represents the activation probability of the jth neuron in the hidden layer. Assuming a value ρ^*, $\hat{\rho}_j$ is expected to approximate ρ^*. The difference between $\hat{\rho}_j$ and ρ^* can be measured by the KL scatter, as is shown in the following equation.

$$KL(\rho^*||\hat{\rho}_j) = \rho^* \log\frac{\rho^*}{\hat{\rho}_j} + (1-\rho^*)\log\frac{1-\rho^*}{1-\hat{\rho}_j} \tag{19}$$

The original 1800-dimensional data obtained after data preprocessing are input into the SCDAE model for feature extraction and noise reduction. To demonstrate the powerful denoising ability of the SCDAE model, noise can also be added at the same time.

Because the SCDAE model is a three-layer DAE model stacked in a convolution mode, it has very strong anti-noise ability, and the model itself is more suitable for deep network. Therefore, this paper uses the SCDAE model to extract the features of the traffic data and perform noise reduction.

B. CONSTRUCTION OF THE CNN-BILSTM-ATTENTION MODEL

1) EXTRACTING SPATIAL FEATURES OF TRAFFIC DATA BASED ON CNN

The 1800-dimensional features obtained in the previous stage are mapped into a grayscale image of size $60 * 60$, which is then fed into the CNN network. In order for CNN to better identify network traffic data, a CNN is formed by combining multiple size convolutional kernels [25]. It improves the efficiency of extracting features from the data in the dataset as well as ensuring the accuracy of feature recognition. The following four sizes of convolution kernels are used: 5∗5, 8∗8, 1∗4 and 1∗2. The convolution is defined as shown in the following equation. The definition formula of convolution is shown below.

$$s(i,j) = (X * W)(i,j) = \sum\nolimits_m \sum\nolimits_n x(i+m, j+n)w(m,n) \tag{20}$$

In the above formula, W is the convolution kernel and X is the input. If X is the input matrix, then W is the corresponding convolution kernel matrix.

In CNN, the ReLU activation function is used. Its formula is shown below.

$$ReLU(x) = \max\{0, x\} = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \tag{21}$$

In the above equation, it is a linear function in the interval of $x \geq 0$. This formula not only overcomes the problem of gradient disappearance, but also accelerates the convergence speed of the model. In the interval of N, it will make the output of some neurons to 0, which increases the sparsity of the network and can make each neuron to maximize its screening effect.

Finally, it is sent to the output layer of the CNN. The output layer of the CNN uses a fully-connected layer, which consists of 1800 neurons. The purpose is to maintain the same dimensionality as the original traffic data after extracting the spatial features.

Through the training process, it is known that the network neurons will be deleted randomly during one training process. The number of neurons removed at each level can be set, and the deleted neurons are due to dropout. These layers where the neural units are removed are called Dropout layers. The definition of the neural network after applying the Dropout method is as follows.

$$\begin{cases} m^n = Bernoulli(s) \\ \tilde{x}^n = m^n * x^n \\ a_i^{n+1} = w_i^{n+1}\tilde{x}^n + b_i^{n+1} \\ x_i^{n+1} = f(a_i^{n+1}) \end{cases} \tag{22}$$

In the above equation, s represents the retention probability of a neuron and m^n represents a random vector satisfying the Bernoulli distribution. x^n represents the output value vector of the neuron in the nth layer of the neural network, and $\tilde{x}^n$ represents the output value vector of the neuron in the nth layer of the neural network after random blocking. b_i^{n+1} is the bias of the ith neuron at layer $n+1$, and a_i^{n+1} represents the input value of the activation function of the ith neuron at layer $n+1$.

2) MINING TEMPORAL FEATURES USING BILSTM

Intrusion detection encounters problems that are often time-series in nature, such as Advanced Persistent Threat. Attackers consciously collect important data assets on servers for compression, encryption, and packaging during internal horizontal penetration and long-term latency. The data is then sent back to the attacker through a hidden data channel. Due to the existence of such long-term latent attack types, a bidirectional LSTM can better capture the information in the before and after sequences compared to a unidirectional LSTM, and thus an intrusion detection model can be better constructed using a bidirectional LSTM.

To detect attacks more effectively and accurately, the model needs to detect not only previously trained information, but also information trained later. Therefore, a BiLSTM network is used in this paper to capture long-range dependent features. The model consists of LSTM modules connected in two directions and is capable of multiple shared weights in the front-to-back network.

In this paper, two-stage BiLSTM is used. The first phase learns the characteristics of the data and the correlation between the data. The second phase performs deeper learning of multiple features of the data.

At each time step, the output of the BiLSTM module is determined by a forgot gate, an input gate, an output gate, and an updated cell state. Each gate is determined by the output of

the previous module and the input of the current moment. The three gates collaborate on the selection of information about the network structure properties, the forgetting work and the updating of the cell state.

The BiLSTM network structure has four layers: the input layer, the forward LSTM layer, the backward LSTM layer and the output layer [26]. The input layer is responsible for encoding the input data to meet the input requirements of the network. The forward LSTM layer is responsible for extracting the sequence information transmitted forward from the input sequence. The backward LSTM layer is responsible for extracting the sequence information transmitted backward from the input sequence. The output layer integrates the output information from the forward LSTM and the backward LSTM. Given the input sequence, the forward transmission and the backward transmission process of BiLSTM are calculated as shown in the following equations.

$$\begin{cases} \vec{i}_t = \sigma\left(\vec{W}_i \cdot \left[\vec{h}_{t-1}, \vec{x}_t\right] + \vec{b}_i\right) \\ \vec{f}_t = \sigma\left(\vec{W}_f \left[\vec{h}_{t-1}, \vec{x}_t\right] + \vec{b}_f\right) \\ \vec{C}_t = \vec{f}_t * \vec{C}_{t-1} + \vec{i}_t * \tilde{C}_t \\ \vec{O}_t = \sigma\left(\vec{W}_o \left[\vec{h}_{t-1}, \vec{x}_t\right] + \vec{b}_o\right) \\ \vec{h}_t = \vec{O}_t * tanh\left(\vec{C}_t\right) \end{cases} \tag{23}$$

$$\begin{cases} \overleftarrow{i}_t = \sigma\left(\overleftarrow{W}_i \cdot \left[\overleftarrow{h}_{t-1}, \overleftarrow{x}_t\right] + \overleftarrow{b}_i\right) \\ \overleftarrow{f}_t = \sigma\left(\overleftarrow{W}_f \left[\overleftarrow{h}_{t-1}, \overleftarrow{x}_t\right] + \overleftarrow{b}_f\right) \\ \overleftarrow{C}_t = \overleftarrow{f}_t * \overleftarrow{C}_{t-1} + \overleftarrow{i}_t * \tilde{C}_t \\ \overleftarrow{O}_t = \sigma\left(\overleftarrow{W}_o \left[\overleftarrow{h}_{t-1}, \overleftarrow{x}_t\right] + \overleftarrow{b}_o\right) \\ \overleftarrow{h}_t = \overleftarrow{O}_t * \tanh\left(\overleftarrow{C}_t\right) \end{cases} \tag{24}$$

In the above equation, W is the weight matrix and b is the bias vector. $\vec{h}_t$ and $\overleftarrow{h}_t$ represent the LSTM outputs in two directions at time t, respectively, and both are connected to the same output. The output vector h_t of BiLSTM can be expressed by the following equation.

$$h_t = \vec{h}_t \oplus \vec{h}_t \tag{25}$$

In the above equation, $\oplus$ represents the combination of BiLSTM output in five ways: sum, mul, ave, concat, and none.

In the BiLSTM network, the input gate, the forgetting gate and the output gate in the LSTM network structure are all activated by the sigmoid function, which is shown in the following formula.

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{26}$$

When generating the candidate value vector in the hidden layer of the LSTM network, the tanh activation function is used for the nonlinear mapping transformation, which is represented in the following equation.

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{27}$$

The Sigmoid function can set the output value between 0 and 1 and implement a nonlinear transformation. The output value between 0 and 1 enables the three gating units to achieve an open or closed state, thus realizing their control functions. The tanh activation function enhances the capacity of the nonlinear model by making only one linear transformation and nonlinear mapping of the data. The results obtained by mapping with the tanh function are distributed in the interval of [-1,1]. The advantage is that the function is perfectly symmetric about the origin, and the gradient of the function at the origin is maximum, which can make the model converge faster.

3) INTRODUCTION OF SELF-ATTENTION MECHANISM

The self-attention mechanism is a variant of the attention mechanism, in which the self-attention mechanism function should be defined. This method first initializes the parameters, then creates a trainable weight through the build function, and writes the functional logic of the layer through call(x). The dot product between Q and K is calculated by entering Q, K, V. To prevent its result from being too large, this result is then divided by the dimensionality of the query and key vectors as the initial scale [27]. The distribution of probabilities is obtained using Softmax, then the matrix V is multiplied to obtain a representation of the summation of weights, and finally the function is used to define the shape change logic.

After the BiLSTM, the usual classification task uses the output vector of the last time step or uses the output vector of all time steps. However, not all packets in a session have the same level of importance. In order to make the final temporal features pay more attention to the important content, Self-Attention will be introduced in this paper to further process the output of the upper layer BiLSTM.

In model training, there needs to be a focus on the input features. Model training allows the model to save more time to focus on its input features that need attention. The Self-Attention Layer is added to the model to meet the requirements needed for the model. The formula of attention is shown below.

$$c_j = \sum_{i=1}^{T} \alpha_{ij} h_i \tag{28}$$

In the above equation, α_{ij} represents the learning attention weight and h_i represents the candidate state. The main role of learning attention weights is to be able to automatically capture the association of h_i and c_j. The vector C can be solved by α_{ij} and used as input into the decoder. The vector c_j represents the weighted sum of all query states and attentions of the encoder at each position j of the decoder. The formula for the main expression of the attention layer is as follows.

$$\begin{cases} attention = Soft\max(Dense(Dense(x, y_{t-1}))) \\ context = \sum_{i=1}^{m} (attention_i * x_i) \end{cases} \tag{29}$$

Building the attention layer requires updating the data first. The updated data is passed through the model to obtain

TABLE 1. Description of NSL-KDD dataset.

Label Type	KDDTrain+	Train Set Scale(%)	KDDTest+	Test Set Scale(%)
Normal	67343	53.46	9711	43.08
Dos	45927	36.46	7458	33.08
Probe	11656	9.25	2421	10.74
R2L	995	0.79	2753	12.21
U2R	52	0.04	200	0.89

each output context. The first time we use the initialization parameter role input x. Then each output obtained by the model. In order to ensure that attention is given to the input with preference for each prediction, the output is obtained. Finally, its attention layer is defined into the model.

C. CLASSIFICATION MODULE

The classification module consists of a fully-connected layer with Softmax functions to implement multiple classification tasks [28]. The Softmax function converts the output values of each cell into a probability distribution ranging from 0 to 1. It is a linear classifier with the following equation.

$$\hat{y} = \frac{\exp(Out^j)}{\sum \exp(Out^i)} \tag{30}$$

In the above equation, Out^i represents the output of the ith neuron in the fully connected layer, and Out^j represents the output of the jth neuron in the fully connected layer. $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_N\}$ is the complete set of classes, where N represents the total number of classes, and the output with the highest probability indicates the category of input values.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. DATA SET FOR THE EXPERIMENT

NSL-KDD dataset is a classic dataset that has been used up to date in the field of anomaly detection [29]. It is an improved version of KD99 dataset and contains part of the original dataset. The NSL-KDD dataset solves some of the inherent problems of KDD99 and also retains the structure of the original dataset. There are roughly 4900000 network connection records in the dataset, and the raw data contains 3925650 attacks and 972781 normal traffic data. Each network connection is viewed as a vector, and each vector contains 41 features and 1 classification identifier.

The training set KDDTrain+ of the NSL-KDD dataset has 125,973 network connection records. The test set KDDTest+ has 22,543 network connection records. The description of the NSL-KDD dataset is shown in Table 1.

The CIC-IDS2018 dataset was published by the Canadian Institute for Cybersecurity Research (CIC) in 2018 [30]. The dataset contains six types of attacks: brute force cracking, botnets, Dos, DDos, Web attacks and network infiltration. The CIC-IDS2018 dataset contains 3,227,424 flows, with 267,839 normal flows and 5,493,385 attack flows. In order to improve the training efficiency, the attack traffic and the collected normal traffic in the CIC-IDS2018 dataset are filtered and combined in this paper, and the final distribution of the data obtained is shown in Table 2.

TABLE 2. Description distribution of CIC-IDS 2018 dataset.

Classification	Number of train sets	Number of test sets
Normal	68563	42300
Benign	46300	15686
Bot	15623	6325
Brute Force	14680	3587
DDOS	36890	10238
DOS	55636	23657
PortScan	12566	8566
Infiltration	8966	3255
Web Attack	17852	9633

CIC-IDS2017 dataset is a network intrusion detection dataset designed, collected and processed by the Canadian Institute for Cybersecurity Research in 2017 [31]. The dataset contains 7 types of attacks. The dataset collects a total of 2830743 network traffic data, including 2273097 normal network traffic data and 557,646 other attack types of network traffic data.

The Canadian Institute for Cybersecurity Research provided the CIC-DDoS2019 public dataset [32]. This dataset contains both normal traffic and PCAP files for the latest common DDoS attacks. The content of this dataset is tested by the CIC agency using different DDos attack methods in different time periods within two days.

B. DATA PREPROCESSING

Both the NSL-KDD and CIC-IDS 2018 datasets contain two types of features: numeric types and character types. Since the model proposed in this paper cannot handle character type data, it is necessary to convert the character type features into numeric features that can be accepted by the model. The process of preprocessing is to numeric the character type features and then normalize the data.

(1) Numerical processing of character type features

The NSL-KDD dataset has 2 distinct features, it consists of 3 character features and 38 numeric features. There are also a large number of character type features in the CIC-IDS 2018 dataset. To carry out the comparison experiments, One-Hot coding is used to encode the two data sets and establish a one-to-one mapping between the symbol vectors and the corresponding numerical features. The features encoded by One-Hot not only handle the features with non-continuous values, but also make the distance between features more reasonable.

(2) Normalization process

After numerical processing, the features values of network traffic data are different greatly. Without normalization, the magnitude of the gradient keeps decreasing as backpropagation proceeds [33]. The speed of learning weights in intrusion detection models is slow, and the complex features of network traffic data cannot be extracted well, nor can deep learning be achieved. Thus, the training effect of intrusion detection model is affected. Therefore, network traffic data must be

normalized in the data preprocessing stage. This paper uses the Min-Max standardized processing method to compress data between [0,1], and the formula is as follows.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{31}$$

In the above equation, x' is the normalized data and x is the current data. $x_{\min}$ is the minimum data value in the current feature attribute, and $x_{\max}$ is the maximum data value in the current feature attribute.

C. EVALUATION INDICATORS

In order to effectively evaluate the performance of the intrusion detection model, the experimental results are evaluated by confusion matrix [34]. The confusion matrix can clearly describe the predicted true-false and actual true-false situations. Based on the confusion matrix, this paper uses Accuracy, Precision, Recall, F1-Score and False Positive Rate (FPR) as the evaluation indicators for the performance of the detection model. The calculation formula of each indicator is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{32}$$

$$Precision = \frac{TP}{TP + FP} \tag{33}$$

$$Recall = \frac{TP}{TP + FN} \tag{34}$$

$$FPR = \frac{FP}{TN + FP} \tag{35}$$

$$F1 - Score = \frac{2 \times \Pr ecision \times Recall}{\Pr ecision + Recall} \tag{36}$$

In the above equation, TP stands for True Positive, where the prediction is true and the true value is also true; TN stands for True Negative (TN), where the prediction is false and the true value is also false; FP stands for False Positive, where the prediction is true and the true value is false. FN stands for False Negative, where the prediction is false and the true value is true. If the Accuracy and F1-Score of a classifier are higher, and the False Positive Rate (FPR) is lower, the classification effect of the classifier is better [35].

D. ANALYSIS OF EXPERIMENTAL RESULTS

1) ANALYSIS OF EXPERIMENTAL RESULTS BASED ON THE NSL-KDD DATASET

This experiment compares the classical classification models commonly used in intrusion detection with the model proposed in this paper. The comparison models include KNN [36], RF [37], CNN, BiLSTM and CNN-BiLSTM, and each method is regarded as a classifier. In this experiment, the training set KDD Train+ is used to train the CNN-BiLSTM-Attention model and other five intrusion detection models. The trained models are compared on the test set KDD Test+ for accuracy testing and the results are shown in Table 3 and Figure 2. The experimental results show that the accuracy of the proposed classification model is 93.26% and the recall rate is 94.26% compared with other classifiers. The experimental results in Figure 2 show that the four evaluation indicators of the proposed model are higher than those of other classification models.

TABLE 3. Classification performance comparison by different models on KDD test+.

Different Models	AC(%)	Precision(%)	Recall(%)	F1- Score (%)
KNN	79.54	78.29	79.29	78.39
RF	78.36	89.29	83.19	83.24
CNN	85.57	86.39	89.26	90.13
BiLSTM	82.58	85.27	87.37	87.59
CNN-BiLSTM	89.36	91.37	92.39	93.37
CNN-BiLSTM-Attention	93.26	95.17	94.26	96.28

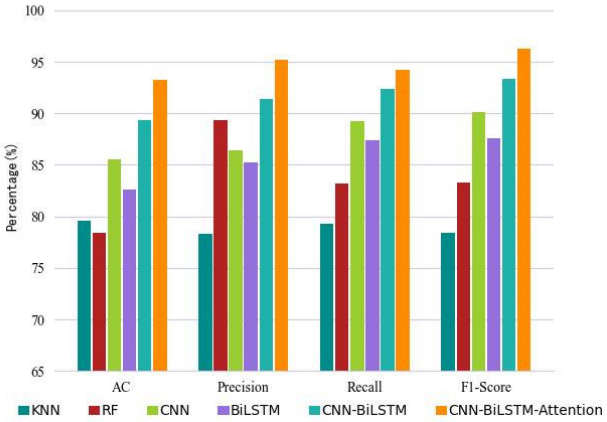

FIGURE 2. Classification results comparison on KDD test+.

TABLE 4. F1-Score for each class of different models on KDD test+.

F1-Score of Models	Normal(%)	DOS(%)	Probe(%)	R2L(%)	U2R(%)
KNN	80.49	81.29	65.46	12.46	16.26
RF	85.69	83.75	68.58	25.89	34.89
CNN	91.38	90.41	76.27	57.48	19.75
BiLSTM	88.59	87.93	67.24	59.29	23.28
CNN-BiLSTM	95.37	93.28	75.38	62.74	34.29
CNN-BiLSTM-Attention	98.26	95.78	80.34	68.97	46.28

F1-Score is the balance between precision and recall. It can be considered as the summed average of precision and recall. The results of the F1-Score test comparison experiments on the test set KDD Test+ are shown in Table 4 and Figure 3. The results show that F1-Score reaches 98.26% for categories labeled Normal. From Figure 3, it is more intuitive to see that for U2R classes with very little data, the classification effect is also significantly improved compared with other methods.

The false positive rate is also tested and compared in this paper, as shown in Table 5 and Figure 4. The experimental results show that the model proposed in this paper has the

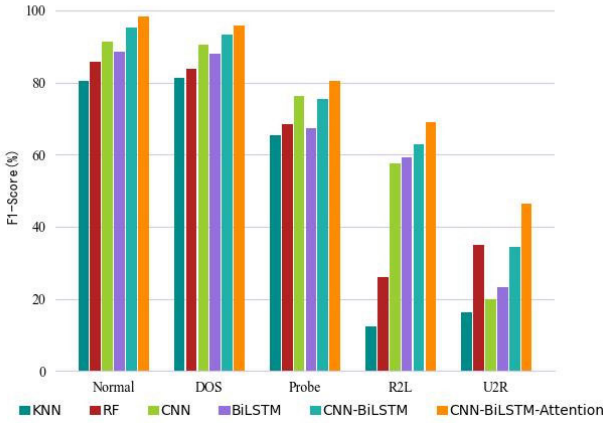

FIGURE 3. Comparison of F1-Score for each class on KDD test+.

TABLE 5. FPR for each class of different models on KDD test+.

FPR of Models	Normal(%)	DOS(%)	Probe(%)	R2L (%)	U2R (%)
KNN	15.38	16.75	11.98	54.37	64.72
RF	11.27	12.58	16.75	53.16	66.57
CNN	8.14	8.46	8.28	62.87	57.31
BiLSTM	9.26	8.73	8.86	51.25	68.28
CNN-BiLSTM	6.15	7.11	6.38	42.59	50.28
CNN-BiLSTM-Attention	2.57	2.83	3.18	35.96	38.36

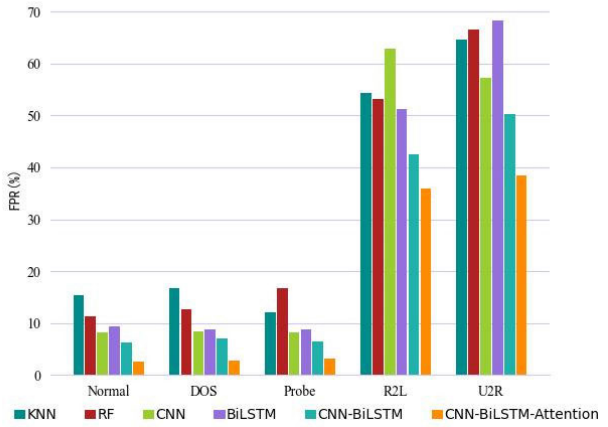

FIGURE 4. Comparison of FPR for each class on KDD test+.

lowest false positive rate compared with other models on Normal, DOS, Probe, R2L, U2R labels.

The CNN-BiLSTM-Attention model has higher accuracy, higher F1-Score and lower false positive rate for all kinds of attacks detection. Therefore, the classification performance of the CNN-BiLSTM-Attention model proposed in this paper is superior to the other five intrusion detection models (KNN, RF, CNN, BiLSTM and CNN-BiLSTM).

2) ANALYSIS OF EXPERIMENTAL RESULTS BASED ON CIC-IDS 2018 DATASET

In order to further verify the method proposed in this paper, we also carried out experiments on CIC-IDS2018 dataset, and the results are shown in Table 6. The classification accuracy

TABLE 6. Classification performance comparison by different models on CIC-IDS2018.

Different Models	AC(%)	Precision(%)	Recall(%)	F1- Score (%)
KNN	82.86	80.53	74.22	77.29
RF	83.57	84.58	83.28	86.59
CNN	84.86	86.84	78.29	81.44
BiLSTM	78.18	88.24	76.18	83.27
CNN-BiLSTM	85.31	85.37	82.46	87.28
CNN-BiLSTM-Attention	88.27	91.54	89.13	90.18

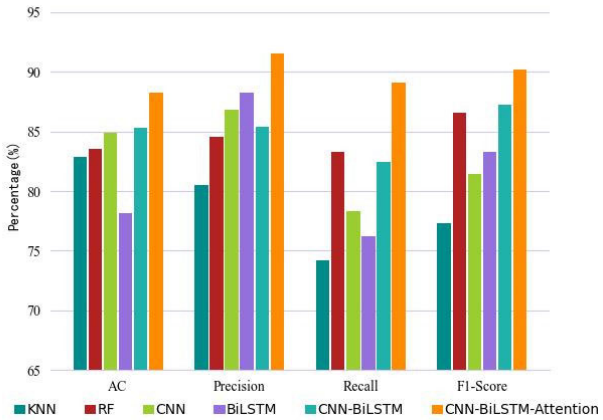

FIGURE 5. Classification results comparison on CIC-IDS2018.

of the proposed method is 88.27%, which is 5.41%, 4.7%, 3.41%, 10.09% and 2.96% higher than KNN, RF, CNN, BiLSTM and CNN-BilstM, respectively. It is also clear from Figure 5 that the model proposed in this paper is also higher than the other models in terms of accuracy, recall and F1-Score. From the classification results, the model proposed in this paper is effective and still obtains better classification results than other methods when faced with more classes of attack data.

The detailed results for each category of all classification models are shown in Table 7. As can be seen from Table 7, although the results of our novel model on Bot, Brute Force and PortScan are basically the same as those of other models. However, the classification effect on Normal, Benign, DDOS, DOS, Infiltration, and Web Attack is improved more obviously, and the F1-Score can reach 86.37%, 80.57%, 65.28%, 93.85%, 42.86%, and 76.28%, respectively. Compared with CNN-BiLSTM, the model in this paper improves 0.86%, 7.15%, 4.76%, 4.58%, 27.52%, and 15.76% on these categories, respectively. It can be seen from the bar figure 6 that CNN-BiLSTM-Attention has higher classification results than other models for almost all categories.

This paper also conducted a comparative experiment on the CIC-IDS2018 dataset for false positive rate testing, as shown in Table 8 and Figure 7. The experimental results show that the proposed model has the lowest false positives on all labels except Bot and Brute Force when compared with other models.

TABLE 7. F1-Score for each class of different models on CIC-IDS2018.

F1-Score of Models	KNN (%)	RF (%)	CNN (%)	BiLSTM (%)	CNN-BiLSTM (%)	CNN-BiLSTM-Attention (%)
Normal	78.23	84.21	80.34	82.51	85.51	86.37
Benign	67.25	68.23	70.12	74.23	73.42	80.57
Bot	8.27	10.42	16.27	14.28	20.58	18.25
Brute Force	14.21	30.26	35.28	28.57	33.28	34.59
DDOS	25.89	55.29	49.18	48.27	60.52	65.28
DOS	87.23	90.24	87.21	88.15	89.27	93.85
PortScan	18.29	40.27	35.28	32.51	37.25	38.97
Infiltration	39.26	35.27	30.24	36.21	15.34	42.86
Web Attack	24.29	50.23	70.25	65.21	60.52	76.28

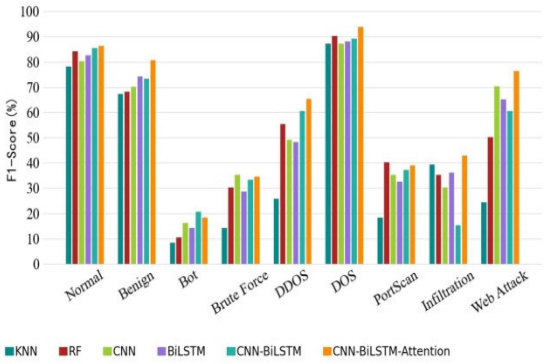

FIGURE 6. Comparison of F1-Score for each class on CIC-IDS2018.

TABLE 8. FPR for each class of different models on CIC-IDS2018.

FPR of Models	KNN (%)	RF (%)	CNN (%)	BiLSTM (%)	CNN-BiLSTM (%)	CNN-BiLSTM-Attention (%)
Normal	28.37	32.58	11.27	26.37	16.37	8.26
Benign	29.32	19.24	23.34	22.31	12.36	7.39
Bot	47.39	42.31	53.26	46.23	36.56	38.68
Brute Force	56.39	54.28	51.36	60.29	43.23	46.56
DDOS	31.48	30.26	22.38	25.34	12.36	6.38
DOS	33.38	21.38	25.33	20.36	14.29	7.18
PortScan	52.88	53.69	46.29	43.26	39.27	37.46
Infiltration	28.36	32.56	23.85	25.31	11.39	9.22
Web Attack	25.34	21.39	15.32	19.36	12.39	6.83

It can be seen from the experimental results that the false positive rate of the proposed CNN-BiLSTM-Attention model is the lowest compared with other five classical models. Therefore, based on the above experimental results, it shows that the model has higher accuracy, higher F1 score and lower false positive rate for all types of attack detection in the CIC-IDS2018 dataset.

3) ANALYSIS OF EXPERIMENTAL RESULTS BASED ON CIC-IDS 2017 DATASET

In this paper, comparative experiments are carried out on the CIC-IDS2017 dataset, and the experimental results are shown

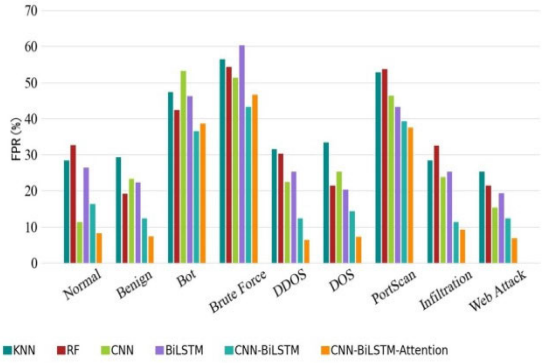

FIGURE 7. Comparison of FPR for each class on CIC-IDS2018.

TABLE 9. Classification performance comparison by different models on CIC-IDS2017.

Different Models	AC(%)	Precision(%)	Recall(%)	F1- Score (%)
KNN	79.43	82.34	83.92	80.19
RF	85.24	80.47	86.78	88.29
CNN	86.73	88.17	79.69	84.34
BiLSTM	83.75	91.33	82.18	87.33
CNN-BiLSTM	87.26	87.57	85.47	89.21
CNN-BiLSTM-Attention	90.31	93.28	88.24	91.33

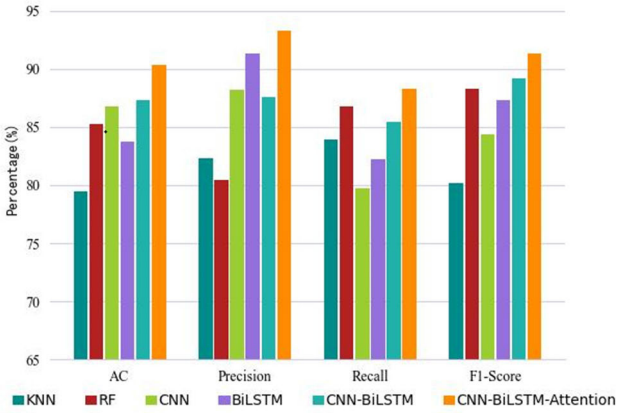

FIGURE 8. Classification results comparison on CIC-IDS2017.

in Table 9 and Figure 8. The classification precision of the proposed model is 93.28%, which is 5.71 %, 1.95%, 5.11%, 12.81% and 10.94% higher than CNN-bilSTM, BiLSTM, CNN, RF and KNN, respectively. In addition, the classification accuracy of this paper is 90.31%, which is 3.05%, 6.56%, 3.58%, 5.07% and 10.88% higher than CNN-bilSTM, BiLSTM, CNN, RF and KNN, respectively. It can also be clearly seen from Figure 8 that the proposed model is also higher than other models in terms of recall and F1-Score.

As can be seen from Table 10, although the results of the proposed method on Web Attack and Botnet are basically the same as those of other methods, the effect on Normal, Brute Force, Heartbleed, Infiltration, DDOS and DOS is

TABLE 10. F1-Score for each class of different models on CIC-IDS2017.

F1-Score of Models	KNN (%)	RF (%)	CNN (%)	BiLSTM (%)	CNN-BiLSTM (%)	CNN-BiLSTM-Attention (%)
Normal	80.12	85.37	83.33	81.36	86.34	91.52
Brute Force	77.29	75.28	80.29	82.22	81.37	85.41
Web Attack	75.26	77.69	65.21	70.25	76.28	78.26
Heartbleed	40.22	50.23	38.21	44.23	48.12	55.69
Infiltration	62.36	77.26	80.29	83.24	76.25	88.23
Botnet	10.26	24.79	11.89	23.28	25.14	25.39
DDOS	16.38	28.11	28.51	19.26	26.31	35.89
DOS	66.98	68.12	76.33	74.23	72.15	80.57

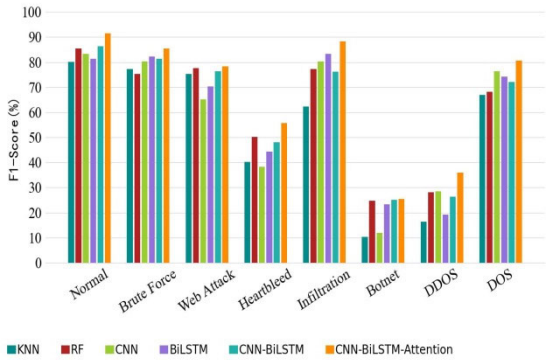

FIGURE 9. Comparison of F1-Score for each class on CIC-IDS2017.

TABLE 11. FPR for each class of different models on CIC-IDS2017.

FPR of Models	KNN (%)	RF (%)	CNN (%)	BiLSTM (%)	CNN-BiLSTM (%)	CNN-BiLSTM-Attention (%)
Normal	27.59	36.58	23.98	19.36	22.56	13.69
Brute Force	52.36	39.67	50.22	48.36	45.39	36.89
Web Attack	13.58	15.69	34.57	22.98	11.58	8.23
Heartbleed	41.23	42.34	28.37	48.04	36.14	25.36
Infiltration	59.36	43.89	62.87	56.29	50.41	44.12
Botnet	35.69	47.06	28.39	23.56	42.59	18.87
DDOS	24.38	11.39	20.56	15.69	8.26	6.07
DOS	19.36	13.54	30.51	24.33	16.32	10.26

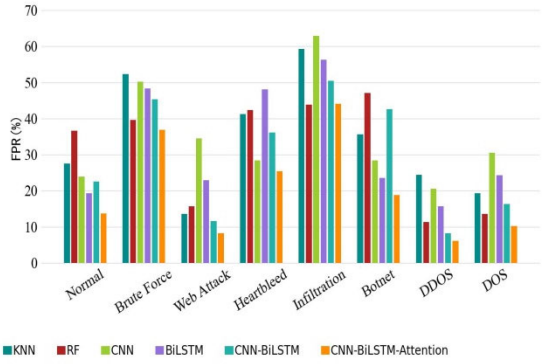

FIGURE 10. Comparison of FPR for each class on CIC-IDS2017.

TABLE 12. Classification performance comparison by different models on CIC-DDoS2019.

Different Models	AC(%)	Precision(%)	Recall(%)	F1-Score (%)
KNN	80.73	78.25	72.36	76.32
RF	88.36	76.82	85.47	79.36
CNN	78.32	87.23	81.74	87.64
BiLSTM	83.56	90.41	76.23	85.21
CNN-BiLSTM	91.33	82.34	78.36	82.39
CNN-BiLSTM-Attention	93.26	94.17	88.23	91.71

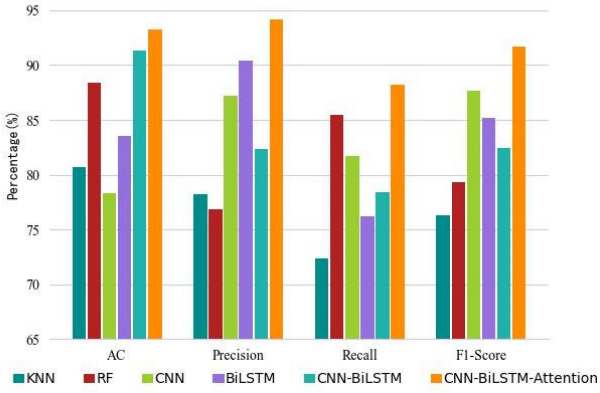

FIGURE 11. Classification results comparison on CIC-DDoS2019.

significantly improved. The value of F1-Score can reach 91.52%, 85.41%, 55.69%, 88.23%, 35.89%, 80.57%, respectively. Compared with CNN-BiLSTM, the proposed method improves 5.18 %, 4.04%, 7.57%, 11.98%, 9.58%, 8.42% in these categories, respectively. The results in Figure 9 show that the classification results of CNN-BiLSTM-Attention are all higher than those of other methods.

This paper also conducted a false positive rate test comparison on the CIC-IDS 2017 dataset, and the results are shown in Table 11 and Figure 10. The experimental results show that compared with other models, the proposed model has the lowest false positive rate on all labels except Infiltration.

From the experimental Figures 8-10, it can be seen that the CNN-BiLSTM-Attention model proposed in this paper has higher accuracy, higher F1-score and lower false alarm rate for various types of attack detection in the CIC-IDS2017 dataset.

4) ANALYSIS OF EXPERIMENTAL RESULTS BASED ON CIC-DDOS2019 DATASET

Finally, the experiments are carried out on the CIC-DDoS2019 dataset to compare the CNN-BiLSTM-Attention model and other five baseline intrusion detection models. The experimental results are shown in Table 12 and Figure 11. The results show that the accuracy rate of the proposed classification algorithm reaches 93.26%, and the precision rate reaches 94.17%.

TABLE 13. F1-Score for each class of different models on CIC-DDoS2019.

F1-Score of Models	KNN (%)	RF (%)	CNN (%)	BiLSTM (%)	CNN-BiLSTM (%)	CNN-BiLSTM-Attention (%)
Normal	75.39	81.36	76.32	85.32	77.14	88.36
DDOS	52.37	60.29	58.27	56.39	62.42	68.37
DOS	32.19	35.38	20.31	27.36	29.71	38.26
Benign	28.71	20.96	41.34	31.28	37.26	46.39
Brute Force	37.24	57.39	40.28	56.29	49.84	58.31
Web Attack	60.19	63.33	50.83	66.24	58.91	70.23
Infiltration	86.32	87.39	74.49	81.29	79.85	90.83
PortScan	18.39	32.71	22.39	35.91	28.75	36.29
Botnet	42.52	40.73	51.82	47.53	54.83	57.26

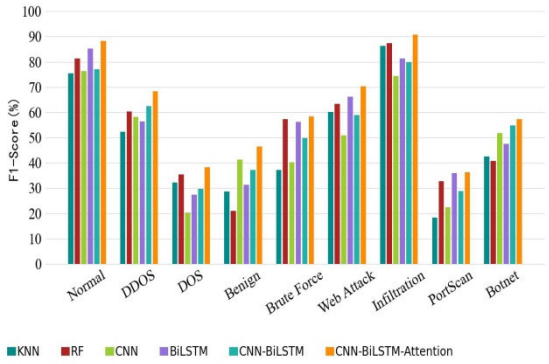

FIGURE 12. Comparison of F1-Score for each class on CIC-DDoS2019.

TABLE 14. FPR for each class of different models on CIC-DDoS2019.

FPR of Models	KNN (%)	RF (%)	CNN (%)	BiLSTM (%)	CNN-BiLSTM (%)	CNN-BiLSTM-Attention (%)
Normal	14.25	12.76	24.27	18.37	10.37	8.36
DDOS	52.38	39.72	43.58	36.58	48.29	32.48
DOS	38.84	43.58	33.86	32.57	46.92	27.39
Benign	37.94	19.82	28.43	33.85	24.73	16.83
Brute Force	16.93	18.72	13.82	25.53	22.08	10.56
Web Attack	40.32	49.85	42.93	47.85	37.83	36.85
Infiltration	41.87	28.61	38.53	32.54	35.93	25.83
PortScan	12.96	18.53	28.94	23.52	10.76	7.53
Botnet	38.14	26.83	44.24	22.71	34.52	18.72

The experimental results in Table 13 show that the results of the proposed model on Brute Force and PortScan are basically equal to those of other methods. However, the effect is improved obviously on Normal, DDOS, DOS, Benign, Web Attack, Infiltration and Botnet. The F1-Score can reach 88.36%, 68.37%, 38.26%, 46.39%, 70.23%, 90.83%, 57.26%, respectively. Compared with BiLSTM, the proposed method improves 3.04%, 11.98%, 10.9%, 15.11%, 3.99%, 9.54%, 9.73% in these categories, respectively. The results in Figure 12 show that the classification effect of the proposed models is superior to other models.

This paper also conducted a false positive rate test comparison on the CIC-DDoS2019 dataset, as shown in Table 14 and

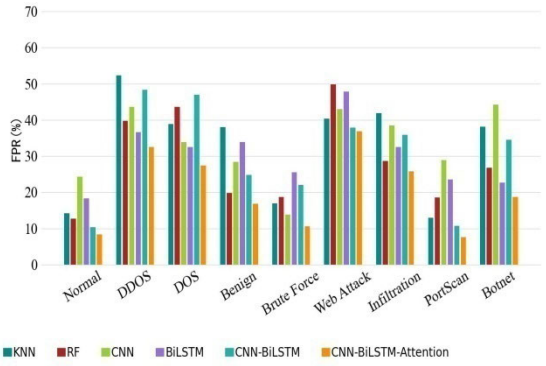

FIGURE 13. Comparison of FPR for each class on CIC-DDoS2019.

Figure 13. The experimental results show that the proposed model has the lowest false positive rate on all labels compared with other models.

From the experimental Figures 11-13, it can be seen that the accuracy rate of the proposed CNN-BiLSTM-Attention model in CIC-DDoS2019 dataset for various types of attack detection reaches 93.26%, and the precision rate reaches 94.17%. The F1-score is higher and the false positive rate is lower on most labels.

V. CONCLUSION

In the Internet era, how to strengthen network security is a key issue to be studied urgently. Intrusion detection technology uses active defense to protect the network. It is a widely used technology and management means in network security. Intrusion detection technology uses the corresponding algorithms to build a model, and uses the model to train and test the network traffic data to detect the presence of attacks.

Traditional intrusion detection methods can only learn the network traffic data at a shallow level, but cannot learn its deep meaning, so they can not accurately learn the characteristics of network traffic data. Deep learning is proposed for further feature learning from a large number of disordered high dimensional data, and its advantage is that it can set up a learning model to select the optimal features by setting reasonable training parameters. The application of deep learning to intrusion detection systems has become an inevitable trend.

Through research and analysis, some problems are found: intrusion detection lacks the ability of automatic feature extraction, and the detection efficiency is low. The ability to detect attacks is not strong, the recognition accuracy is not high, the precision rate is not high, and the false positive rate is high. To solve these problems, this paper presents a hierarchical intrusion detection model that combines multiple deep learning models with attention mechanism.

The main functions of this model include: The SCDAE model is used to reduce the dimension of data and extract features; Spatial features of the data are extracted based on CNN. After arranging the spatial features in time, the Bi LSTM is used to mine the temporal features of the network traffic data.

In feature extraction of intrusion detection models, not only the relationship between features is considered at the spatial level, but also the law of change at the time level. Thus, the accuracy of the classification is improved; On the basis of the combination of CNN and BiLSTM, the output of each time step is weighted and summed by combining the Attention mechanism to retain the important information; Finally, the session-level feature vectors obtained by hierarchical feature extraction are input to the classification module of the fully connected layer to obtain the detection results. In order to verify the model, the proposed intrusion detection model is compared with five classical models on four data sets of CIC-DDoS2019, CIC-IDS2017, CIC-IDS2018 and NSL-KDD. The experimental results show that the proposed model has higher classification accuracy and lower false positive rate, which proves that the model has higher application value.

The limitations of this study include: The proposed model was only experimented on existing public datasets when performing the work on network traffic classification and intrusion detection. The types of attacks in these datasets are known, and unknown types of attacks are not considered; The current intrusion detection is still based on the model training with a large amount of data. The intrusion and attack behaviors in the data have already existed and been discovered. There is a lack of detection capability for unknown brand new attacks.

In future studies, we can consider directly using the original network traffic data to improve the application ability of the model. In addition, the features in different datasets can be divided into different categories and then processed using different processing methods. The detection speed will be used as the main evaluation indicator so that the attack traffic data in the network can be detected more effectively and quickly.

REFERENCES

[1] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019.

[2] J. Lee and K. Park, "GAN-based imbalanced data intrusion detection system," *Pers. Ubiquitous Comput.*, vol. 25, no. 1, pp. 121–128, Feb. 2021.

[3] C. Tang, Y. Xiang, Y. Wang, J. Qian, and B. Qiang, "Detection and classification of anomaly intrusion using hierarchy clustering and SVM," *Secur. Commun. Netw.*, vol. 9, no. 16, pp. 3401–3411, Nov. 2016.

[4] R. C. Staudemeyer, "Applying long short-term memory recurrent neural networks to intrusion detection," *South Afr. Comput. J.*, vol. 56, pp. 136–154, Jul. 2015.

[5] B. Yan and G. Han, "Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system," *IEEE Access*, vol. 6, pp. 41238–41248, 2018.

[6] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[7] S. M. Kasongo and Y. Sun, "A deep long short-term memory based classifier for wireless intrusion detection system," *ICT Exp.*, vol. 6, no. 2, pp. 98–103, Jun. 2020.

[8] R. Zhao, J. Yin, Z. Xue, G. Gui, B. Adebisi, T. Ohtsuki, H. Gacanin, and H. Sari, "An efficient intrusion detection method based on dynamic autoencoder," *IEEE Wireless Commun. Lett.*, vol. 10, no. 8, pp. 1707–1711, Aug. 2021.

[9] V. Kumar, H. Chauhan, and D. Panwar, "K-means clustering approach to analyze NSL-KDD intrusion detection dataset," *Int. J. Soft*, vol. 3, no. 4, pp. 2231–2307, Sep. 2013.

[10] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset," *IEEE Access*, vol. 8, pp. 32150–32162, 2020.

[11] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2018.

[12] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231–48246, 2018.

[13] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.

[14] A. Telikani, A. H. Gandomi, K.-K. R. Choo, and J. Shen, "A cost-sensitive deep learning-based approach for network traffic classification," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 1, pp. 661–670, Mar. 2022.

[15] X. Wang, S. Yin, H. Li, J. Wang, and L. Teng, "A network intrusion detection method based on deep multi-scale convolutional neural network," *Int. J. Wireless Inf. Netw.*, vol. 27, no. 4, pp. 503–517, Dec. 2020.

[16] L. Zhang, Z. Huang, W. Liu, Z. Guo, and Z. Zhang, "Weather radar echo prediction method based on convolution neural network and long short-term memory networks for sustainable e-agriculture," *J. Cleaner Prod.*, vol. 298, May 2021, Art. no. 126776.

[17] Y. Li and B. Zhang, "An intrusion detection algorithm based on deep CNN," *Comput. Appl. Softw.*, vol. 37, no. 4, pp. 324–328, 2020.

[18] L. Lv, Z. Wu, J. Zhang, L. Zhang, Z. Tan, and Z. Tian, "A VMD and LSTM based hybrid model of load forecasting for power grid security," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6474–6482, Sep. 2022.

[19] L. Zhang, C. Xu, Y. Gao, Y. Han, X. Du, and Z. Tian, "Improved Dota2 lineup recommendation model based on a bidirectional LSTM," *Tsinghua Sci. Technol.*, vol. 25, no. 6, pp. 712–720, Dec. 2020.

[20] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Inf. Sci.*, vol. 513, pp. 386–396, Mar. 2020.

[21] L. Lv, J. Chen, Z. Zhang, B. Wang, and L. Zhang, "A numerical solution of a class of periodic coupled matrix equations," *J. Franklin Inst.*, vol. 358, no. 3, pp. 2039–2059, Feb. 2021.

[22] H. Xu and Y. Lv, "Mining and application of tourism online review text based on natural language processing and text classification technology," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–13, May 2022.

[23] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32464–32476, 2020.

[24] Z. Wu, J. Wang, L. Hu, Z. Zhang, and H. Wu, "A network intrusion detection method based on semantic re-encoding and deep learning," *J. Netw. Comput. Appl.*, vol. 164, Aug. 2020, Art. no. 102688.

[25] L. Zhang, S. Tang, and L. Lv, "An finite iterative algorithm for sloving periodic Sylvester bimatrix equations," *J. Franklin Inst.*, vol. 357, no. 15, pp. 10757–10772, Oct. 2020.

[26] Z. Ma, J. Li, Y. Song, X. Wu, and C. Chen, "Network intrusion detection method based on FCWGAN and BiLSTM," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–17, Apr. 2022.

[27] L. Lv, S. Tang, and L. Zhang, "Parametric solutions to generalized periodic Sylvester bimatrix equations," *J. Franklin Inst.*, vol. 357, no. 6, pp. 3601–3621, Apr. 2020.

[28] F. J. Abdullayeva, "Advanced persistent threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm," *Array*, vol. 10, Jul. 2021, Art. no. 100067.

[29] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020.

[30] J. Zhang, Y. Ling, X. Fu, X. Yang, G. Xiong, and R. Zhang, "Model of the intrusion detection system based on the integration of spatial-temporal features," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101681.

[31] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116.

[32] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8.

[33] L. Lv, Z. Wu, L. Zhang, B. B. Gupta, and Z. Tian, "An edge-AI based forecasting approach for improving smart microgrid efficiency," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7946–7954, Nov. 2022.
[34] H. Xu, G. Fan, and Y. Song, "Novel key indicators selection method of financial fraud prediction model based on machine learning hybrid mode," *Mobile Inf. Syst.*, vol. 2022, pp. 1–12, Mar. 2022.
[35] H. Xu, G. Fan, and Y. Song, "Application analysis of the machine learning fusion model in building a financial fraud prediction model," *Secur. Commun. Netw.*, vol. 2022, pp. 1–13, Mar. 2022.
[36] A. Meryem and B. E. Ouahidi, "Hybrid intrusion detection system using machine learning," *Netw. Secur.*, vol. 2020, no. 5, pp. 8–19, May 2020.
[37] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Comput. Netw.*, vol. 188, Apr. 2021, Art. no. 107840.

HONGSHENG XU received the M.S. degree from Henan University, China, in 2007. He is currently with the Henan Key Laboratory for Big Data Processing and Analytics of Electronic Commerce and an Associate Professor with the College of Electronic Commerce, Luoyang Normal University. His research interests include artificial intelligence, deep learning, intrusion detection systems, knowledge graph, and attention mechanism.

LIBO SUN received the M.S. degree from the University of Mysore, India, in 2012. He is currently with the Binghamton University, State University of New York, Binghamton, NY, USA, and the College of Electronic Commerce, Luoyang Normal University. His research interests include deep learning, knowledge graph, and intrusion detection systems.

GANGLONG FAN received the M.S. degree from the Huazhong University of Science and Technology, China. He is currently with the Henan Key Laboratory for Big Data Processing and Analytics of Electronic Commerce and a Professor with the College of Electronic Commerce, Luoyang Normal University. His research interests include deep learning, machine learning, and intrusion detection systems.

WANXING LI received the M.S. degree from the Wuhan Institute of Technology, China, in 2019. She is currently with the Henan Key Laboratory for Big Data Processing and Analytics of Electronic Commerce and a Lecturer with the College of Electronic Commerce, Luoyang Normal University. Her research interests include deep learning, knowledge graph, supply chain management, E-commerce, and intrusion detection systems.

GUOFANG KUANG received the M.S. degree from Harbin Engineering University, China, in 2007. He is currently an Associate Professor with the School of Information Technology, Luoyang Normal University. His research interests include artificial intelligence, machine learning, intrusion detection systems, and big data analysis.