Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064270

ARTICLE

Intelligent Detection of Abnormal Traffic Based on SCN-BiLSTM

Lulu Zhang, Xuehui Du*, Wenjuan Wang, Yu Cao, Xiangyu Wu and Shihao Wang

Henan Province Key Laboratory of Information Security, PLA Information Engineering University, Zhengzhou, 450001, China

*Corresponding Author: Xuehui Du. Email: dxh37139@136.com

Received: 10 February 2025; Accepted: 14 April 2025; Published: 09 June 2025

ABSTRACT: To address the limitations of existing abnormal traffic detection methods, such as insufficient temporal and spatial feature extraction, high false positive rate (FPR), poor generalization, and class imbalance, this study proposed an intelligent detection method that combines a Stacked Convolutional Network (SCN), Bidirectional Long Short-Term Memory (BiLSTM) network, and Equalization Loss v2 (EQL v2). This method was divided into two components: a feature extraction model and a classification and detection model. First, SCN was constructed by combining a Convolutional Neural Network (CNN) with a Depthwise Separable Convolution (DSC) network to capture the abstract spatial features of traffic data. These features were then input into the BiLSTM to capture temporal dependencies. An attention mechanism was incorporated after SCN and BiLSTM to enhance the extraction of key spatiotemporal features. To address class imbalance, the classification detection model applied EQL v2 to adjust the weights of the minority classes, ensuring that they received equal focus during training. The experimental results indicated that the proposed method outperformed the existing methods in terms of accuracy, FPR, and F1-score and significantly improved the identification rate of minority classes.

KEYWORDS: Convolutional neural network; depthwise separable convolution; bidirectional long and short-term memory network; class imbalance; abnormal traffic detection

1 Introduction

In the digital era, network security has become a global concern with the continuous evolution of cyberattacks, highlighting the need to secure the network environment. These attacks not only threaten personal privacy but can also have a severe impact on business operations and national security. Although traditional security measures offer protection, they are often insufficient for increasingly sophisticated attacks. Network anomaly traffic detection methods are commonly used to identify abnormal traffic and enable timely responses, thereby ensuring stable operation and security of the network. Consequently, abnormal traffic detection has become a key research focus in this field [1].

Abnormal network traffic detection involves a thorough analysis of network traffic by utilizing traffic classification or statistical methods to identify and address anomalies promptly. Traditional detection methods have widely employed machine learning-based approaches to detect abnormal traffic. For instance, researchers have applied various machine learning techniques, such as random forest [2], K-nearest neighbor [3], support vector machine [4,5], Naive Bayes [6], and decision tree [7], to investigate abnormal network traffic detection, and have achieved promising results. However, these methods require manual design and selection of traffic features.

To address the challenges of traditional machine learning feature set design, researchers have turned to deep learning for abnormal traffic detection, which enables automatic feature extraction from network traffic and identification of unusual behaviors. Deep learning-based techniques have shown superior detection capabilities compared with conventional machine learning models [8]. For example, researchers have utilized Convolutional Neural Network (CNN), Deep Neural Network (DNN), and Recurrent Neural Network (RNN) to overcome the limitations of shallow classifiers and improve the accuracy of intrusion detection [9,10]. Despite advancements in deep learning methodologies, challenges remain in improving classification accuracy and addressing the imbalance in dataset categories. Although CNN is effective at processing large datasets and extracting robust features, the pooling stage may lead to loss of critical details in anomalous data. In addition, both RNN and CNN may struggle to capture long-range dependencies, which affects the generalizability and performance of the models. Moreover, numerous current models both failing to consider the spatiotemporal complexity of traffic data and overlook the efficiency of the model itself. Particularly deep learning models, complex neural network architectures require significant memory and processing power, rendering them unsuitable for edge-computing devices with limited resources [11]. Moreover, the imbalance in many datasets can increase the false positive rate, thereby reducing the overall accuracy of deep learning-based approaches.

To address the challenges outlined above, this study proposed a novel method for detecting abnormal network traffic using a combination of Stacked Convolutional Network (SCN) and Bidirectional Long Short-Term Memory (BiLSTM). The SCN constructed by integrating a CNN and Depthwise Separable Convolution (DSC) extracted the spatial features of abnormal traffic, whereas BiLSTM captured the temporal features. An attention mechanism was incorporated to prioritize the critical spatial and temporal features. To address the common issue of class imbalance in the dataset, Equalization Loss v2 (EQL v2) [12] was applied to enhance the learning of minority classes, thereby improving model accuracy. The key contributions of this study are as follows.

(1) The regular convolutional layer of CNN was combined with DSC to construct the SCN, which extracted the spatial features from traffic data while reducing the number of parameters and computational workload, thereby simplifying the complexity of the model. BiLSTM bidirectional processing was employed to capture the temporal dependencies between traffic features and extract contextual information. Additionally, an attention mechanism was introduced to enhance the model to capture the critical spatial and temporal features, ultimately improving classification accuracy.

(2) To address the class imbalance issue in multi-class classification tasks, the EQL v2 loss function was designed to improve the detection precision for minority class data. By combining the SCN-BiLSTM feature extraction model, which included the attention mechanism, with a class imbalance-oriented classification detection model, this approach aimed to enhance the detection performance on imbalanced datasets by addressing both insufficient spatial and temporal feature extraction and the class imbalance problem.

(3) The efficacy of the anomaly detection approach was validated using the NSL-KDD dataset. The empirical results demonstrated that the proposed method outperformed the other approaches, demonstrating significant improvements in multiclassification accuracy, recall, and F1-score.

The paper is organized as follows: Section 2 reviews related research work. Section 3 details the proposed methodology. Section 4 describes the experimental results and analyses. Section 5 concludes our work and looks forward to future research directions.

2 Related Work

As artificial intelligence research has deepened, particularly with the rapid development of machine learning and deep learning [13,14], the application of these methods in abnormal traffic detection has become increasingly popular.

Compared with machine learning, deep learning can leverage neural networks to automatically extract features from different types of network traffic, addressing the challenge of heavy reliance of machine learning techniques on expert experience [15]. As deep learning has evolved, an increasing number of deep learning-based models have shown exceptional performance [16]. Li et al. [17] applied an image transformation technique to convert network traffic data into an image form, enabling the automatic learning of graphical features through a CNN model for traffic detection. Khan et al. [18] proposed a CNN-based traffic detection method that employed three convolutional layers with convolutional and pooling operations to effectively capture feature relationships, using the softmax function to classify the extracted features with an accuracy of 99.23%. Temporal features, such as timestamps and durations, in traffic data have led some researchers to consider using RNNs for analysis. However, RNNs are limited by their short-term memory and inability to analyze longer sequences. To address this, Long Short-Term Memory (LSTM) network serves as a specialized form of RNN that can learn long-term dependencies. Donkol and Hussein [19] proposed an intrusion detection model combining feature selection with an enhanced LSTM and RNN architecture. While this model achieved a high detection rate, its computational cost remained significantly high. Imrana et al. [20] suggested that traditional models have a high FPR for specific attack types such as U2R and R2L. To improve this, they proposed a BiLSTM-based network model, and experimental data demonstrated that this method outperformed the traditional LSTM and other advanced models in terms of classification performance. Similarly, Zhang et al. [21] proposed a network intrusion detection model based on BiLSTM with a multi-head attention mechanism. The model assigns attention weights to features and employs BiLSTM to capture long-distance dependencies, achieving superior accuracy on several benchmark datasets.

These methods demonstrated that CNN and BiLSTM are effective in improving the classification performance of abnormal traffic detection. However, a thorough review of current research on traffic detection using deep learning models reveals that most existing models focus on either spatial or temporal features without adequately balancing them. For example, CNN in the above study excels in capturing spatial feature correlations, whereas BiLSTM is more suited for analyzing temporal features. In addition, these networks often suffer from high computational complexity, long training times, and other limitations. To address these issues, this study proposed a feature extraction model that combined CNN, DSC, and BiLSTM.

Meanwhile, it is found that these methods do not effectively address the issue of dataset class imbalance. When deep learning models handle imbalanced data, they struggle to accurately identify certain attacks owing to the overwhelming number of normal samples compared with abnormal samples and the significant variation in the proportion of each attack type within the abnormal samples. For instance, Dong et al. [22] proposed an abnormal traffic detection framework based on deep reinforcement learning, which achieved good results. However, it still falls short in detecting low-quantity abnormal traffic such as R2L and U2R. Therefore, addressing class imbalance is crucial for improving detection performance. Techniques for addressing this issue primarily involve data- and algorithmic-level approaches. Data-level methods aim to achieve more balanced class representation by manipulating a dataset through oversampling and undersampling. For example, the SMOTE method was employed by Wei et al. [23] to generate representative samples for minority classes, alleviating the imbalance problem of network traffic. However, these methods carry the risks of data loss, overfitting of the training set, and increased computational demands. In contrast, algorithmic-level methods address class imbalance by incorporating specialized algorithms during the training phase, without modifying the original dataset composition. For instance, cost-sensitive learning

maintains an imbalanced distribution of dataset by assigning different weights to different samples while improving the recognition of classes with lower frequencies. Dina et al. [24] proposed the application of the focal loss (FL) function to address category imbalance, thereby enhancing the model classification performance by increasing the influence of difficult-to-classify samples. Tan et al. [25] introduced the Equalization Loss (EQL) weighted loss function to adjust the weights of positive and negative samples. However, the improvement in detection accuracy from the above studies is limited. Therefore, this study employed EQL v2 to address class imbalance, providing independent and balanced weight adjustments during the training process for each class, thereby enhancing classification accuracy for minor classes.

In summary, practical traffic detection scenarios require attention not only to the issue of unbalanced data categories but also to the extraction of spatial and temporal features within traffic data to improve detection accuracy. Therefore, designing a more efficient deep learning model that balances detection performance across various categories while effectively capturing spatial and temporal features to enhance accuracy has become a key challenge in this field.

3 Proposed Methodology

3.1 Overview of Feature Extraction Model

Currently, most abnormal traffic detection methods focus on a single-network perspective and fail to fully explore the temporal and spatial dimensions inherent in traffic datasets. This study proposed a lightweight feature extraction model, SCN-BiLSTM, with a fusion attention mechanism that effectively addressed the limitations of traditional detection models by leveraging both the temporal and spatial information in traffic data, resulting in more comprehensive and accurate abnormal traffic detection.

3.1.1 Spatial Feature Extraction Module with SCN

In deep learning, CNN has become a key technology for efficient feature extraction in tasks such as target recognition, text classification, and anomaly detection. Conventional CNN consists of three main components: a convolutional layer, a pooling layer, and a fully connected layer [26]. The convolutional layer performs convolutional operations by moving the kernel over the input data to efficiently extract local features, making it the most crucial layer in the CNN structure. The pooling layer reduces the dimensionality of the feature maps, thereby decreasing the computational load and the parameter volume. After the convolutional and pooling layers extract the relevant features, the fully connected layer synthesizes these features for final classification or regression analysis. Although the convolutional layer excels in local feature extraction, its ability to capture global feature correlations is limited, which may result in the loss of important semantic details when processing complex data, thereby affecting the final results.

DSC extracts spatial features within a single channel by decomposing the regular convolutional layer into depthwise convolution to capture spatial features within each channel and pointwise convolution to combine cross-channel features. This two-step process first applies depth-wise convolution to independently convolve each input channel, significantly reducing the computational cost and focusing on feature extraction within each channel. The pointwise convolution then performs inter-channel feature fusion, refining, and compressing the features. This approach ensures that the model remains lightweight while maintaining a performance comparable to that of regular convolution, significantly reducing computation and model parameters. Although DSC offers advantages in terms of computational efficiency and model size, it is less effective than regular convolution in capturing certain complex feature patterns, which may reduce the model's expressive capacity.

To address these issues, this study introduced an SCN for spatial feature extraction. The SCN consisted of four main layers: a regular convolutional layer, a depthwise convolutional layer, a pointwise convolutional layer, and a max pooling layer. As the first layer, the regular convolutional layer captured the local patterns and underlying features of the sequence from the original input by setting hyperparameters such as the number of filters and the size of the convolutional kernel, providing a rich feature base for subsequent processing. To retain more detail and spatial information, a pooling layer was not adopted after this layer. The depth-wise convolutional layer applied the convolution kernel to each channel separately, reducing the parameter count while preserving channel information. The pointwise convolutional layer combined the outputs of depthwise convolution, mixing information across channels to enhance the nonlinear processing capacity of the model. Finally, the max pooling layer reduced the spatial dimensions and decreased the input size for subsequent layers, which helped the model learn more representative features and mitigate overfitting. The principle of SCN is illustrated in Fig. 1.

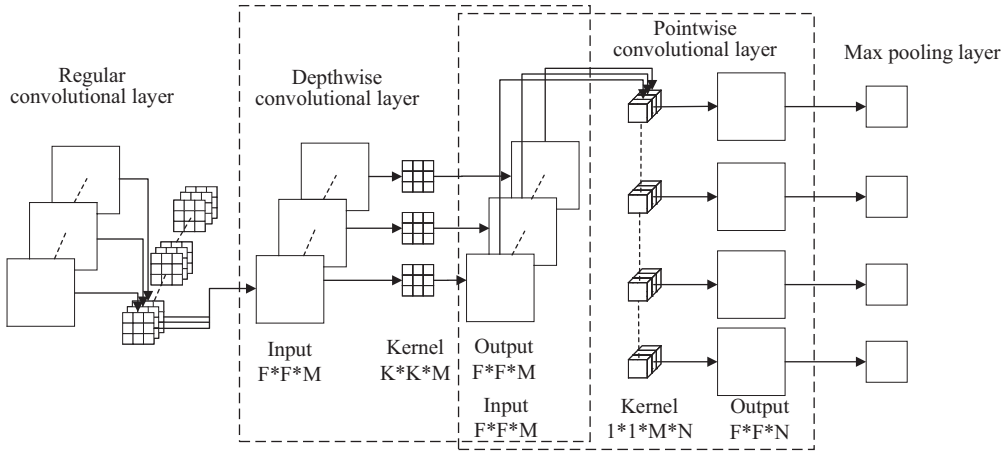

Figure 1: Principles of stacked convolutional network

The formulas for the DSC computation and parameter count are given in Eqs. (1)–(5). The input feature dimension was assumed to be denoted as $F * F * M$, the convolution kernel size as $K * K$, the output feature dimension as $F * F * N$, and M and N represent the numbers of input and output channels, respectively.

$$A1 = K * K * M * N * F * F \tag{1}$$

$$A2 = K * K * M * N \tag{2}$$

$$B1 = K * K * M * F * F + M * N * F * F \tag{3}$$

$$B2 = K * K * M + 1 * 1 * M * N \tag{4}$$

$$C1 = (1/N) + (1/K^2) \tag{5}$$

The calculation amount of the regular convolution is $A1$, and its parameter number is $A2$. For DSC, the calculation amount is $B1$, and the parameter number is $B2$. The ratio of the calculation amount and parameter number between DSC and regular convolution is $C1$. Therefore, the calculation amount and parameter number of DSC are $(1/N) + (1/K^2)$ of those in regular convolution, significantly reducing both the parameters.

The regular convolutional layer excels in extracting local features and capturing more complex feature combinations. The depth-wise and point-wise convolutional layers focus on efficiently integrating the cross-channel information. The max pooling layer reduces the input dimensions for subsequent layers while preserving the key feature information. An SCN composed of these network layers can enhance the network's feature-learning ability while maintaining a low parameter count.

3.1.2 Temporal Feature Extraction Module with BiLSTM

LSTM is commonly used in network traffic detection models because of its strong long-term memory capability [27]. By incorporating gate control mechanisms and cell states, such as the forgetting gate, input gate, and output gate, the LSTM network can regulate which information is retained or forgotten, thereby enabling it to preserve the long-term memory. The structure of the LSTM model is shown in Fig. 2.

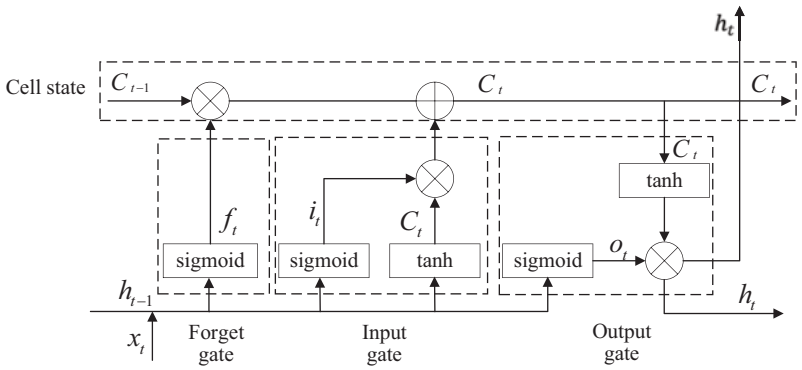

Figure 2: LSTM structure

The forget gate f_t is generated from the current input x_t and the hidden state h_{t-1} from the previous time step. It is responsible for eliminating redundant or irrelevant information and controlling information retained in the cell state C_{t-1}. The formula for calculating the forget gate is given by Eq. (6):

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{6}$$

The input gate i_t is generated from the current input x_t and the hidden state h_{t-1} from the previous time step. It selectively adds temporary information $\widetilde{C}_t$ to the cell state C_{t-1}. The calculation formulas for the input gate are shown in Eqs. (7) and (8).

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{7}$$

$$\widetilde{C}_t = tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{8}$$

The cell state C_t at the current moment is generated by combining the information $f_t \times C_{t-1}$ to be retained from the previous moment with the information $i_t \times \widetilde{C}_t$ to be added at the current moment. The calculation formula for the cell state is shown in Eq. (9).

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t \tag{9}$$

The output gate o_t filters the information C_t processed by the forget and input gates, outputs the hidden state h_t at the current time step and passes it to the next LSTM unit. The calculation formulas for the output gate are shown in Eqs. (10) and (11).

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{10}$$

$$h_t = o_t \times tanh\left(C_t\right) \tag{11}$$

where σ is the sigmoid function; W_f, W_i, W_o, and W_c respectively represent the weight matrices for the forget gate, input gate, output gate, and cell state; and b_f, b_i, b_o, and b_c denote the bias terms for the forget gate, input gate, output gate, and cell state, respectively.

Although LSTM is effective at capturing long-term dependencies, it can only handle one-way dependencies in time-series data and may not fully capture contextual information. To address this, this study introduced BiLSTM, which processed both forward and backward temporal dependencies in traffic data. This bidirectional mechanism allowed the BiLSTM to capture the temporal features across different timeframes and directions, thereby enhancing the detection capabilities of the model. The BiLSTM model is illustrated in Fig. 3.

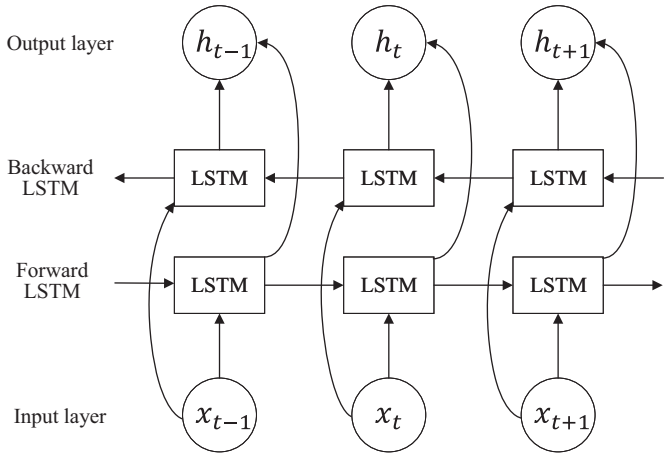

Figure 3: BiLSTM structure

By integrating forward and reverse LSTM units, BiLSTM considers both past and future contexts at each time step, generating richer hidden states and enhancing the understanding of the time-series data. The forward LSTM unit processes the data in the original time-series order, whereas the reverse LSTM unit analyzes the same series in reverse. Both units share the same input $(x_1, x_2, \ldots, x_t, \ldots)$ but operate independently, cooperatively generating hidden states at each moment. The hidden states from both directions are combined to form a comprehensive hidden state sequence $(h_1, h_2, \ldots, h_t, \ldots)$, which is then passed to the next layer of the model.

3.1.3 Attentional Mechanisms

The attention mechanism is a key technique in deep learning and has been widely applied in various fields in recent years. It enhances the detection rates and classification accuracy in complex tasks by assigning higher weights to key features, allowing for better capture of important information while disregarding irrelevant data.

The attention mechanism determines the importance of each value by calculating the dot product between the query (Q) and key set (K), applying the softmax function to transform the dot product into a probability distribution, and then obtaining the output through weighted summation. The query (Q) and key-value pair $(K-V)$ were derived by multiplying the input feature (X) by the respective weight matrices (W^Q, W^K, and W^V), which were learned automatically during the training phase of the model. The calculation formulas are shown in Eqs. (12)–(14).

$$Q = XW^Q \tag{12}$$

$$K = XW^K \tag{13}$$

$$V = XW^V \tag{14}$$

The formula for calculating the attention value is shown in Eq. (15).

$$Attention\left(Q,K,V\right) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)\cdot V \tag{15}$$

where K^T denotes the transpose of K, and d_k represents the dimensionality of the key vector. In addition, the scaling factor $\frac{1}{\sqrt{d_k}}$ is introduced. The purpose of scaling is to prevent the dot product operation in high-dimensional space from yielding excessively large values, which could cause the gradient of the softmax function to approach zero, thereby affecting numerical stability during the training process.

This study integrated the attention mechanism with SCN and BiLSTM to construct an SCN-BiLSTM feature extraction model, enhancing the ability of SCN to focus on the most important channels and spatial locations while improving the capacity of BiLSTM to understand the sequence context. The effectiveness of this approach was validated by subsequent experiments.

3.2 Overview of Traffic Classification Model

This section provides a detailed introduction to EQL v2 in the class imbalance-oriented classification detection model and intelligent detection framework for abnormal traffic based on SCN-BiLSTM.

3.2.1 Class Imbalance Processing

This study used the EQL v2 loss function to address the class imbalance issue. EQL v2 applies a gradient reweighting mechanism to balance the gradients of the positive and negative samples for each category. Adjusting the importance of different categories, particularly increasing the weight of samples from underrepresented classes, effectively mitigated the negative impact of class imbalance on the model's classification results.

The underlying concept of EQL v2 gradient reweighting is as follows. Each classifier's positive and negative gradients are independently weighted based on the cumulative gradient ratio between the positive and negative classes within the classifier.

The expressions for the positive and negative gradients concerning the output Z_j of each class in relation to the loss L are shown in Eqs. (16) and (17).

$$\nabla_{z_j}^{pos}\left(L\right) = \frac{1}{|I|}\sum_{i\in I} y_j^i\left(p_j^i - 1\right) \tag{16}$$

$$\nabla_{z_j}^{neg}\left(L\right) = \frac{1}{|I|}\sum_{i\in I}\left(1 - y^i{}_j\right)p^i{}_j \tag{17}$$

where p_j^i is the estimated probability of the i-th data sample in class j; Z_j is the output that is not activated by the sigmoid function; I is all data samples; and y^i is the one-hot ground truth label for the i-th data sample.

To actualize the concept of gradient-guided balanced reweighting, the ratio of the cumulative positive to negative gradients for class j up to iteration t was initially defined as ${g_j}^{(t)}$. During this iteration, the weight ${q_j}^{(t)}$ for the positive gradient and the weight ${r_j}^{(t)}$ for the negative gradient were calculated using Eqs. (18) and (19).

$$q_j^{(t)} = 1 + \alpha\left(1 - f\left(g_j^{(t)}\right)\right) \tag{18}$$

$$r_j^{(t)} = f\left(g_j^{(t)}\right) \tag{19}$$

where $f(\cdot)$ is the mapping function with $f(x) = \frac{1}{1+e^{-\gamma(x-\mu)}}$.

After determining the weights ${q_j}^{(t)}$ and ${r_j}^{(t)}$ for the positive and negative gradients, they were applied to the positive and negative gradients of the current batch. The results of the reweighted gradients are shown in Eqs. (20) and (21).

$$\nabla_{z_j}^{pos\prime}\left(L^{(t)}\right) = q_j^{(t)}\nabla_{z_j}^{pos}\left(L^{(t)}\right) \tag{20}$$

$$\nabla_{z_j}^{neg\prime}\left(L^{(t)}\right) = r_j^{(t)}\nabla_{z_j}^{neg}\left(L^{(t)}\right) \tag{21}$$

The cumulative ratio of the positive to negative gradients was then updated for the subsequent iteration, $t+1$, as shown in Eq. (22).

$$g_j^{(t+1)} = \frac{\sum_{t=0}^{T}\left|\nabla_{z_j}^{pos\prime}\left(L^{(t)}\right)\right|}{\sum_{t=0}^{T}\left|\nabla_{z_j}^{neg\prime}\left(L^{(t)}\right)\right|} \tag{22}$$

By employing this method, EQL v2 dynamically adjusts the ratio of positive to negative gradients during the training phase, allowing the model to learn more equitably from different categories of samples and enhance its ability to recognize minority classes.

3.2.2 Intelligent Detection Framework for Abnormal Traffic

Fig. 4 illustrates the intelligent detection framework for abnormal traffic based on SCN-BiLSTM architecture. The framework consists of three main components: a data processing module, a feature extraction module, and a classification detection module.

The data processing module involves data cleaning, one-hot encoding, and data normalization to ensure that the input data meet the requirements for model training. Data cleaning removes invalid entries from the dataset, one-hot encoding converts the non-numeric features into a format suitable for processing, and data normalization applies the min-max normalization method to scale all values and standardize the dataset.

The feature extraction module addresses the challenge of fully exploring the temporal and spatial features of traffic data by using the SCN-BiLSTM model integrated with an attention mechanism. This model efficiently extracts complex temporal and spatial features through deep learning of traffic data. First, an SCN combining regular convolutional and DSC layers can be applied to capture the spatial features of the input data. A pooling layer reduces the spatial dimensions of the features, thereby minimizing the network parameters and computation while mitigating the risk of overfitting. The attention mechanism is then employed to focus on key features and enhance model expressiveness without significantly increasing complexity. The results from the attention layer are fed into the BiLSTM layer, which consists of 64 neurons,

to capture contextual information and temporal dependencies from the input sequence. This bidirectional processing improves the ability of the model to detect complex patterns in data. Finally, the attention mechanism was reapplied after the BiLSTM layer to emphasize the most influential parts of the sequence, enhancing the model to understand the sequence context.

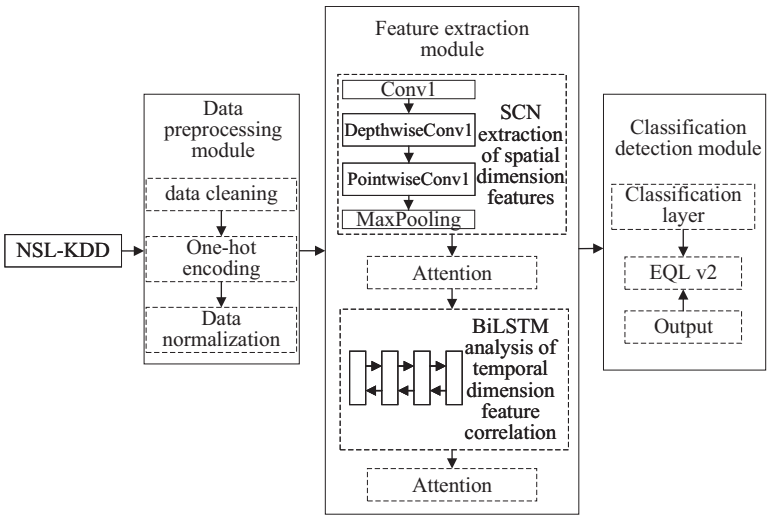

Figure 4: Intelligent detection framework for abnormal traffic based on SCN-BiLSTM

The classification detection module was designed to address the class imbalance issue in the dataset by implementing a class imbalance-oriented classification model. The output layer uses the Softmax function to convert the raw output into a probability distribution, whereas the EQL v2 loss function refines the predictions by calculating the loss value, performing backpropagation, and updating parameters.

4 Experiment and Results

In this paper, the experimental environment are running on a server with RTX 4090 GPU and 32 GB RAM using Python3.9 + TensorFlow 2.10.0 + Keras 2.10.0.

4.1 Experimental Data and Evaluation Metrics

The performance of network traffic anomaly detection is significantly influenced by the selection of the dataset. NSL-KDD, created by the Canadian Institute for Cybersecurity (CIC) in 2009, is widely used in research as a benchmark dataset for evaluating the effectiveness of intrusion detection technologies. Derived from KDD Cup 99 [28], it addresses the redundancy issues of the original data and covers four major attack types (DoS, Probe, R2L, U2R) as well as normal traffic. The class imbalance problem, such as U2R attacks accounting for only 0.04% of the data, rigorously evaluates the model's ability to detect minority classes. Additionally, the dataset includes detailed documentation and annotations, describing the data attributes, attack types, and ground truth labels. This information enhances the interpretability of results and enables researchers to better understand the strengths and limitations of their methods.

To train and validate the data and assess the effectiveness of the proposed approach, a Stratified 10-Fold Cross Validation methodology was employed with a training-to-test set ratio of 9:1. The distributions and ratios are listed in Table 1.

Table 1: Sample distribution and proportion of NSL-KDD dataset

Category	Training dataset	Teating dataset	Percentage (%)
Normal	121,217	13469	53.46
DoS	82,669	9185	36.46
Probe	20,980	2332	9.25
R2L	1791	199	0.79
U2R	94	10	0.04
Total	226,751	25195	100.00

To thoroughly evaluate the performance of the proposed method on an imbalanced dataset, various metrics were adopted, including Accuracy, Recall, FPR, and F1-score. These metrics were derived from the confusion matrix, as shown in Table 2.

Table 2: Confusion matrix

	Predicted positive	**Predicted negative**
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

TP refer to instances that are correctly predicted as positive, whereas FN are actual positives incorrectly predicted as negative. FP are instances incorrectly identified as positive when they are actually negative, and TN are correctly identified as negative.

The calculation formulas for each evaluation metric are provided in Eqs. (23)–(27):

Accuracy represents the proportion of correctly classified samples to the total number of samples, with a higher accuracy indicating better model performance.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{23}$$

Precision was defined as the ratio of correctly predicted positive samples to total number of predicted positive samples.

$$Precision = \frac{TP}{TP + FP} \tag{24}$$

FPR refers to the proportion of actual negative samples incorrectly predicted as positive. A lower FPR indicates better accuracy in predicting negative instances.

$$FPR = \frac{FP}{FP + TN} \tag{25}$$

Recall refers to the true-positive rate, representing the ratio of correctly identified positive samples to the total number of positive samples. A higher recall rate indicates better performance in capturing positive cases.

$$Recall = \frac{TP}{TP + FN} \tag{26}$$

The F1-score is a metric that balances precision and recall, offering a comprehensive evaluation of the trade-off between these two measures in the classification models. A higher F1-score indicates a more balanced and robust model.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{27}$$

4.2 Analysis of Experimental Results

To minimize experimental biases and errors, the dataset was preprocessed. And 10-Fold Cross Validation was employed during training, which reduces errors caused by random data partitioning through averaging multiple evaluations. Furthermore, the EQL v2 loss was utilized to automatically adjust the weights of hard-to-classify samples, thereby reduce the missed detection of minority classes.

To validate the effectiveness of the proposed approach, the experimental results were analyzed from three perspectives: (**A**) The impact of varying network architectures on the results. (**B**) The influence of the attention mechanism on the outcomes, and (**C**) A comparative analysis of the classification performance of the proposed method and several commonly used methods.

(**A**) The impact of varying network architectures on the results.

First, five distinct models (a), (b), (c), (d), and (e) were designed for multi-classification experiments (Fig. 5).

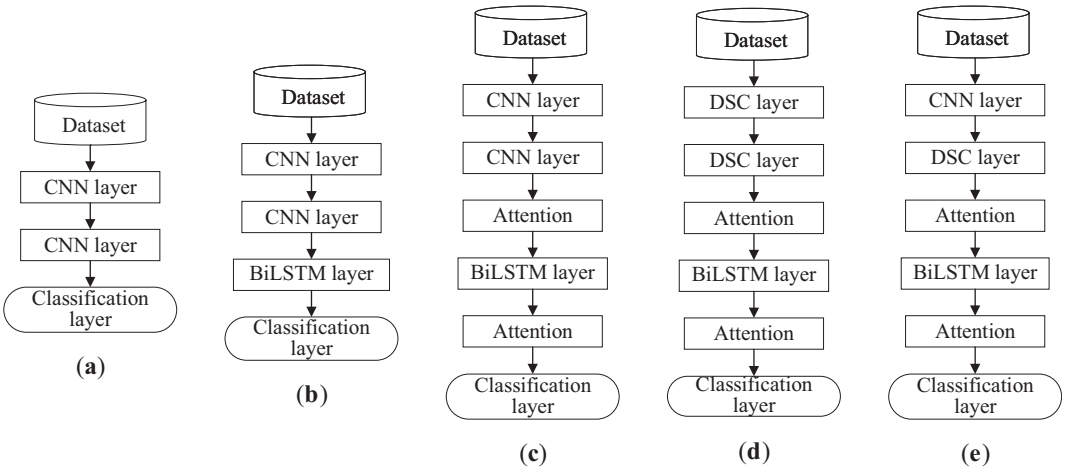

Figure 5: Five different model structures: **(a)** CNN and CNN in tandem; **(b)** CNN, CNN, and BiLSTM in tandem; **(c)** CNN, CNN, and BiLSTM with fused attention mechanism; **(d)** DSC, DSC, and BiLSTM with fused attention mechanism; **(e)** SCN and BiLSTM with fused attention mechanism

The detailed experimental results are presented in Table 3. An analysis of these results revealed that model (e) achieved an accuracy rate of 99.81%, recall rate of 99.70%, and F1-score of 99.81%, outperforming models (a), (b), (c), and (d). Model (e) replaced the second layer of the CNN with a DSC layer in the CNN-CNN-BiLSTM model by integrating the attention mechanism, as shown in model (c). A comparison of the parameters and training time for both models is provided in Table 4, demonstrating that the SCN combined with CNN and DSC in the model (e) significantly reduced the number of parameters and improved training efficiency. Consequently, this study adopted the SCN-BiLSTM feature extraction model (e), which integrates the attention mechanism, for temporal and spatial feature extraction.

Table 3: Experimental results for models (a), (b), (c), (d), and (e) and the models in this paper (%)

Model	Acc	Recall	F1-score	FPR
CNN-CNN (a)	97.83	96.30	97.54	0.46
CNN-CNN-BiLSTM (b)	99.80	99.65	99.80	0.05
CNN-CNN-BiLSTM integrated attention mechanism (c)	99.80	99.66	99.79	0.07
DSC-DSC-BiLSTM integrating attention mechanism (d)	99.77	99.70	99.77	0.15
SCN-BiLSTM integrating attention mechanism (e)	99.81	99.70	99.81	0.08
Ours	99.94	99.95	99.94	0.07

Table 4: Comparison of the number of model parameters and training time

Model	Parameters	Training time (min)
CNN-CNN-BiLSTM integrated attention mechanism (c)	333,573	564
SCN-BiLSTM integrating attention mechanism (e)	79,685	475

To evaluate the contribution of the EQL v2 loss function in addressing class imbalance on the NSL-KDD dataset, three different loss functions, including FL, Cross-Entropy (CE), and EQL v2, were tested on the same model (e). The experimental results presented in Table 5 and Fig. 6 indicated that the classification detection model using the EQL v2 loss function achieved precision rates of 99.92%, 99.97%, 99.95%, 97.98%, and 90.90% for the five classes (Normal, DoS, Probe, R2L, and U2R), respectively, outperforming the models using other loss functions. Notably, the precision improvements for minority classes R2L and U2R were particularly significant. These results demonstrated that the EQL v2 loss function effectively enhanced precision in detecting rare abnormal traffic types. Therefore, EQL v2 was adopted as the loss function for the classification detection model in this study to address the unbalanced categories.

Table 5: Comparison of detection results of different loss functions for five data types (%)

Model	Normal	DoS	Probe	R2L	U2R	Acc	Recall	F1-score	FPR
SCN-BiLSTM integrating attention mechanism+FL	99.91	99.95	99.31	93.46	70.00	99.81	99.70	99.81	0.08
SCN-BiLSTM integrating attention mechanism+CE	99.92	99.85	99.69	92.96	80.00	99.82	99.70	99.82	0.06
Ours	99.92	99.97	99.95	97.98	90.90	99.94	99.95	99.94	0.07

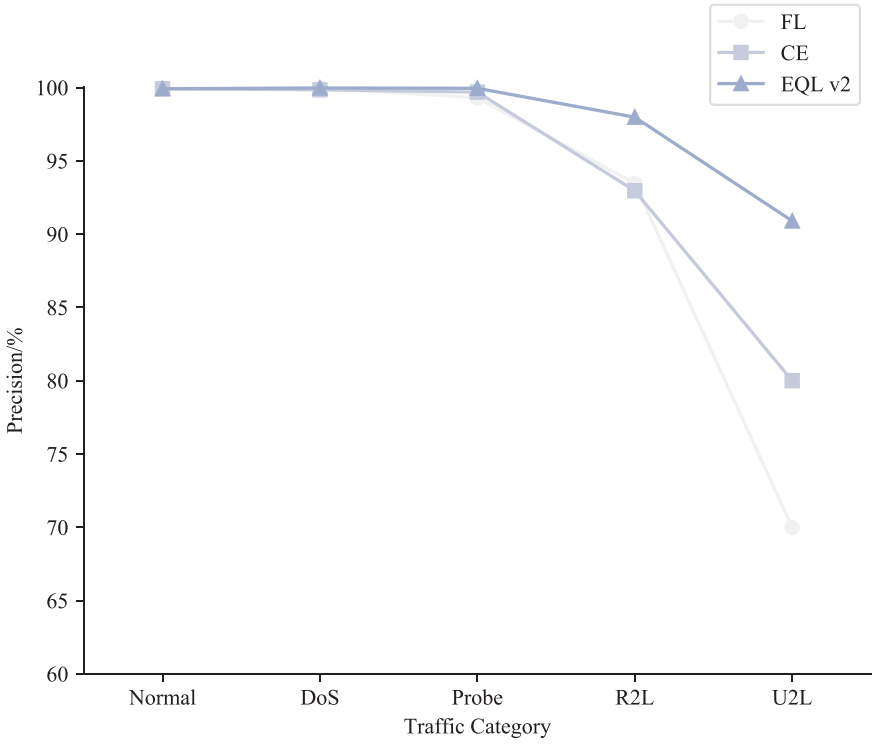

Figure 6: Precision of different loss functions for five data types

The intelligent detection method for abnormal traffic based on SCN-BiLSTM proposed in this study integrated the SCN-BiLSTM feature extraction model with an attention mechanism and an unbalanced category-oriented classification detection model. This method achieved an accuracy rate of 99.94%, recall rate of 99.95%, F1-score of 99.94%, and FPR of 0.07%. Compared with model (b), this approach exhibited a maximum increase in accuracy of 2.11%, recall of 3.65%, and F1-score of 2.4%. Although the FPR was slightly higher by 0.02% than that of the model (b), these results demonstrated the strong detection capability of the proposed method on the NSL-KDD dataset.

(**B**) The influence of the attention mechanism on the outcomes.

To evaluate the significance of the attention mechanism, this study compared the results of multiclass classification with and without the attention mechanism. The experimental results presented in Table 6 demonstrated that the inclusion of the attention mechanism improved the accuracy, recall, and F1-score by 1.6%, 3.41%, and 2%, respectively, while FPR decreased by 0.01%. These findings indicated that the attention mechanism significantly enhanced the overall detection performance of the proposed method.

Table 6: Effect of attentional mechanisms on experimental results (%)

Availability of attention mechanism	Acc	Recall	F1-score	FPR
Y	99.94	99.95	99.94	0.07
N	98.34	96.54	97.94	0.08

(**C**) A comparative analysis of the classification performance of the proposed method and several commonly used methods.

The efficacy of the proposed method was evaluated and compared with eight commonly used techniques, including KNN, RF, SVM, ICVAE-DNN [29], CD-2 [30], I-SiamIDS [31], 1DCNN-BiLSTM [32], and CANET [33], using the NSL-KDD dataset. The results of comparing the multiclass classification performance of the proposed method with other methods on the NSL-KDD dataset are presented in Table 7 and Figs. 7–9.

Table 7: Comparison of detection results of different models for five data types (%)

Model	Normal	DoS	Probe	R2L	U2R	Acc	Recall	F1-score	FPR
KNN	92.78	82.25	59.40	3.56	3.50	76.51	64.19	75.68	7.22
RF	97.37	80.24	58.53	7.55	0.50	76.49	60.69	74.62	2.63
SVM	92.82	74.85	61.71	0.00	0.00	72.28	56.73	69.97	7.18
ICVAE-DNN	87.04	77.87	79.89	23.17	11.50	75.43	72.86	82.92	12.96
CD-2	85.00	88.00	69.00	78.00	55.00	83.00	65.00	68.00	–
I-SiamIDS	81.52	82.09	74.50	65.62	37.78	80.00	67.44	66.54	6.16
1DCNN-BiLSTM	95.47	98.21	74.55	80.08	79.05	–	89.17	87.09	2.18
CANET	99.90	99.94	99.52	92.46	70.00	99.81	99.71	99.79	0.09
Ours	99.92	99.97	99.95	97.98	90.90	99.94	99.95	99.94	0.07

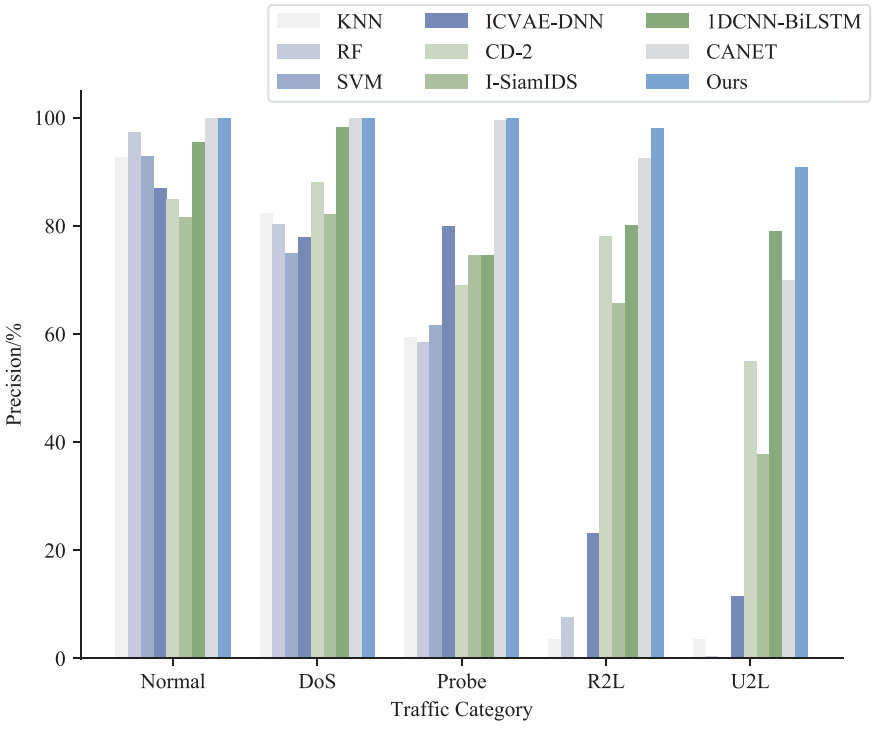

Figure 7: Precision of different methods for five data types

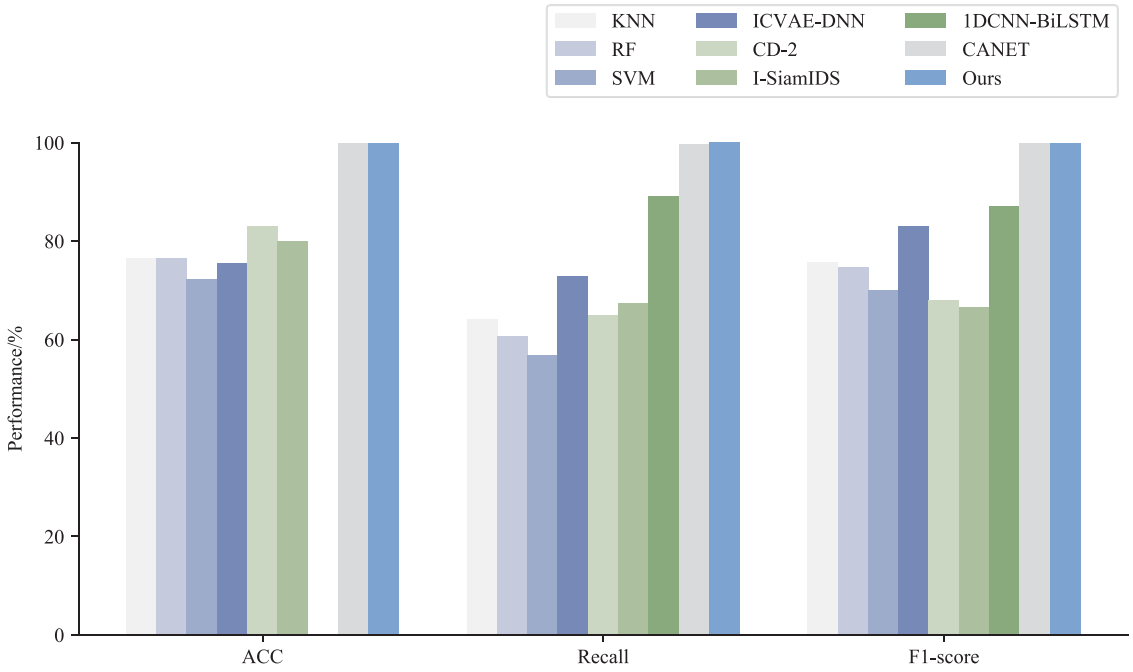

Figure 8: Comparison of ACC, Recall, and F1-score results on NSL-KDD by different methods

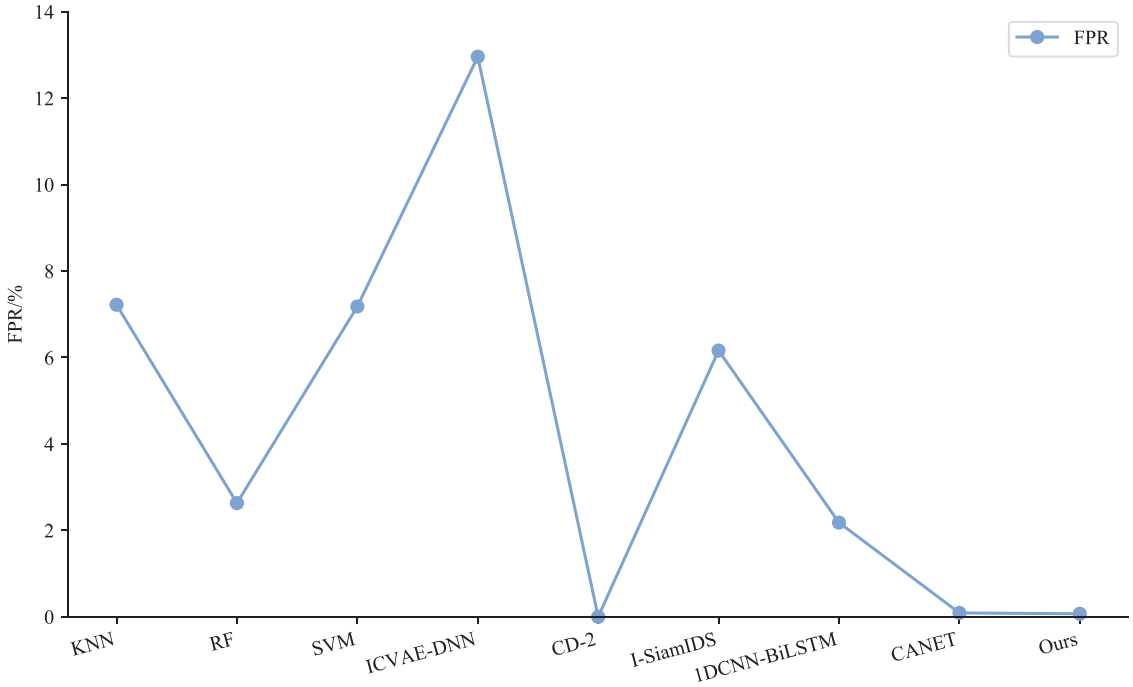

Figure 9: Comparison of FPR results of different methods on NSL-KDD

The intelligent detection method for abnormal traffic based on SCN-BiLSTM proposed in this study achieved optimal accuracy, recall, F1-score, and FPR on the NSL-KDD dataset. Compared with other alternative approaches, the proposed method improved the accuracy by 0.13%–27.66%, recall by 0.24%–43.22%, and F1-score by 0.15%–33.40%, while reducing the FPR by 0.02%–12.89%. These results demonstrated that

the proposed method not only enhanced the detection accuracy but also maintained a low FPR, further validating its efficacy in abnormal traffic detection.

The experimental results demonstrated that this approach effectively captured the features of network traffic data, enabled more accurate detection of network intrusion behavior, and enhanced the performance of abnormal traffic detection.

4.3 Threats to Validity

In this section, we have identified some threats to the validity of our research. (1) Although NSL-KDD is widely used in traffic anomaly detection, it is derived from data collected under specific conditions and has inherent limitations. For instance, its attack types and network environments are relatively fixed, which may not encompass emerging attack types in modern networks. Additionally, the traffic features in the dataset may differ from those in real-world networks, potentially affecting the model's detection performance in practical applications. (2) During model training, hyperparameters were adjusted to achieve optimal learning performance. However, hyperparameter tuning risks overfitting, where the model performs well on the training set but fails to generalize to new datasets or real-world scenarios. Moreover, hyperparameter selection impacts the model's generalization ability, as different settings may lead to varying performance across datasets. (3) Our research is based on a specific network environment and attack types, which may differ from real-world conditions. Factors such as network topologies, protocols, and user behaviors can influence the performance of traffic anomaly detection models. Furthermore, as network technologies and attack methods evolve, new attack types may emerge, requiring updates and adjustments to our model to maintain its effectiveness.

5 Conclusion

The application of deep learning models to abnormal network traffic detection has become a growing trend. This study leveraged the advantages of the SCN-BiLSTM feature extraction model integrated with an attention mechanism and class-oriented unbalanced classification detection model using EQL v2. A combination of these methods was applied to an intelligent detection framework for abnormal traffic to effectively extract both the spatial and temporal features of traffic data. This approach enhanced detection accuracy, reduced the computational complexity, and improved the model training efficiency. In addition, it significantly boosted the detection accuracy of minority class samples, thereby addressing the class imbalance problem in the dataset. However, this method has only been tested on the NSL-KDD dataset, and network anomaly traffic detection usually needs to be conducted in a real-time environment, which requires anomaly traffic detection methods based on deep learning to respond within milliseconds to prevent the spread of potential attacks. To address these limitations, future work will prioritize the following directions: (1) Developing models using traffic datasets with realistic samples and diverse scenarios to improve generalizability. (2) Exploring optimization techniques such as model compression, parameter pruning, knowledge distillation, and edge computing to enhance real-time detection capabilities, ensuring adaptability to high-throughput and low-latency network environments.

Acknowledgement: The authors thank all research members who provided support and assistance in this study.

Funding Statement: This work was supported by the National Natural Science Foundation of China (Grant No. 62102449).

Author Contributions: Research conception and design: Lulu Zhang; Data collection: Lulu Zhang, Xuehui Du; Result analysis and interpretation: Lulu Zhang, Wenjuan Wang; Manuscript preparation: Lulu Zhang, Xuehui Du, Yu Cao, Xiangyu Wu, Shihao Wang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The dataset used in this paper is the public dataset NSL-KDD dataset, accessed as follows: https://github.com/HoaNP/NSL-KDD-DataSet (accessed on 13 April 2025).

Ethics Approval: Not applicable.

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

References

1. Fu Y, Wang K, Duan XY, Liu TT. A review of research on anomaly traffic detection for software-defined networks. J Communicat. 2024;45(3):208–26. (In Chinese).
2. Zhou JY, He PF, Qiu RF, Chen G, Wu WG. Research on intrusion detection by fusing random forest and gradient boosting tree. J Softw. 2021;32(10):3254–65. (In Chinese).
3. Ding H, Chen L, Dong L, Fu Z, Cui X. Imbalanced data classification: a KNN and generative adversarial networks-based hybrid approach for intrusion detection. Future Gener Comput Syst. 2022;131(7):240–54. doi:10.1016/j.future.2022.01.026.
4. Jyoti DK, Prakash VS, Vinay K. A novel adaptive optimization framework for SVM hyper-parameters tuning in non-stationary environment: a case study on intrusion detection system. Expert Syst Appl. 2023;213(2):119189. doi:10.1016/j.eswa.2022.119189.
5. Wang S, Jin Z. Intrusion detection classification algorithm based on fuzzy SVM model. Applicat Res Comput. 2020;37(2):501–4. (In Chinese).
6. Panigrahi R, Borah S, Pramanik M, Bhoi AK, Barsocchi P, Nayak SR, et al. Intrusion detection in cyber-physical environment using hybrid Naïve Bayes—Decision table and multi-objective evolutionary feature selection. Comput Commun. 2022;188(21):133–44. doi:10.1016/j.comcom.2022.03.009.
7. Liu J, Yinchai W, Siong TC, Li X, Zhao L, Wei F. A hybrid interpretable deep structure based on adaptive neuro-fuzzy inference system, decision tree, and K-means for intrusion detection. Sci Rep. 2022;12(1):20770. doi:10.1038/s41598-022-23765-x.
8. Akgun D, Hizal S, Cavusoglu U. A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. Comput Secur. 2022;118(1):102748. doi:10.1016/j.cose.2022.102748.
9. Yang Z, Liu X, Li T, Wu D, Wang J, Zhao Y, et al. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Comput Secur. 2022;116(4):102675. doi:10.1016/j.cose.2022.102675.
10. Sharma B, Sharma L, Lal C, Roy S. Explainable artificial intelligence for intrusion detection in IoT networks: a deep learning based approach. Expert Syst Appl. 2024;238(1):121751. doi:10.1016/j.eswa.2023.121751.
11. Rizvi S, Scanlon M, McGibney J, Sheppard J. Deep learning based network intrusion detection system for resource-constrained environments. In: International Conference on Digital Forensics and Cyber Crime 2022; 2022 Nov 16–18; Boston, MA, USA. p. 355–67.
12. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021; 2021 Jun 20–25; Nashville, TN, USA. p. 1685–94.
13. Ilyas MU, Alharbi SA. Machine learning approaches to network intrusion detection for contemporary internet traffic. Computing. 2022;104(5):1061–76. doi:10.1007/s00607-021-01050-5.
14. Saba T, Rehman A, Sadad T, Kolivand H, Bahaj SA. Anomaly-based intrusion detection system for IoT networks through deep learning model. Comput Electr Eng. 2022;99(5):107810. doi:10.1016/j.compeleceng.2022.107810.
15. Wang W, Zhu M, Wang J, Zeng X, Yang Z. End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI); 2017 Jul 22–24; Beijing, China. p. 43–8.

16. Cai Z, Xiong Z, Xu H, Wang P, Li W, Pan Y. Generative adversarial networks: a survey toward private and secure applications. CM Comput Surv. 2021;54(6):132. doi:10.1145/3459992.
17. Li Z, Qin Z, Huang K, Yang X, Ye S. Intrusion detection using convolutional neural networks for representation learning. In: Liu D, Xie S, Li Y, Zhao D, El-Alfy ES, editors. Neural information processing. ICONIP 2017. Lecture notes in computer science. Cham, Switzerland: Springer International Publishing; 2017. p. 858–66.
18. Khan RU, Zhang X, Alazab M, Kumar R. An improved convolutional neural network model for intrusion detection in networks. In: 2019 Cybersecurity and Cyberforensics Conference (CCC); 2019 May 8–9; Melbourne, VIC, Australia. p. 74–7.
19. Donkol H, Hussein M. Optimization of intrusion detection using likely point PSO and enhanced LSTM-RNN hybrid technique in communication networks. IEEE Access. 2023;11:9469–82. doi:10.1109/ACCESS.2023.3240109.
20. Imrana Y, Xiang Y, Ali L, Abdul-Rauf Z. A bidirectional LSTM deep learning approach for intrusion detection. Expert Syst Appl. 2021;185(8):115524. doi:10.1016/j.eswa.2021.115524.
21. 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.
22. Dong S, Xia Y, Wang T. Network abnormal traffic detection framework based on deep reinforcement learning. IEEE Wirel Commun. 2024;31(3):185–93. doi:10.1109/MWC.011.2200320.
23. Wei W, Chen Y, Lin Q, Ji J, Wong K, Li J. Multi-objective evolving long-short term memory networks with attention for network intrusion detection. Appl Soft Comput. 2023;139(3):110216. doi:10.1016/j.asoc.2023.110216.
24. Dina AS, Siddique AB, Manivannan D. A deep learning approach for intrusion detection in Internet of Things using focal loss function. Intern Things. 2023;22(1):100699. doi:10.1016/j.iot.2023.100699.
25. Tan J, Wang C, Li B, Li Q, Ouyang W, Yin C, et al. Equalization loss for long-tailed object recognition. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020 Jun 13–19; Seattle, WA, USA. p. 11662–71.
26. ElSayed MS, Le-Khac NA, Albahar MA, Jurcut A. A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique. J Netw Comput Appl. 2021;191(4):103160. doi:10.1016/j.jnca.2021.103160.
27. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. doi:10.1162/neco.1997.9.8.1735.
28. Bala R, Nagpal R. A review on kdd cup99 and nsl-kdd dataset. Int J Adv Comput Res. 2019;10(2):64–7. doi:10.26483/ijarcs.v10i2.6395.
29. Yang Y, Zheng K, Wu C, Yang Y. Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Sensors. 2019;19(11):2528. doi:10.3390/s19112528.
30. Yoo J, Min B, Kim S, Shin D, Shin D. Study on network intrusion detection method using discrete pre-processing method and convolution neural network. IEEE Access. 2021;9:142348–61. doi:10.1109/ACCESS.2021.3120839.
31. Bedi P, Gupta N, Jindal V. I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems. Appl Intell. 2021;51(2):1133–51. doi:10.1007/s10489-020-01886-y.
32. Yin ZN, Ma HL, Hu T. Traffic anomaly detection method based on joint attention mechanism and one-dimensional convolutional neural network-bidirectional long and short-term memory network model. J Electr Inform Technol. 2023;45(10):3719–28. (In Chinese).
33. 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.