International Journal of Information Technologies and Systems Approach
Volume 16 • Issue 3

A Network Intrusion Detection Method Based on Improved Bi-LSTM in Internet of Things Environment

Xingliang Fan, Chongqing Vocational College of Applied Technology, China*

Ruimei Yang, School of Big Data and Intelligent Engineering, Chongqing University of Foreign Business and Economics, China

ABSTRACT

When performing malicious network attack detection, traditional intrusion detection methods show their disadvantage of low accuracy and high false detection rate. To address these problems, this paper proposes a novel network intrusion detection scheme based on an improved bi-directional long short-term memory (Bi-LSTM) model under the emerging internet of things (IoT) environment. Firstly, this paper analyzes Bi-LSTM model. Then, it introduces a two-layer attention network structure into Bi-LSTM network. Finally, the corresponding network intrusion detection system is constructed based on the improved Bi STM model. Through simulation experiments, the proposed network intrusion detection method and other three methods are compared under five identical databases. Experimental results show that the false detection rate and detection accuracy of the proposed method are optimal on all sample data, the detection accuracy reaches 97.24% and the false detection rate drops to 5.13%.

KEYWORDS

Attention Mechanism, Bi-LSTM, Feature Selection, Internet of Things, Network Intrusion Detection

1. INTRODUCTION

With the rapid progress of network technology in recent years, the world is in the era of Internet plus. The Internet has become an indispensable tool for people's daily communication. The information network has also gone deep into all aspects of the economic field, and people, goods and commerce have been gradually interconnected through the information network. Although the Internet can bring great help to human life, it also brings security risks too serious to be ignored (Gamage et al., 2020; Ošlejšek et al., 2021; C et al., 2022), such as distributed denial of service, ransomware and other malicious network attacks that take advantage of the availability characteristics of the Internet of Things (IoT) platform. Such attacks are becoming more and more complex and mature and may threaten data integrity, privacy and availability (Gamage et al., 2020; Dong et al., 2020; Asvija et al.,

DOI: 10.4018/IJITSA.319737 *Corresponding Author

2021). Malicious network activities always threaten people's information infrastructure, application and data security, and bring many serious consequences, such as server downtime, unauthorized illegal access, information leakage, tampering and destruction (Hewitt et al., 2021; Nie et al., 2022). Therefore, in order to protect people's lives and even national security, highly accurate and timely security detection of network information is crucial (Singh et al., 2021).

As an important component of the network security protection system, network intrusion detection (NID) technology can effectively identify abnormal data in various complex network environments (Bondgulwar et al., 2021; Chen et al., 2019; Akbulut et al., 2019). It is an effective way to ensure the security of computer networks by stopping various malicious network activities from causing more harm. NID usually builds a network activity model based on machine learning (ML) methods, and detects malicious network activities by evaluating the differences between network intrusion activities and normal behavior (Kaur et al., 2020; Mahmoud et al., 2021). ML models for the purpose of NID are basically implemented based on decision trees, Bayesian network models and support vector machines, which can enhance the classification performance of network activities to some degree, making the security protection systems smarter and more efficient (Liu Wei, 2021; Xie et al., 2019). However, the traditional methods also have the problems of high false positive rate and low accuracy. Deep learning (DL) models are a subset of ML research. Compared with shallow learning model, DL model has stronger fitting ability. Applying deep learning technology to NID has become the research priority for many scholars (Chandramohan et al., 2020).

2. RELATED WORK

In recent decades, DL technology has developed rapidly. Numerous researchers have tried to apply DL technology to network intrusion detection and made some research achievements. According to the characteristics of network traffic, reference (Damasevicius et al., 2021) proposed a NID method based on federated learning that allows multiple ISP or other institutions to conduct joint in-depth learning training while retaining local data. But this method neglects the time sequence of behavior sequence, leading to its low detection rate. Reference (Liu et al., 2021) analyzed the problems of existing NID deep neural network models such as low categorization accuracy and long training time. An integrated deep intrusion detection model was developed based on Stacked Denoising Autoencoder and Extreme Learning Machine to achieve timely response to intrusion, and small batch gradient descent method was used to train and optimize the network. However, this method takes no consideration of the influence of network on the false positive detection rate. Reference (Hu et al., 2021) proposed a NID method based on deep migration learning by modeling the deep migration learning problem and using unsupervised learning deep self-coder for migration learning. However, this method has obvious deficiency in feature extraction of high-dimensional features, making the model run poorly. Reference (Du et al., 2020) developed an intelligent intrusion detection system to promote interoperability between various network communication protocols used in the IoT against network attacks in IoT environment by using DL algorithms to identify malicious traffics in the IoT network. However, this method did not consider the impact of association between features on classification and recognition, and the model still has room for further improvement. Reference (Du et al., 2019) proposed a novel NID method based on semantic re-coding as well as learning of network traffic by taking advantage of the differences between normal traffics and malicious traffics in multiple semantic dimensionalities. However, the network model introduced by this method is a shallow network, and the recognition accuracy is limited. Reference (Hu et al., 2020) proposed an improved NID model based on convolutional depth confidence network to address the issue of low efficiency and scalability of intrusion inspection classification algorithm. However, this method has a low utilization rate of payload data in network traffic, and it is difficult to detect malicious information contained in payload data. Reference (Dong et al., 2021) introduced the multi-scaled residual time convolution (MS Res) block to fine tune the network capability of learning temporal-

spatial representation, and proposed a semi-supervised DL method for intrusion inspection (SS Peep ID) by considering the order characteristics of IoT traffic data. However, the training effect of this method on high-dimensional data is poor.

Recently, several DL-based methods have been proposed to improve the feature extraction ability of abnormal traffic in IoT environment. For industrial IoT security, reference (Li et al., 2022) proposed an intelligent NID system based on improved bidirectional short-term memory network (Bi-LSTM), and the network training speed is accelerated through the Batch Normalization mechanism. However, this method does not take into account the weight calculation of different data attribute features. For marine IoT sensors, reference (Hou et al., 2023) proposed a NID framework combining conditional variational autoencoder (VAE) and Bi-LSTM. The important attack features are extracted from monitored traffic data by generating labeled virtual samples, and the feature extraction ability for rarely appeared attack types is enhanced through loss function improvements. However, this method shows obvious sensitivity deviation to different attack types and is only applicable to specific network environments.

To alleviate the low accuracy and high false detection rate of traditional NID methods for malicious attacks, this paper proposes a NID method based on improved Long Short-term Memory (LSTM) network in IoT scenarios. The basic idea is as follows: (1) The Bi-LSTM connects all the hidden layers of the model to an identical output layer. (2) In the process of Bi-LSTM network coding to form the overall weight eigenvector of network flow, a two-layer attention network structure is introduced. (3) The NID model is constructed based on the improved LSTM network. Compared with the NID method, the innovation of proposed method lies in:

1. Bi-LSTM network is adopted to replace traditional neural network to solve the gradient disappearing issue, and the learning of long-term temporal-spatial context information is realized.
2. In Bi-LSTM network, a two-layered attention mechanism has been incorporated to uniformly encode the input byte word vector, eliminating the problem that some key byte information will affect classification results.
3. Based on the improved long short -term memory network, a NID model is constructed, which improves the detection accuracy to a certain extent.

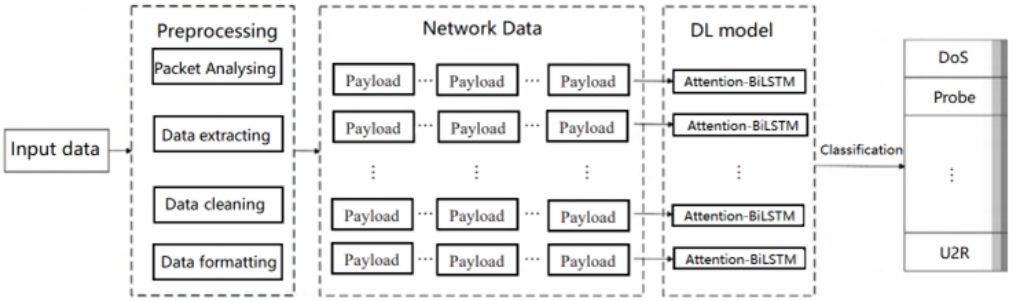
3. PROPOSED NID FRAMEWORK AND RELATED MODEL STRUCTURE

3.1 Network Intrusion Detection Framework

The NID framework based on attention-BiLSTM proposed in this paper is shown in Fig. 1. Firstly, data preprocessing is performed on the raw traffic data, including data packet analysis, data cleaning and other operations, and input data is extracted. Then the attention-BiLSTM is constructed and trained, and finally the model classification result is obtained. The proposed framework is an end-to-end DL model, which automatically learns the feature information of the input data, and realizes the in-depth mining of attack information through the Bi-LSTM neural network. And by integrating the double-layered attention mechanism, the ability to extract important features is improved to achieve accurate detection of attacks of various types and different lengths.

Data preprocessing mainly includes data analysing, data extracting, data cleaning, and data formating steps. First, packet analysis is performed on the raw network data, and the data packets of malicious behaviors are collected based on time stamps, IP addresses, port numbers, etc., then protocol analysis is performed to identify the protocols used by the data packets and clarify the payload data structures and characteristics that can be extracted. In data extracting step, the payload data above the transport layer of each packet is extracted, and useless data such as IP headers are discarded. In traffic cleaning step, the data packets with payload data length of zero are removed, and redundant packets are deleted. Finally, the traffic data is converted to character data type.

Figure 1.
The proposed NID framework



3.2 Bi-LSTM Network

The standard RNN is defined as an artificial neural network, which has the ability to simulate discrete time dynamic systems, but at the same time, one of the main defects of RNN is that it cannot learn long-term context information due to gradient disappearance. This is mainly due to the long-time interval between obtaining input and making decisions, which hinders the ability of RNN to learn distance dependence.

LSTM network algorithm is an extended version of RNN, which adopts the concept of related cell gate. LSTM solves the gradient disappearing problem and can retain context information for a long period. The concept of Bi-LSTM is derived from bidirectional RNN, which processes input sequences by using two different hidden layers, forward and backward input directions. Bi-LSTM structure including multiple continuous time steps is shown in Fig. 2.

In Fig. 2, $x\left(t\right)$ denotes the sequence of input vectors, and $y\left(t\right)$ denotes the sequence of generated output vectors. For a specific time interval t_a, a vector of the input sequence is processed at each time step $t \in \left[1, t_a\right]$, and the internal state vector is defined as $r\left(t\right) = x\left(t\right), \forall t \in \left[1, t_a\right]$. Bi-LSTM can connect all the hidden layers to an identical output layer. The typical limitation of RNN is that only the previous input data sequence context can be used, while Bi-LSTM makes up for this limitation by allowing data to flow forward and backward. Bi-LSTM network uses the following formula to estimate the output sequence $\vec{r}\left(t\right)$ of the hidden layer in the forward direction, $\vec{r}\left(t\right)$ and $y\left(t\right)$ of the hidden layer in the backward direction by iterating the forward network layer from $t = 1$ to $t = t_a$ and the backward network layer from $t = t_a$ to $t = 1$, and then updating the final value:

$$\vec{r}\left(t\right) = R\left(W_{\vec{j}} X_t + D_{\vec{j}} r_{\vec{j}}\left(t-1\right) + b_{\vec{j}}\right) \tag{1}$$

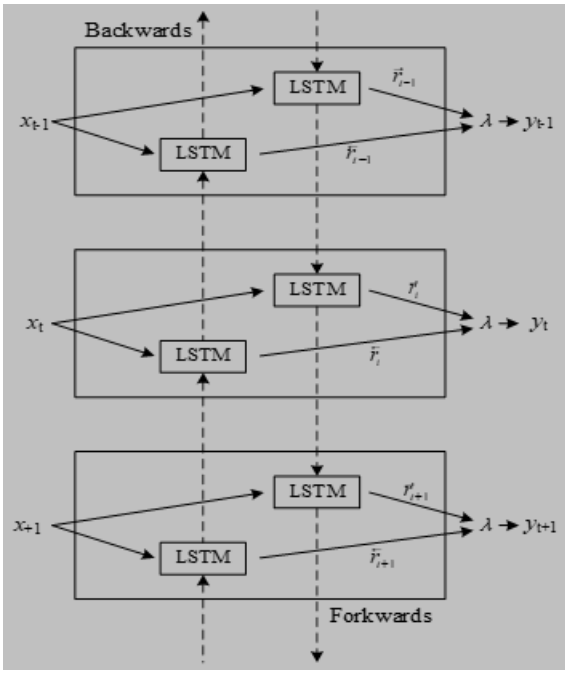
$$\vec{r}\left(t\right) = R\left(W_{\vec{j}} X_t + D_{\vec{j}} r_{\vec{j}}\left(t-1\right) + b_{\vec{j}}\right) \tag{2}$$

$$y\left(t\right) = U_{\vec{j}} r_{\vec{j}}\left(t\right) + U_{\vec{j}} r_{\vec{j}}\left(t\right) + b_y \tag{3}$$

where, W_j refers to the weight matrix between the input layer and the hidden layers, D_j refers to the weight matrix from two successive hidden states $r_j\left(t-1\right)$ and $r_j\left(t\right)$ quality inspection orders, b_j refers to the offset vector in the hidden layer, and λ_j refers to the activation function from which the hidden state is generated. U_j is the weight matrix between the hidden layers and the output layer, b_y is the offset vector in the output layer, and σy is the activation function for the output layer.

The calculation method of final output vector $y\left(t\right)$ is shown in equation (4):

Figure 2.
Bi-LSTM structure with multiple consecutive time steps



$$y\left(t\right) = \lambda_{y}\left(\vec{r}, \overleftarrow{r}\right) \tag{4}$$

The λ_{y} function concatenates the output sequences of neurons in the hidden layer. In RNN training phase, Bi-LSTM is used and characterized by back propagation error learning.

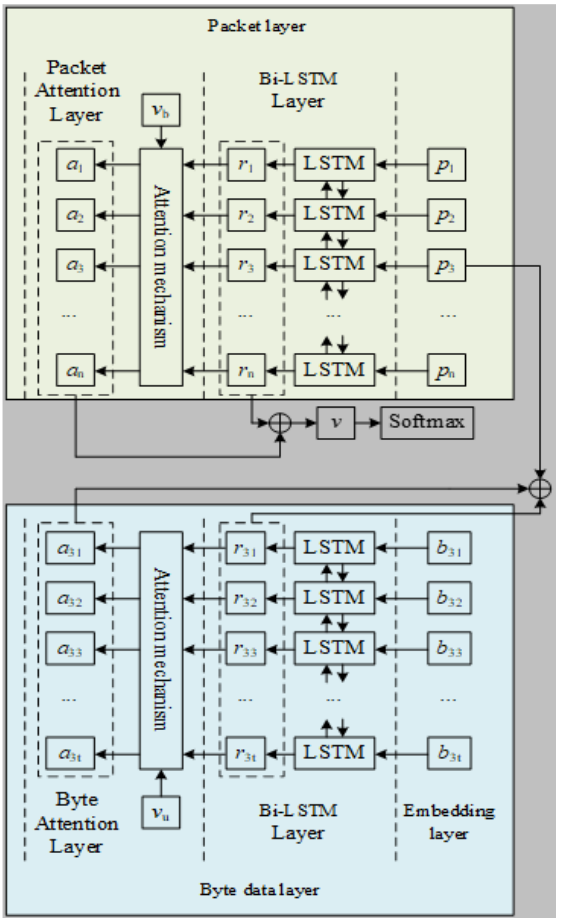
3.3 Bi-LSTM With Double Attention Mechanism

The proposed model is a Bi-LSTM network structure with two-layer attention. According to the characteristics of byte data packet network flow, the model first uses a Bi-LSTM neural network and a byte attention layer Encode byte word vector data with weight information to form packet feature representation. Then, through Bi-LSTM neural network and packet attention mechanism, the overall weight eigenvector of network flow is encoded. Finally, SoftMax function is used for categorization. The network structure is shown in Fig. 3.

In Fig. 3, the input data in word embedding layer is composed of byte sequences of data packets. Similar to word processing in natural language processing, the word embedding layer needs to be used for input coding. Due to the limited range of byte data values, the one pot encoding method is selected here. In byte Bi-LSTM network layer, Bi-LSTM neural network is used to encode word vectors of byte data, learn feature information of byte data, generate forward and reverse feature vectors, and generate feature vectors of data packets after connecting them. The standard Bi-LSTM network model in the byte attention coding layer uniformly encodes the input byte word vector. The contribution of each word vector to the generated intermediate encoding vector is basically the same, but ignoring some key byte information will affect the categorization results. In order to solve this problem, we introduce the byte attention mechanism to calculate the distribution of byte data weights and highlight the byte information with significant influences.

After the data packet vector with byte weight is obtained in the Bi-LSTM network layer of data packet, Bi-LSTM neural network encodes the data packet vector, learns the feature information of

Figure 3.
Bi-LSTM neural network model structure



data packet, generates the forward and reverse feature vectors, and generates the feature vectors of network flow after connecting them. The packet attention encoding layer is similar to the byte information feature encoding. The standard Bi-LSTM network model also has an important impact on the classification results by ignoring some key packet information. By introducing the packet attention mechanism and calculating the weight distribution of packets, the important contribution of packet information will be highlighted.

Finally, the network flow vector is input into the SoftMax classifier. It is linearized into a vector whose length is equal to the quantity of class labels, and cross entropy loss is utilized to minimize the loss function. The advantages of the Bi-LSTM neural network model are listed as follows:

1. Preprocessing technique is utilized to transform symbolic features into numeric values, which can improve the conversion efficiency compared with other preprocessing methods.
2. Taking advantage of CNN model for the parallel extraction of local features, the attribute features are extracted to alleviate the information loss of local features.
3. By utilizing the advantages of LSTM for processing longer sequential data, Bi-LSTM model is used to extract features with long-term dependencies, and the impact of forward and backward directions for each attribute within the sequential data is considered to reduce the false alarm rate of NID.

4. An attention block is incorporated to differentiate the significance of different attributes to pay more attention on more significant features, so as to obtain satisfactory intrusion inspection performance.

4. IMPROVED ATTENTION-BILSTM BASED NETWORK INTRUSION DETECTION METHOD

The diagram of the proposed Attention-BiLSTM based NID model is shown in Fig. 4.

The training and verification of attention-BiLSTM algorithm has the following four main stages:

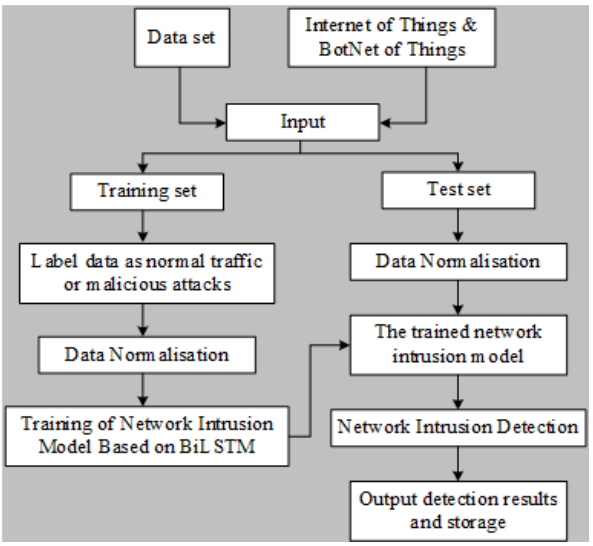
1. **Data input:** At this stage, data will be input into the system, including such datasets as UNSW-NB15 and BoT-IoT. These two datasets contain a large number of security events and compliance observations in the cloud network.
2. **Division of training set and test set:** In this stage, the dataset is divided into training subset and test subset to determine the classification efficiency of attention-BiLSTM algorithm for attack and anomaly observation.

Data standardization: At this stage, the training subset and test subset are standardized to a specific range, for example, to facilitate the use of attention-BiLSTM model for effective fitting data. The Min-Max transformation function shown in equation (5) is used for standardization:

$$y_i\left(x\right) = y_i\left(x\right) - \frac{\min\left[y\left(x\right)\right]}{\max\left[y\left(x\right)\right]} - \min\left[y\left(x\right)\right] \tag{5}$$

where $\min[\ ]$ and $\max[\ ]$ functions represent the maximal and minimal values of each value y_i of feature x in the original set, respectively.

Figure 4.
The NID method based on attention-BiLSTM



4. **Model development: In** this stage, the NID model based on atttention-BiLSTM is built to train and verify its efficiency in classifying attack events.

First, Keras deep learning library in Python is used to establish the attention-BiLSTM model. Each dataset is partitioned into 3 different groups: training, verification and testing accounting for 80%, 10% and 10% respectively. The trained model is used to process each row of the test dataset to verify the model, and the resulting rows are classified as normal or attack records.

5. EXPERIMENT RESULTS AND ANALYSIS

5.1 Experimental Data and Settings

The experiment was conducted in the Ubuntu 17.04 LTS environment, using Keras2.0.2 neural network library to build a network, and using the well-known open source Artificial Intelligence (AI) platform TensorFlow 1.1.0 from Google as the back-end computing framework. In the experiment, the weight and offset of embedded layer and convolution layer are initialized with Normal (0, 0.005) distribution, the weight and offset of LSTM layer and full connection layer are initialized with glorot_normal(0) distribution. The mini-batch method is used to update the network in small batches. When training the Bi-LSTM based NID model, the mini batch-size is set to 64. When training the sequence representation model, the mini batch-size is set to 512. Because the training sample set used for NID model is small, the number of training iteration epoch of network is set to 50, and the rule of early stop is set when $Loss \leq 0.08$ is on the training dataset. When training the sequence representation model, the number of iteration rounds epoch is set to 500. Adam learning algorithm is used to learn model parameters. The specific experimental environment configuration is shown in Table. 1.

In the field of NID, few datasets can be used to train the deep learning algorithm, so KDDcup99 dataset (Tavallaee M.,et al., 2009) is selected for training. KDDcup99 dataset contains five different categories of attacks: Normal, Probe, DoS, U2R and R2L. The data includes 41 fixed feature properties and 1 categorical identifier, which indicates whether the record is unusual. KDDcup99 has 5 million records, and 20% of all data is used as test set and training set. The specific information of the experimental data is shown in Table. 2.

5.2 Experimental Super Parameter Analysis

5.2.1 Experimental Analysis of Convolution Kernel Length

First, we will analyze the effect of convolution kernel length. Eight groups of different convolution kernel lengths are set for experiments to analytically determine the impact of the convolution kernel size on the NID performance of the network. During the experiment, the three performance indicators, namely, missing detection rate, false detection rate and detection cost, were used as the evaluation criteria, and the command embedding dimension was set to 100. The experiment results are shown in Fig. 5.

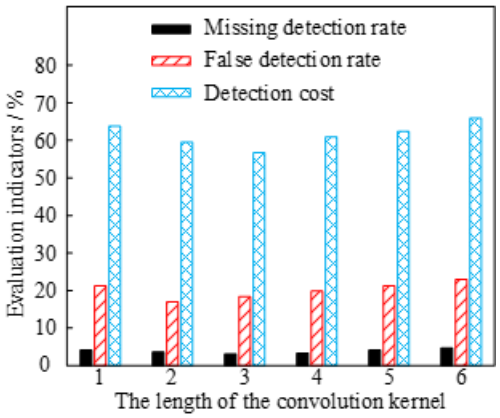
Table 1.
Experiment Configuration

Name	Configuration
Operating system	Ubuntu 17.04 LTS
CPU	Interl Cove i5-3470
Memory	8 GB
Programming language	Python 3.5
DL Framework	TensorFlow1.1.0

Table 2.
Test set and raining set

Type of attack	The amount of data		Proportion	
	Training set	Test set	Training set	Test set
DoS	393, 250	376,800	78.65%	75.36%
Normal	101,550	93750	20.31%	18.75%
Probe	5,050	10,650	1.01%	2.13%
R2L	1000	18,400	0.2%	3.68%
U2R	100	400	0.02%	0.08%

Figure 5.
The effect of convolution kernel length on network performance



We can see from Fig. 5 that the three different metrics exhibit a rising trend followed by a downward trend. When the length of the convolution kernel is set to 2, the missing detection rate reaches the lowest value of 18.2%, and in the case when the length of the convolution kernel is set to 3, the false detection rate and detection cost reach the lowest values of 2.5% and 57.3% respectively. This shows that when the length of convolution kernel is 2, the network has the best detection accuracy for attack samples, but it also sacrifices the detection accuracy for normal samples. In contrast, when the length of convolution kernel is 3, although it does not achieve the best detection rate for attacks, the lowest false detection rate is realized. That is, the ratio of correct samples being wrongly divided is smaller, so the cost of comprehensive index detection is the lowest. Therefore, if the length of convolution kernel is set to 3, the network has the best comprehensive performance. From the figure, we can further draw the following conclusions: (1) When the command sequence has strong local relevance, that is, the context window is 1, and only one command is considered at a time, the situation degenerates to the situation where local correlation is ignored. At this time, the three indicators of the network are all the highest, that is, the detection performance of the network is the lowest. (2) The length of convolution kernel should be kept within a certain range, which will damage the detection performance of the network to a certain extent. This is because the long convolution kernel length will introduce data sparsity, and it will also make it difficult to train the model with too many parameters.

5.2.2 Comparative Analysis of Loss Function

The loss values of the proposed models are compared and analyzed under different batch sizes and learning rates. The results are shown in Fig. 6 and Fig. 7 respectively.

We can see from Fig. 6 that when the batch size is set to 16, the loss value of the proposed model reaches the minimum. In addition, we can see from Fig. 7 that when the learning rate is set to 5, the loss value of the proposed model reaches the minimum. Thus, in order to obtain better performance results, the batch size of the proposed model is set to 16, and the learning rate is determined to be 5.

5.3 Comparative Analysis

To verify the superiority of the proposed NID method, the error detection rate and detection accuracy are respectively taken as the evaluation metrics. For the proposed NID method based on improved LSTM and the methods in reference (Hu et al., 2021), reference (Liu et al., 2021) and reference (Hu et al., 2021), 10 tests were conducted to calculate the average value using DoS, Normal, Probe, R2L and U2R attack types contained in KDDcup99 dataset, respectively. The final calculation results of average false detection rate and detection accuracy of different algorithms are shown in Fig. 8 and Fig. 9.

Figure 6.
Loss values under different batch sizes

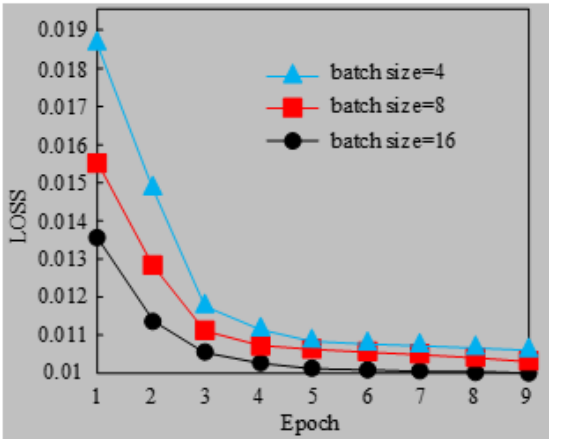


Figure 7.
Loss values under different learning rates

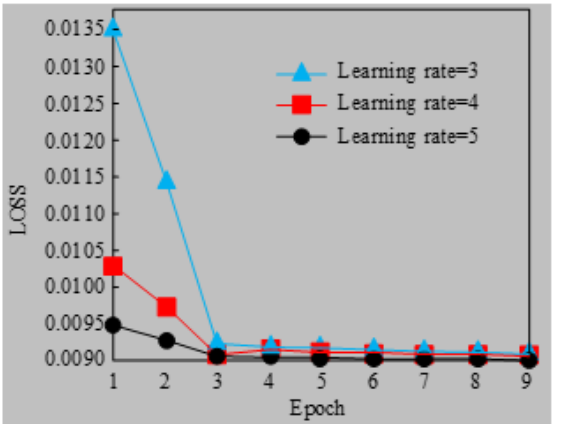


Figure 8.
Comparison of precision of different methods

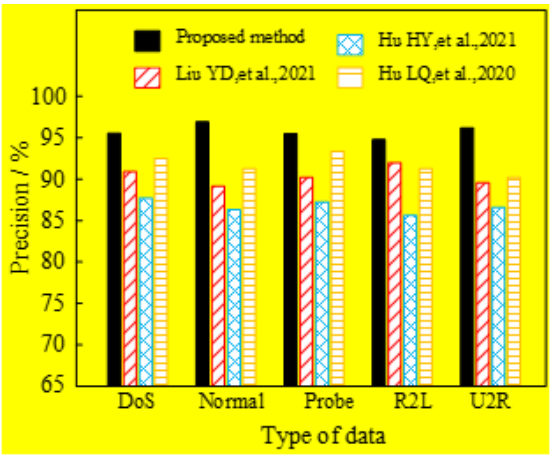
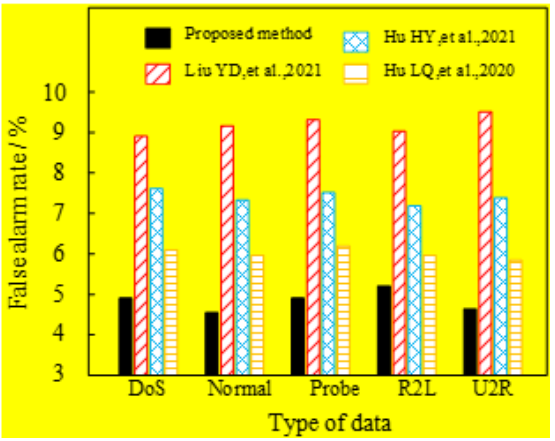


Figure 9.
Comparison of false alarm rates of different methods



We can see from Fig. 8 and Fig. 9 that when five types of sample data are used respectively, the proposed NID method is significantly better than the other three comparison methods in terms of false detection rate and detection accuracy. In terms of overall performance, the detection accuracy of (Hu et al., 2021) is the lowest, the focus of this method is to realize distributed intrusion detection through the federated learning mechanism and solve the problem of local data privacy, but the results proved that its detection accuracy cannot meet the security requirements. The method of (Liu et al., 2021) has the highest false positive rate, because this method is based on the improved Autoencoder model, which cannot deal with the data redundancy and duplicate problems in the dataset, and it has poor data mining ability for minority attack types in the case of unbalanced training samples. The method of (Hu et al., 2020) encodes network traffic from the perspective of semantic analysis, improves the ability to distinguish malicious traffics and normal traffics, and achieves suboptimal performance in both detection accuracy and false positive rates. However, compared with the method in this paper, this method lacks the ability of malicious information mining at the byte level. The lowest average detection rate of the proposed algorithm in five types of samples is 94.98%, and the highest average

detection rate is 97.24%. In comparison with the other three algorithms, the proposed algorithm has a great improvement. The highest average false detection rate of the proposed algorithm in five types of samples is 5.13%, and the lowest average false detection rate is 4.63%, which is lower than the other three comparison algorithms. This is because the Bi-LSTM network can eliminate the problem of gradient disappearance and reduce the time interval from obtaining input to making decisions. In addition, the introduction of the two-layer attention mechanism supports the calculation of the weight distribution of different bytes. On this basis, the bytes with large weight are highlighted to solve the impact of vital byte information on the classification performance, thus improving the detection accuracy of the network intrusion algorithm.

6. CONCLUSION

In order to solve the problems of low accuracy and high false detection rate in using traditional NID methods for malicious attacks, this paper proposes a NID method based on improved LSTM in IoT scenarios. From the experiment results, we can conclude that using Bi LSTM network instead of traditional neural networks can address the gradient disappearing problem, reduce the time interval from obtaining input to making decisions, and improve the detection accuracy. The double-layer attention mechanism is introduced into Bi LSTM network to compute the distribution of byte data weights, give prominence to the byte information with significant influence on model performance, and alleviate the impact of certain key byte information on the categorization results.

The future work will focus on how to use more in-depth learning models (such as DBN, DNN, etc.) to enhance the performance of feature learning. From the perspective of network adaptation, we can achieve the improvement of the accuracy of NID and acceleration of the convergence speed at the same time.

REFERENCES

Abdel-Basse, M., Chakrabortty, R. K., & Hawash, H. (2022). Semi-Supervised Spatiotemporal Deep Learning for Intrusions Detection in IoT Networks. *IEEE Internet of Things Journal*, *8*(15), 12251–12265. doi:10.1109/JIOT.2021.3060878

Akbulut, A., Elmasry, W., & Zaim, A. H. (2019). Empirical study on multiclass classification-based network intrusion detection. *Computational Intelligence*, *35*(4), 919–954. doi:10.1111/coin.12220

Asvija, B., Bijoy, M. B., & Eswari, R. (2021). Security Threat Modelling With Bayesian Networks and Sensitivity Analysis for IAAS Virtualization Stack. (JOEUC). *Journal of Organizational and End User Computing*, *33*(4), 44–69. doi:10.4018/JOEUC.20210701.oa3

Bondgulwar, S., & Gulghane, S., &bShingate, V. (2021*). A Survey on Intrusion Detection System Using Machine Learning Algorithms. International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, Coimbatore, INDIA.

Chandramohan, D., Manimaran, A., & Shrinivas, S. G. (2020). A comprehensive novel model for network speech anomaly detection system using deep learning approach. *International Journal of Speech Technology*, *23*(2), 305–313. doi:10.1007/s10772-020-09693-z

Chawla, S., & Thamilarasu, G. (2019). Towards Deep-Learning-Driven Intrusion Detection for the Internet of Things. *Sensors (Basel)*, *19*(9), 121–129. PMID:31035611

Chen, Z., Rao, X., & Xu, P. (2019). Network Intrusion Detection with Incomplete Information Based on Deep Learning. *Netinfo Security*, *10*(6), 53–60.

Damasevicius, R., Toldinas, J., & Venckauskas, A.. (2021). A Novel Approach for Network Intrusion Detection Using Multistage Deep Learning Image Recognition. *Electronics (Basel)*, *10*(15), 212–220.

Deshmukh, C. D., Mark, E., & Sukte, R. R. (2022). Efficient Cryptographic Protocol Design for Secure Sharing of Personal Health Records in the Cloud. (IJITSA). *International Journal of Information Technologies and Systems Approach*, *15*(1), 1–16. doi:10.4018/IJITSA.304810

Dong, Y. S., He, J., & Wang, R. (2020). Real-Time Network Intrusion Detection System Based on Deep Learning. *10th IEEE International Conference on Software Engineering and Service Science (ICSESS),* (pp. 1-4). IEEE.

Dong, Z. X., Shang, C. J., & Wen, W. M. (2021). An intrusion detection model using improved convolutional deep belief networks for wireless sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, *36*(1), 20–31. doi:10.1504/IJAHUC.2021.112980

Du Guozhen, L. M., & Ji, Z. (2020). Network intrusion detection based on deep transfer learning. *Jisuanji Yingyong Yanjiu*, *37*(9), 2811–2814.

Gamage, S., & Samarabandu, J. (2020). Deep learning methods in network intrusion detection: A survey and an objective comparison. *Journal of Network and Computer Applications*, *169*(5), 88–96. doi:10.1016/j.jnca.2020.102767

Gamage, S., & Samarabandu, J. (2020). Deep learning methods in network intrusion detection: A survey and an objective comparison. *Journal of Network and Computer Applications*, *169*(12), 38–46. doi:10.1016/j.jnca.2020.102767

Hewitt, B., & White, G. (2021). Factors Influencing Security Incidents on Personal Computing Devices. (JOEUC). *Journal of Organizational and End User Computing*, *33*(4), 185–208. doi:10.4018/JOEUC.20210701.oa9

Hou, T., Xing, H., Liang, X., Su, X., & Wang, Z. (2023). A Marine Hydrographic Station Networks Intrusion Detection Method Based on LCVAE and CNN-BiLSTM. *Journal of Marine Science and Engineering*, *11*(1), 221. doi:10.3390/jmse11010221

Hu, H. Y., Xu, C. H., & Tang, Z. Y. (2021). A federated learning method for network intrusion detection. *Concurrency and Computation*, *34*(10), 212–220.

Hu, L. Q., Wu, Z. D., & Wang, J. J. (2020). A network intrusion detection method based on semantic Re-encoding and deep learning. *Journal of Network and Computer Applications*, *164*(23), 23–30.

Kaur, S., & Singh, M. (2020). Hybrid intrusion detection and signature generation using Deep Recurrent Neural Networks. *Neural Computing & Applications*, *32*(12), 7859–7877. doi:10.1007/s00521-019-04187-9

Li, A., & Yi, S. (2022). Intelligent intrusion detection method of industrial Internet of things based on CNN-BiLSTM. *Security and Communication Networks*, *2022*(1), 1–8. doi:10.1155/2022/5448647

Liu, Y. D., He, D. J., & Wang, Z. D. (2021). Intrusion detection methods based on integrated deep learning model. *Computers & Security*, *103*(12), 45–52.

Mahmoud, Q. H., & Ullah, I. (2021). Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access: Practical Innovations, Open Solutions*, *9*(15), 103906–103926.

Nie, L. S., Wu, Y. X., Wang, X. J., Guo, L., Wang, G., Gao, X., & Li, S. (2022). Intrusion Detection for Secure Social Internet of Things Based on Collaborative Edge Computing: A Generative Adversarial Network-Based Approach. *IEEE Transactions on Computational Social Systems*, *9*(1), 134–145. doi:10.1109/TCSS.2021.3063538

Ošlejšek, R., & Pitner, T. (2021). Optimization of Cyber Defense Exercises Using Balanced Software Development Methodology. (IJITSA). *International Journal of Information Technologies and Systems Approach*, *14*(1), 136–155. doi:10.4018/IJITSA.2021010108

Singh, N. B., Singh, M. M., Sarkar, A., & Mandal, J. K. (2021). A novel wide & deep transfer learning stacked GRU framework for network intrusion detection. *Journal Of Information Security And Applications*, *61*(21), 55–63. doi:10.1016/j.jisa.2021.102899

Tavallaee M, Bagheri E, Lu W, et al.,(2009) A detailed analysis of the KDD CUP 99 data set. 2009 IEEE Symposium On Computational Intelligence For Security And Defense Applications, (pp. 1-6).

Wei, L. (2021). Stacked Non-symmetric Deep Auto-encoder Detecting Network Intrusion. *Kongzhi Gongcheng*, *28*(9), 1879–1885.

Xie, X., Yang, X., & Zhang, S. (2019). Intrusion detection method based on a deep convolutional neural network. *Qinghua Daxue Xuebao. Ziran Kexue Ban*, *59*(1), 44–52.