Information Dynamics and Applications
https://www.acadlore.com/journals/IDA

Enhanced Method for Monitoring Internet Abnormal Traffic Based on the Improved BiLSTM Network Algorithm

Li Yan[1*], Hongzhang Han[2], Zhong Li[3]

[1] School of Information Engineering, Changzhou Vocational Institute of Industry Technology, 213164 Changzhou, China
[2] School of Computer Engineering, Jiangsu University of Technology, 213001 Changzhou, China
[3] Scientific Research Department, Jiangsu Joint Institute of Changzhou Liuguojun Branch, 213000 Changzhou, China

* Correspondence: Li Yan (yanli@ciit.edu.cn)

Received: 10-08-2024 **Revised:** 11-06-2024 **Accepted:** 11-20-2024

Citation: L. Yan, H. Z. Han, and Z. Li, "Enhanced method for monitoring Internet abnormal traffic based on the improved BiLSTM network algorithm," *Inf. Dyn. Appl.*, vol. 3, no. 4, pp. 211–222, 2024. https://doi.org/10.56578/ida030401.

Abstract: The complexity and variability of Internet traffic data present significant challenges in feature extraction and selection, often resulting in ineffective abnormal traffic monitoring. To address these challenges, an improved Bidirectional Long Short-Term Memory (BiLSTM) network-based approach for Internet abnormal traffic monitoring was proposed. In this method, a constrained minimum collection node coverage strategy was first applied to optimize the selection of collection nodes, ensuring comprehensive data coverage across network nodes while minimizing resource consumption. The collected traffic dataset was then transformed to enhance data validity. To enable more robust feature extraction, a combined Convolutional Neural Network (CNN) and BiLSTM model was employed, allowing for a comprehensive analysis of data characteristics. Additionally, an attention mechanism was incorporated to weigh the significance of attribute features, further enhancing classification accuracy. The final traffic monitoring results were produced through a softmax classifier, demonstrating that the proposed method yields a high monitoring accuracy with a low false positive rate of 0.2, an Area Under the Curve (AUC) of 0.95, and an average monitoring latency of 5.7 milliseconds (ms). These results indicate that the method provides an efficient and rapid response to Internet traffic anomalies, with a marked improvement in monitoring performance and resource efficiency.

Keywords: Internet; Abnormal traffic monitoring; Improved BiLSTM network; Attention mechanism

1 Introduction

In today's world, the Internet has become an indispensable part of people's lives with its popularity and development. From online shopping and business exchanges to cloud computing and data storage, various data and businesses are continuously transmitted on the network, building an inseparable digital world [1]. However, it is this convenient and efficient Internet world that also has many security risks and threats, such as Distributed Denial of Service (DDoS) attacks, botnets, malware, etc., which can destroy network communications, steal sensitive information, and seriously affect the security of the network. For website owners, network administrators and ordinary users, protecting network security has become an urgent and important task. For network operators, cloud service providers and large enterprises, it is crucial to maintain the stability of the network and business [2, 3]. Abnormal traffic may affect the normal operation of the business and bring economic losses and reputation risks to the enterprise. Therefore, in the modern Internet environment, abnormal traffic monitoring has become an important means to ensure network security and business stability. Through the real-time monitoring and analysis of network traffic, abnormal conditions can be found in time, and corresponding measures can be taken quickly to ensure the normal operation of the network. At the same time, with the development of cloud computing and other technologies, the amount of network traffic data has increased sharply, and the traditional anomaly detection methods have been unable to meet the needs of real-time and accurate monitoring. Therefore, it is necessary to study and develop more efficient and intelligent anomaly traffic monitoring methods [4]. In addition, with the wide application of new technologies such as the Internet of Things (IoT) and 5G, the requirements for network stability and security are becoming higher and

higher. Therefore, it is of great significance to strengthen the research and application of Internet abnormal traffic monitoring methods. In this context, people from all walks of life put forward higher requirements on how to better monitor and deal with the problem of abnormal network traffic.

Qiu and Wang [5] proposed an IoT abnormal traffic monitoring method based on deep learning in an edge computing environment. Firstly, the data was preprocessed by data cleaning, normalization, oversampling and undersampling, and dataset segmentation to obtain a dataset with balanced data distribution. Secondly, the feature information calculation method based on data increment was used to extract feature information from a dynamic data stream. Finally, the CNN was used to extract the local features of the data, and the Bidirectional Gated Recurrent Unit (BiGRU) was used to extract the correlation of long series of data. The two networks worked together to complete the final abnormal traffic monitoring. However, data cleaning, normalization, oversampling, undersampling and other processing operations in this method may introduce additional time and computational costs. In the face of large-scale Internet traffic, data preprocessing may become time-consuming and difficult to expand, which limits the application of the method in actual large-scale scenarios. Duan et al. [6] proposed a network traffic anomaly monitoring method based on the multi-scale residual classifier. A sliding window was used to divide the network traffic into sub-sequences with different observation scales. The wavelet transform technology was used to obtain the time-frequency information of each sub-sequence on multiple decomposition scales. The stacked automatic encoder (SAE) was designed to learn the distribution of the input data. The constructed feature space was used to calculate the reconstruction error vector, and the multi-path residual group was used to learn the characteristic information of different scales in the reconstruction error vector. The traffic anomaly monitoring was completed by the lightweight classifier. Although this method uses SAE to learn the distribution of input data and uses the multipath residual group to learn the feature information, there may be some feature patterns that are difficult to capture for complex network traffic data. This may result in inaccurate identification of abnormal traffic in some cases. Liu et al. [7] used the Graph Neural Network (GNN) with the MLP to detect abnormal traffic in the IoT through the distributed anomaly detection module of the perceptron. However, in the Internet environment, the relationship between devices is complex and diverse, and the existing GNN model may not be able to fully express the complex relationship between nodes, resulting in the omission of the final monitoring results and an unsatisfactory monitoring effect. Moreira et al. [8] proposed a new monitoring method to effectively monitor malicious activities and traffic on the Internet. This method combines CNN and reinforcement learning (RL) technology to achieve an intelligent and adaptive packet sampling rate in the high-performance network interface to complete the Internet abnormal traffic monitoring. This method reduces the cost of the monitored entity, but the training of the RL model usually has certain instability, especially in the dynamic internet environment. There are a variety of abnormal conditions and traffic patterns, and it may be difficult for the training model to generalize to different scenarios or new abnormal types, resulting in limited monitoring effect.

Therefore, based on the above research, in order to improve the effect of Internet abnormal traffic monitoring, an Internet abnormal traffic monitoring method based on the improved BiLSTM network algorithm was proposed to improve the intelligent and automatic management level of the Internet and promote the sustainable and healthy development of the Internet.

2 Collection of the Internet Traffic Data

The monitoring of security vulnerabilities on the Internet is carried out before the program runs, mainly aiming for the source code monitoring of the network device program. In the actual industrial control environment, relevant methods can be directly used to monitor code fragments. However, in the process of traffic monitoring, it is necessary to collect and analyze the traffic data generated in the environment in real time [9] because there are many terminals and network device nodes on the Internet environment. In order to ensure the monitoring of abnormal traffic on the Internet, the traffic data collection method on the Internet was first designed in this study. The structure of traffic data collection on the Internet is shown in Figure 1.

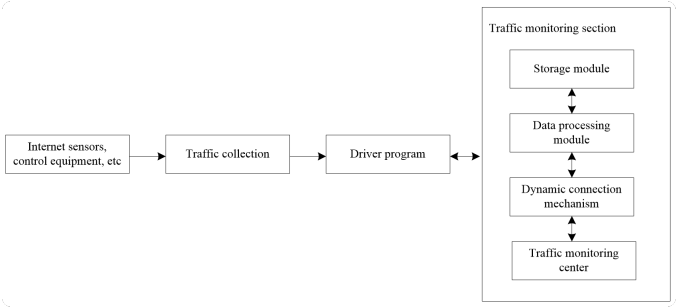

Figure 1. Schematic diagram of traffic data collection structure on the Internet

The traffic data collection on the Internet mainly includes three parts: the traffic collection part, the driving part and the traffic monitoring part. Firstly, the flow data generated by the sensors and control devices on the Internet is collected by the flow collection device. Secondly, the collected flow data is transferred to the storage module in the system through the driver to facilitate the review of abnormal flow data. Then it is transferred to the data processing module to preliminarily process the flow data and convert its data format. Finally, the processed results are uploaded to the flow monitoring center through the dynamic connection mechanism to realize the real-time monitoring of abnormal flow on the Internet.

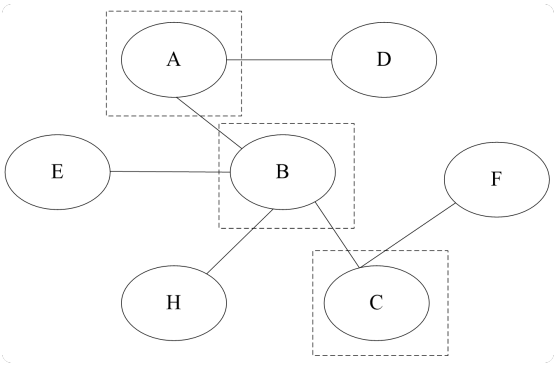

Figure 2. Minimum acquisition node coverage strategy

In the process of traffic data collection, the collection node should be selected first. The collection optimization of the Internet nodes starts from the spatial dimension and can be understood as a minimum vertex coverage problem. The constraint condition is to set the sampling weight at the response time. The minimum collection node coverage strategy based on the constraint condition is shown in Figure 2.

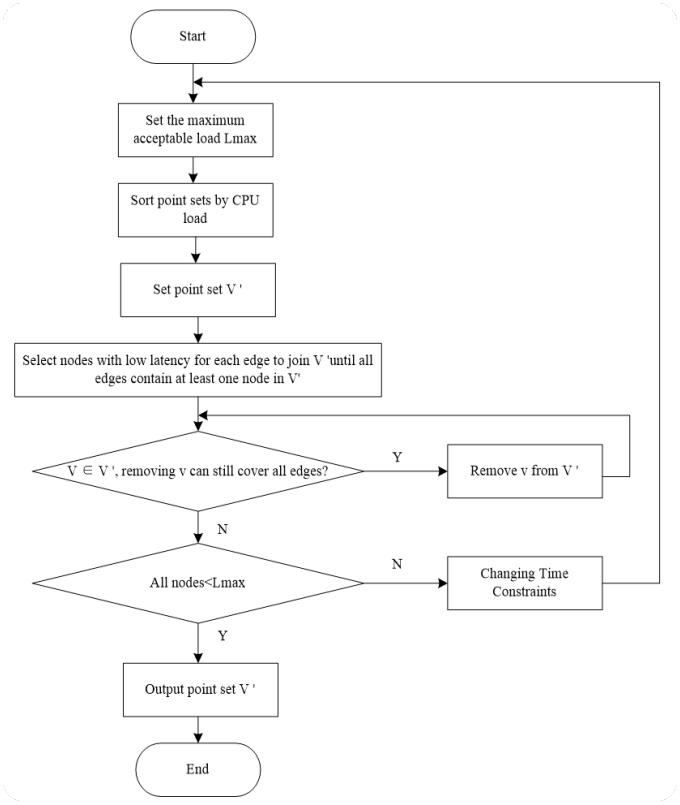

Figure 3. Flow chart of selecting the Internet traffic data collection nodes

In Figure 2, the traffic data of seven nodes can be collected. However, due to the connectivity between Internet devices, the information of device D is included in the connected device A. If the two devices are collected separately, a large amount of redundant information can appear, causing a waste of computer resources. In order to ensure that the traffic data of all nodes on the Internet can be collected and the consumption of resources is small, a point set

that includes all device connection links can be found. In the above figure, to get the traffic data of all nodes, the traffic data of devices A, B and C can only be collected, forming the minimum collection node coverage strategy. In the process of Internet device traffic collection, different nodes consume different time to collect traffic data. This difference is caused by the different real-time Central Processing Unit (CPU) loads of different nodes. In order to realize the rapid collection of traffic information on the Internet, traffic data from nodes with low CPU loads should be collected.

To sum up, the minimum vertex coverage strategy can be described as finding the smallest point set in figure $G = \{V, E\}$, so that each edge e in E has at least one vertex in V', and each vertex in point set V' meets the load constraint [10], thus completing the selection of acquisition nodes on the Internet. The specific process is shown in Figure 3.

When collecting abnormal Internet traffic, a connection network needs to be built first, ensuring that all nodes that need to collect information are covered. Then, the traffic data of which nodes do not need to be collected was filtered through the connection relationship between nodes, and these nodes were excluded. If all nodes cannot be excluded after filtering, whether these nodes meet the collection time constraints should be judged. If the conditions are met, the minimum node coverage scheme can be obtained. If the conditions are not met, it is necessary to modify the time constraint and repeat the above process until a satisfactory node coverage scheme is found. Finally, the output collection of Internet traffic data contains nodes that meet the time constraints.

3 Design of the Internet Abnormal Traffic Monitoring Method Based on the Improved BiLSTM Network Algorithm

3.1 Internet Traffic Data Conversion Processing

In order to ensure the validity of the collected data, the preprocessing was carried out, mainly including digitization, normalization and standardization. In order to increase the processability of the data and make the data more standardized and operable in the analysis process, the attribute mapping method was used to carry out the digitization of the collected Internet traffic data. Then normalization was conducted. Because there are differences in different numerical ranges, and if there are abnormal values (extreme values) in the data, their existence can cause great interference to subsequent operations. Normalization processing can limit the value range of all values to a small range to avoid the influence of abnormal values and improve the stability of subsequent operations [11–13]. Therefore, the data were normalized to be in the interval [0,1] as follows:

$$y = \frac{x - m_i}{m_a - m_i} \tag{1}$$

where, x is the attribute of the intrusion in the dataset, and m_a and m_i are the attributes with the maximum and minimum values, respectively.

Finally, standardization was conducted. In order to make the data have a unified scale and be easier to understand and process, Eq. (2) was used to complete the standardization process, providing a better basis for subsequent analysis.

$$y' = \frac{y - AVG_j}{STAD_j} \tag{2}$$

where, AVG_j is the average value, and $STAD_j$ is the average absolute deviation.

Thus, the Internet traffic data conversion was completed through the above processing, and the data conversion process is shown in Figure 4.

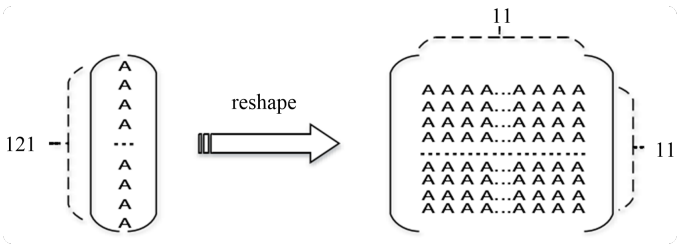

Figure 4. Schematic diagram of the data conversion process

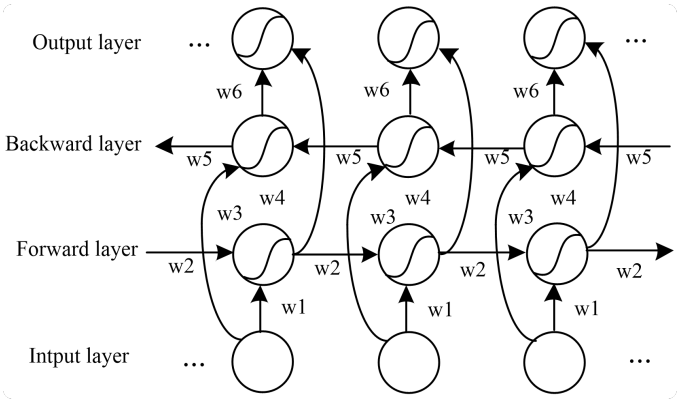

Figure 5. Partial deployment of the BiLSTM network

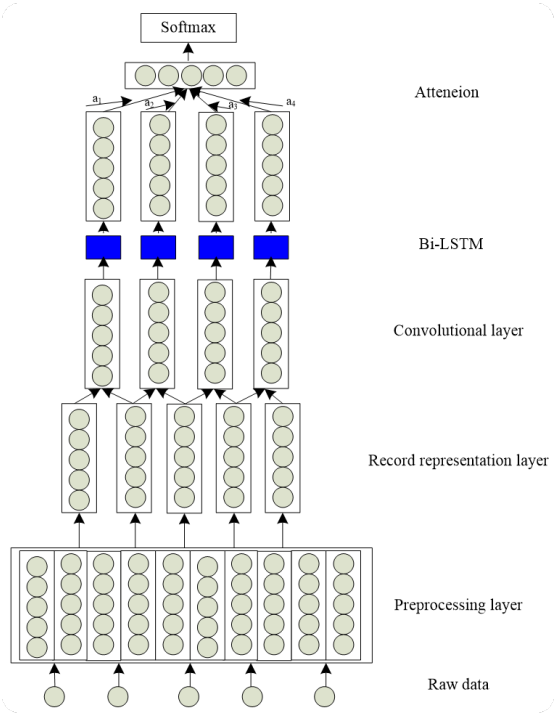

Figure 6. Model structure

3.2 Abnormal Traffic Monitoring Using the CNN-BiLSTM Network Algorithm with Attention Mechanism

3.2.1 BiLSTM model structure

The BiLSTM model is composed of forward and reverse LSTM. When processing sequence data, the partial BiLSTM model expands along the time axis, as shown in Figure 5.

In the figure, W_2 and W_5 are the weights of forward and reverse feature extraction, respectively. In the BiLSTM model, after preprocessing and encoding the data through the input layer, the data enters two parallel LSTM hidden layers at the same time: the forward and reverse LSTM hidden layers. The forward LSTM hidden layer is responsible for feature extraction from front to back in the order of the input sequence. It captures the forward dependency of each attribute point in the sequence by analyzing the sequence information before the current position. The reverse LSTM hidden layer takes the opposite direction, extracts features from the back to the front, and pays attention to the sequence information after each attribute point, thereby capturing the backward dependency. Both forward and backward dependencies can be effectively captured and integrated, which improves the model's ability to understand and process sequential data. Finally, the features extracted from the forward and reverse LSTM hidden layers are spliced together and passed to the output layer. The output layer performs the final prediction or classification task according to these characteristics. Through the structure design and feature extraction of the BiLSTM model, the information in the sequence data can be better used to improve the performance and accuracy of the model. However, BiLSTM is mainly good at capturing long-term dependencies in time series data, and its ability to extract

local features in data is relatively weak [14, 15]. Therefore, in order to improve the comprehensiveness of feature extraction, CNN was introduced to help the BiLSTM model extract data features more comprehensively and improve the performance of the model in abnormal traffic detection. In abnormal traffic monitoring, some specific features may play a decisive role in discovering abnormal traffic. Therefore, in order to further improve the accuracy of abnormal traffic monitoring and reduce the false alarm rate, an attention mechanism was introduced into the CNN-BiLSTM network to help the model learn and pay attention to more important features in the data, avoid the model being submerged by a large amount of irrelevant information, and improve the ability of the model to identify and use key information [16], thereby better adapting to abnormal traffic monitoring tasks in different scenarios.

3.2.2 Abnormal traffic monitoring based on the improved BiLSTM network algorithm

The structure of the CNN-BiLSTM model with attention mechanism is shown in Figure 6.

Based on Figure 6, the specific implementation process of Internet abnormal traffic monitoring is described below.

After collecting the Internet traffic data according to Section 2, the data is processed according to Section 3.1. Suppose the data sequence after preprocessing is $s = [y'_1, y'_2, \ldots, y'_n]$ as input, $v_{y'_n}$ is the embedded representation of y'_n in s, and the final embedded representation is $V_s = \lfloor v_{y'_1}, v_{y'_2}, \ldots, v_{y'_n} \rfloor$.

Then, the convolution layer and pooling layer in CNN are used for processing [17, 18]. After recording and representation, the result is input to the convolution layer and the convolution operation is performed on V_s. The convolution principle is shown in Figure 7.

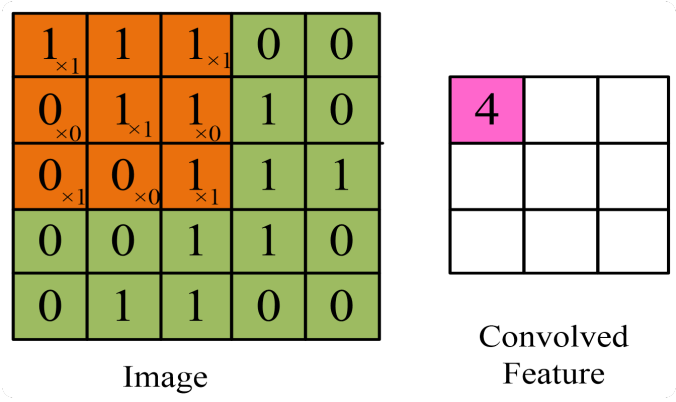

Figure 7. Schematic diagram of the working principle of the convolution layer

In the figure, yellow indicates that the convolution kernel size is $3 * 3$, and green means that the size is $5 * 5$. This process is data feature extraction, which is accomplished by continuously sliding blocks of different colors. The size of convolution kernel is not a rigid rule, but the best experimental results are obtained by selecting the size of convolution kernel through experiments, aiming to select the size with the best experimental performance. The length of the slide means that several feature points should be separated for one extraction. Thus, the convolution operation is performed to obtain the feature h_i^d as follows:

$$h_i^d = f\left(W_d \cdot V_i + b_d\right) \tag{3}$$

where, f is the ReLU function; W_d is the convolution kernel size; and b_d is the offset. For the specific feature V_i recorded in V_s, the winding operation was performed. In order to extract more comprehensive local features, d different convolution kernel sizes W was set to extract V_s features respectively, producing the output features H^d.

$$H^d = \left[h_1^d, h_2^d, \ldots, h_{n-d_r+1}^d\right] \tag{4}$$

Then, features H^d were superimposed to obtain feature sequence $H_s = \lfloor h_1, h_2, \ldots, h_{n-d_r+1} \rfloor$.

Next, the feature sequence H_s was pooled. The pooled operation does not affect the number of data features obtained by convolution, but only cuts each feature more or less, reducing the feature dimension and successfully solving the fitting problem [19]. The commonly used pooling is divided into average pooling and maximum pooling. In order to retain more information, reduce the risk of over-fitting, and improve the stability and position invariance, average pooling was used for pooling. H^d was divided into M small blocks to obtain the eigenvector p^{dM} as follows:

$$p^{dM} = chunkAve\left\{H^d\right\} = \left[h_{m_1}^d, h_{m_2}^d, \ldots, h_{m_M}^d\right] \tag{5}$$

Thus, the average values of all modules were spliced to obtain $P_s = \lfloor p_{m_1}, p_{m_2}, \ldots, p_M \rfloor$, where p_{m_i} is the vector obtained after the average pooling of block m_i.

In order to capture the long-distance dependence feature, P_s was input into the BiLSTM model, which is controlled by the forgetting gate (f_t), the input gate (i_t), the output gate (o_t) and a cell state update to complete the selection of attribute information, forgetting and cell state update. On the time step t, the forward part extracts the feature of p_t as follows:

$$\text{Forward direction LSTM} \Rightarrow \begin{cases} i_t = \sigma\left(W_i \cdot [h_{t-1}, p_t] + b_i\right) \\ f_t = \sigma\left(W_f \cdot [h_{t-1}, p_t] + b_f\right) \\ q_t = \tanh\left(W_q \cdot [h_{t-1}, p_t] + b_q\right) \\ o_t = \sigma\left(W_o \cdot [h_{t-1}, p_t] + b_o\right) \\ c_t = f_t * c_{t-1} + i_t * q_t \\ b_t = o_t * \tanh\left(c_t\right) \end{cases} \tag{6}$$

On time step t, the reverse part extracts features from p_t as follows:

$$\text{Reverse LSTM} \Rightarrow \begin{cases} i_t = \sigma\left(W_i \cdot [h_{t+1}, p_t] + b_i\right) \\ f_t = \sigma\left(W_f \cdot [h_{t+1}, p_t] + b_f\right) \\ q_t = \tanh\left(W_q \cdot [h_{t+1}, p_t] + b_q\right) \\ o_t = \sigma\left(W_o \cdot [h_{t+1}, p_t] + b_o\right) \\ c_t = f_t * c_{t+1} + i_t * q_t \\ b_t = o_t * \tanh\left(c_t\right) \end{cases} \tag{7}$$

where, σ is the sigmoid activation function, tanh is the hyperbolic tangent function, i_t is the selection operation for input information, f_t is the forgetting operation for the information to be forgotten in the previous module, c_t is used to determine which information should be stored in the current cell state, o_t is the output gate to select the output information, h_{t-1} is the output of the previous module, and p_t is the input at the current moment [20].

At time step t, the final output feature vector P_t of the BiLSTM layer is as follows:

$$P_t = [\text{forward direction LSTM, reverse LSTM}] \tag{8}$$

In the process of Internet abnormal traffic monitoring, the introduction of an attention mechanism can help the CNN-BiLSTM model learn and focus on more important features in the data, avoid the model being submerged by a large number of irrelevant information, improve the ability of the CNN-BiLSTM model to identify and generalize key information, and ensure the performance of classification monitoring. The specific steps are as follows:

Firstly, a multi-layer perceptron (MLP) was used to interpret the attributes of P_t and obtain the attribute representation of u_t. Then the importance of the attribute representation u_t relative to the context attribute u_w was calculated, obtaining the standard importance weight a_t. Finally, P_t was weighted and summed to obtain the advanced representation of the attribute v, and u_w was randomly initialized during the training process. The processing process in the attention mechanism layer is shown in the following Eqs. (9), (10), and (11):

$$u_t = \tanh\left(W_w P_t + b_w\right) \tag{9}$$

$$a_t = soft\max\left(u_t^T, u_w\right) \tag{10}$$

$$v = \sum a_t P_t \tag{11}$$

where, W_w is the context vector.

Finally, the result v was input into the softmax classifier to get the classification result of Internet abnormal traffic and complete the monitoring.

4 Experimental Testing

4.1 Experimental Environment and Parameter Settings

To verify the monitoring effectiveness of the proposed method, the monitoring method based on deep learning in the study by Qiu and Wang [5] and the monitoring method based on the multi-scale residual classifier in the study by Duan et al. [6] were selected as comparative methods for testing. The testing environment is a Windows 10 64-bit operating system, using AMD Ryzen 5 4600U with a Radeon Graphics 2.10 GHz processor with 16.0 GB of RAM, a programming environment in Python 3.8, and a learning framework in Tensorflow 1.0.

During the testing process, the minimum collection node coverage strategy was used to select the collection nodes, and then the data collection structure in Figure 1 was used for collection. Approximately 97018 data packets were collected, with a total of 4.2 MB of data. The collection rate reached 200 Kbps, with a delay of less than 50 ms. The collected dataset has 26 dimensional features and a label column, with a total of eight different types of data. Except for one type of normal traffic, the other seven types are all abnormal traffic. The quantity statistics corresponding to the traffic types in the dataset are shown in Figure 8.

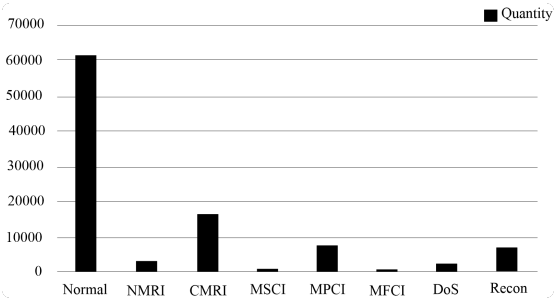

Figure 8. Quantity statistics chart based on dataset traffic types

In Figure 8, "normal" represents normal traffic and corresponds to data labels of 0, while the other seven types of abnormal traffic correspond to data labels of 1-7. From the quantity statistics chart, it can be seen that there is far more normal traffic than abnormal traffic. To ensure the reliability of the experimental test results, relevant testing parameters were set, as shown in Table 1.

Table 1. Parameter settings

Parameter	Numerical Value
Attention mechanism activation function	softmax
Regularization parameters	0.01
Convolutional kernel size	3×3
Number of convolution kernels	64
CNN activation function	ReLU
Number of LSTM hidden layer units	128
BiLSTM layers	2
Learning rate	0.001
Dropout rate	0.2
Input sequence length	128

4.2 Indicator Setting

On the basis of the above settings, the false alarm rate indicator, AUC indicator, and average monitoring delay indicator were selected to conduct comparative tests on the proposed method, the monitoring method based on deep learning in the study by Qiu and Wang [5], and the monitoring method based on the multi-scale residual classifier in the study by Duan et al. [6], aiming to complete the verification of monitoring performance.

Among them, false positive rate refers to the proportion of negative samples wrongly classified as positive to all negative samples, that is, the proportion of false positive samples. It is an important indicator that can be used to measure the degree of misjudgment of a method on normal data. A higher false alarm rate means that the method is more likely to misclassify normal instances as abnormal instances in monitoring, resulting in a large number of false alarm situations, thereby reducing the credibility and practicality of the method. The mathematical expression for false alarm rate is as follows:

$$FA = \frac{FP}{N} \tag{12}$$

where, FP is the normal number of samples that are falsely reported as an intrusion, and N is the total number of samples.

The AUC metric is an important measure for evaluating the performance of classification methods, specifically referring to the area covered below the Receiver Operating Characteristic (ROC) curve. The ROC curve, as a visualization tool, comprehensively evaluates the classification performance of a method by displaying the relationship

between true positive rate and false positive rate under different classification thresholds. The range of AUC values is between 0 and 1, and the larger the AUC value, the better the performance of the method. When the AUC of the method is 0.5, it indicates that the prediction and random selection of the method are no different; when the AUC is greater than 0.5, it indicates that the method's prediction is better than random selection; when the AUC approaches 1, it indicates that the method has strong discriminative ability.

The average monitoring delay refers to the average time delay of monitoring methods for monitoring and responding to abnormal traffic events. Usually measured in milliseconds, it is one of the efficiency indicators for monitoring methods to handle abnormal traffic events. The average monitoring delay directly affects the real-time and sensitivity of monitoring methods to abnormal traffic events. A lower average monitoring delay can enable the method to detect and respond to abnormal situations faster, helping to reduce potential losses and risks.

Therefore, the above multiple testing evaluation indicators were considered comprehensively in order to provide comprehensive performance evaluation results and complete the performance verification of the proposed method.

4.3 Result Analysis

4.3.1 Analysis of false alarm rate results

To verify the monitoring accuracy of the proposed method, 6,000 sample data points were randomly selected from the collected data. A comparative test was conducted on the false alarm rate index between the proposed method, the deep learning-based monitoring method in the study by Qiu and Wang [5], and the monitoring method based on the multi-scale residual classifier in the study by Duan et al. [6]. The results are shown in Figure 9.

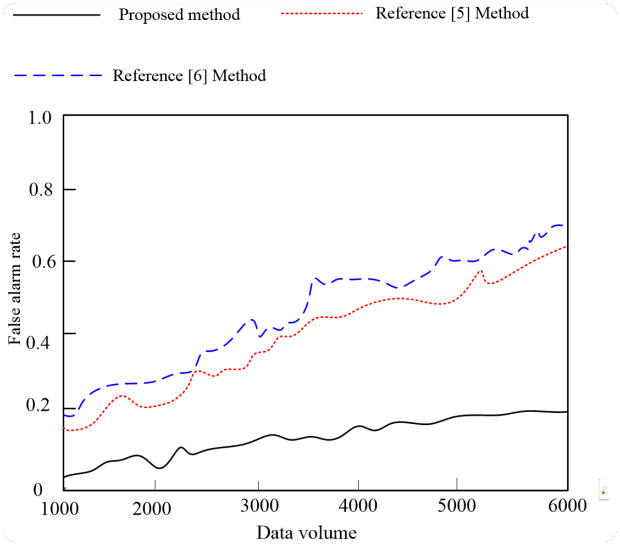

Figure 9. Comparison of false alarm rates

According to the results in Figure 9, with the increase in data volume, the false alarm rate obtained by the three methods for Internet abnormal traffic monitoring shows an upward trend, but the false alarm rate of the proposed method is lower than that of the other two comparison methods and can always be maintained below 0.2. When the data volume reached 6000, the false alarm rate of the proposed method was only 0.18, while the false alarm rates of the other two methods [5, 6] were 0.61 and 0.63, respectively. By comparing the results of the three methods, it can be concluded that the proposed method shows a low false alarm rate in terms of Internet abnormal data monitoring, has good monitoring accuracy, and can accurately complete the task of monitoring Internet abnormal data.

4.3.2 AUC value analysis

Next, to further validate the monitoring effectiveness of the proposed method, a comparative test was conducted on the AUC value indicator, including the proposed method and the other two methods [5, 6]. The results are shown in Figure 10.

According to the results in Figure 10, there are certain differences in AUC values obtained by the three methods, and their results are all greater than 0.5, which indicates that the three methods can effectively monitor abnormal Internet traffic. Further comparison of the AUC values of these three methods reveals that the proposed method has a higher AUC value than the other two methods. Moreover, the AUC value of the proposed method is close to 1, indicating that the method has strong ability to distinguish between normal and abnormal flow, and it can more accurately distinguish between normal and abnormal flow. Compared with the other two methods, the proposed method shows a better monitoring effect in terms of Internet abnormal traffic monitoring.

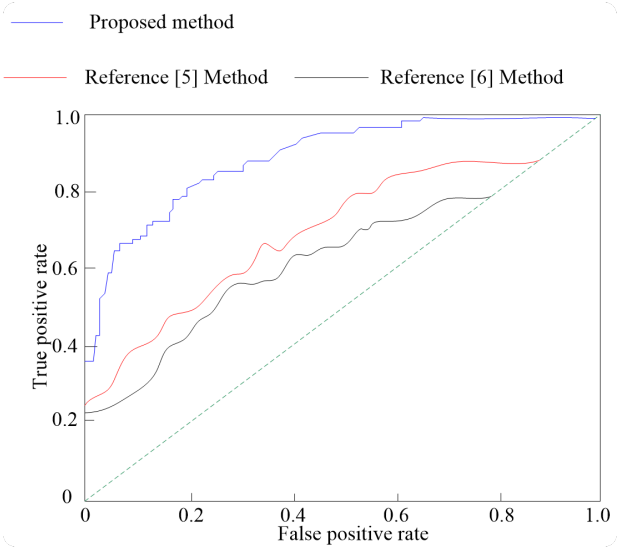

Figure 10. Comparison results of AUC values

4.3.3 Analysis of average monitoring delay results

On the basis of the above tests, to verify the reliability of the proposed method, a comparative test was conducted on the average monitoring delay index for the proposed method and the other two methods [5, 6]. The results are shown in Table 2.

Table 2. Comparison results of average monitoring delay

Data Volume	Proposed Method (ms)	Method from Reference [5] (ms)	Method from Reference [6] (ms)
1000	2.2	9.7	10.1
2000	2.7	10.8	11.4
3000	3.1	11.9	12.6
4000	3.8	13.3	13.9
5000	4.6	14.5	15.3
6000	5.7	15.8	16.6

According to the results in Table 2, with the increase in data volume, the proposed method and the other two methods [5, 6] were used to monitor Internet abnormal traffic, and the average monitoring delay shows an upward trend. However, compared with the average monitoring delay results of the three methods, the proposed method has a lower average monitoring delay and a smaller growth rate. When the data volume reaches 6000, the average monitoring delay of the proposed method is only 5.7 ms, while the average monitoring delay of the other two methods [5, 6] is 15.8 ms and 16.6 ms, respectively. Comparing the average monitoring delay results of the three methods, it can be concluded that the proposed method can effectively reduce the average monitoring delay, has high real-time and sensitivity to abnormal traffic events, can detect and respond to abnormal situations faster, and helps reduce potential losses and risks.

5 Conclusion

In order to improve the accuracy and real-time performance of abnormal traffic monitoring, an Internet abnormal traffic monitoring method based on the improved BiLSTM network algorithm was proposed. First, the minimum collection node coverage strategy based on constraints was adopted to select the collection node, aiming to ensure that the traffic data of all nodes on the Internet can be collected and the resources consumed are small. Secondly, the collected dataset was transformed to ensure its validity. Then, in order to optimize the performance of Internet abnormal traffic monitoring, CNN and BiLSTM models were introduced to extract data features more comprehensively. Finally, the attention mechanism was used to calculate the importance of attribute features to improve the performance of the final classification monitoring. The results show that the false alarm rate is 0.2, the AUC value is 0.95, and the average monitoring delay is 5.7 ms when using the proposed method for monitoring. This shows that the proposed method has high monitoring accuracy, a fast response rate to abnormal traffic, and can better maintain the security and stability of the Internet system.

Funding

The paper was funded by the High-end Training Support Project for Professional Leaders in Higher Vocational Colleges in Jiangsu Province (Grant No.: 2024GRFX031).

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Z. P. Lu and K. B. Shi, "Anomaly detection method for TBM construction based on improved VMD-XGBoost-BILSTM combined model," *Earth Sci. Inform.*, vol. 16, no. 4, pp. 4273–4284, 2023. https://doi.org/10.1007/s12145-023-01101-9

[2] A. J. Hashim, M. A. Balafar, and J. Tanha, "NEAE: NeuroEvolution AutoEncoder for anomaly detection in internet traffic data," *J. Supercomput.*, vol. 80, no. 5, pp. 6746–6777, 2024. https://doi.org/10.1007/s11227-023-05715-0

[3] L. Duan, J. X. Zhou, Y. Wu, and W. Y. Xu, "A novel and highly efficient botnet detection algorithm based on network traffic analysis of smart systems," *Int. J. Distrib. Sens. Netw.*, vol. 18, no. 3, p. 15501477211049910, 2022. https://doi.org/10.1177/15501477211049910

[4] C. D. Xuan, "Detecting APT attacks based on network traffic using machine learning," *J. Web Eng.*, vol. 20, no. 1, pp. 171–190, 2021. https://doi.org/10.13052/jwe1540-9589.2019

[5] L. C. Qiu and L. Wang, "Abnormal traffic detection method of Internet of Things based on deep learning in edge computing environment," *J. Circuits Syst. Comput.*, vol. 32, no. 16, p. 2350283, 2023. https://doi.org/10.1142/S0218126623502833

[6] X. Y. Duan, Y. Fu, and K. Wang, "Network traffic anomaly detection method based on multi-scale residual classifier," *Comput. Commun.*, vol. 198, no. 6, pp. 206–216, 2023. https://doi.org/10.1016/j.comcom.2022.10.024

[7] Y. Z. Liu, Y. S. Zou, Y. Wu, H. Y. Zhang, and G. F. Ding, "A novel abnormal detection method for bearing temperature based on spatiotemporal fusion," *Proc. Inst. Mech. Eng. F J. Rail Rapid Transit*, vol. 236, no. 3, pp. 317–333, 2022. https://doi.org/10.1177/09544097211022105

[8] R. Moreira, L. F. R. Moreira, and F. de Oliveira Silva, "An intelligent network monitoring approach for online classification of darknet traffic," *Comput. Electr. Eng.*, vol. 110, p. 108852, 2023. https://doi.org/10.1016/j.compeleceng.2023.108852

[9] J. J. Zhao, X. Y. Jing, Z. Yan, and W. Pedrycz, "Network traffic classification for data fusion: A survey," *Inf. Fusion*, vol. 72, pp. 22–47, 2021. https://doi.org/10.1016/j.inffus.2021.02.009

[10] A. A. S. Shaikh, M. Bhargavi, and C. P. Kumar, "An optimised Darknet traffic detection system using modified locally connected CNN - BiLSTM network," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 43, no. 2, pp. 87–96, 2023. https://doi.org/10.1504/IJAHUC.2023.131361

[11] Q. Zhang, Y. J. Bu, B. Chen, S. R. Zhang, and X. Y. Lu, "Research on phishing webpage detection technology based on CNN-BiLSTM algorithm," *J. Phys.: Conf. Ser.*, vol. 1738, no. 1, p. 012131, 2021. https://doi.org/10.1088/1742-6596/1738/1/012131

[12] W. H. Wei, Y. Chen, Q. Z. Lin, J. K. Ji, K. C. Wong, and J. Q. Li, "Multi-objective evolving long–short term memory networks with attention for network intrusion detection," *Appl. Soft Comput.*, vol. 139, p. 110216, 2023. https://doi.org/10.1016/j.asoc.2023.110216

[13] M. Jiao and D. Q. Wang, "The Savitzky-Golay filter based bidirectional long short-term memory network for SOC estimation," *Int. J. Energy Res.*, vol. 45, no. 13, pp. 19 467–19 480, 2021. https://doi.org/10.1002/er.7055

[14] C. Li, Y. N. Zhang, G. H. Zhao, and Y. R. Ren, "Hourly solar irradiance prediction using deep BiLSTM network," *Earth Sci. Inform.*, vol. 14, pp. 299–309, 2021. https://doi.org/10.1007/s12145-020-00511-3

[15] Z. D. Tian and P. F. Song, "A novel network traffic combination prediction model," *Int. J. Commun. Syst.*, vol. 35, no. 7, p. e5097, 2022. https://doi.org/10.1002/dac.5097

[16] T. V. Geetha and A. J. Deepa, "A FKPCA-GWO WDBiLSTM classifier for intrusion detection system in cloud environments," *Knowl.-Based Syst.*, vol. 253, p. 109557, 2022. https://doi.org/10.1016/j.knosys.2022.109557

[17] G. Beguš and A. Zhou, "Interpreting intermediate convolutional layers of generative CNNs trained on waveforms," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 3214–3229, 2022. https://doi.org/10.1109/TASLP.2022.3209938

[18] J. Y. Tian, J. T. Zhou, and J. Duan, "Hierarchical services of convolutional neural networks via probabilistic selective encryption," *IEEE Trans. Serv. Comput.*, vol. 16, no. 1, pp. 343–355, 2023. https://doi.org/10.1109/TSC.2021.3136601

[19] W. C. Yeh, Y. P. Lin, Y. C. Liang, C. M. Lai, and C. L. Huang, "Simplified swarm optimization for hyperparameters of convolutional neural networks," *Comput. Ind. Eng.*, vol. 177, p. 109076, 2023. https://doi.org/10.1016/j.cie.2023.109076

[20] D. W. Xia, N. Yang, S. Y. Jian, Y. Hu, and H. A. Li, "SW-BiLSTM: A Spark-based weighted BiLSTM model for traffic flow forecasting," *Multimed. Tools Appl.*, vol. 81, no. 17, pp. 23 589–23 614, 2022. https://doi.org/10.1007/s11042-022-12039-3