IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 March 2024, accepted 26 March 2024, date of publication 4 April 2024, date of current version 19 April 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3384528*

RESEARCH ARTICLE

# Network Intrusion Detection Method Based on CNN-BiLSTM-Attention Model

**WEI DAI, XINHUI LI, WENXIN JI, AND SICHENG HE**
School of Electronics and Information Engineering, Liaoning Technical University, Huludao 125105, China
Corresponding author: Wei Dai (daiwei0084@126.com)

**ABSTRACT** To address the issue of low detection accuracy and high false positive rate in existing network intrusion detection methods, this paper proposes an intrusion detection model based on CNN-BiLSTM-Attention. Firstly, CNN is used to extract the spatial features from the intrusion data; Secondly, BiLSTM is used to mine the temporal features from the intrusion data further; Thirdly, the attention mechanism is used to assign different weights to the extracted spatiotemporal features and then enhance the role of important features in the calculation process, which can improve the classification accuracy of the model. In addition, for the problem of class imbalance existing in network intrusion data, Equalization Loss v2 is introduced as the loss function of the CNN-BiLSTM-Attention model, making the model pay more attention to minority class data during the training process, thereby improving the detection rate of the model for the minority class data. Finally, comparative experiments are conducted on NSL-KDD, UNSW-NB15, and CIC-DDoS2019 datasets. The experimental results show that the CNN-BiLSTM-Attention model outperforms the other models in terms of accuracy, detection rate, and false positive rate.

**INDEX TERMS** Network intrusion detection, CNN, BiLSTM, attention mechanism, EQL v2, class imbalance.

## I. INTRODUCTION

The situation of network security is becoming increasingly severe, and network attackers will use intelligent attack methods to invade network systems, further obtaining private information or damaging network systems. Intrusion detection systems can detect various attack behaviors in the network, which is a hot research direction in the field of network security [1]. The current intrusion detection system faces two key problems. The first problem is that the detection rate of most intrusion detection models for the minority attack categories is too low due to an unbalanced distribution across various categories within the intrusion data. The second issue is that most intrusion detection models do not extract enough spatial or temporal features of the intrusion data and lack a comprehensive consideration of network intrusion data information, resulting in low accuracy and efficiency of intrusion detection. Therefore, how to effectively detect network attacks in real network environments is an important problem that intrusion detection systems need to solve.

At present, there are mainly two types of methods to solve class imbalance problems: data-level methods and cost-sensitive learning methods. The data-level approach mainly deals with imbalanced training datasets to balance the distribution of data between different categories. Oversampling minority class data and downsampling majority class data are two common methods [2], [3], [4]. Literature [5] proposes a method to solve the problem of class imbalance, which combines the Synthetic Minority Over Sampling Technique (SMOTE) technology and Gaussian Mixture Model (GMM) technology. The principle of this method is to oversample minority class data through SMOTE and undersample majority class data through GMM, ultimately achieving the goal of balancing various types of data. This method combines the advantages of oversampling and undersampling, reducing time costs and avoiding the loss of important information. Literature [6] oversamples the samples of the minority class data using Adaptive Synthesis (ADASYN) technology, increasing the sample number of the

The associate editor coordinating the review of this manuscript and approving it for publication was Nurul I. Sarkar.

minority class data to achieve the same proportion of all types of intrusion data. Then, the LightGBM model is used to build a classifier. The results indicate that ADASYN technology effectively solves the problem of the low detection rate of minority class data caused by imbalanced training sets. Literature [7] proposes a Generative Adversarial Network (GAN) method, which can generate new attack class samples in a limited number of the minority attack class data, and then input the generated new samples into a detection model based on a Convolutional Neural Network (CNN). Finally, experiments are conducted on the NSL-KDD [8], KDDCup99 [9], and UNSW-NB15 [10] datasets, and the results show that this GAN method solves the limitations caused by small sample attack data. Although the above methods improve the detection rate of intrusion detection models for minority class data, in real life, there is a certain degree of skewness in the class distribution of many intrusion data, and relying solely on data-level methods may reduce the accuracy of certain types of intrusion data. In the field of network intrusion detection, cost-sensitive learning methods have the advantages of high efficiency and low time cost when dealing with imbalanced data, while not requiring additional data processing [11], [12], [13]. Cost-sensitive learning methods assign a larger cost factor to minority class data, thereby improving the detection rate of classifiers for minority class data. Literature [14] uses the focal loss function to assign a larger misclassification cost to minority class samples to achieve the effective classification of unbalanced intrusion data. Experiments show that the focal loss function can effectively improve the detection ability of the classifier for minority class data.

Traditional machine learning techniques have been widely applied in the field of intrusion detection, mainly including Random Forest [15], Support Vector Machine (SVM) [16], [17], Adaptive Boosting (AdaBoost) [18], [19], Decision Tree (DT) [20], [21], and K-Nearest Neighbor (KNN) [22], [23]. However, traditional machine learning techniques cannot identify large-scale network intrusion attacks efficiently and accurately. In the face of large-scale intrusion data, deep learning techniques are more suitable for network intrusion detection due to their powerful automatic feature extraction capabilities [24], [25], [26]. In recent years, the research on deep learning in the field of intrusion detection has become increasingly widespread. There are four commonly used deep learning algorithms in the field of intrusion detection: intrusion detection models based on CNN [27], [28], intrusion detection models based on RNN [29], [30], intrusion detection models based on LSTM [31], [32], and intrusion detection models based on BiLSTM [33]. CNN can automatically extract local features from original data and mine potential unknown intrusion behavior features, but it cannot learn sequence correlation. RNNs can only learn sequence information within a limited step size and cannot learn long-term dependent information. LSTM can only learn long-term dependency information from one direction and cannot fully extract the temporal features of the data. Compared with LSTM, BiLSTM is a combination of forward LSTM and backward LSTM. BiLSTM can consider the impact of forward and backward information on current information, reducing the false alarm rate of intrusion detection. The attention mechanism has been widely applied in many research fields. It is a method that mimics human insight into the outside world, grabbing important parts from panoramic objects and performing feature processing. The attention mechanism is an efficient resource allocation mechanism. Recently, attention mechanisms have become highly popular in the field of intrusion detection due to their powerful learning ability in sequence data [34]. Literature [35] proposes a network intrusion detection model based on BiLSTM and a multi-head attention mechanism. This model enhances attention to certain feature vectors through attention mechanisms and captures long-distance dependencies of feature vectors through BiLSTM. The model was tested on the NSL-KDD, KDDCup99, and CIC-IDS2017 [36] datasets, and the results showed that the model performed well in terms of accuracy. Network intrusion data has a typical hierarchical structure, but many researchers choose a single model in the field of intrusion detection to fully extract features, resulting in the low detection rate of most current intrusion detection methods. Therefore, some researchers have proposed methods that combine the advantages of different basic models to achieve efficient and accurate feature extraction. Literature [37] combines CNN and RNN, and this method significantly improves detection accuracy compared to a single CNN structure or a single RNN structure. Literature [38] establishes an intrusion detection model combining CNN and LSTM and conducts multi-classification and binary classification performance comparison experiments on the UNSW-NB15, CIC-IDS2017, and WSN-DS [39] datasets. The results show that this model has higher detection accuracy than a single LSTM model. Literature [40] proposes a network intrusion detection model based on CNN-BiLSTM. The model first uses CNN to extract spatial features of intrusion data and then uses BiLSTM to extract temporal features of the data. Compared with the CNN-LSTM model, the CNN-BiLSTM model has higher detection accuracy on the NSL-KDD and UNSW-NB15 datasets. Literature [41] proposes an intrusion detection model that combines CNN and the attention mechanism, where a single-layer CNN-Attention structure is used to extract local spatiotemporal features of intrusion data, while a multi-layer CNN-Attention structure can more fully learn multi-level spatiotemporal features. The model has good binary and multi-classification performance on intrusion datasets. This model has been compared with other deep learning models in terms of accuracy, detection rate, and false positive rate, and the results fully demonstrate that this model has better detection performance. Although the intrusion detection model using deep learning methods has the advantage of processing

large-scale data, the models are prone to training for the majority class data, which reduces the detection rate for the minority class data. Literature [42] proposes a hierarchical intrusion detection model, which is called SCDAE-CNN-BiLSTM-Attention, which integrates multiple deep learning models. The main process of this hierarchical model is as follows. Firstly, the Stacked Convolutional Denoising Autoencoders (SCDAE) model is used to perform feature denoising on high-dimensional intrusion data, so that the original data is replaced with new feature samples. Secondly, CNN is used to extract the spatial features of the intrusion data; Thirdly, BiLSTM is used to fully mine the temporal features of the intrusion data; Fourthly, the Self-attention mechanism is introduced to summarize the important intrusion information by weighting the spatiotemporal features at each time step. Finally, the softmax classifier is used to obtain the classification results of the SCDAE-CNN-BiLSTM-Attention model. By conducting comparative experiments on four intrusion datasets, NSL-KDD, CIC-IDS2017, CIC-IDS2018 [43], and CIC-DDoS2019 [44], the results show that the SCDAE-CNN-BiLSTM-Attention model has high classification accuracy and low false positive rate. The SCDAE-CNN-BiLSTM-Attention method proposed in this paper is designed under the assumption that the intrusion data is credible and does not contain noise and redundancy. However, in reality, intrusion data often contains noise and redundancy, which will affect intrusion detection. model performance. Literature [45] proposes the DDoS attack detection method based on CNN-AttBiLSTM. Firstly, before the CNN-AttBiLSTM model, a combination of the Random Forest and Pearson correlation analysis algorithms is used to select important features to reduce redundant data. Secondly, ID-CNN and BiLSTM are used to extract spatial and temporal features, respectively. Thirdly, the self-attention mechanism is introduced to automatically calculate the weights of various spatiotemporal features and screen out the features that are important to the classification results. Finally, the softmax classifier is used to classify the intrusion data. The CNN-AttBiLSTM model has a high accuracy and a low false positive rate in DDoS attack detection. However, before, the CNN-AttBiLSTM model, the RFP method is used for feature selection, which may result in the loss of useful intrusion information, thereby reducing the detection performance of the CNN-AttBiLSTM model for DDoS attacks. Literature [46] proposes a network intrusion detection method based on TCN-BiGRU-Attention. Firstly, Temporal Convolutional Network (TCN) and Bidirectional Gated Recurrent Unit (BiGRU) are used to simultaneously extract the spatiotemporal features of network traffic data. Secondly, the spatiotemporal features are assigned different weight parameters accordingly through the self-attention mechanism, thereby maximizing the retention of the prominent features. Finally, softmax is used as a classifier to identify network traffic data. The experimental results show that the TCN-BiGRU-Attention model achieves good multi-classification results on the CSE-CIC-IDS2018 dataset. However, the TCN-BiGRU-Attention model does not solve the class imbalance problem existing in the network traffic data, which makes the model have poor classification results for the minority attack samples.

Based on the above analysis, aim at the problem of low detection accuracy of existing network intrusion detection methods, this paper proposes an intrusion detection model based on CNN-BiLSTM-Attention from the perspective of network traffic data having temporal correlations and spatial correlations and the existence of importance differences between different features. This model combines CNN and BiLSTM to extract the spatiotemporal features of network intrusion data. Considering the differences in importance between different spatiotemporal features, the attention mechanism is used to screen out the features that have a significant impact on the classification results, thereby improving the classification performance of the model. In addition, to solve the class imbalance problem that exists in the network intrusion dataset, EQL v2 is selected to participate in the training of the model, which makes the model increase the attention to the minority class data, thus improving the detection accuracy of the model for the minority class data. This paper conducts experiments on three public datasets, and the results show that the CNN-BiLSTM-Attention model achieves better classification results on all three datasets.

## II. CNN-BiLSTM-ATTENTION MODEL

Intrusion detection models based on CNN can automatically extract local features in the intrusion data, avoiding the process of manual feature extraction. However, Intrusion detection models based on CNN cannot learn the temporal correlation between the intrusion data. Intrusion detection models based on BiLSTM can discover persistent attack behavior by extracting the bidirectional temporal features in the intrusion data. However, the meanings represented by the features at each moment are different, and the classifier cannot pay more attention to important features, which may lead to the loss of important information and an increase in the false negative rate. The attention mechanism can selectively focus on features that have a greater impact on the classification results, assigning higher weights to the features that need to be focused on, while assigning lower weights to other features, thereby improving the classification accuracy of the model. Therefore, combining the advantages of CNN, BiLSTM, and attention mechanism, this paper proposes an intrusion detection model based on CNN-BiLSTM-Attention.

The structure of the CNN-BiLSTM-Attention model is shown in Figure 1, the CNN-BiLSTM-Attention model is mainly divided into the Input layer, the CNN layer, the BiLSTM layer, the Attention layer, the Fully Connected layer, and the Output layer.

The Input layer takes the preprocessed network intrusion data as the input to the CNN-BiLSTM-Attention model, and the dimension of the input data is usually one-dimensional.

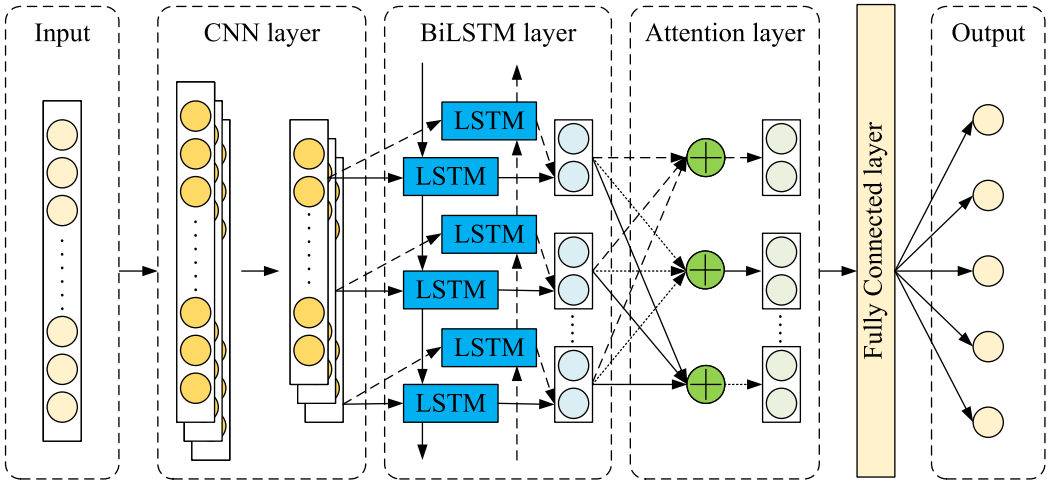


**FIGURE 1.** Framework of the CNN-BiLSTM-Attention model.

The BiLSTM layer captures bi-directional temporal features in the intrusion data by learning the correlation of the output features from the CNN layer. The number of nodes in the BiLSTM layer is set to 128 to prevent errors caused by different vector dimensions.

The Attention layer appears after the BiLSTM layer. The Attention layer distinguishes the importance of different features by assigning corresponding attention weights for features at different moments, thereby strengthening the influence of key features on classification results.

In the Fully Connected layer, a flattened layer is firstly used to reshape the multi-dimensional features into a 1D vector, then a Dense layer is used to integrate the flattened features, and finally, the integrated result is passed to the output layer for classification.

The Output layer is the last layer of the model, which is the classification layer. In this paper, the softmax function is used as a classifier to achieve the classification for network intrusion data, that is, the probability scores of various types of attacks are calculated by the softmax function.

### A. CONVOLUTIONAL NEURAL NETWORK

CNN can automatically extract the internal features in the intrusion data [47]. At the same time, CNN has the advantages of local connectivity and weight sharing, which can well solve the problem of excessive computational complexity. In intrusion detection systems, CNN can obtain effective local features by performing multiple convolution and pooling operations on the intrusion data [48].

CNN framework consists of two one-dimensional Convolutional layers, two one-dimensional MaxPooling layers, and two one-dimensional Batch Normalization layers.

Figure 2 shows the implementation process of CNN. The Convolutional layer is used to extract high-dimensional local features from the input data. The size of the convolutional kernel is set to 64, the sliding step of the convolutional kernel is set to 1, and the padding method is set to the same

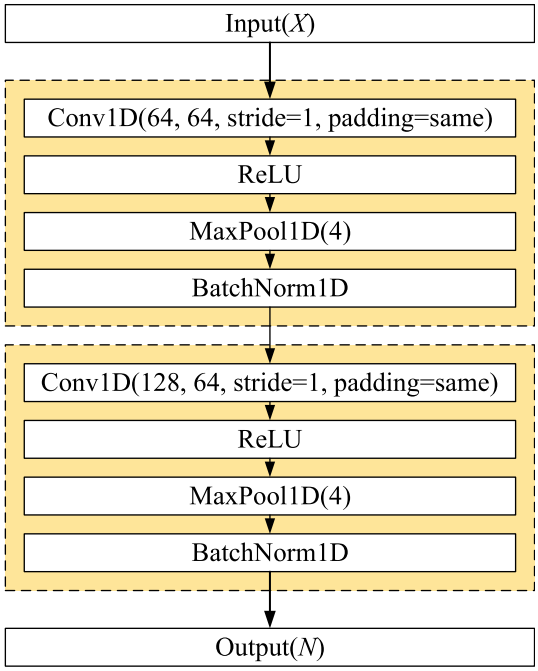


**FIGURE 2.** Framework of CNN.

to ensure that the size of the input and output vectors are consistent. The ReLU function is introduced to add non-linear factors to the network, making the model capable of handling more complex classification tasks. The maximum pooling method is chosen, on the one hand, it can perform feature dimensionality reduction, on the other hand, it makes the network extract the features more efficiently. The pooling step size is set to 4. The Batch Normalization layer is added after each pooling layer to ensure that the output feature values of the pooling layer follow the standard normal distribution, thereby accelerating the convergence speed of the network.

$X$ represents the input data of the CNN, and $N$ represents the output feature of the CNN.

### B. BIDIRECTIONAL LONG SHORT-TERM MEMORY

CNN is unable to extract the temporal features of intrusion data, mainly because the output of forward neurons in the CNN structure can only be transmitted to the next layer of neurons, and cannot analyze the temporal correlation among datas [49], [50]. As an optimization of RNN, the LSTM network can effectively capture long-distance dependencies in time series data [51]. Meanwhile, the LSTM network has added three "gate" structures, which can effectively solve the problems of gradient vanishing and exploding caused by RNN [52]. LSTM can better detect intrusion behavior over some time by learning temporal features in network intrusion data. The structure of the LSTM unit is shown in Figure 3.

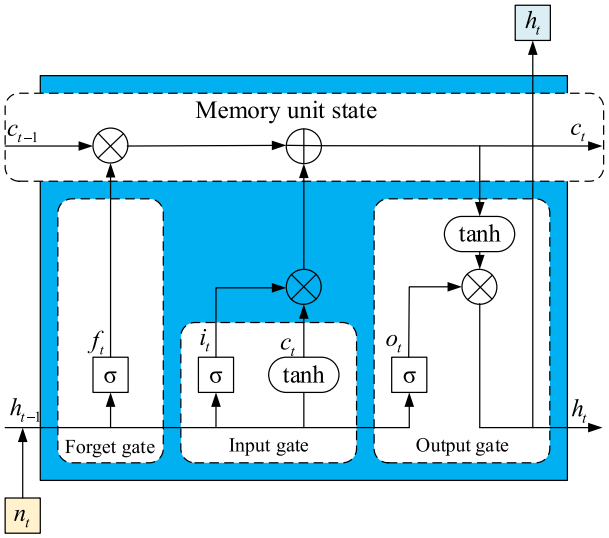


**FIGURE 3.** Structure of LSTM unit.

The principle of LSTM is to complete the retention and updating of the state information of memory units through the forgetting gates, the input gates, and the output gates, so that the useful information is retained, and the useless information is forgotten.

The forget gate determines which part of information needs to be forgotten. The forget gate reads the output information $h_{t-1}$ at the time $t-1$ and the input information $n_t$ at time $t$, it maps the read information to a range of [0,1] through the sigmoid function, and it determines which information is forgotten based on the numerical value, where "1" represents that all the information is remembered, and "0" represents that all the information is forgotten. The output of the forget gate $f_t$ is as follows.

$$f_t = \sigma\left(W_f n_t + U_f h_{t-1} + b_f\right) \tag{1}$$

where $W_f$ is the weight matrix between the input layer and the forget gate, $U_f$ is the weight matrix between the hidden layer and the forget gate, $b_f$ is the bias vector of the forget gate.

The input gate determines which part of the information needs to enter the memory unit state $c_t$ at time $t$. By normalizing $h_{t-1}$ and $n_t$, the output $i_t$ of the input gate is obtained to preserve important information. A temporary state $\widetilde{c}_t$ is generated using the tanh activation function. The state $c_{t-1}$ at time $t-1$ is updated to the state $c_t$ at time $t$. The updated process of the input gate is as follows.

$$\begin{cases} i_t = \sigma\left(W_i n_t + U_i h_{t-1} + b_i\right) \\ \widetilde{c}_t = \tanh\left(W_c n_t + U_c h_{t-1} + b_c\right) \\ c_t = f_t c_{t-1} + i_t \widetilde{c}_t \end{cases} \tag{2}$$

where $W_i$ and $U_i$ are the connection weights of the input gate, $b_i$ is the bias vector of the input gate, $W_c$ and $U_c$ are the weight matrices of the memory unit, $b_c$ is the bias vector.

The output gate determines the output information at the $t$ moment. The same normalization operation is applied to $h_{t-1}$ and $n_t$ to get the output of the output gate $o_t$. The state $c_t$ processed by the tanh function is multiplied with $o_t$ to get the output information $h_t$ of the LSTM unit at the time $t$. The updated process of the output gate is as follows.

$$\begin{cases} o_t = \sigma\left(W_o n_t + U_o h_{t-1} + b_o\right) \\ h_t = o_t \tanh\left(c_t\right) \end{cases} \tag{3}$$

where $W_o$ and $U_o$ are the connection weights of the output gate, $b_o$ is the bias vector of the output gate.

The attack behavior in network intrusion data is not completed at a single point in time but will be sent continuously for some time, with a certain degree of temporal correlation, and there may be some correlation between the data before and after. Therefore, by analyzing the correlation of time series in intrusion data, intrusion behavior can be better detected. However, LSTM can only transmit time series information about intrusion data unidirectionally and cannot learn from both directions simultaneously. Therefore, in this paper, BiLSTM is used to extract the bidirectional time series information of intrusion data, to improve the classification accuracy of intrusion detection [53]. BiLSTM is a combination of forward LSTM and backward LSTM, and the structure of BiLSTM is shown in Figure 4. Assuming the input feature of BiLSTM at time $t$ is $n_t$, the forward output feature of LSTM at time $t$ is $\overrightarrow{h_t}$, the backward output feature of LSTM at time $t$ is $\overleftarrow{h_t}$, and the final output feature $l_t$ of BiLSTM at time $t$ is obtained by combining $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$. The updated process of the BiLSTM is as follows.

$$\begin{cases} \overrightarrow{h_t} = LSTM\left(n_t, \overrightarrow{h_{t-1}}\right) \\ \overleftarrow{h_t} = LSTM\left(n_t, \overleftarrow{h_{t+1}}\right) \\ l_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \end{cases} \tag{4}$$

### C. ATTENTION MECHANISM

The principle of attention mechanism is based on a signal connection mechanism in the human brain when observing

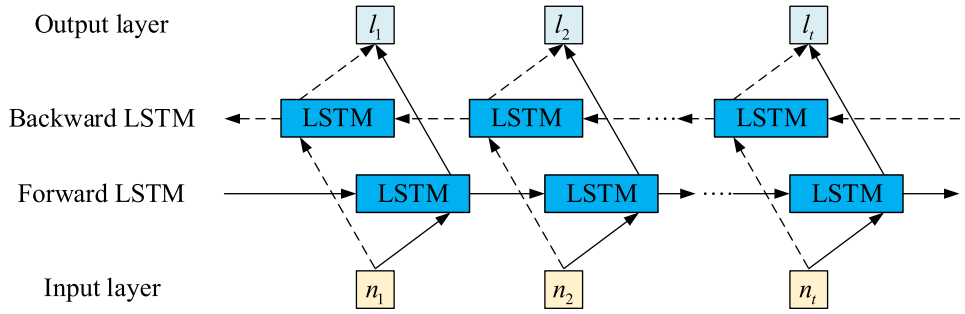


**FIGURE 4.** Structure of BiLSTM.

things, which is to quickly scan the global image to obtain the target area that needs to be focused on, and then invest more attention in this area to obtain more detailed information of this target area, while suppressing information outside the target area [54]. Attention mechanism has the advantages of few training parameters, low computational complexity, parallel computation, and fast operation speed, so they have been widely used in various fields recently. In the field of deep learning, the basic idea of the attention mechanism is to select the information that is more critical to the current task from the input information. Reference [55] It assigns the corresponding weights according to the importance of each input information, thereby quickly choosing out the high-value information and ignoring other useless information. In the field of intrusion detection, although BiLSTM can extract temporal correlations from intrusion data, the influence of features at different time points on classification results varies. Therefore, an attention mechanism is introduced after the BiLSTM layer. The attention mechanism can assign corresponding weights based on the importance of features at different time points to classification results, thereby capturing more valuable information. The calculation process of the attention mechanism is divided into the following three steps.

*Step 1( Calculate Attention Weights):* Calculate the attention weights $\widetilde{\alpha_{ti}}$ between the output features $l_t$ of BiLSTM at time $t$ and the output features $l_i$ at time $i$.

$$\widetilde{\alpha_{ti}} = \tanh\left(Wl_t + Ul_i + b\right), i = 1, 2, \cdots, t-1, t \quad (5)$$

where $W$,$U$, and $b$ are the learning parameters of the model respectively, which are constantly updated during the model training process.

*Step 2( Normalization):* After calculating the weights of each part, the values are not within a certain range, so it is necessary to normalize them. The normalized weights $\alpha_{ti}$ are obtained by using the softmax function.

$$\alpha_{ti} = \text{softmax}(\widetilde{\alpha_{ti}}) = \frac{\exp\left(\widetilde{\alpha_{ti}}\right)}{\sum_{i=1}^{t} \widetilde{\alpha_{ti}}}, i = 1, 2, \cdots, t-1, t \quad (6)$$

*Step 3 (Weighted Sum):* Weighted summation of each part can obtain the output $s_t$ of the attention layer at time $t$.

$$s_t = \sum_{i=1}^{t} \alpha_{ti} l_i, i = 1, 2, \cdots, t-1, t \quad (7)$$

As can be seen from the calculation results of $s_t$, the result value of each $s_t$ is related to the entire input vector, which is also one reason why the attention mechanism can accelerate the computation in parallel.

### D. LOSS FUNCTION

The loss function is mainly applied to the softmax classification layer of the intrusion detection model, which is used to measure the difference between the target predicted value and the true value, thereby guiding the model to optimize in a more accurate direction. The higher the loss value indicates that the greater the difference between the real value and the predicted value, the worse the performance of the model, and the goal of the intrusion detection model training is to minimize the loss value as much as possible. Cross Entropy (CE) loss function is commonly used in intrusion detection models, but it does not consider the problem of imbalanced datasets, which may cause the model to be prone to overfitting problems during the training process [56], [57], [58]. To address this problem, this paper applies EQL v2 to the CNN-BiLSTM-Attention model [59]. EQL v2 is a relatively new type of loss function, which was proposed by researchers in 2021. The principle of EQL v2 is to balance the positive and negative sample gradients of each category using a gradient reweighting mechanism. Specifically, this method increases the loss weight of the minority class so that the model no longer ignores the training of minority class data, thereby improving the detection accuracy of the model for minority class samples.

The working principle of EQL v2 is to calculate the corresponding weight by accumulating the positive and negative gradient ratios output by the classifier during each backpropagation. It should be noted that for the category $j$, other categories can be considered as negative samples, and the positive and negative samples of each category correspond to a weight. When the model is iterated $t$ times,

the calculation equation of EQL v2 is as follows.

$$L_t = -\frac{1}{N}\sum_{i\in N}\left(\sum_{j=1}^{C}\beta_j^t\left(y_{ij}\ln p_{ij} + \left(1-y_{ij}\right)\ln\left(1-p_{ij}\right)\right)\right) \tag{8}$$

among them, $\beta_j^t$ represents the weight corresponding to the class $j$ when the number of iterations for training the model is $t$, and the specific calculation equation is as follows.

$$\beta_j^t = \begin{cases} \beta_j^{t(\text{ pos })} = 1 + \alpha\left(1 - \mathrm{f}\left(g_j^t\right)\right), & \text{if } y_j > 0, \\ \beta_j^{t(\text{ neg })} = \mathrm{f}\left(g_j^t\right), & \text{if } y_j = 0, \end{cases} \tag{9}$$

$$\mathrm{f}(x) = \frac{1}{1+e^{-\gamma(x-\mu)}} \tag{10}$$

where, $\alpha$, $\mu$ and $\gamma$ are hyperparameters; $g_j^t$ represents the cumulative positive and negative gradient ratio of the classifier corresponding to the category $j$ when the number of iterations of training is $t-1$; $\beta_j^{k(pos)}$ represents the weight corresponding to the positive sample of the category $j$ when the number of iterations of training is $k$; $\beta_j^{k(neg)}$ represents the weight corresponding to the negative sample of the category $j$ when the number of iterations of training is $k$.

When the number of iterations of training is $t$, the update process of the classifier $\omega_j$ corresponding to the category $j$ is as follows.

$$\omega_j^{t+1} = \omega_j^t - \frac{1}{N}\left(\beta_j^{t(pos)}\sum_{i\in N_j^{\text{pos}}}\left(1-p_{ij}\right) + \beta_j^{t(neg)}\sum_{i\in N_j^{\text{neg}}}p_{ij}\right)\frac{\partial z_j}{\partial \omega_j} \tag{11}$$

EQL v2 mainly adjusts the weights of the positive and negative samples corresponding to a category through equation 9 and equation 10. For the minority categories whose cumulative positive and negative gradient ratios are less than the ratio $\mu$, the less the ratio of positive and negative gradients, the larger $\beta_j^{k(pos)}$ and the smaller $\beta_j^{k(neg)}$ in equation 11, which results in the larger the positive gradient corresponding to positive samples and the smaller the negative gradient corresponding to negative samples. From equation 9, it can be seen that for categories with a positive and negative gradient ratio larger than $\mu$, the weight corresponding to positive samples is also larger than 1. Therefore, the improved classification loss function does not suppress the majority category. Finally, EQL v2 adjusts the positive and negative gradient ratio of the minority classes with varying weights to improve the detection performance of the model for the minority classes.

## III. EXPERIMENTS

### A. EXPERIMENTAL DATA

The NSL-KDD dataset is an improved version of the KDDCup99 dataset, which eliminates the duplicate instances in the KDDCup99 dataset. The data types of the NSL-KDD dataset can be classified into five categories, namely Normal, DoS, Probe, U2R, and R2L. The data distribution of the NSL-KDD dataset is shown in Table 1.

**TABLE 1. Data distribution of the NSL-KDD dataset.**

| Class | Number of Train set | Number of Test set |
|---|---|---|
| Normal | 67343 | 9711 |
| DoS | 45927 | 7458 |
| Probe | 11656 | 2421 |
| R2L | 995 | 2754 |
| U2R | 52 | 200 |
| Total | 125973 | 22544 |

The UNSW-NB15 dataset combines real modern normal network traffic with contemporary comprehensive attack activities, comprehensively and deeply reflecting the modern network intrusion situation. The data types of the UNSW-NB15 dataset can be divided into 10 categories, namely Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Normal, Reconnaissance, Shellcode, and Worms. The data distribution of the UNSW-NB15 dataset is shown in Table 2.

The CIC-DDoS2019 dataset was established by the Canadian Institute for Cybersecurity. To simulate real modern network attacks, the CIC-DDoS2019 dataset is added to various DDoS attacks. The data types of the CIC-DDoS2019 dataset can be divided into 15 categories. namely Normal and DrDoS-DNS, DrDoS-LDAP, DrDoS-MSSQL, DrDoS-NetBIOS, DrDoS-NTP, DrDoS-SNMP, DrDoS-SSDP, DrDoS-UDP, SYN, TFTP, LDAP, NetBIOS, MSSQL, Portmap, UDP, and UDP- Lag. The data distribution of the CIC-DDoS2019 dataset is shown in Table 3.

**TABLE 2. Data distribution of the UNSW-NB15 dataset.**

| Class | Number of Train set | Number of Test set |
|---|---|---|
| Normal | 56000 | 37000 |
| Generic | 40000 | 18871 |
| Exploits | 33393 | 11132 |
| Fuzzers | 18184 | 6062 |
| DoS | 12264 | 4089 |
| Reconnaissance | 10493 | 3496 |
| Analysis | 2000 | 677 |
| Backdoor | 1746 | 583 |
| Shellcode | 1131 | 378 |
| Worms | 130 | 44 |
| Total | 175341 | 82332 |

**TABLE 3. Data distribution of the CIC-DDoS2019 dataset.**

| Class | Number | Class | Number |
|---|---|---|---|
| Normal | 20803 | SYN | 11102 |
| DrDoS-DNS | 11453 | TFTP | 2205 |
| DrDoS-LDAP | 1733 | LDAP | 540 |
| DrDoS-MSSQL | 8044 | NetBIOS | 1000 |
| DrDoS-NetBIOS | 11543 | MSSQL | 4356 |
| DrDoS-NTP | 30102 | Portmap | 1060 |
| DrDoS-SNMP | 2713 | UDP | 12131 |
| DrDoS-SSDP | 13214 | UDP-lag | 208 |
| DrDoS-UDP | 10387 | | |

There is a class imbalance problem in the NSL-KDD, UNSW-NB15, and CIC-DDoS2019 datasets, which seriously affects the training and testing of the intrusion detection model for the minority classes. The minority class U2R in the NSL-KDD dataset accounts for 0.04% of the training samples, while the majority class Normal accounts for 53.46%. The minority classes Shellcode and Backdoor in the UNSW-NB15 dataset account for 0.65% and 1.01% respectively of the training samples, while the majority class Normal accounts for 32.27%. Similarly, in the CIC-DDo2019 dataset, there is also imbalanced data distribution between the minority classes and the majority classes.

Since the original intrusion dataset cannot be directly used as input for the model, data preprocessing is required. Data preprocessing mainly includes the following two parts.

Data digitization. One-hot encoding technology is used to convert discrete non-numeric features into numeric features to solve the problem that the model can only transmit numerical data.

Data standardization. The Min-Max normalization method is used to normalize the data to [0,1] interval, to improve the convergence speed of the model.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{12}$$

where, $x'$ is the normalized data, $x$ is the current data, $x_{min}$ is the minimum data value, and $x_{max}$ is the maximum data value.

### B. EXPERIMENTAL SETUP

In the experiment, the operating system is Ubuntu 20.04.3, the CPU is Intel Xeon Platinum 8375C, the GPU is NVIDIA GeForce RTX 4090, the programming language is Python 3.10, and the deep learning framework is TensorFlow 2.10.

Adam is chosen as the optimizer of the model. The epoch is set to 50, and the learning rate on the NSL-KDD, UNSW-NB15, and CIC-DDoS2019 datasets are set to 0.0005, 0.0015, and 0.0010.

The evaluation indicators used in this paper include accuracy (Acc), detection rate (DR), and false positive rate (FPR). The accuracy can evaluate the ability of the model that accurately classify the normal and attack data, the detection rate reflects the ability of the model to detect the attack class data, and the false positive rate is used to evaluate the misclassification ability of the model.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$DR = \frac{TP}{TP + FN} \tag{14}$$

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

where True Positive (TP) denotes the number of data detected as attack data and the detection result is correct; True Negative (TN) denotes the number of data detected as attack data but its detection result is wrong and the data is actually normal data; False Positive (FP) denotes the number of data detected as normal data and the detection result is correct; and False Negative (FN) denotes the number of data detected as normal data but its detection result is wrong and the data is actually the attack data.

### C. EXPERIMENTAL EVALUATION

In order to verify the multi-classification performance of the CNN-BiLSTM-Attention model for network intrusion detection, the Stratified K-Fold Cross Validation method is used in this paper to conduct multi-classification experiments on the NSL-KDD, UNSW-NB15, and CIC-DDoS2019 datasets respectively, with the range of K-values is 2 to 10. The multi-classification results are as follows. As shown in Table 4, the best accuracy of the NSL-KDD dataset at K = 10 is 99.79%, the highest detection rate is 99.83%, and the lowest false positive rate is 0.17%. Similarly, as shown in Table 5, the optimal performance of the UNSW-NB15 dataset is also achieved at K = 10. As shown in Table 6, the best accuracy and highest detection rate of the CIC-DDoS2019 dataset are obtained at K = 8, which are 99.84% and 99.99% respectively. For the three datasets, the best results of the model proposed in this paper almost all occur at K = 10. This is because as the K-value increases, the number of training samples for various attack categories also increases, thereby improving the classification performance of the model. At the same time, the CNN-BiLSTM-Attention model can achieve both high average accuracy and low average false positive rates, which further proves the feasibility and effectiveness of the CNN-BiLSTM-Attention model.

**TABLE 4. Multi-classification results on NSL-KDD dataset.**

| K | Acc(%) | DR(%) | FPR(%) |
|---|---|---|---|
| 2 | 99.43 | 99.50 | 0.57 |
| 4 | 99.64 | 99.75 | 0.41 |
| 6 | 99.69 | 99.76 | 0.26 |
| 8 | 99.72 | 99.73 | 0.28 |
| 10 | **99.79** | **99.83** | **0.17** |
| Average | 99.65 | 99.71 | 0.34 |

**TABLE 5. Multi-classification results on UNSW-NB15 dataset.**

| K | Acc(%) | DR(%) | FPR(%) |
|---|---|---|---|
| 2 | 81.55 | 93.72 | 8.03 |
| 4 | 85.49 | 96.51 | 4.72 |
| 6 | 87.32 | 97.41 | 2.65 |
| 8 | 88.36 | 98.15 | 2.26 |
| 10 | **88.84** | **98.52** | **1.82** |
| Average | 86.31 | 96.86 | 3.90 |

In order to verify intuitively the detection ability of the CNN-BiLSTM-Attention model for each attack in NSL-KDD dataset, this paper conducts experiments using the K-Fold Cross Validation method. Figure 5 visually shows the trend of the detection rate of each category with the change of the K-value in NSL-KDD datasets. In Figure 5, the detection

**TABLE 6.** Multi-classification results on CIC-DDoS2019 dataset.

| K | Acc(%) | DR(%) | FPR(%) |
|---|---|---|---|
| 2 | 99.28 | 99.92 | 0.15 |
| 4 | 99.54 | 99.97 | 0.07 |
| 6 | 99.79 | 99.98 | 0.02 |
| 8 | **99.84** | **99.99** | 0.03 |
| 10 | **99.84** | **99.99** | **0.00** |
| Average | 99.66 | 99.97 | 0.05 |

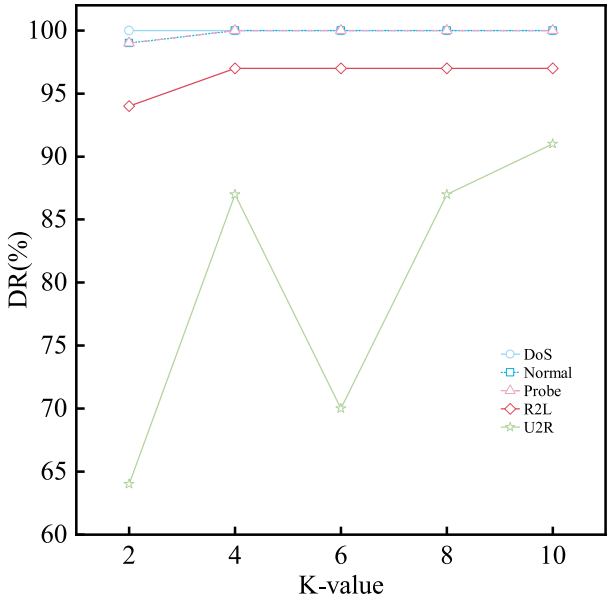


**FIGURE 5.** Detection rate for every class on NSL-KDD dataset.

rate of the CNN-BiLSTM-Attention model for DoS, Probe, and R2L attacks is all over 95%, and the optimal detection rate for U2R attacks is increased to over 90%. The results show that the CNN-BiLSTM-Attention model can accurately identify various attack categories in the NSL-KDD dataset.

To verify the detection ability of the CNN-BiLSTM-Attention model for each attack in the UNSW-NB15 dataset, this paper conducts experiments using the K-Fold Cross Validation method. Figure 6 visually shows the trend of the detection rate of each category with the change of the K-value in UNSW-NB15 datasets. In Figure 6, the detection rate of the CNN-BiLSTM-Attention model for most attack classes can all reach 80%, such as Exploits, Fuzzers, Generic, Reconnaissance, and Shellcode, while the detection rate for the Worms attack can also reach 60%. The results indicate that the CNN-BiLSTM-Attention model can accurately detect most attack categories in the UNSW-NB15 dataset.

To verify the detection ability of the CNN-BiLSTM-Attention model for each attack in the CIC-DDoS2019 dataset, this paper conducts experiments using the K-Fold Cross Validation method. Figure 7 visually shows the trend of the detection rate of each category with the change of the K-value in the CIC-DDoS2019 datasets. As shown in

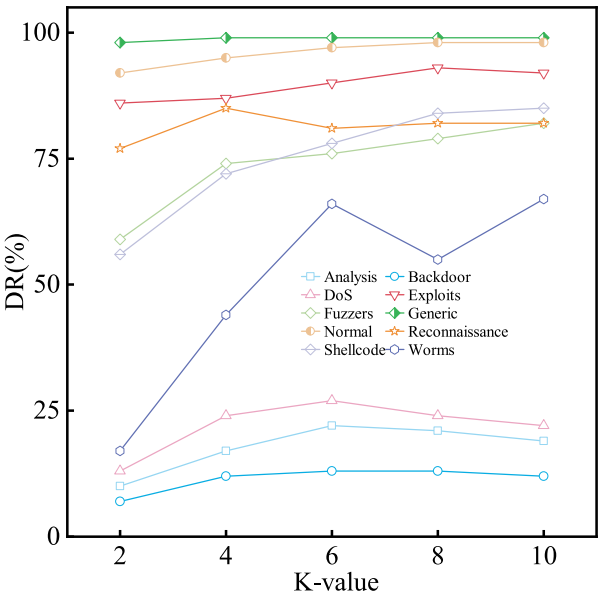


**FIGURE 6.** Detection rate for every class on UNSW-NB15 dataset.

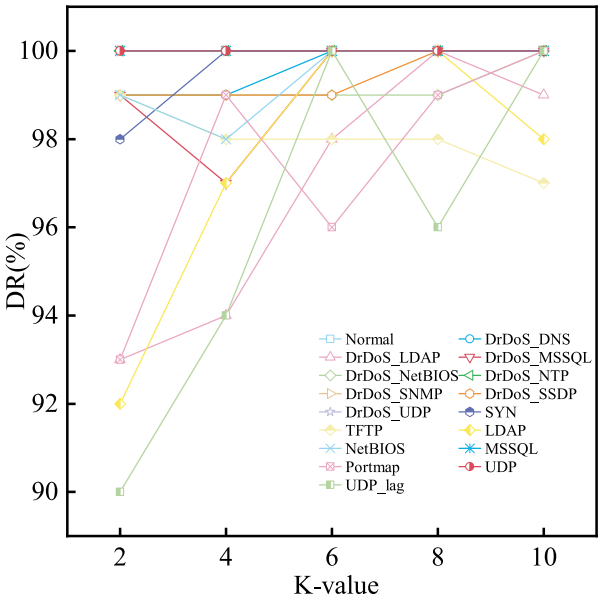


**FIGURE 7.** Detection rate for every class on CIC-DDoS2019 dataset.

Figure 7, the detection rate of the CNN-BiLSTM-Attention model for various attack categories can reach over 90%, and it still has a high detection rate for the minority attack categories. For example, the detection rate for LDAP attacks can reach 100%, while the detection rate for UDP lag attacks can still reach 100%. The results indicate that the CNN-BiLSTM-Attention model can accurately detect various types of attacks in the CIC-DDoS2019 dataset.

### D. COMPARATIVE EXPERIMENTS

To verify the superiority of the CNN-BiLSTM-Attention model on the UNSW-NB15 dataset, this paper compares

the CNN-BiLSTM-Attention model with other intrusion detection models, such as Random Forest, SVM, AdBoost, CNN-LSTM, and CNN-BiLSTM. As shown in Table 7, compared with traditional machine learning models, the CNN-BiLSTM-Attention model has achieved better results in various performance indicators, which further illustrates the obvious advantages of deep learning in the field of intrusion detection. Compared with the CNN-LSTM model, the accuracy and detection rate of the CNN-BiLSTM-Attention model increased by 6.63% and 16.10% respectively, while the false positive rate decreased by 0.34%; compared with the CNN-BiLSTM model, the accuracy and detection rate of the CNN-BiLSTM Attention model increased by 5.74% and 6.00% respectively, while the false positive rate decreased by 5.81%. The reason is that the CNN-BiLSTM-Attention model not only combines the advantages of CNN and BiLSTM but also extracts more important spatiotemporal features through the attention mechanism, which results in better classification performance. Moreover, the CNN-BiLSTM-Attention model solves the problem of low detection performance of most intrusion detection models for minority class samples, thereby improving the overall detection accuracy of the model.

**TABLE 7. Multi-classification results on UNSW-NB15.**

| Model | Acc(%) | DR(%) | FPR(%) |
|---|---|---|---|
| AdaBoost | 73.19 | 91.13 | 22.11 |
| SVM | 74.80 | 83.71 | 7.73 |
| RF | 84.59 | 92.24 | 3.01 |
| CNN-LSTM | 82.20 | 82.41 | 2.22 |
| CNN-BiLSTM | 82.09 | 92.51 | 6.09 |
| CNN-BiLSTM-Attention | **88.83** | **98.51** | **1.88** |

To verify the superiority of the CNN-BiLSTM-Attention model on the CIC-DDoS2019 dataset, this paper compares the CNN-BiLSTM-Attention model with some intrusion detection models, such as CANET, SCDAE-CNN-BiLSTM-Attention, Random Forest, AdaBoost, DT, and KNN, as shown in Table 8. N/A indicates no such data. Compared with CANET, accuracy increased by 0.26%, and false positive rate decreased by 0.05%. Although the detection rate only increased by 0.02%, it will also contribute to future research. Compared with the SCDAE-CNN-BiLSTM-Attention model, the accuracy of the CNN-BiLSTM Attention model has increased by 6.58%. The reason may be that in the SCDAE-CNN-BiLSTM-Attention model, SCDAE is used to apply Feature denoise to the original traffic data, which leads to the loss of important network traffic information and further affects the classification accuracy of the CNN-BiLSTM-Attention model for traffic data. Meanwhile, compared to the SCDAE-CNN-BiLSTM-Attention model, the detection rate of the CNN-BiLSTM-Attention model is improved by 11.76%, because EQL v2 is used as the loss function of the model to improve the detection rate for minority class data, which further improves the overall detection rate of the CNN-BiLSTM-Attention model. For several common machine learning models, the CNN-BiLSTM-Attention model has significant improvements in accuracy, detection rate, and false positive rate. The comparative experimental results show that the CNN-BiLSTM-Attention model is advanced on the CIC-DDoS2019 dataset.

**TABLE 8. Multi-classification results on CICI-DDoS2019.**

| Model | Acc(%) | DR(%) | FPR(%) |
|---|---|---|---|
| RF | 99.28 | 99.85 | 0.10 |
| AdaBoost | 54.56 | 48.40 | 12.38 |
| KNN | 90.66 | 99.77 | 0.47 |
| DT | 99.75 | 99.91 | 0.54 |
| CANET | 99.58 | 99.97 | 0.06 |
| SCDAE-CNN-BiLSTM-Attention | 93.26 | 88.23 | N/A |
| CNN-BiLSTM-Attention | **99.84** | **99.99** | **0.01** |

To verify the superiority of the CNN-BiLSTM-Attention model on the NSL-KDD dataset, this paper compares the CNN-BiLSTM-Attention model with other intrusion detection models, such as Pelican [60], Lunet [61], CNN-LSTM, CNN-BiLSTM, CANET, and SCDAE-CNN-BiLSTM-Attention. N/A indicates no such data. It can be seen from Table 9 that the CNN-BiLSTM-Attention model outperforms other models in the accuracy, detection rate, and false positive rate, thereby verifying the superiority of the CNN-BiLSTM-Attention model.

**TABLE 9. Multi-classification results on NSL-KDD.**

| Model | Acc(%) | DR(%) | FPR(%) |
|---|---|---|---|
| Pelican | 99.21 | 99.13 | 0.65 |
| Lunet | 99.14 | 99.02 | 0.61 |
| CNN-BiLSTM | 99.22 | 99.88 | 0.43 |
| CANET | 99.77 | 99.72 | 0.18 |
| SCDAE-CNN-BiLSTM-Attention | 93.26 | 94.26 | N/A |
| CNN-BiLSTM-Attention | **99.79** | **99.83** | **0.17** |

To fully verify the detection performance of the CNN-BiLSTM-Attention model for each class, this paper calculates the detection rate for each category in the UNSW-NB15 dataset, and the CNN-BiLSTM-Attention model is compared with the CNN-BiLSTM model and the CANET model, the results are shown in Table 10. Compared to the CANET model, the CNN-BiLSTM-Attention model improves the detection rate for all categories except DoS and Worms. Among them, the minority attack category Analysis improves by 4%, the minority attack category Backdoor improves by 6%, and the minority attack category Shellcode improves by 5%. Compared with the CNN-BiLSTM model, except for Worms, the CNN-BiLSTM-Attention model significantly improves the detection rate for all minority categories, with Analysis improving by 19%, Backdoor improving by 13%, and Shellcode improving by 92%. The reason why the detection performance of Worms does not improve is that the CNN-BiLSTM model oversampled it. In summary, the CNN-BiLSTM-Attention model not only improves the overall

**TABLE 10. Comparison of detection rate for each class on models.**

| Class | CNN-BiLSTM-Attention | CANET | CNN-BiLSTM | Number |
|---|---|---|---|---|
| Normal | 99% | 99% | 95% | 9300 |
| Generic | 99% | 99% | 98% | 5887 |
| Exploits | **92%** | 85% | 92% | 4452 |
| Fuzzers | **85%** | 84% | 54% | 2425 |
| DoS | 27% | **46%** | 5% | 1635 |
| Reconnaissance | **85%** | 84% | 73% | 1399 |
| Analysis | **25%** | 21% | 6% | 268 |
| Backdoor | **20%** | 14% | 7% | 232 |
| Shellcode | **92%** | 87% | 0% | 151 |
| Worms | 61% | 89% | **100%** | 18 |

**TABLE 11. Comparison of the loss function in the UNSW-NB15 dataset for each class detection rate.**

| | EQL v2 | CE | Number |
|---|---|---|---|
| Normal | **99%** | 97% | 9300 |
| Generic | **99%** | 96% | 5887 |
| Exploits | **92%** | 91% | 4452 |
| Fuzzers | **84%** | 80% | 2425 |
| DoS | 27% | **33%** | 1635 |
| Reconnaissance | **84%** | 82% | 1399 |
| Analysis | **25%** | 22% | 268 |
| Backdoor | **20%** | 14% | 232 |
| Shellcode | **92%** | 90% | 151 |
| Worms | 61% | **83%** | 18 |

detection rate but also significantly improves the detection rate for the minority attack classes.

To verify the classification performance of intrusion detection models using different loss functions for minority class data, this paper compares CNN-BiLSTM-Attention model with different loss functions, EQL v2 and CE in the UNSW-NB15 dataset, as shown in Table 11.

In Table 11, CNN-BiLSTM-Attention model with EQL v2 is more obviously advantageous than CNN-BiLSTM-Attention model with CE for the detection rate of the minority class Analysis, Backdoor, and Shellcode. Among them, for the detection rate of Backdoor, CNN-BiLSTM-Attention model with EQL v2 is 6% higher than CNN-BiLSTM-Attention model with CE. Therefore, it can be concluded that CNN-BiLSTM-Attention model using EQL v2 can effectively improve the detection performance for the minority classes, thereby solving the class imbalance problem in the dataset.

## IV. CONCLUSION

In response to the low classification accuracy of traditional intrusion detection models, this paper proposes CNN-BiLSTM-Attention model. Firstly, CNN can automatically extract the spatial features of the intrusion data; Secondly, BiLSTM is used to learn bidirectional temporal features for the intrusion data, and then the attention mechanism is introduced after BiLSTM to assign weights to different temporal features to strengthen the influence of important features on the classification results. At the same time, EQL v2 is selected as the loss function of the CNN-BiLSTM-Attention model, so that the model pays more attention to the minority class data during the training process, furthermore enhancing the recognition accuracy of the model for the minority class data. The experimental results are as follows.

Multi-classification performance experiments are conducted on three datasets, which achieve high accuracy and low false positive rate, thereby verifying the effectiveness of the CNN-BiLSTM-Attention model on large-scale network intrusion data; The result of comparison experiment show that CNN-BiLSTM-Attention model with EQL v2 has a higher detection rate in identifying the minority attack class data, which then proves that the EQL v2 can solve the class imbalance problem existing in the intrusion dataset.

CNN-BiLSTM-Attention model has achieved good detection results, but it has certain limitations. The model relies on labeled data during intrusion detection, and when labeled data is scarce, it can seriously affect the detection performance of the model. In future research, unsupervised or semi-supervised intrusion detection methods can be used to improve the recognition ability of new unknown attack data.

## REFERENCES

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, Dec. 2019.

[2] H. Chindove and D. Brown, "Adaptive machine learning based network intrusion detection," in *Proc. Int. Conf. Artif. Intell. Appl.*, Dec. 2021, pp. 1–6.

[3] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives," in *Proc. IEEE 3rd Int. Conf. Comput., Commun. Secur. (ICCCS)*, Oct. 2018, pp. 1–8.

[4] T. Liu, X. Zhu, W. Pedrycz, and Z. Li, "A design of information granule-based under-sampling method in imbalanced data classification," *Soft Comput.*, vol. 24, no. 22, pp. 17333–17347, Nov. 2020.

[5] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107315.

[6] J. Liu, Y. Gao, and F. Hu, "A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM," *Comput. Secur.*, vol. 106, Jul. 2021, Art. no. 102289.

[7] Y. Lu, "Intrusion detection classification method based on generative adversarial networks," in *Proc. 3rd Int. Conf. Frontiers Electron., Inf. Comput. Technol. (ICFEICT)*, May 2023, pp. 344–349.

[8] S. S. Khan and A. B. Mailewa, "Detecting network transmission anomalies using autoencoders-SVM neural network on multi-class NSL-KDD dataset," in *Proc. IEEE 13th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Mar. 2023, pp. 835–843.

[9] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Feature selection and classification in multiple class datasets: An application to KDD cup 99 dataset," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5947–5957, May 2011.

[10] V. Kumar, A. K. Das, and D. Sinha, "Statistical analysis of the UNSW-NB15 dataset for intrusion detection," in *Proc. Comput. Intell. Pattern Recognit. (CIPR)*. Singapore: Springer, 2020, pp. 279–294.

[11] Q. Xu, S. Lu, W. Jia, and C. Jiang, "Imbalanced fault diagnosis of rotating machinery via multi-domain feature extraction and cost-sensitive learning," *J. Intell. Manuf.*, vol. 31, no. 6, pp. 1467–1481, Aug. 2020.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[13] H. Peng, C. Wu, and Y. Xiao, "CBF-IDS: Addressing class imbalance using CNN-BiLSTM with focal loss in network intrusion detection system," *Appl. Sci.*, vol. 13, no. 21, p. 11629, Oct. 2023.

[14] M. Mulyanto, M. Faisal, S. W. Prakosa, and J.-S. Leu, "Effectiveness of focal loss for minority classification in network intrusion detection systems," *Symmetry*, vol. 13, no. 1, pp. 4–16, Dec. 2020.

[15] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 5, pp. 649–659, Sep. 2008.

[16] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE Access*, vol. 6, pp. 33789–33795, 2018.

[17] P. Tao, Z. Sun, and Z. Sun, "An improved intrusion detection algorithm based on GA and SVM," *IEEE Access*, vol. 6, pp. 13624–13631, 2018.

[18] W. Hu, J. Gao, Y. Wang, O. Wu, and S. Maybank, "Online AdaBoost-based parameterized methods for dynamic distributed network intrusion detection," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 66–82, Jan. 2014.

[19] Q. Wang and X. Wei, "The detection of network intrusion based on improved AdaBoost algorithm," in *Proc. 4th Int. Conf. Cryptography, Secur. Privacy*, Jan. 2020, pp. 84–88.

[20] J. R. Quinlan, "Simplifying decision trees," *Int. J. Man-Mach. Stud.*, vol. 27, no. 3, pp. 221–234, Sep. 1987.

[21] S. Sahu and B. M. Mehtre, "Network intrusion detection system using J48 decision tree," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Aug. 2015, pp. 2023–2026.

[22] H. Benaddi, K. Ibrahimi, and A. Benslimane, "Improving the intrusion detection system for NSL-KDD dataset based on PCA-fuzzy clustering-KNN," in *Proc. 6th Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2018, pp. 1–6.

[23] K. Atefi, H. Hashim, and M. Kassim, "Anomaly analysis for the classification purpose of intrusion detection system with K-Nearest neighbors and deep neural network," in *Proc. IEEE 7th Conf. Syst., Process Control (ICSPC)*, Dec. 2019, pp. 269–274.

[24] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M. L. Shyu, S. C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, 2018.

[25] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018.

[26] K. B. Lee and H. S. Shin, "An application of a deep learning algorithm for automatic detection of unexpected accidents under bad CCTV monitoring conditions in tunnels," in *Proc. Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. (Deep-ML)*, Aug. 2019, pp. 7–11.

[27] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1222–1228.

[28] H. Yang and F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," *IEEE Access*, vol. 7, pp. 64366–64374, 2019.

[29] 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.

[30] M. A. Khan, "HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system," *Processes*, vol. 9, no. 5, p. 8341, 2021.

[31] S. A. Althubiti, E. M. Jones, and K. Roy, "LSTM for anomaly-based network intrusion detection," *Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2018, pp. 1–3.

[32] N. Gupta, V. Jindal, and P. Bedi, "LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108076.

[33] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115524.

[34] M. Tan, A. Iacovazzi, N. M. Cheung, and Y. Elovici, "A neural attention model for real-time network intrusion detection," in *Proc. IEEE 44th Conf. Local Comput. Netw. (LCN)*, Oct. 2019, pp. 291–299.

[35] J. Zhang, X. Zhang, Z. Liu, F. Fu, Y. Jiao, and F. Xu, "A network intrusion detection model based on BiLSTM with multi-head attention mechanism," *Electronics*, vol. 12, no. 19, p. 4170, Oct. 2023.

[36] R. Singh and G. Srivastav, "Novel framework for anomaly detection using machine learning technique on CIC-IDS2017 dataset," in *Proc. Int. Conf. Technol. Advancements Innov. (ICTAI)*, Nov. 2021, pp. 632–636.

[37] C. Yue, L. Wang, D. Wang, R. Duo, and X. Nie, "An ensemble intrusion detection method for train Ethernet consist network based on CNN and RNN," *IEEE Access*, vol. 9, pp. 59527–59539, 2021.

[38] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022.

[39] N. Singh, D. Virmani, and X.-Z. Gao, "A fuzzy logic-based method to avert intrusions in wireless sensor networks using WSN-DS dataset," *Int. J. Comput. Intell. Appl.*, vol. 19, no. 3, Sep. 2020, Art. no. 2050018.

[40] J. Sinha and M. Manollas, "Efficient deep CNN-BiLSTM model for network intrusion detection," in *Proc. 3rd Int. Conf. Artif. Intell. Pattern Recognit.*, Jun. 2020, pp. 223–231.

[41] K. Ren, S. Yuan, C. Zhang, Y. Shi, and Z. Huang, "CANET: A hierarchical CNN-attention model for network intrusion detection," *Comput. Commun.*, vol. 205, pp. 170–181, May 2023.

[42] H. Xu, L. Sun, G. Fan, W. Li, and G. Kuang, "A hierarchical intrusion detection model combining multiple deep learning models with attention mechanism," *IEEE Access*, vol. 11, pp. 66212–66226, 2023.

[43] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 big data," *J. Big Data*, vol. 7, no. 1, pp. 1–19, Dec. 2020.

[44] D. Akgun, S. Hizal, and U. Cavusoglu, "A new DDoS attacks intrusion detection model based on deep learning for cybersecurity," *Comput. Secur.*, vol. 118, Jul. 2022, Art. no. 102748.

[45] J. Zhao, Y. Liu, Q. Zhang, and X. Zheng, "CNN-AttBiLSTM mechanism: A DDoS attack detection method based on attention mechanism and CNN-BiLSTM," *IEEE Access*, vol. 11, pp. 136308–136317, 2023.

[46] F. Teng, Y. Song, and X. Guo, "Attention-TCN-BiGRU: An air target combat intention recognition model," *Mathematics*, vol. 9, no. 19, p. 2412, Sep. 2021.

[47] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli, "A new intrusion detection system based on fast learning network and particle swarm optimization," *IEEE Access*, vol. 6, pp. 20255–20261, 2018.

[48] A. Shah, M. Shah, A. Pandya, R. Sushra, R. Sushra, M. Mehta, K. Patel, and K. Patel, "A comprehensive study on skin cancer detection using artificial neural network (ANN) and convolutional neural network (CNN)," *Clin. eHealth*, vol. 6, pp. 76–84, Dec. 2023.

[49] H. Liu, B. Lang, M. Liu, and H. Yan, "CNN and RNN based payload classification methods for attack detection," *Knowl.-Based Syst.*, vol. 163, pp. 332–341, Jan. 2019.

[50] M. Sheikhan, Z. Jadidi, and A. Farrokhi, "Intrusion detection using reduced-size RNN based on feature grouping," *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1185–1190, Sep. 2012.

[51] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters," *Proc. Comput. Sci.*, vol. 125, pp. 676–682, Jan. 2018.

[52] M. Khan, M. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry*, vol. 11, no. 4, p. 583, Apr. 2019.

[53] M. Imani, "Alzheimer's diseases diagnosis using fusion of high informative BiLSTM and CNN features of EEG signal," *Biomed. Signal Process. Control*, vol. 86, Sep. 2023, Art. no. 105298.

[54] K. Sethi, Y. V. Madhav, R. Kumar, and P. Bera, "Attention based multi-agent intrusion detection systems using reinforcement learning," *J. Inf. Secur. Appl.*, vol. 61, Sep. 2021, Art. no. 102923.
[55] A. R. Javed, S. U. Rehman, M. U. Khan, M. Alazab, and T. G. Reddy, "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1456–1466, Apr. 2021.
[56] A. S. Shukla and R. Maurya, "Entropy-based anomaly detection in a network," *Wireless Pers. Commun.*, vol. 99, no. 4, pp. 1487–1501, Apr. 2018.
[57] W. Han, J. Xue, and H. Yan, "Detecting anomalous traffic in the controlled network based on cross entropy and support vector machine," *IET Inf. Secur.*, vol. 13, no. 2, pp. 109–116, Mar. 2019.
[58] X. Kan, Y. Fan, Z. Fang, L. Cao, N. N. Xiong, D. Yang, and X. Li, "A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network," *Inf. Sci.*, vol. 568, pp. 147–162, Aug. 2021.
[59] J. Tan, X. Lu, G. Zhang, C. Yin, and Q. Li, "Equalization loss v2: A new gradient balance approach for long-tailed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1685–1694.
[60] P. Wu, H. Guo, and N. Moustafa, "Pelican: A deep residual network for network intrusion detection," in *Proc. 50th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2020, pp. 55–62.
[61] P. Wu and H. Guo, "LuNet: A deep neural network for network intrusion detection," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2019, pp. 617–624.

**WEI DAI** received the Ph.D. degree in safety administration from Liaoning Technical University, in 2018. She is currently a Full Lecturer with Liaoning Technical University. Her representative work has been published in high-level international/national and journals/conferences, such as *China Safety Science Journal*, *Chinese Journal of Sensors and Actuators*, *Journal of Liaoning Technical University* (Natural Science), and *Journal of Nanoelectronics and Optoelectronics*. She has gained one science and technology advancement medal and obtained a state patent. Her research interest includes the Internet of Things security situational awareness in open pit mines.

**XINHUI LI** was born in Huludao, Liaoning, China, in 1997. She received the bachelor's degree in communication engineering from Dalian Ocean University. She is currently pursuing the master's degree with the School of Electronics and Information Engineering, Liaoning Technical University. Her research interest includes the Internet of Things security situational awareness in open pit mines.

**WENXIN JI** was born in Harbin, Heilongjiang, China, in 2000. She received the bachelor's degree in communication engineering from Changshu Institute of Technology. She is currently pursuing the master's degree with the School of Electronics and Information Engineering, Liaoning Technical University. Her research interest includes the Internet of Things security situational awareness in open pit mines.

**SICHENG HE** was born in Hengshui, Hebei, China, in 1999. He received the bachelor's degree in communication engineering from Liaoning Technical University, where he is currently pursuing the master's degree with the School of Electronics and Information Engineering. His research interest includes information and signal processing.