Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.068372

ARTICLE

A Dual-Attention CNN-BiLSTM Model for Network Intrusion Detection

Zheng Zhang[1,2], **Jie Hao**[2], **Liquan Chen**[1,*], **Tianhao Hou**[2] **and Yanan Liu**[2]

[1]School of Cyber Science and Engineering, Southeast University, Nanjing, 211199, China
[2]School of Network Security, Jinling Institute of Technology, Nanjing, 211169, China

*Corresponding Author: Liquan Chen. Email: Lqchen@seu.edu.cn

Received: 27 May 2025; Accepted: 22 August 2025; Published: 10 November 2025

ABSTRACT: With the increasing severity of network security threats, Network Intrusion Detection (NID) has become a key technology to ensure network security. To address the problem of low detection rate of traditional intrusion detection models, this paper proposes a Dual-Attention model for NID, which combines Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) to design two modules: the FocusConV and the TempoNet module. The FocusConV module, which automatically adjusts and weights CNN extracted local features, focuses on local features that are more important for intrusion detection. The TempoNet module focuses on global information, identifies more important features in time steps or sequences, and filters and weights the information globally to further improve the accuracy and robustness of NID. Meanwhile, in order to solve the class imbalance problem in the dataset, the EQL v2 method is used to compute the class weights of each class and to use them in the loss computation, which optimizes the performance of the model on the class imbalance problem. Extensive experiments were conducted on the NSL-KDD, UNSW-NB15, and CIC-DDos2019 datasets, achieving average accuracy rates of 99.66%, 87.47%, and 99.39%, respectively, demonstrating excellent detection accuracy and robustness. The model also improves the detection performance of minority classes in the datasets. On the UNSW-NB15 dataset, the detection rates for Analysis, Exploits, and Shellcode attacks increased by 7%, 7%, and 10%, respectively, demonstrating the Dual-Attention CNN-BiLSTM model's excellent performance in NID.

KEYWORDS: Network intrusion detection; class imbalance problem; deep learning

1 Introduction

With the rapid development of information technology, the network has become an indispensable part of people's lives and work, and a large amount of data is transmitted and stored on the network. However, network security threats are also increasing, posing serious risks to personal privacy, corporate assets, and national security. The current network traffic intrusion detection is facing a surge in the scale of attacks, with the global average weekly attacks reaching 1925 incidents. In addition, new and unknown attacks account for more than 30% of threats. Technological applications, such as machine learning and deep learning, are mainstream. However, the data imbalance problem is significant, resulting in low detection accuracy of existing models. In terms of industry demand, the cloud security market is increasing by 25% annually, and China accounts for 9.5% of the total, which is a promising prospect for the future.

NID can identify potentially malicious behaviors or security policy violations [1,2], and combining deep learning with intrusion detection has become a hot research topic in the domain of network security [3–5].

Early machine learning techniques, especially supervised learning (e.g. Random Forest [6], Decision Trees [7], Support Vector Machines SVM [8], k Nearest Neighbour KNN [9], etc.) can be used to automate the identification of normal and abnormal behaviors by learning features from historical data. This reduces the need for manual feature extraction, thus improving the efficiency and accuracy of NID [10]. With the surge in network traffic and the diversification of attack types, early machine learning techniques are increasingly showing their limitations in detecting large-scale network intrusions both efficiently and accurately. In contrast, deep learning techniques are more suitable for processing large-scale intrusion data due to their powerful automatic feature extraction capabilities, thus improving the effectiveness of NID [11]. However, a single neural network model may suffer from poor generalization ability and overfitting in NID. Different underlying models often have their advantages when dealing with specific layers of data or specific types of features, and combining multiple models can be more effective in deal with diverse network attacks.

The class imbalance problem [12] is also an important factor that affects the detection rate of intrusion detection models, which struggle to cope with attacks from a few classes due to the imbalance in the distribution of the classes in the intrusion data. There are two main approaches to address the class imbalance problem: data-level methods and cost-sensitive learning methods. Data-level methods cope with the class imbalance problem by changing the dataset structure, which typically include oversampling [13] and undersampling [14]. Relying solely on data-level methods for intrusion detection can alter the original feature distribution of traffic data, rendering the features less accurate and reliable. This makes it difficult for the model to correctly identify previously unknown attack patterns hidden in the data, thereby significantly increasing the risk of under-detecting such attacks. Cost-sensitive learning methods [15] improve the recognition ability of minority classes by assigning a higher cost to minority classes, which makes the model pay more attention to minority classes during the training process. Thus, the cost-sensitive learning method has the advantages of high efficiency and low time cost when dealing with unbalanced data.

Due to the problem of low detection rate of intrusion detection models, this paper proposes a Dual-Attention CNN-BiLSTM model for NID, which combines the advantages of CNN, BiLSTM, and attention mechanism, and designs FocusConv and TempoNet modules to automatically extract local and global features for dual-attention hierarchical feature fusion. Meanwhile, to solve the class imbalance problem in the dataset, the EQL v2 cost-sensitive method [16–18] is used, which can optimize the class imbalance problem. Experimental results show that the Dual-Attention CNN-BiLSTM model achieves good classification results on the datasets and improves the detection rate of minority class attacks. The main contributions of this paper can be summarized as:

- Combining CNN [19,20] and the attention mechanism [21], the FocusConV module is proposed, which can automatically adjust and weight the local features extracted by CNN to highlight the more important local information, suppress those irrelevant or redundant local features, and help CNN to focus more on those local features that are meaningful for intrusion detection.
- Combining BiLSTM [22] and the attention mechanism, the TempoNet module is proposed to focus on the global information and help the model determine which features in the time steps or sequences are more important for the final intrusion detection task to filter and weigh the information globally to further enhance the accuracy and robustness of NID.
- The performance of the model on the class imbalance problem is optimized by using the EQL v2 method, which calculates the category weights for each category. In this paper, experiments are conducted on these public datasets, NSL-KDD [23,24], UNSW-NB15 [25,26], and CIC-DDos2019 [27,28]. The model achieves the highest accuracy of 99.72%, the highest detection rate of 99.78%, and the lowest false-positive rate of 0.25% for multiclassing on the NSL-KDD dataset. The highest accuracy of multiple subcategories on the UNSW-NB15 dataset reaches 89.07%, the highest detection rate reaches 98.90%,

and the lowest false-positive rate reaches 1.46%. On the CIC-DDos2019 dataset, the highest accuracy rate for multi-classification reached 99.74%, the highest detection rate reached 100%, and the lowest false positive rate reached 0.03%. In addition, there is a significant improvement in the detection rate for a few categories of attacks, especially for Analysis and Exploits attacks, both of which see a 7% improvement, and Shellcode attacks show a 10% improvement.

The rest of the paper is organized as follows. Section 3 elaborates on the Dual-Attention Guided Hierarchical Feature Fusion method. Section 4 presents the detailed experimental results and analyses. Section 5 presents the conclusion of the work and areas for improvement.

2 Related Work

Early machine learning techniques have been widely used in the field of NID with their unique advantages. For example, supervised learning algorithms work by mining features from a large amount of historical data and constructing classification models to distinguish between normal network behaviors and intrusion behaviors. They reduce the workload of manual feature extraction to some extent, improve detection efficiency, and make NID shift from relying on manual experience judgment to automated detection. However, the increasingly complex network traffic environment makes feature extraction difficult, and these methods difficult to extract effective features from the vast and complex data accurately.

Insufficient feature extraction often leads to low model detection accuracy, which has driven widespread research into deep learning-based methods in the field of NID. The application of the Attention Mechanism in deep learning has yielded significant results. In NID, the importance of each feature in network traffic or logs can change depending on the time or context. The attention mechanism adaptively assigns higher weights to crucial features and reduces the influence of irrelevant or noisy ones, thereby enhancing the accuracy and robustness of the model. Literature [29] proposes a deep learning model based on CNN for detecting Denial of Service (DoS) attacks to address the difficulty of detecting advanced attacks with traditional NIDs. The model processes KDDCup99 [30] and CSE-CIC-IDS 2018 [31] data by converting symbolic data to numerical data, scaling features, and converting them into RGB or grayscale images as CNN model input. Experiments show its performance significantly outperforms that of RNN models. However, when dealing with multi-class, more complex advanced DoS attacks, its performance will decline, and its generalization ability needs to be enhanced. The main reasons for this phenomenon are: Advanced DoS attack features are more complex and diverse, with higher feature overlap between different attack types. Although CNN can extract local spatial features, it is difficult to capture the subtle differences and global features between these attacks. Second, the number of categories increases in multi-class classification, and the sample distribution is unbalanced. Insufficient samples of minority classes lead to the model's weak recognition ability for them. Literature [32] proposes a deep learning intrusion detection method (RNN-IDS) based on RNN [33], which is proven to be superior to machine learning methods in terms of detection accuracy by experimental comparison. However, the model training time is long and cannot solve the gradient explosion problem.

When handling large-scale and complex network attack data, the hybrid model can better adapt to diverse attack scenarios, improve accuracy and detection rates, and effectively mitigate the overfitting problem of single models. Literature [34] proposes a deep learning model combining a 1-dimensional CNN and a BiLSTM for NID. 1-D CNN and Max Pooling layers are used for fast spatial learning and feature extraction, and the BiLSTM layer learns temporal features. Experiments are conducted on NSL-KDD and UNSW-NB15 datasets, and the model performs better in metrics such as detection rate, false positives, and accuracy compared to other models. However, due to class imbalance in the dataset, the model has insufficient learning of the features of minority classes, relying on limited samples, which restricts its generalization ability. Meanwhile, there are differences in the complexity of features among different attack types, and the model lacks the ability to capture low-discriminability features. Furthermore, in multi-class classification, minority classes are easily dominated by the features of majority classes, making it difficult to learn their unique patterns, which affects the detection effect of such attacks. These factors together lead to the model being greatly affected by the balance of the dataset, having insufficient detection ability for some attack types, and weak generalization ability. Literature [35] proposes a NID model that combines a multi-head attention mechanism and a BiLSTM to effectively improve detection accuracy. The model consists of an Embedding Layer, Multihead Attention Mechanism, BiLSTM, and Dense Layer. The Multihead Attention Mechanism assigns weights to different features, and the BiLSTM captures long-range dependencies. Using the KDDCUP99, NSLKDD, and CICIDS2017 datasets, the model achieves accuracies of 98.29%, 95.19%, and 99.08%, respectively, with higher accuracy and F1 scores compared to other models. Due to the excessive number of normal samples in the dataset, oversampling (SMOTE algorithm) and undersampling are adopted to balance the data distribution. However, the sampling process has strong randomness, which may accidentally delete important information contained in most samples, thereby affecting the model's adaptability to complex scenarios. Literature [36] proposes a hierarchical CNN-attention network CANET for NID, which effectively solves the problems of high false alarm rates and category imbalance of existing methods. The CA block within the model consists of a CNN and an attention mechanism. The combination of the two can learn spatiotemporal features at multiple levels. Experiments show that CANET exhibits high accuracy and detection rate, low false-positive rate on different datasets, and an improved detection rate for minority class attacks. But it has a poor ability to distinguish between attack categories with similar features in the dataset. Literature [37] proposes a NID model that fuses CNN and Gated Recurrent Unit (GRU) to effectively solve the feature redundancy and sample imbalance problems of the existing models. Convolutional Block Attention Module (CBAM) is introduced to assign weights to the features by Channel Attention Module and Spatial Attention Module, extract spatial features by using the 2D convolution of CNN, and fuse CNN and GRU to comprehensively learn data features. Experiments show that high accuracy and precision are achieved on multiple datasets. However, the model suffers from a large number of parameters, long running time, and limited improvement in detection accuracy for a few samples. The main reasons for these phenomena are that the model integrates CNN, GRU, and CBAM attention mechanisms, and the superposition of parameters leads to a large overall scale. The ADRDB hybrid sampling and RFP feature selection in the preprocessing stage involve complex calculations, which further extend the training and inference time when processing large-scale datasets. The minority samples generated by ADASYN may deviate from the real distribution, the RFP feature selection may ignore the key features of minority classes, and the attention mechanism is not optimized for minority classes.

The class imbalance problem is an important factor that affects the detection rate of intrusion detection models. There are two main approaches to address the class imbalance problem: data-level methods and cost-sensitive learning methods. Data-level methods cope with the class imbalance problem by changing the structure of the dataset. Literature [38] investigated the impact of resampling techniques on the performance of Artificial Neural Network (ANN) multi-class classifiers for NID datasets and found that these techniques affect model performance in various ways. The experiments show that, in highly imbalanced datasets, oversampling and undersampling significantly improve recall, while macro precision increases or decreases. RURO and RU-SMOTE perform better at identifying fewer classes and improving macro recall. In moderately imbalanced datasets, resampling has little effect on model performance. However, purely relying on data-level methods will distort the distribution of traffic data features, amplify the risk of unknown attack leakage detection, and lead to the failure of the detection model in real attack and defense scenarios.

Cost-sensitive learning methods improve the recognition of minority classes by assigning them a higher cost, encouraging the model to focus more on these classes during training. Literature [39] proposes CostDeepIoT, a hybrid machine learning model that incorporates cost-sensitive learning and multi-task learning to effectively address class imbalance and unknown attack detection. A multi-task Support Vector Machine (SVM) classifier is integrated with a Stacked Auto-Encoder (SAE) to extract high-level features. The multi-task SVM employs a one-to-one approach, treating each binary SVM model as a separate task for learning feature weights and predicting attack types. The hinge loss function is enhanced with class-specific costs, adjusting penalty values to prioritize correct classification of minority attack classes, while the model is updated through parameter adjustment. Experiments show that the model performs well on the UNSW-NB15 and BoT-IoT datasets, outperforming in detecting minority-class and unknown attacks. Compared with data-level methods, cost-sensitive learning methods do not require adjustment of the original structure of the dataset. They not only avoid the distortion of traffic data feature distribution caused by operations such as resampling, but also directly guide the model to focus on minority class samples during the training process by assigning higher weights to minority classes in the loss function. This improves the recognition effect of minority classes while maintaining higher training efficiency and lower time costs.

In summary, to address the problem of low model detection rates caused by insufficient feature extraction and the impact of dataset imbalance, this paper proposes a Dual-Attention CNN-BiLSTM model for NID and designs FocusConv and TempoNet modules to automatically extract local and global features for hierarchical feature fusion with dual attention. Then, to solve the class imbalance problem, the EQL v2 method is used to optimize the class imbalance problem.

3 Dual-Attention Guided Hierarchical Feature Fusion

This section elaborates on the Dual-Attention Guided Hierarchical Feature Fusion. First, the overall architecture of the Dual-Attention CNN-BiLSTM model is presented, as shown in Fig. 1. Next, Section 3.1 discusses intrusion detection issues and conducts a case study. The local feature extraction mechanism is introduced in Section 3.2, followed by the global feature extraction mechanism in Section 3.3, and finally, loss optimization is described in Section 3.4.

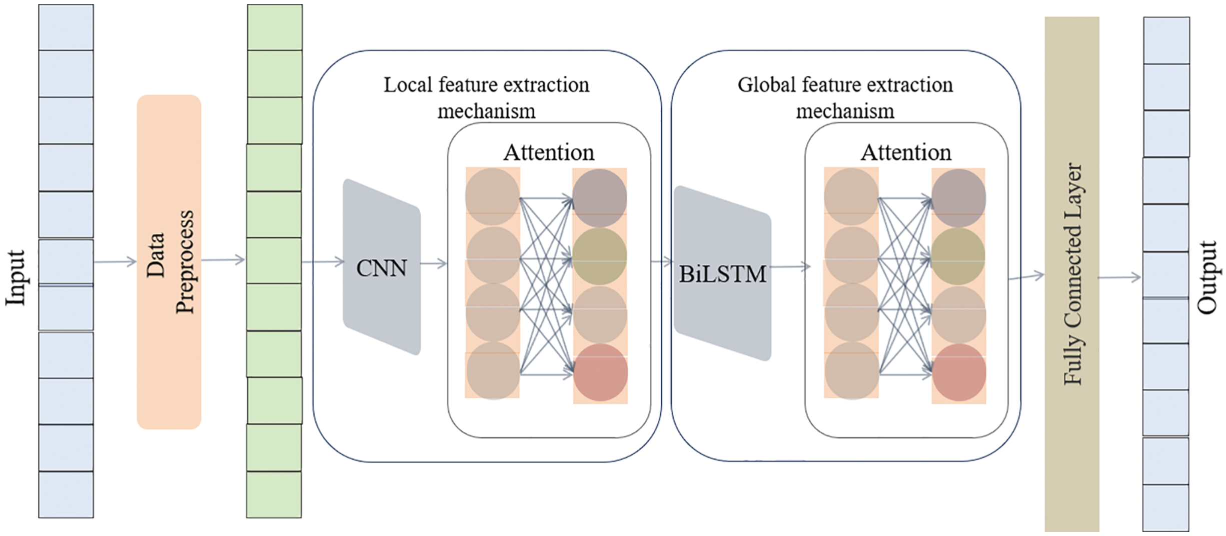

Figure 1: Overall architecture of Dual-Attention CNN-BiLSTM model

The raw data "Input" first enters the "Data Preprocess" module for preprocessing, and then feeds into the FocusConV module, where the convolutional neural network "CNN" extracts local features. The extracted features go into the module containing the "Attention" mechanism, which helps the CNN to focus more on local features that are meaningful for intrusion detection, and then into the TempoNet module. The data is processed by a BiLSTM, then passed through an Attention module that focuses on global information and helps the model determine which features in a time step or sequence are most important for the final intrusion detection task. Finally, the data is passed through the Fully Connected Layer to produce the final "Output".

3.1 Problem Description and Case Study

To clarify the design motivation of the dual-attention guided hierarchical feature fusion method, this section first describes the problem of intrusion detection and analyses a case study involving a SYN Flood DDoS attack (which includes both local packet-level and global temporal anomalies). The reasons for designing the method proposed in this paper is shown in Fig. 2.

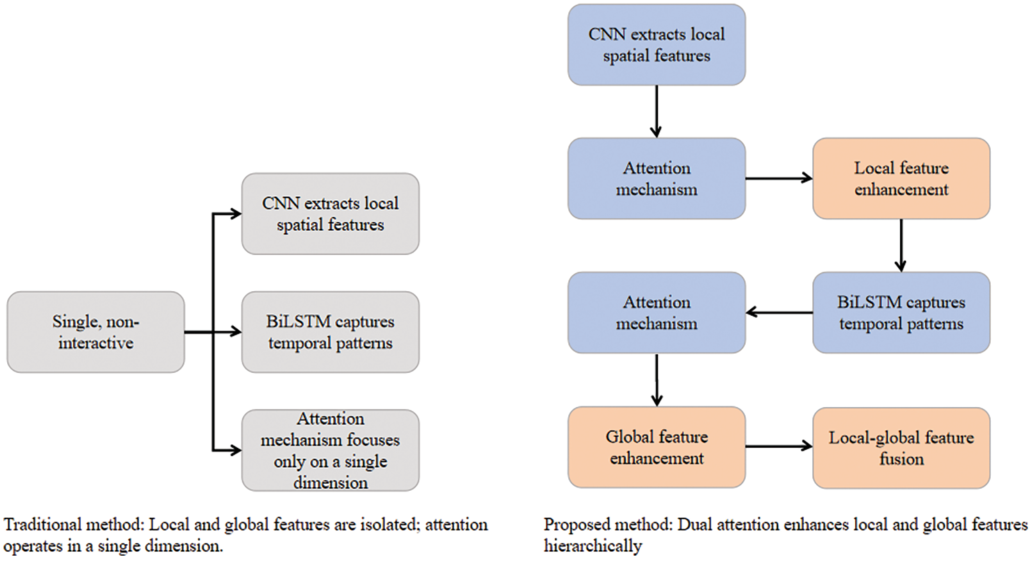
Single, non-interactive
CNN extracts local spatial features
BiLSTM captures temporal patterns
Attention mechanism focuses only on a single dimension
Traditional method: Local and global features are isolated; attention operates in a single dimension.
CNN extracts local spatial features
Attention mechanism
Local feature enhancement
BiLSTM captures temporal patterns
Attention mechanism
Global feature enhancement
Local-global feature fusion
Proposed method: Dual attention enhances local and global features hierarchically

Figure 2: The advantages of the method proposed in this paper

Network intrusion detection requires capturing two complementary types of features from traffic data. Local features: fine-grained patterns within individual packets (e.g., TCP flags, port numbers), reflecting spatial-dimensional anomalies; Global features: long and short-term temporal dependencies in traffic sequences (e.g., sudden increases in connection rates, periodic attack patterns), reflecting temporal-dimensional anomalies.

Attack scenario setup: The attacker controls a botnet and launches a SYN Flood attack on the target server (IP: 192.168.1.100, port 80). Local packet features: Each data packet has TCP flags SYN = 1 and ACK = 0, with a fixed destination port of 80 and randomly forged source IP addresses; global temporal features: Traffic increases from a normal 100 connections per second to 10,000 connections per second at time step t = 10, and continues for five consecutive time steps.

A pure CNN model can extract local SYN flag bits and other features but cannot detect the temporal surge at t = 10, mistakenly classifying the attack as a 'normal traffic peak'; A pure BiLSTM model can model temporal traffic surges but cannot distinguish between 'SYN Flood' and 'legitimate connection surges' because it ignores the local malicious flags (SYN = 1, ACK = 0); Single attention models can weight key features in a single dimension, but focusing solely on the temporal dimension (surge points) overlooks abnormal packet structures, while focusing on the spatial dimension (SYN flags) ignores the temporal criticality of surges. Therefore, existing methods face three major bottlenecks:

(1) CNN excels at extracting local spatial features but cannot model long-term temporal dependencies; BiLSTM can capture temporal patterns but tends to overlook critical local details of packets.

(2) Most attention mechanisms focus on a single dimension (spatial or temporal) and cannot dynamically weigh the importance of local and global features.

(3) The hierarchical association between local packet features and global sequence patterns is not effectively utilized, leading to underutilized feature complementarity.

3.2 Local Feature Extraction Mechanism

The role of CNN in IDS is to help the system accurately identify attacks through automated feature extraction, hierarchical abstraction, and powerful pattern recognition capabilities. Combining CNN with the attention mechanism can improve the model's focus on important features, which in turn improves the detection accuracy and efficiency. Based on this local feature extraction mechanism, a new spatial feature extraction module, the FocusConv module is designed, which combines the CNN and the Attention mechanism by adding an Attention layer after the CNN layer. The module structure is shown in Fig. 3.

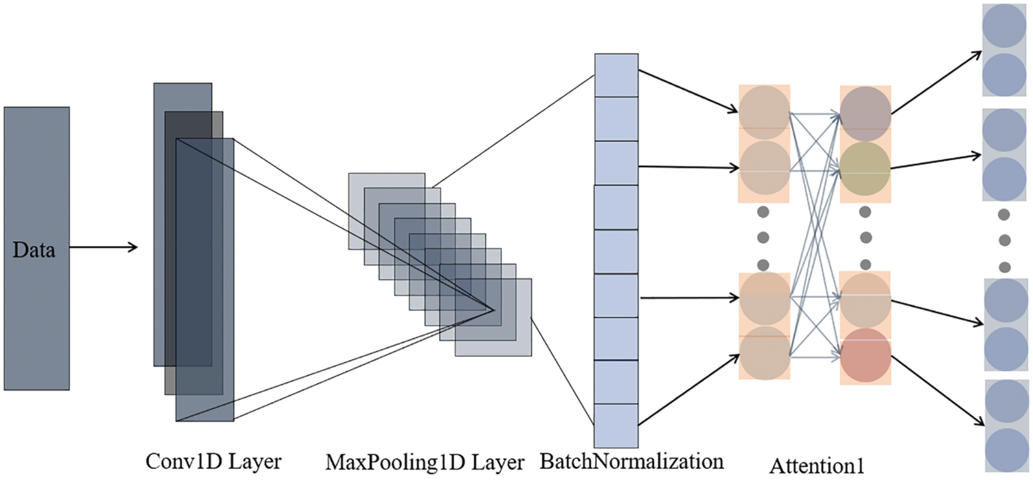

Figure 3: Structure of FocusConv module

The module firstly extracts the local features in the input data through a one-dimensional convolution operation at the convolution layer. The size of the convolution kernel is set to 128, stride = 1 ensures that the convolution kernel slides point by point, and padding = same keeps the size of the output feature maps in line with the inputs. The ReLU linear rectification function is used to introduce non-linear factors and enhance the expressive ability. A one-dimensional maximum pooling layer (pooling window size of 10) is used to downsample the data to reduce the data dimensionality and reduce the computational volume while retaining the main features. Finally, with the BatchNorm1D layer, each small batch of data is normalized in terms of feature dimensions to accelerate model training convergence speed, improve model stability and generalization ability, and obtain the final output.

The output of the CNN layer processing is then processed by Attention, and the result can be expressed as:

$$\mathrm{F}_{\mathrm{att}} = \mathrm{Softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d}_k}}\right) \bullet \mathrm{V} \tag{1}$$

F_{att} is the feature graph weighted by the attention mechanism. Q is the query, K is the key, and V is the value. Softmax is a normalization operation on the attention weights.

After being processed by the FocusConV module, the model can dynamically adjust the local feature weights to highlight key attack features and suppress noise, improving the sensitivity to traffic attacks and the global correlation ability of complex attack chains in intrusion detection.

3.3 Global Feature Extraction Mechanism

CNN is good at extracting local higher-order features from raw data, while BiLSTM avoids information loss, especially long-distance dependencies, through its LSTM mechanism. Combining the two can capture more comprehensive information and improve detection accuracy.

Combining both bidirectional long and short-term memory networks and an Attention mechanism, it aims to effectively capture the temporal information and global dependencies of the sequence data to enhance the recognition of attack patterns. BiLSTM is used to process and capture the bidirectional dependencies of the temporal data, while the Attention mechanism further helps the model to focus on the important global features. Based on this global feature extraction mechanism, a new temporal feature extraction module, the TempoNet module, is designed to combine the BiLSTM and Attention mechanism by adding an Attention layer after the BiLSTM. The module structure is shown in Fig. 4.

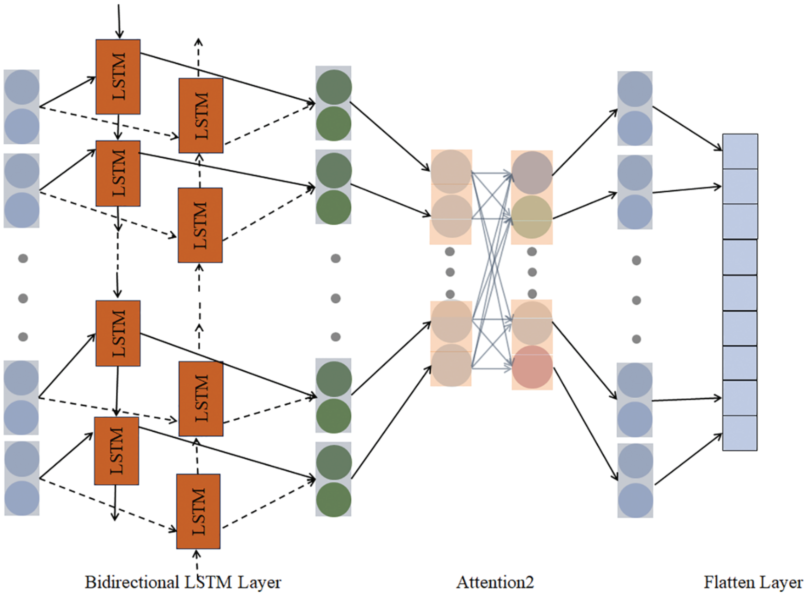

Figure 4: TempoNet module structure

The output processed by the FocusConV module is firstly processed by BiLSTM, which controls the flow and retention of information using three gates (input, oblivion, and output) and a memory unit. The special feature of BiLSTM is that it consists of two LSTM networks: a left-to-right (forward) LSTM and a right-to-left (reverse) LSTM. Eventually, the outputs of the two LSTMs are combined to better capture the bidirectional dependencies of the sequence.

The oblivion gate determines which information should be discarded at the current time step t. The output is shown below.

$$f_t = \sigma\left(W_f\left[h_t - 1, x_t\right] + b_f\right) \tag{2}$$

The input gate determines which new information should be stored in the memory cell, and the output is shown below.

$$\mathbf{i}_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{3}$$

The output gate determines what is output from the memory cell, and the output is shown below.

$$O_t = \sigma\left(Wo\left[h_t - 1, x_t\right] + b_o\right) \tag{4}$$

For BiLSTM, the forward LSTM will update the hidden state gradually from the beginning to the end of the sequence, while the reverse LSTM will update the hidden state from the end to the beginning of the sequence. The final output of each time step will combine the forward and reverse hidden states, which can be either spliced or averaged, and the output is shown below.

$$h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{5}$$

$\vec{h}_t$ denotes the output of forward LSTM and $\overleftarrow{h_t}$ denotes the output of reverse LSTM.

The input sequence passes through the BiLSTM layer and generates a sequence of hidden states containing bi-directional information, with the hidden state h_t encoding the current location and historical/future information at each time step, as follows.

$$H = \{h_1, h_2, \cdots, h_T\} \tag{6}$$

The hidden state sequence is then processed by the Attention layer to calculate the attention weights for each time step, and the BiLSTM output features are weighted and summed to obtain the final weighted output as follows. Calculate attention weights and weighted sum output:

$$\alpha_t = \frac{\exp(Q \cdot K^T)}{\sum_{t=1}^{T} \exp(Q \cdot K^T)} \tag{7}$$

$$F_{att} = \sum_{t=1}^{T} \alpha_t \cdot h_t \tag{8}$$

The similarity weights between all time steps are obtained after processing and can dynamically aggregate global key information. The final processing is done by the Flatten layer, which spreads the attention output into vectors.

$$Z_{flat} = \text{Flatten}(F_{att}) \tag{9}$$

The flattened high-dimensional vectors can be directly fed into the fully connected layer for classification, which is suitable for multi-class intrusion detection tasks.

After the TempoNet module, which combines bidirectional temporal modeling and dynamic feature weighting, the design balances local pattern recognition and global contextual correlation in complex attack detection to further enhance the accuracy and robustness of NID.

3.4 Loss Optimisation

In intrusion detection datasets, class imbalance is a common issue, and the standard cross-entropy loss can cause the model to be overly biased in favor of the majority class, ignoring the samples of the minority class. The EQL v2 method combines class balancing and adjustments to the class frequency to address the model's bias in favor of the majority class and to avoid overfitting by calculating the class weights for each class and using them in the loss calculation. Therefore, the EQL v2 method is chosen to be applied as a loss function in the Dual-Attention CNN-BiLSTM model.

EQL v2 method boosts the weighting for positive gradients while lowering it for negative gradients independently based on the cumulative ratio of the positive and negative gradients for each classifier, as shown in the following equation.

$$grad_{pos} = \nabla L(\hat{y}, y) \tag{10}$$

$$grad_{neg} = \nabla L(\hat{y}, y) \tag{11}$$

$\hat{y}$ is the predicted probability, and y is the true label.

Gradient-based balanced recalibration defines a ratio of cumulative positive and negative gradients, which represents the cumulative ratio of positive and negative gradients of the task up to the iteration, as shown in the following equation.

$$R_t^{(j)} = \frac{\sum_{i=1}^{t} grad_{pos}^{i}}{\sum_{i=1}^{t} grad_{neg}^{i}} \tag{12}$$

In the tenth iteration, the weights of the positive and negative gradients can be calculated using the following equation.

$$\beta_t^{(pos)} = 1 + \alpha(1 - f(R_t^{(j)})) \quad \text{(weights of positive gradients)} \tag{13}$$

$$\beta_t^{(neg)} = f(R_t^{(j)}) \quad \text{(weights of negative gradients)} \tag{14}$$

$f(\cdot)$ is a variant of the Sigmoid function, and α is a scaling factor.

After obtaining the positive and negative gradient weights, these weights are applied to adjust the positive and negative gradients of the current batch as shown in the following equation.

$$grad_{pos}^{(j)} = \beta_t^{(pos)} \cdot grad_{pos} \tag{15}$$

$$grad_{neg}^{(j)} = \beta_t^{(neg)} \cdot grad_{neg} \tag{16}$$

Finally, the cumulative positive and negative gradient ratios at the next iteration $t + 1$ are updated as shown in the following equation.

$$R_{t+1}^{(j)} = \frac{\sum_{i=1}^{t+1} grad_{pos}^{i}}{\sum_{i=1}^{t+1} grad_{neg}^{i}} \tag{17}$$

Using the EQL v2 method as a loss function, the gradient ratios of positive and negative samples are dynamically balanced, and the weights of the gradients are gradually adjusted during the training process so as to cope with the challenge of data imbalance and to improve the generalization ability of the model.

4 Experimental Results and Analyses

4.1 Data Sets

In this section, the performance of the Dual-Attention CNN-BiLSTM model will be evaluated on three publicly available datasets, NSL-KDD, UNSW-NB15, and CIC-DDos2019. The data distributions of these datasets are shown in Tables 1–3.

Table 1: Data distribution of the NSL - KDD dataset

Form	Number of training sessions	Number of tests
Normal	67,343	9711
Dos	45,927	7458
Probe	11,656	2421
R2L	995	2754
U2R	52	200
Total	125,973	22,544

Table 2: Data distribution of the UNSW-NB15 dataset

Form	Number of training sessions	Number of tests
Normal	56,000	37,000
Generic	40,000	18,871
Exploits	33,393	11,132
Fuzzers	18,184	6062
Dos	12,264	4089
Reconnaissance	10,493	3496
Analysis	2000	677
Backdoor	1746	583
Shellcode	1131	378
Worms	130	44
Total	175,341	82,332

Table 3: Data distribution of the CIC-DDos2019 dataset

Form	Number	Number of training sessions	Number of tests
BENIGN	29,803	26,823	2980
DrDoS-DNS	11,453	10,307	1146
DrDoS-LDAP	1733	1560	173
DrDoS-MSSQL	8044	7239	805
DrDoS-NetBIOS	11,543	10,389	1154

(Continued)

Table 3 (continued)

Form	Number	Number of training sessions	Number of tests
DrDoS-NTP	30,102	27,092	3010
DrDoS-SNMP	2713	2441	272
DrDoS-SSDP	13,214	11,893	1321
DrDoS-UDP	10387	9348	1039
SYN	11,102	9992	1110
TFTP	2295	2066	229
LDAP	549	494	55
NetBIOS	1090	981	109
MSSQL	4356	3920	436
Portmap	1060	954	106
UDP	12,131	10,918	1213
UDP-lag	208	187	21

The data types of the NSL-KDD dataset are five categories: Normal, Dos, Probe, U2R, and R2L. The majority class Normal accounts for 53.46% and the minority class U2R accounts for only 0.04% of the data in the dataset.

The majority class Normal in the UNSW-NB15 dataset accounts for 32.27%, and the minority classes Shellcode and Backdoor account for 0.65% and 1.01% of the training samples, respectively, which shows that there is an imbalance in the number of classes in the dataset.

In the CIC-DDos2019 dataset, BENIGN and DrDoS-NTP are the two largest categories, accounting for 19.64% and 19.83%, respectively. Attack types such as UDP-lag and LDAP account for less than 1% of the total training set, forming a severe long-tail distribution.

Before training and testing, we preprocess raw data to improve its quality, which can speed up model convergence and improve model accuracy. Data preprocessing includes the following two steps:

(1) One-Hot Encoding: Some features in the dataset cannot be fed directly into the neural network, so they are converted to digital form, and one-hot encoding is used to convert the text symbols.

(2) Normalization: Applying min-max normalization to scale the data range to $[0, 1]$ avoids the excessive influence of a large range of features on the model learning process.

4.2 Parameterisation and Assessment Indicators

In the experiments, Adam was used as the optimizer to optimize the weights for training, with the learning rate set to 0.001. A stratified K-Fold cross-validation strategy was also adopted for training and testing, and the model was trained for 100 epochs in the baseline architecture. All experiments were conducted using Keras. The experimental hardware and environment are shown in Table 4 below.

Table 4: The experimental hardware and environment

Hardware environment	Software environment
16 GB RAM	Windows 10 operating system
Intel (R) Core (TM) i7-10750H CPU @ 2.60 GHz	Python 3.9.18

(Continued)

Table 4 (continued)

Hardware environment	Software environment
NVIDIA GeForce GTX 1650	Keras 2.10.0 Numpy 1.26.4 scikit-learn 1.5.1 Matplotlib 3.9.2

Accuracy (ACC), Detection Rate (DR), and False Positive Rate (FPR) are taken as metrics to evaluate the performance of the model. ACC can be used to evaluate the ability of the model to predict normal and attack data, DR is used to evaluate the ability of the model to detect attack data, and FPR is used to evaluate the misclassification of the model's normal data. The formulas are shown below.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

$$DR = \frac{TP}{TP + FN} \tag{19}$$

$$FPR = \frac{FP}{FP + TN} \tag{20}$$

TP and TN are the number of correctly categorized attacks and the number of data where attack data is detected but the detection results are wrong and the data is actually normal data, respectively; FP is the number of actual normal data that is misclassified as an attack, and FN is the number of attacks that are incorrectly categorized as normal data.

4.3 Dichotomous and Multichotomous Results

Dichotomous and multicategorical classifications are used to evaluate the performance of the model. Dichotomous categories: the model determines whether a sample is an attack or a normal category. Multi-categorical classification: the model predicts whether the sample is a normal or a category attack in the dataset.

Table 5 shows the dichotomous category results of the model for the NSL-KDD, UNSW-NB15, and CIC-DDos2019 datasets under different stratified K-Fold cross-validation. The average accuracy (ACC%) for the NSL-KDD dataset is 99.68%, the detection rate (DR%) is 99.69%, and the FPR is 0.31%. For the UNSW-NB15 dataset, the average ACC is 97.72%, DR is 98.15%, and FPR is 1.85%. The average ACC for the CIC-DDos dataset is 99.93%, DR is 99.95%, and FPR is 0.05%.

Table 5: Dichotomous category results

K	NSL-KDD			UNSW-NB15			CIC-DDos2019		
	ACC%	DR%	FPR%	ACC%	DR%	FPR%	ACC%	DR%	FPR%
2	99.50	99.38	0.62	94.98	95.88	4.12	99.82	99.87	0.13
4	99.64	99.68	0.32	97.43	97.84	2.16	99.93	99.96	0.04
6	99.65	99.72	0.28	98.36	98.73	1.27	99.96	99.97	0.03

(Continued)

Table 5 (continued)

K	NSL-KDD			UNSW-NB15			CIC-DDos2019		
	ACC%	DR%	FPR%	ACC%	DR%	FPR%	ACC%	DR%	FPR%
8	99.77	99.79	0.21	98.73	98.92	1.08	99.95	99.96	0.04
10	99.84	99.86	0.14	99.09	99.39	0.61	99.98	100.00	0.00
On average	99.68	99.69	0.31	97.72	98.15	1.85	99.93	99.95	0.05

As can be seen in Fig. 5, for the NSL-KDD dataset, the best DR is 99.86% and the FPR is 0.14%, for UNSW-NB15, the best DR is 99.39% and the FPR is 0.61%, and for the CIC-DDos2019 dataset, the best DR reached 100% and the FPR is 0%. As the number of rounds increases, the accuracy and detection rate of the model continues to increase, the false positive rate decreases, and the model's performance reaches its best when K is 10, which indicates that the Dual-Attention CNN-BiLSTM model has a strong ability to differentiate between normal and attacking traffic, and proves the effectiveness of the Dual-Attention CNN-BiLSTM model.

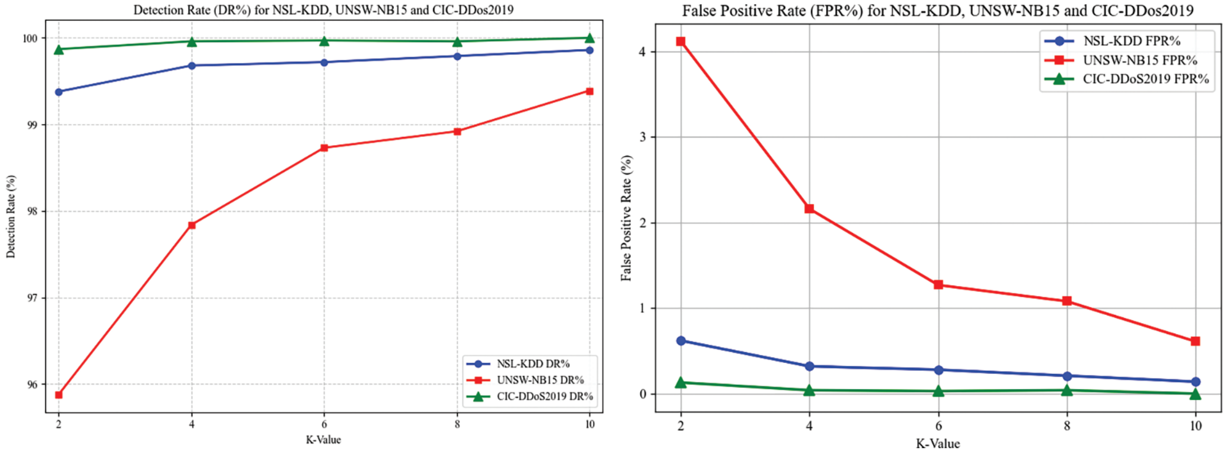

Figure 5: Dataset detection rate (DR%) and false positive rate (FPR%)

Table 6 shows the multi-category results of the model for the NSL-KDD, UNSW-NB15, and CIC-DDos2019 datasets using different stratified K-Fold cross-validation. The average accuracy (ACC%) for the NSL-KDD dataset is 99.66%, the detection rate (DR%) is 99.72%, and the FPR is 0.37%. For the UNSW-NB15 dataset, the average ACC is 87.47%, DR is 97.73%, and FPR is 2.96%. For the CIC-DDos2019 dataset, the average ACC is 99.39%, DR is 96.96%, and FPR is 0.13%.

Table 6: Multi-category results

K	NSL-KDD			UNSW-NB15			CIC-DDos2019		
	ACC%	DR%	FPR%	ACC%	DR%	FPR%	ACC%	DR%	FPR%
2	99.54	99.56	0.43	83.90	95.48	6.41	98.77	99.89	0.11
4	99.68	99.75	0.38	87.08	97.34	3.05	99.24	99.96	0.03
6	99.64	99.78	0.45	88.33	98.23	2.01	99.57	99.94	0.07

(Continued)

Table 6 (continued)

K	NSL-KDD			UNSW-NB15			CIC-DDos2019		
	ACC%	DR%	FPR%	ACC%	DR%	FPR%	ACC%	DR%	FPR%
8	99.72	99.73	0.25	89.07	98.90	1.85	99.65	99.99	0.10
10	99.70	99.78	0.32	88.95	98.71	1.46	99.74	100.00	0.10
On average	99.66	99.72	0.37	87.47	97.73	2.96	99.39	99.96	0.13

Fig. 6 shows the confusion matrix and the accuracy of each category for multi-class classification of the NSL-KDD dataset. From the confusion matrix plot, it can be seen that most of the samples are concentrated on the diagonal of the matrix, which indicates that the model can detect network traffic attacks well and the overall detection performance of the model is high. From the graph of detection rate by category, it can be seen that even a few categories have a high detection rate.

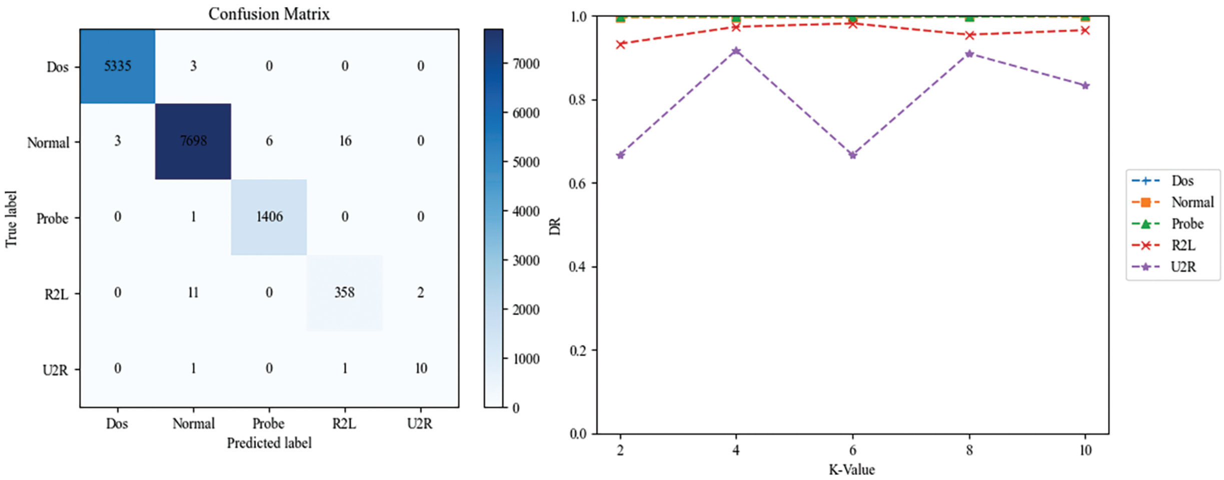

Figure 6: NSL-KDD confusion matrix and detection rate for each category

Fig. 7 shows the confusion matrix and the detection rate for each category for the UNSW-NB15 dataset in multi-class classification. We can see that the model can detect the attacks Exploits, Fuzzers, Generic, Reconnaissance, and Shellcode very well. In this round of detection, the attack classes are very unbalanced, even though the number of Shellcode and Worms samples is very small, the model still maintains a high detection rate for Shellcode and Worms with a maximum of 90.73% and 88.24%, respectively, which proves that the model can deal with the class imbalance problem very well and achieves good results.

Fig. 8 shows the confusion matrix and detection rates of each category in the CIC-DDos2019 dataset. The model has very high detection rates for all 17 categories, even for minority attack categories, indicating that the model has excellent detection capabilities for attacks of different scales and with different protocol characteristics.

The model maintains high accuracy and detection rates across these three datasets, demonstrating excellent generalization capabilities. At the same time, as the K value increases, the model's detection capabilities for various categories also continue to improve, demonstrating strong robustness.

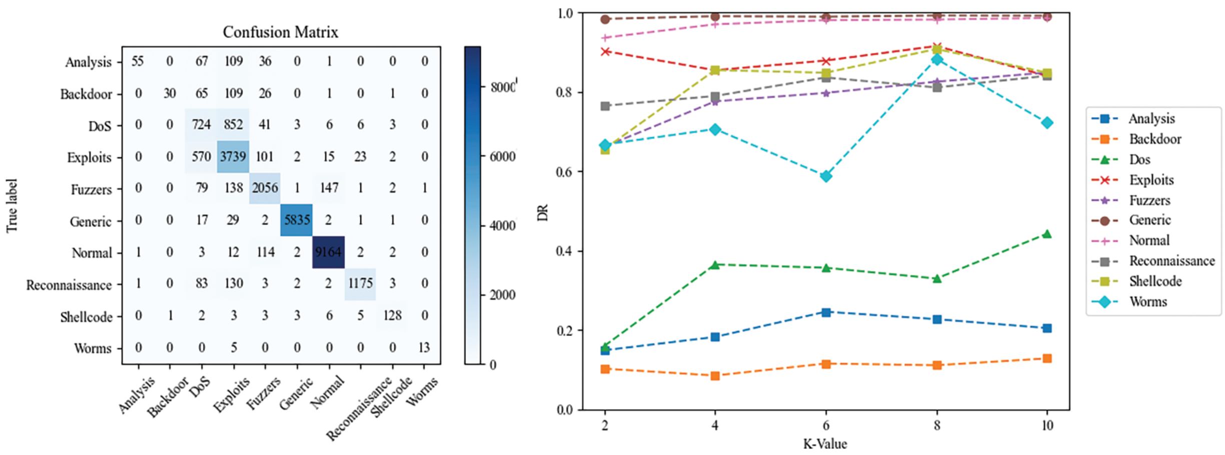

Figure 7: UNSW-NB15 confusion matrix and detection rates by category

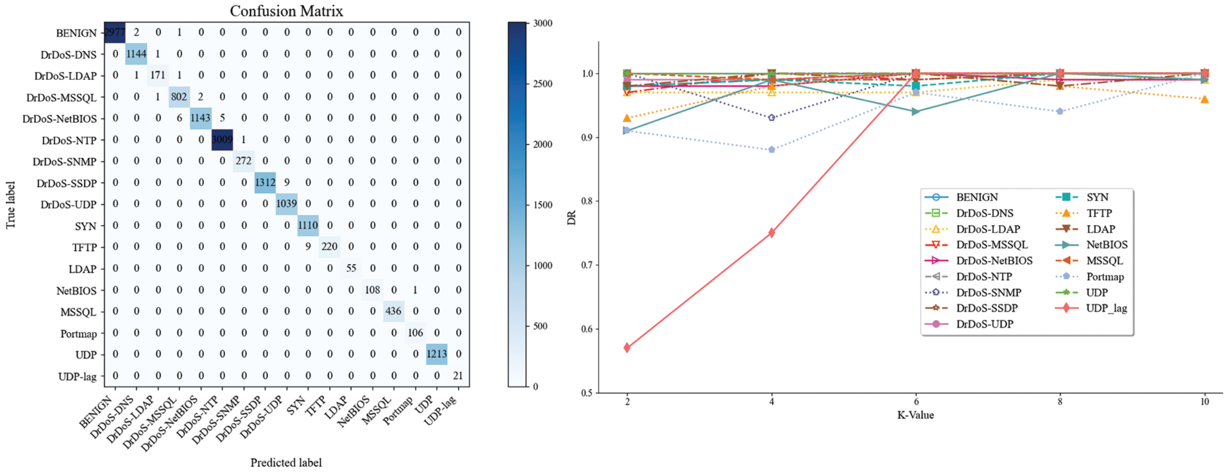

Figure 8: CIC-DDos2019 confusion matrix and detection rates by category

4.4 Comparative Experiments

4.4.1 Model Comparison Experiments across Different Datasets

To verify the advantages of the Dual-Attention CNN-BiLSTM model on the NSL-KDD dataset, this paper compares the Dual-Attention CNN-BiLSTM model with other intrusion detection models: the Pelican [40], the Lunet [41], the Multi-Head Attention-BiLSTM (MHAB), CNN-BiLSTM, CNN-GRU, as shown in Table 7. The Dual-Attention CNN-BiLSTM model has a better improvement in key performance metrics compared to the other models.

On the UNSW-NB15 dataset, this paper compares the Dual-Attention CNN-BiLSTM model with other intrusion detection models: the SVM [42], RF, AdaBoost [43], CNN-BiLSTM, and CNN-GRU, as shown in Table 8. The Dual-Attention CNN-BiLSTM model improves accuracy by 6.87%, detection rate by 6.2%, and false-positive rate by 4.63% compared to the CNN-BiLSTM model. For the more advanced CNN-GRU model, the accuracy is also improved by 2.7% and the detection rate is improved by more than 12%, which confirms that the Dual-Attention CNN-BiLSTM model is effective in intrusion detection.

Table 7: Comparison results of NSL-KDD dataset models

Model	ACC (%)	DR (%)	FPR (%)
Pelican	99.21	99.13	0.65
Lunet	99.14	99.02	0.61
MHAB	95.19	98.00	–
CNN-BiLSTM	99.22	98.88	0.43
CNN-GRU	99.69	99.69	–
This method	99.72	99.73	0.25

Table 8: Model comparison results for the UNSW-NB15 dataset

Model	ACC (%)	DR (%)	FPR (%)
SVM	74.80	83.71	7.73
RF	84.59	92.24	3.01
AdaBoost	73.19	91.13	22.11
CNN-BiLSTM	82.08	92.51	6.09
CNN-GRU	86.25	86.25	–
This method	88.95	98.71	1.46

On the CIC-DDos2019 dataset, this paper compares the Dual-Attention CNN-BiLSTM model with other intrusion detection models: KNN [44], RF, AdaBoost, RTIDS [45], and CANET, as shown in Table 9. Compared to the advanced CANET model, the Dual-Attention CNN-BiLSTM model shows slight improvements in accuracy, detection rate, and false positive rate, particularly achieving a detection rate of 100%.

Table 9: Model comparison results for the CIC-DDos2019 dataset

Model	ACC (%)	DR (%)	FPR (%)
KNN	90.66	99.77	0.47
RF	99.28	99.85	0.10
AdaBoost	54.56	48.40	12.38
RTIDS	98.58	98.66	-
CANET	99.21	99.95	0.12
This mothed	99.74	100.00	0.10

4.4.2 Comparative Experiments on Detection Capability for Class Imbalance in Datasets

To verify the ability of the Dual-Attention CNN-BiLSTM model to address class imbalance in the dataset, this model is compared with the CANET and the CNN-BiLSTM model for each class of the UNSW-NB15 dataset, as shown in Table 10. This model significantly improves detection for Analysis, Exploits, Fuzzers, and Shellcode attacks compared to the more advanced CANET model, while only the Worms attack lags behind the CNN-BiLSTM model. In summary, the Dual-Attention CNN-BiLSTM model enhances the detection of minority classes and exhibits strong overall detection performance.

Table 10: Comparison results of the UNSW-NB15 dataset model by category

Category	This mothed	CANET	CNN-BiLSTM	Number
Analysis	25%	21%	6%	268
Backdoor	14%	14%	7%	232
Dos	44%	46%	5%	1635
Exploits	92%	85%	92%	4452
Fuzzers	85%	84%	54%	2425
Generic	99%	99%	98%	5887
Normal	99%	99%	95%	9300
Reconn	84%	84%	73%	1399
Shellcode	91%	87%	0%	151
Worms	89%	89%	100%	18

4.4.3 Comparative Experiments on Detection Capability for Minority Classes Using EQL v2 Method

To verify whether the EQL v2 method improves the model's ability to detect minority classes, it is chosen to compare the detection rate of each category with cross-entropy (CE), as shown in Table 11. There is a significant improvement in the detection rate of each category, especially for Analysis and Exploits attacks by 7% and for Shellcode attacks by 10%, and only Fuzzers and Worms attacks lag behind the standard cross entropy. In summary, the EQL v2 method improves the model's ability to detect minority classes and optimizes the dataset imbalance.

Table 11: Comparison results between EQL v2 and CE categories

	EQL v2	CE	Number
Analysis	25%	18%	268
Backdoor	14%	13%	232
Dos	44%	40%	1635
Exploits	92%	85%	4452
Fuzzers	85%	86%	2425
Generic	99%	99%	5887
Normal	99%	98%	9300
Reconn	84%	84%	1399
Shellcode	91%	81%	151
Worms	89%	94%	18

4.4.4 Comparative Experiments on Model Architecture Efficiency and Performance

To validate the architectural advantages of the Dual-Attention CNN-BiLSTM model, we compared it with advanced models such as Autoformer, Informer, Transformer, and Reformer, as well as the lightweight CANET model, in terms of training and validation processing speed and performance, as shown in Table 12. Transformer-based models, such as Autoformer and Informer, although they introduce improved attention mechanisms, are fundamentally constrained by the quadratic time complexity of global self-attention. In contrast, the method proposed in this paper significantly reduces computational complexity by leveraging the linear temporal modeling capabilities of BiLSTM and local attention, thereby substantially reducing training time while maintaining high accuracy and detection rates. The lightweight model CANET achieves a 28% higher training processing speed than this method due to its lightweight design. However, as a pure CNN+Attention architecture, its verification speed is 89% slower than this method, primarily due to the lack of temporal modeling capabilities. In the UNSW-NB15 dataset, Exploit-class attacks rely on temporal features such as the time intervals between connected sequences and port access frequencies for detection. Since CANET does not incorporate a sequence modeling layer, it exhibits a high false negative rate for such attacks. Therefore, the Dual-Attention CNN-BiLSTM model maintains high training processing speed while also performing well in accuracy and detection rate. With a verification processing speed of 82 us/sample, it supports real-time inference capabilities and is suitable for applications requiring low latency, such as network attack detection.

Table 12: Comparison of model training and validation processing speeds and detection performance

UNSW-NB15 Dataset (Multi-classification, Same Environment, batch = 256)				
Model	**Training speed (us/sample)**	**Validation speed (us/sample)**	**ACC (%)**	**DR (%)**
Autoformer	292 us/sample	165 us/sample	74.89%	69.43%
Informer	705 us/sample	42 us/sample	81.67%	92.38%
Transformer	485 us/sample	147 us/sample	80.42%	90.20%
Reformer	624 us/sample	143 us/sample	76.06%	83.77%
CANET	229 us/sample	155 us/sample	87.53%	97.84%
This method	316 us/sample	82 us/sample	88.95%	98.71%

5 Conclusion

Due to dataset class imbalance and low detection rate of intrusion detection models, this paper proposes a Dual-Attention CNN-BiLSTM model and designs two modules: the FocusConV and TempoNet modules. The FocusConV module automatically adjusts and weights the local features extracted by the CNN to focus on the more important local features. The TempoNet module captures global information by identifying essential features across time steps, emphasizing long-range contextual dependencies. Meanwhile, the EQL v2 method is used to optimize the performance of the model on the class imbalance problem. Several experiments were conducted on the NSL-KDD, UNSW-NB15, and CIC-DDos2019 datasets, and the experimental results show that the Dual-Attention CNN-BiLSTM model achieves a high accuracy rate, a high detection rate, and a low false-positive rate on these datasets, which verifies the feasibility of the model in the face of network intrusion. However, the study still has certain limitations: model training takes a long time. In terms of model structure optimization, we could explore more efficient attention mechanism variants or combine other advanced neural network structures to reduce the computational complexity of the model and thus shorten the training time. The model has not been optimized for real-time deployment scenarios, and its latency constraints and processing efficiency in actual network environments have not been explored. Future research could utilize lightweight techniques and GPU/TPU hardware acceleration to reduce inference latency, validate real-time processing capabilities under high-speed networks; test response times in actual traffic, balance detection accuracy and real-time performance through sliding windows and online learning; assess resource consumption on edge devices, explore integration solutions for existing NID systems, and validate applicability in real network environments.

Acknowledgement: The authors would like to appreciate Southeast University, Jinling Institute of Technology, for supporting this research work.

Funding Statement: This research was supported by the High-Level Talent Foundation of Jinling Institute of Technology (grant number. JIT-B-202413).

Author Contributions: Zheng Zhang: Conceptualization, Methodology, Software, Writing—Original Draft. Jie Hao: Validation, Formal Analysis. Liquan Chen: Supervision. Tianhao Hou: Data Curation, Software. Yanan Liu: Writing—Reviewing and Editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data openly available in a public repository. Data available within the article.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Chen CM, Chen YL, Lin HC. An efficient network intrusion detection. Comput Commun. 2010;33(4):477–84. doi:10.1016/j.comcom.2009.10.010.
2. Abdulganiyu OH, Ait Tchakoucht T, Saheed YK. A systematic literature review for network intrusion detection system (IDS). Int J Inf Secur. 2023;22(5):1125–62. doi:10.1007/s10207-023-00682-2.
3. Shone N, Ngoc TN, Phai VD, Shi Q. A deep learning approach to network intrusion detection. IEEE Trans Emerging Topics Comput Intell. 2018;2(1):41–50. doi:10.1109/tetci.2017.2772792.
4. Liu H, Lang B. Machine learning and deep learning methods for intrusion detection systems: a survey. Appl Sci. 2019;9(20):4396. doi:10.3390/app9204396.
5. Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F. Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans Emerg Telecomm Technol. 2021;32(1):e4150. doi:10.1002/ett.4150.
6. Zhang J, Zulkernine M, Haque A. Random-forests-based network intrusion detection systems. Trans Sys Man Cyber Part C. 2008;38(5):649–59. doi:10.1109/tsmcc.2008.923876.
7. Rai K, Devi MS, Guleria A. Decision tree based algorithm for intrusion detection. Int J Adv Netw Appl. 2016;7(4):2828.
8. Hasan MAM, Nasser M, Pal B, Ahmad S. Support vector machine and random forest modeling for intrusion detection system (IDS). J Intell Learn Syst Appl. 2014;6(1):45–52. doi:10.4236/jilsa.2014.61005.
9. Liao Y, Vemuri VR. Use of k-nearest neighbor classifier for intrusion detection. Comput Secur. 2002;21(5):439–48.
10. Zhang C, Jia D, Wang L, Wang W, Liu F, Yang A. Comparative research on network intrusion detection methods based on machine learning. Comput Secur. 2022;121:102861.
11. Gamage S, Samarabandu J. Deep learning methods in network intrusion detection: a survey and an objective comparison. J Netw Comput Appl. 2020;169(2):102767. doi:10.1016/j.jnca.2020.102767.
12. Abdulhammed R, Faezipour M, Abuzneid A, AbuMallouh A. Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. IEEE Sensors Lett. 2018;3(1):1–4. doi:10.1109/lsens.2018.2879990.
13. Ahmed HA, Hameed A, Bawany NZ. Network intrusion detection using oversampling technique and machine learning algorithms. PeerJ Comput Sci. 2022;8(1):e820. doi:10.7717/peerj-cs.820.
14. Silva BR, Silveira RJ, da Silva Neto MG, Cortez PC, Gomes DG. A comparative analysis of undersampling techniques for network intrusion detection systems design. J Commun Inf Syst. 2021;36(1):31–43. doi:10.14209/jcis.2021.3.
15. Zhang G, Wang X, Li R, Song Y, He J, Lai J. Network intrusion detection based on conditional Wasserstein generative adversarial network and cost-sensitive stacked autoencoder. IEEE Access. 2020;8:190431–47. doi:10.1109/access.2020.3031892.
16. Tan J, Lu X, Zhang G, Yin C, Li Q. Equalization loss v2: a new gradient balance approach for long-tailed object detection. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 20–25; Nashville, TN, USA. p. 1685–94.

17. Liu T, Fu Y, Wang K, Duan X, Wu Q. A multiscale approach for network intrusion detection based on variance-covariance subspace distance and EQL v2. Comput Secur. 2025;148(3):104173. doi:10.1016/j.cose.2024.104173.
18. Dai W, Li X, Ji W, He S. Network intrusion detection method based on CNN-BiLSTM-attention model. IEEE Access. 2024;12:53099–111. doi:10.1109/access.2024.3384528.
19. Mohammadpour L, Ling TC, Liew CS, Aryanfar A. A survey of CNN-based network intrusion detection. Appl Sci. 2022;12(16):8162. doi:10.3390/app12168162.
20. Gan B, Chen Y, Dong Q, Guo J, Wang R. A convolutional neural network intrusion detection method based on data imbalance. J Supercomput. 2022;78(18):19401–34.
21. Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning. Neurocomputing. 2021;452:48–62. doi:10.1016/j.neucom.2021.03.091.
22. Siami-Namini S, Tavakoli N, Namin AS. The performance of LSTM and BiLSTM in forecasting time series. In: 2019 IEEE International Conference on Big Data; 2019 Dec 9–12; Los Angeles, CA, USA. p. 3285–92.
23. Revathi S, Malathi A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. Int J Eng Res Technol. 2013;2(12):1848–53.
24. For Cybersecurity (CIC) CI. CIC-NSL-KDD Dataset [Internet]. GitHub; 2009 [cited 2025 Aug 21]. Available from: https://github.com/HoaNP/NSL-KDD-DataSet.
25. Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS); 2015 Nov 10–12; Canberra, ACT, Australia. p. 1–6.
26. Moustafa N, Slay J. UNSW-NB15 network intrusion detection dataset [Internet]. UNSW Canberra Cybersecurity Research Group; 2015 [cited 2025 Aug 21]. Available from: https://research.unsw.edu.au/projects/unsw-nb15-dataset.
27. Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST); 2019 Oct 1–3; Chennai, India. p. 1–8.
28. Sharafaldin I, Lashkari AH, Ghorbani AA. CIC-DDoS2019 Dataset [Internet]. Canadian Institute for Cybersecurity (CIC); 2019 [cited 2025 Aug 21]. Available from: http://cicresearch.ca/CICDataset/CICDDoS2019/Dataset/.
29. Kim J, Kim J, Kim H, Shim M, Choi E. CNN-based network intrusion detection against denial-of-service attacks. Electronics. 2020;9(6):916. doi:10.3390/electronics9060916.
30. Bolon-Canedo V, Sanchez-Marono N, Alonso-Betanzos A. Feature selection and classification in multiple class datasets: an application to KDD Cup 99 dataset. Exp Syst Appl. 2011;38(5):5947–57. doi:10.1016/j.eswa.2010.11.028.
31. Gopalan SS, Ravikumar D, Linekar D, Raza A, Hasib M. Balancing approaches towards ML for IDS: a survey for the CSE-CIC IDS dataset. In: 2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA); 2021 Mar 16–18; Sharjah, United Arab Emirates. p. 1–6.
32. Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access. 2017;5:21954–61. doi:10.1109/access.2017.2762418.
33. Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys D Nonlinear Phenom. 2020;404(8):132306. doi:10.1016/j.physd.2019.132306.
34. Sinha J, Manollas M. Efficient deep CNN-BiLSTM model for network intrusion detection. In: Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition; 2020 Sep 25–27; Xiamen, China. p. 223–31.
35. Zhang J, Zhang X, Liu Z, Fu F, Jiao Y, Xu F. A network intrusion detection model based on BiLSTM with multi-head attention mechanism. Electronics. 2023;12(19):4170. doi:10.3390/electronics12194170.
36. Ren K, Yuan S, Zhang C, Shi Y, Huang Z. CANET: a hierarchical cnn-attention model for network intrusion detection. Comput Commun. 2023;205(16):170–81. doi:10.1016/j.comcom.2023.04.018.
37. Cao B, Li C, Song Y, Qin Y, Chen C. Network intrusion detection model based on CNN and GRU. Appl Sci. 2022;12(9):4184. doi:10.3390/app12094184.
38. Bagui S, Li K. Resampling imbalanced data for network intrusion detection datasets. J Big Data. 2021;8(1):6. doi:10.1186/s40537-020-00390-x.

39. Telikani A, Rudbardeh NE, Soleymanpour S, Shahbahrami A, Shen J, Gaydadjiev G, et al. A cost-sensitive machine learning model with multitask learning for intrusion detection in IoT. IEEE Trans Ind Inform. 2023;20(3):3880–90. doi:10.1109/tii.2023.3314208.
40. Wu P, Guo H, Moustafa N. Pelican: a deep residual network for network intrusion detection. In: 2020 50th annual IEEE/IFIP international conference on dependable systems and networks workshops (DSN-W); 2020 Jun 29–Jul 2; Valencia, Spain. p. 55–62.
41. Wu P, Guo H. LuNET: a deep neural network for network intrusion detection. In: 2019 IEEE Symposium Series on Computational Intelligence (SSCI); 2019 Dec 6–9; Xiamen, China. p. 617–24.
42. Ahmad I, Basheri M, Iqbal MJ, Rahim A. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access. 2018;6:33789–95. doi:10.1109/access.2018.2841987.
43. Hu W, Gao J, Wang Y, Wu O, Maybank S. Online adaboost-based parameterized methods for dynamic distributed network intrusion detection. IEEE Trans Cybernetics. 2013;44(1):66–82. doi:10.1109/tcyb.2013.2247592.
44. Benaddi H, Ibrahimi K, Benslimane A. Improving the intrusion detection system for NSL-KDD dataset based on PCA-fuzzy clustering-KNN. In: 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM); 2018 Oct 16–19; Marrakesh, Morocco. p. 1–6.
45. Wu Z, Zhang H, Wang P, Sun Z. RTIDS: a robust transformer-based approach for intrusion detection system. IEEE Access. 2022;10(3):64375–87. doi:10.1109/access.2022.3182333.