J Inf Process Syst, Vol.19, No.5, pp.688~701, October 2023
https://doi.org/10.3745/JIPS.01.0097

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

A Network Intrusion Security Detection Method Using BiLSTM-CNN in Big Data Environment

Hong Wang*

Abstract

The conventional methods of network intrusion detection system (NIDS) cannot measure the trend of intrusion-detection targets effectively, which lead to low detection accuracy. In this study, a NIDS method which based on a deep neural network in a big-data environment is proposed. Firstly, the entire framework of the NIDS model is constructed in two stages. Feature reduction and anomaly probability output are used at the core of the two stages. Subsequently, a convolutional neural network, which encompasses a down sampling layer and a characteristic extractor consist of a convolution layer, the correlation of inputs is realized by introducing bidirectional long short-term memory. Finally, after the convolution layer, a pooling layer is added to sample the required features according to different sampling rules, which promotes the overall performance of the NIDS model. The proposed NIDS method and three other methods are compared, and it is broken down under the conditions of the two databases through simulation experiments. The results demonstrate that the proposed model is superior to the other three methods of NIDS in two databases, in terms of precision, accuracy, F1-score, and recall, which are 91.64%, 93.35%, 92.25%, and 91.87%, respectively. The proposed algorithm is significant for improving the accuracy of NIDS.

Keywords

Big Data, BiLSTM, CNN, Feature Selection, Network Intrusion Detection

1. Introduction

Since the 21st century, the Internet has rapidly developed worldwide and has become widely used, and the growth trend of the Internet is gradually increasing [1,2]. The quick development of the Internet has brought many conveniences to Internet users; however, it has also created many security problems (i.e., tampering, backdoor implantation, and fake websites). In addition, many high-risk vulnerabilities have become public [3-5]. Network security faces several unknown risks and challenges. Malware, security vulnerabilities, denial-of-service attacks, website security, cloud platform security, and other accidents occur frequently [6-8].

Currently, every field uses networks for transactions. Computer networks play an irreplaceable role. Therefore, the security problems of the accompanying network need to be addressed. The methods often used to these problems include firewall and intrusion detection technologies [9,10]. In the social background of big data, due to the diversification of network intrusion means and the gradual complexity

Manuscript received December 7, 2022; first revision March 7, 2023; second revision April 3, 2023; accepted April 11, 2023.
* **Corresponding Author:** Hong Wang (whiam@163.com)
School of Electronic Information, Sichuan Modern Vocational College, Chengdu, China (whiam@163.com)

of the network environment, the existing methods of network intrusion detection system (NIDS) have gradually failed to adapt to the current actual situation [11,12].

Current research methods for NIDS have many limitations, and detection accuracy cannot be guaranteed. Therefore, new NIDS technologies must be selected to achieve the desired effect [13-15].

Research on artificial intelligence and deep learning in human society has been continuously conducted and deepened, and the NIDS technology has developed vigorously. The NIDS method developed on this basis can automatically detect NIDS information and significantly improve the overall performance of NIDS [16,17]. However, there is no suitable method or mature solution for IDS to confirm large-scale distributed and combined intrusion attacks. False alarms are serious, and users frequently drown in massive amounts of alarm message and let slip real alarms. It cannot investigate attacks or prevent attacks without user participation. Therefore, defects in network protocol, design principle, timely response, signature database updating, and timeliness are the biggest challenges faced by conventional intrusion detection algorithms based on machine learning [18,19].

2. Related Research

Studying the deep learning-based NIDS model is meaningful to solve the detection problems that traditional NIDS systems are increasingly encountering. Chen and Miao [20] established an intrusion information sequence model by collecting data from different centers in network intrusion under the background of a sequence model of big data and used path-trend binary weighted semantics to conduct a trend path set for the intrusion path direction to achieve network security intrusion detection. However, this method does not analyze possible information leakage caused by the network itself. Vieira et al. [21] proposed an autonomous NIDS and response method by introducing an operational analysis and a response model. This method solved the problem of large-scale distributed systems that are vulnerable to hijacking and other attacks. Nonetheless, the model runs poorly due to a sever deficiency in the feature extraction ability of this model when face high-dimensional features. Liu et al. [22] analyzed the latest research on NIDS technology, studied the problems in system models and detection algorithms, and analyzed new applications of NIDS technology in the context of big data. However, they did not propose strategies or methods to solve the corresponding problems effectively. A new NIDS method was presented in Viegas et al. [23], and a reliable NIDS model based on ML was established. Moreover, this method has no influence of correlation between recognition and features on classification, this model also can be further improved. For the problem that big data generated by network flows and system events has an effect on the accuracy of NIDS, Wang and Jones [24] analyzed datasets with various data types using the “capacity,” “accuracy,” and “diversity” of big data features in network flows and attacks with R language and its functions. nevertheless, the training performance of this method is poor on high-dimensional. Dasgupta and Saha [25] proposed a NIDS method to solve a problem in which the current NIDS technology is complex. This method introduces the network model, which is shallow, and its accuracy of recognition is limited, Kalinin and Krundyshev [26] proposed an effective NIDS method to improve security NIDS degenerates under the condition of big data input, but it ignores the behavior sequence’s time sequence, resulting in a low rate of detection. An oversampling algorithm was proposed by Fu et al. [27], and it was used as a data augmentation method for solving the issue of unbalanced network intrusion

date. A stacked autoencoder with a dropout structure was used as the data downscaling method to enhance model's generalization ability. The channel attention mechanism was associated with a bidirectional long short-term memory (BiLSTM) network to enhance the network structure. However, owing to the lack of application scenarios, improving the network model further to enhance the accuracy of detection is necessary. Albasheer et al. [28] proposed a NIDS system to predict the intrusion alarm problem by reviewing and analyzing the detection results. However, owing to the limitations of alert correlation, insufficient detection accuracy is a problem. Alavizadeh et al. [29] proposed a method for detecting and classifying different types of network intrusion attacks using deep reinforcement learning models. By combining a deep feed-forward neural network with reinforcement learning, the accuracy of classifying different types of network intrusion attacks can be improved. However, owing to the single-application scenario of the model, this was not applicable. Cao et al. [30] proposed a NIDS model associating a convolutional neural network (CNN) and a gated recurrent unit. The proposed intrusion detection model was rated based on the NSL-KDD, UNSW_NB15, and CIC-IDS2017 datasets, which solved the issue of low multi-classification accuracy of intrusion behavior and low-class imbalance data's detection accuracy in existing intrusion detection models. However, owing to the abundant parameters in the model, the running time was relatively high, which reduced the efficiency of the detection data.

To address the high false detection rate and low accuracy of NIDS methods for malicious attacks, a NIDS method using BiLSTM-CNN is proposed. The basic concept is as follows:

- The detection model is constructed in two stages using feature reduction and an abnormal probability output as the core.
- CNN is used as the basic model to simplify the complexity and reduce computation. The real-time classification speed is improved by introducing BiLSTM to realize the relevance of the input. Comparison with conventional NIDS methods.

3. NIDS Model based on BiLSTM-CNN

3.1 Overall Framework of the Method

The NIDS model proposed herein comprises two CNN structures, as shown in Fig. 1.

Its innovative idea is the first stage of feature reduction and abnormal probability output and the second stage of neural network classification. The first stage uses a BiLSTM network. Because of its good performance, it can be divided into abnormal and normal traffics with probability scores in the initial stage. This probability value was considered as an additional feature in the final classification stage. It is mainly divided into two tasks: constructing the supervised learning BiLSTM model and the output anomaly probability value, P. First, the weight value w and offset value c of unlabeled data are learned in an unsupervised learning method, which are applied to unlabeled data for layer-by-layer unlabeled pre-training. We then used w and c in a supervised learning mode, passed the pre-trained features and label vectors into the softmax regression classifier, and used the backpropagation algorithm to classify the labeled data (x, y). At this stage, the model finishes fine-tuning the pretrained w and c values, and outputs an abnormal probability value P through the softmax classifier.

The model comprises two stages. The BiLSTM conducts feature learning to achieve the best feature representation in the first stage. After the first training stage, traffic is initially classified through proba-

bility score. during the second stage, features are learned and extracted from other data. This improves the accuracy and reduces the number of false positives.

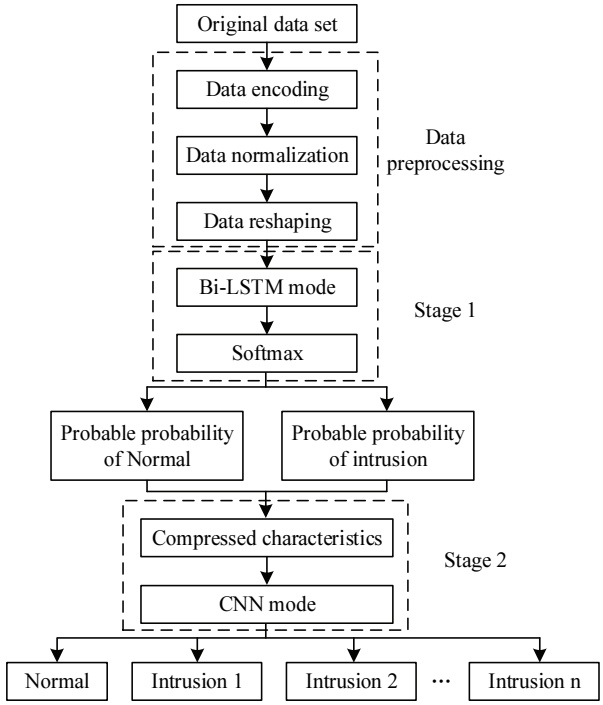

Fig. 1. Structure of intrusion detection model based on BiLSTM-CNN.

3.2 Data Preprocessing

3.2.1 Data encoding

Data processing is difficult owing to the variety and inconsistency of information types. Therefore, it is necessary to encode information that is not of the numerical type, and finally program the numerical type.

3.2.2 Data reshaping

To facilitate the input of the network model, it can reshape traffic sample with $a \times b * r + 1$, then build a matrix N of $b * r$. The construction of its matrix is based on the size of the data volume and the feature data extracted by deep learning, as shown in the following formula:

$$N = \begin{bmatrix} N_{11} & N_{12} & \dots & N_{1r} \\ N_{21} & N_{22} & \dots & N_{2r} \\ \dots & \dots & & \dots \\ N_{b1} & N_{b2} & \dots & N_{br} \end{bmatrix}. \quad (1)$$

3.2.3 Data segmentation

Each model required two datasets. The input data were segmented by combining the content of the

dataset and size of the above matrix. The entire dataset was divided according to the method shown in Fig. 2.

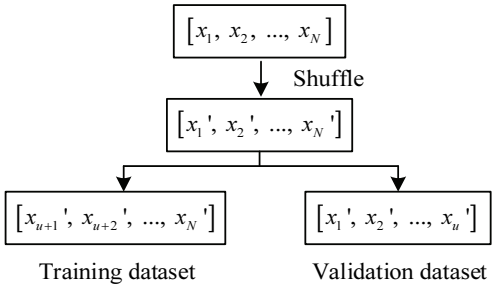

Fig. 2. Data segmentation diagram.

3.3 Normalization

Scaling was applied to features with a wide range of differences between minimum and maximum values. Therefore, according to formula (2), all different values of each feature are mapped in the range of 0 and 1, thereby normalizing the feature.

$$X_{i,j} = \frac{X_{i,j} - X_{i,j\_min}}{Xi,j\_min_{i,j\_max}}, \tag{2}$$

where, $X_{i,j}$ represents the value of the feature in rows i and columns j. X_{i,j_min} represents the minimum value of each feature. X_{i,j_max} represents the maximum value of all data points.

In many cases, feature normalization can reduce the training time of the model. If there are two datasets, d_1 and d_2, in the original dataset, the value range of d_1 is $[1000,5000]$ and that of d_2 is $[1,5]$. Fig. 3 shows a route comparison of the gradient descent before and after normalization. The ellipse represents the graph without normalization of the original data and the circle represents the graph after normalization.

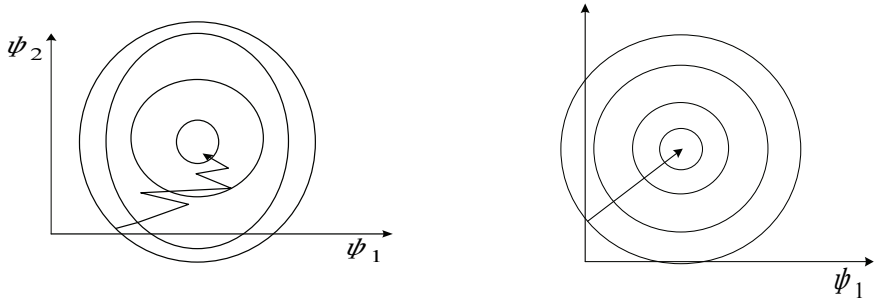

Fig. 3. Comparison graph (a) before and (b) after normalization.

As shown in Fig. 3(a), the speed to calculate the local optimal solution is slow when the data are not normalized, and multiple iterations are required in the training process to reach the local optimal solution; therefore, the model takes a long time to train the data. Combining the feature extraction characteristics

of deep learning networks and data normalization methods can significantly improve the efficiency of data enhancement. After normalization of the original data, as shown in Fig. 3(b), the model can quickly find the local optimal solution in the gradient descent process in only a few or even several iterations.

3.4 BiLSTM

Relative to LSTM, BiLSTM inverts the input sequence and recalculates and outputs using the LSTM model. The final output was a stack of forward and reverse results. Thus, input relevance can be achieved. Fig. 4 shows the structure of the BiLSTM.

Each circular node in Fig. 4 represents an LSTM cell, and the forward and reverse arrows represent different times. Notably, the forward and reverse nodes are not shared. The role of this model is to extract feature data. When used as the output, the two nodes output one result.

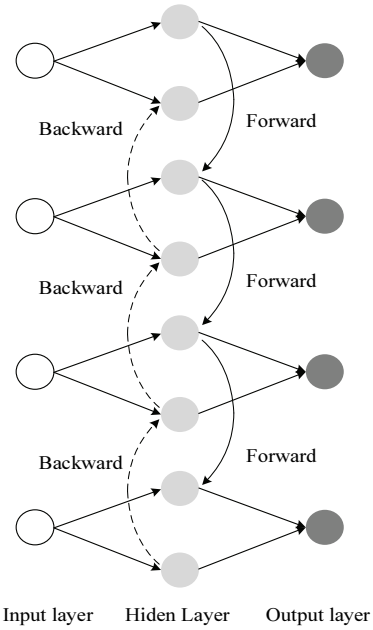

Fig. 4. BiLSTM model structure diagram.

3.5 CNN Model

The basic structure of a CNN comprises five layers. Fig. 5 shows the training process.

The role of the CNN in this study was to further extract the data features of network intrusion target detection and classify the data deeply by decomposing the convolutional layer.

3.6 Selection of Pooling Layer

The pooling layer samples the required features. This can decrease the size of the input data, promote the statistical efficiency, and reduce the demand for parameters, thereby reducing the complexity of the calculation.

The most common pooling methods are maximum and average pooling. Fig. 6 describes the calculation

process of the maximum pooling layer when the feature map size is 2×2 and the step size is two. Maximizing the pooling involves calculating the maximum value of each size 2×2. The purpose of using a pooling layer in this study is to prevent overfitting of the introduced deep neural network, resulting in redundancy and errors in the feature extraction data, which reduces the network detection accuracy.

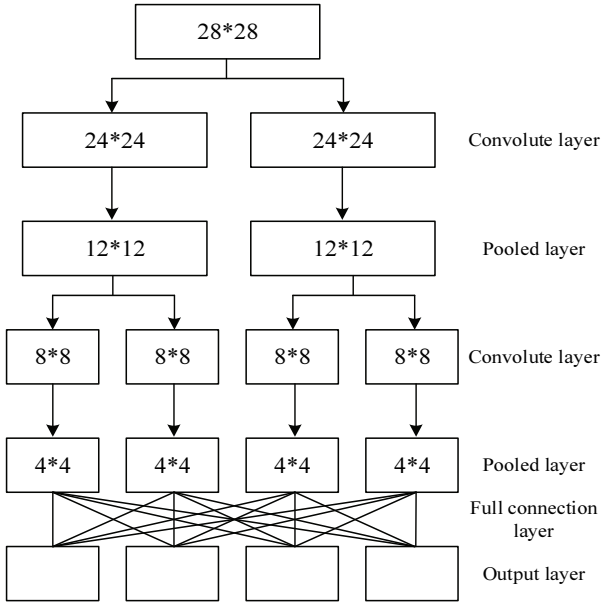

Fig. 5. Convolutional neural network structure diagram.

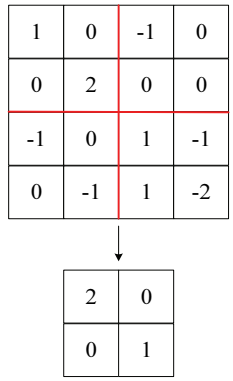

Fig. 6. Schematic diagram of maximum pooling.

3.7 Softmax Layer

The softmax layer, also known as the classification layer, is used for classification or regression calculations as the neural network's last layer. Based on the problem to be solved, softmax was used to categorize the network traffic data. The following formula was used to figure the sample probability x belonging to the category l, where $\varphi(w, c)$ is the parameter of the classification layer and m represents the number of label types of the classified samples. The classification formula is shown in formula (3).

$$P(\hbar = l|x; \varphi(w,c)) = \frac{exp[\varphi_l^T(w,c)x]}{\sum_{i=1}^{m} exp[\varphi_l^T(w,c)x]}, \quad (3)$$

where, h is the predicted sample category, formula (4) is given as follows:

$$L = \frac{1}{n}\sum_{k=1}^{n}(l_k - l_k')^2, \quad (4)$$

where n represents the samples amount; k represents the k-th sample; l_k and l_k' represents the prediction category and real label of k, respectively. L is the calculation error. It is necessary to minimize the calculation error when training neural networks.

4. Experiment

4.1 Experimental Settings

The environment configuration is summarized in Table 1.

Table 1. Experimental environment configuration

Name	Configuration
Operating system	Windows 10
CPU	E3-1505M v6
Processor	Intel Xeon
Hard disk	1 T
Memory	8 GB
Programming language	Python 3.5
Programming environment	Jupyter Notebook+keras+skleam

4.2 Datasets

The common dataset used in NIDS is KDD99, however, this dataset has many drawbacks, particularly its inability to provide the most original data traffic. Therefore, two new NIDS datasets were used for experimental analysis. These two datasets are newer than KDD99 and other NIDS datasets, and the datasets contain more types of traffic with reliable verification and test datasets. The KDD99 dataset artificially extracts features of specific dimensions from the original traffic data without providing the most original traffic package data files. The two datasets adopted here provide the most original traffic PCAP files, which can be used by researchers to mine more information from the file or directly use the original traffic packet data for research.

(1) CICIDS2017 Dataset: The CICIDS2017 collected network information from Monday to Friday. The information on Monday is only positive. From Tuesday to Friday, the attacking network attacks the victim's network. Finally, we mark the flow accurately using five timestamp fields. Benign traffic was extracted from the data. The generated flow was labeled according to the flow method in the CICIDS2017 data to obtain a true and reliable label. Details of the dataset are listed in Table 2.

Table 2. Distribution of various traffic in the CICIDS2017 dataset

Category of flow	Quantity
Benign	359,641
DoS GoldenEye	9,467
DoS Hulk	16,354
DoS Slowhttp	6,521
DoS Slowloris	5,937
SSH Patator	4,683
FTP Patator	6,019
Brute Force	3,035
WebSQL Injection	255
XSS Attack	1,641
BotNet	3,267
Port Scan	176,983
DDoS	18,647

From the dataset, 20,000 benign flows and 16,749 port scan attacks were randomly selected using a downsampling algorithm.

(2) CTU Dataset: The CUT dataset was generated by capturing and marking network information in different situations. Eleven types of flows generated from June 2020 to June 2021 were selected. Details of the dataset are listed in Table 3.

Data-balance processing was not required for the CTU dataset. Although Viaxmr and Trojan BotNet traffic has relatively small proportion in total number, considering that these two types of traffic are relatively common, they are still used for intrusion detection analysis in actual experiments to find suspicious attacks effectively.

Table 3. Distribution of various traffic in CTU dataset

Category of flow	**Quantity**
Benign	76,425
Sathurbot	26,381
Trickster	23,692
TrickBot	72,947
Dridex	8,634
WebCompanion	16,892
Viaxmr	1,806
Trojan	1,821
CoinMiner	20,648
HTBot	17,463
Ursnif	12,594

4.3 Evaluation Indices

In general, the performance evaluation of NIDS methods has four evaluation criteria: F1 (F-measure or F1-score), precision, recall, and accuracy. These evaluation criteria are defined using four functions,

including true negatives (TN), false negatives (FN), true positive (TP), and false positives (FP). TN is the normal sample number predicted as normal samples, FN is the attack number incorrectly predicted as normal samples, TP is the sample number that correctly predicts attack types, FP is the normal number incorrectly predicted as attack samples. These functions can be obtained from confusion matrix C. The element C_{ij} of confusion matrix C represents the sample amount in the class i to be predicted as class j. Moreover, the receiver operating characteristic (ROC) curve and area under the curve (AUC) of ROC can also be used as indices of classifiers or models. The meanings and calculation formulas for the accuracy, precision, recall, F-score, ROC curve, and AUC value were as follows:

$$A = \frac{T_P+T_N}{T_P+T_N+F_P+F_N}, \tag{5}$$

$$P = \frac{T_P}{T_P+F_P}, \tag{6}$$

$$R = \frac{T_P}{T_P+F_N}, \tag{7}$$

$$F = \frac{2PR}{P+R}. \tag{8}$$

The ordinate of the ROC represents the TP rate, which is the correctly classified samples' proportion among all normal samples. The calculation formula is following:

$$TPR = \frac{T_P}{T_P+F_N}. \tag{9}$$

The abscissa of the ROC curve represents the false-positive rate (FPR). FPR is also called specificity, which is negative samples' proportions that are incorrectly categorized as positive among all negative samples. The calculation formula is as follows:

$$FPR = \frac{F_P}{F_P+T_N}. \tag{10}$$

The upper-left corner of the ROC image shows the maximum TP and minimum FP rates.

4.4 Effect of Different Batch Sizes on Model Accuracy

To unify the best batch size, experiments were conducted with batch sizes of 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1,000, according to the optimal parameters determined above. The results are presented in Fig. 7.

Fig. 7 shows that when the batch size was approximately 50 or 500, the model's detection accuracy is high, exceeding 93.30%. Because each iteration's duration was shorter when the batch size was 50 than 500, the batch size was set to 50. When the same accuracy is achieved, the corresponding iteration time must have same direction as the batch size is raised, and the required time of training will be lengthened. This experiment was analyzed by varying the batch size based on the same number of iterations; thus, the accuracy decreased as the batch size increased.

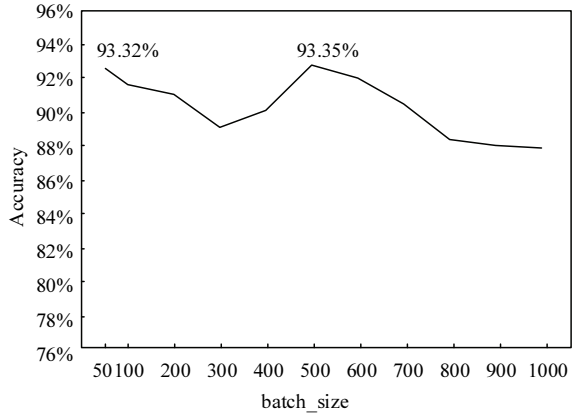

Fig. 7. Accuracy of the model with different batch sizes.

4.5 Effect of Iterations on Model Accuracy

One-third of the training data was used as validation data in each iteration. Fig. 8 is the loss function values during the training process.

Training was performed to reduce the loss completely and make it converge. If the number of iterations of training is not sufficient, the neural network does not converge, indicating that effective features have not been entirely extracted, and redundant information still exists. If iteration's number is extremely large and the loss function value has converged, the training could cause overfitting. As shown in Fig. 8, when the number of iterations exceeds 30, the loss function is maintained at approximately 0.01 without further decline. Therefore, the number of iterations was determined to be 30 based on the loss function diagram and the experimental results of different iterations.

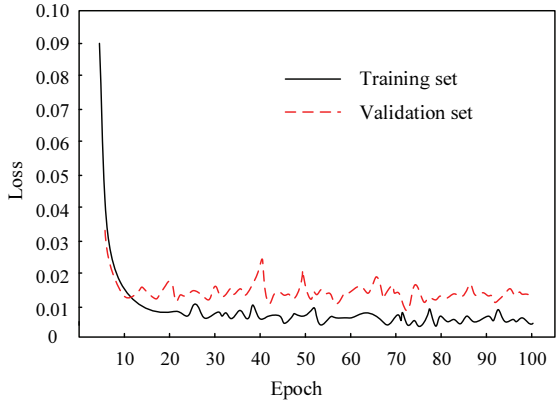

Fig. 8. Accuracy and loss value of model training process.

4.6 Verify the Effectiveness of the Model

The following criteria were used to verify the superiority of the NIDS method: F1-score, accuracy (A), recall (R), and precision (P). For the proposed NIDS method using BiLSTM-CNN and the methods in [20,23,24], an experiment was conducted using the CICIDS2017 and CTU datasets. The final estimation results of the evaluation indices are presented in Table 4 and Fig. 9.

Table 4. Results on CICIDS2017 dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Proposed method	93.35	91.64	91.87	92.25
Chen and Miao [20]	90.21	88.33	88.86	89.64
Viegas et al. [23]	84.52	81.35	82.05	82.97
Wang and Jones [24]	88.45	85.62	86.31	87.06

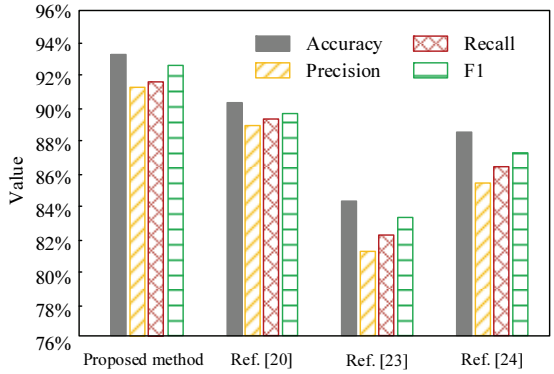

Fig. 9. Results on CTU dataset.

Table 4 and Fig. 9 showed that the proposed NIDS method based on BiLSTM-CNN is superior to the other three comparison methods in A, P, R, and F1, reaching 93.35%, 91.64%, 91.87%, and 92.25%, respectively, when the two datasets are respectively used. The proposed method's accuracies were 3.14%, 8.83%, and 4.90%, higher compared to those of the other comparison methods. The reason is the BiLSTM network can exclude the problem of gradient disappearance and shorten the time interval between acquiring the input and making the decisions. In addition, a pooling layer was added after the convolution layer, which improved the feature sampling process and detection accuracy of the confirmed normal classes.

5. Conclusion

To address the high false detection rate and low accuracy of conventional NIDS methods for malicious attacks, a NIDS method using BiLSTM-CNN in a big data environment is proposed. The result of experiment indicates that the introduction of feature reduction and abnormal probability output in the detection process can effectively prevent overfitting. Introducing a BiLSTM network into a conventional CNN can deal with the problem of gradient disappearance, shorten the time interval from obtaining the input to making decisions, and increase the detection accuracy. If the pooling layer is connected after convolution layer, it will optimize the sampling process of the required features.

Further improving the accuracy of NIDS will be focused in the future: (1) research on the method of feature learning performance to improve the learning performance; (2) from the view of network adaptation, improving the feature extraction performance of the network, and then improving the accuracy of NIDS; and (3) from the perspective of optimizing the model, accelerating the training speed and improving the efficiency of NIDS.

References

[1] F. Faraji Daneshgar and M. Abbaspour, "On the resilience of P2P botnet footprints in the presence of legitimate P2P traffic," *International Journal of Communication Systems*, vol. 32, no. 13, article no. e3973, 2019. https://doi.org/10.1002/dac.3973

[2] A. K. Bhandage and A. Barragan, "Calling in the cavalry: toxoplasma gondii hijacks GABAergic signaling and voltage-dependent calcium channel signaling for Trojan horse-mediated dissemination," *Frontiers in Cellular and Infection Microbiology*, vol. 9, article no. 61, 2019. https://doi.org/10.3389/fcimb.2019.00061

[3] A. Amouri, V. T. Alaparthy, and S. D. Morgera, "A machine learning based intrusion detection system for mobile Internet of Things," *Sensors*, vol. 20, no. 2, article no. 461, 2020. https://doi.org/10.3390/s20020461

[4] D. Li, L. Deng, M. Lee, and H. Wang, "IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning," *International Journal of Information Management*, vol. 49, pp. 533-545, 2019. https://doi.org/10.1016/j.ijinfomgt.2019.04.006

[5] C. L. Ferre, J. B. Carmel, V. H. Flamand, A. M. Gordon, and K. M. Friel, "Anatomical and functional characterization in children with unilateral cerebral palsy: an atlas-based analysis," *Neurorehabilitation and Neural Repair*, vol. 34, no. 2, pp. 148-158, 2020. https://doi.org/10.1177/1545968319899916

[6] C. Qi, H. B. Ly, Q. Chen, T. T. Le, V. M. Le, and B. T. Pham, "Flocculation-dewatering prediction of fine mineral tailings using a hybrid machine learning approach," *Chemosphere*, vol. 244, article no. 125450, 2020. https://doi.org/10.1016/j.chemosphere.2019.125450

[7] B. Yan and G. Han, "Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system," *IEEE Access*, vol. 6, pp. 41238-41248, 2018. https://doi.org/10.1109/ACCESS.2018.2858277

[8] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT)*, New York, NY, 2016, pp. 21-26. https://doi.org/10.4108/eai.3-12-2015.2262516

[9] B. A. Pratomo, P. Burnap, and G. Theodorakopoulos, "Unsupervised approach for detecting low rate attacks on network traffic with autoencoder," in *Proceedings of 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, Glasgow, UK, 2018, pp. 1-8. https://doi.org/10.1109/CyberSecPODS.2018.8560678

[10] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the Internet of Things: a comprehensive investigation," *Computer Networks*, vol. 160, pp. 165-191, 2019. https://doi.org/10.1016/j.comnet.2019.05.014

[11] J. Granjal, J. M. Silva, and N. Lourenço, "Intrusion detection and prevention in CoAP wireless sensor networks using anomaly detection," *Sensors*, vol. 18, no. 8, article no. 2445, 2018. https://doi.org/10.3390/ s18082445

[12] Y. Lv, S. Peng, Y. Yuan, C. Wang, P. Yin, J. Liu, and C. Wang, "A classifier using online bagging ensemble method for big data stream learning," *Tsinghua Science and Technology*, vol. 24, no. 4, pp. 379-388, 2019. https://doi.org/10.26599/TST.2018.9010119

[13] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," in *Proceedings of the 2019 ACM Southeast Conference*, Kennesaw, GA, 2019, pp. 86-93. https://doi.org/10.1145/3299815.3314439

[14] W. Zong, Y. W. Chow, and W. Susilo, "Interactive three-dimensional visualization of network intrusion detection data for machine learning," *Future Generation Computer Systems*, vol. 102, pp. 292-306, 2020. https://doi.org/10.1016/j.future.2019.07.045

[15] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761-768, 2018. https://doi.org/10.1016/j.future.2017.08.043

[16] A. I. Saleh, F. M. Talaat, and L. M. Labib, "A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers," *Artificial Intelligence Review*, vol. 51, pp. 403-443, 2019. https://doi.org/10.1007/s10462-017-9567-1

[17] G. Karatas, O. Demir, and O. K. Sahingoz, "Deep learning in intrusion detection systems," in *Proceedings of 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, Ankara, Turkey, 2018, pp. 113-116. https://doi.org/10.1109/IBIGDELFT.2018.8625278

[18] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "A novel two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373-30385, 2019. https://doi.org/10.1109/ACCESS.2019.2899721

[19] P. Devan and N. Khare, "An efficient XGBoost–DNN-based classification model for network intrusion detection system," *Neural Computing and Applications*, vol. 32, pp. 12499-12514, 2020. https://doi.org/10.1007/s00521-020-04708-x

[20] J. Chen and Y. Miao, "Study on network security intrusion target detection method in big data environment," *International Journal of Internet Protocol Technology*, vol. 14, no. 4, pp. 240-247, 2021. https://doi.org/10.1504/IJIPT.2021.118966

[21] K. Vieira, F. L. Koch, J. B. M. Sobral, C. B. Westphall, and J. L. de Souza Leao, "Autonomic intrusion detection and response using big data," *IEEE Systems Journal*, vol. 14, no. 2, pp. 1984-1991, 2020. https://doi.org/10.1109/JSYST.2019.2945555

[22] H. Liu, Y. Zhang, J. Bi, and M. Xing, "Review of technology based on distributed and collaborative network intrusion detection," *Computer Engineering and Application*, vol. 54, no. 8, pp. 1-6, 2018.

[23] E. Viegas, A. O. Santin, and V. Abreu, "Machine learning intrusion detection in big data era: a multi-objective approach for longer model lifespans," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 366-376, 2021. https://doi.org/10.1109/TNSE.2020.3038618

[24] L. Wang and R. Jones, "Big data analytics in cyber security: network traffic and attacks," *Journal of Computer Information Systems*, vol. 61, no. 5, pp. 410-417, 2021. https://doi.org/10.1080/08874417.2019.1688731

[25] S. Dasgupta and B. Saha, "HMA-ID mechanism: a hybrid mayfly optimisation based apriori approach for intrusion detection in big data application," *Telecommunication Systems*, vol. 80, no. 1, pp. 77-89, 2022. https://doi.org/10.1007/s11235-022-00882-6

[26] M. Kalinin and V. Krundyshev, "Security intrusion detection using quantum machine learning techniques," *Journal of Computer Virology and Hacking Techniques*, vol. 19, pp. 125-136, 2023. https://doi.org/10.1007/s11416-022-00435-0

[27] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics*, vol. 11, no. 6, article no. 898, 2022. https://doi.org/10.3390/electronics11060898

[28] H. Albasheer, M. Md Siraj, A. Mubarakali, O. Elsier Tayfour, S. Salih, M. Hamdan, S. Khan, A. Zainal, and S. Kamarudeen, "Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: a survey," *Sensors*, vol. 22, no. 4, article no. 1494, 2022. https://doi.org/10.3390/s22041494

[29] H. Alavizadeh, H. Alavizadeh, and J. Jang-Jaccard, "Deep Q-learning based reinforcement learning approach for network intrusion detection," *Computers*, vol. 11, no. 3, article no. 41, 2022. https://doi.org/10.3390/computers11030041

[30] B. Cao, C. Li, Y. Song, Y. Qin, and C. Chen, "Network intrusion detection model based on CNN and GRU," *Applied Sciences*, vol. 12, no. 9, article no. 4184, 2022. https://doi.org/10.3390/app12094184

Hong Wang **https://orcid.org/0000-0003-1970-801X**

He received a master's degree of information and communication and graduated from Yunnan University in 2011. He is an associate professor at Sichuan Modern Vocational College. His research interests include network and information security.