Computer Network Security System Optimization Based on Improved Neural Network Algorithm and Data Search

Huizhong Zhang[1], Fanrong Meng[2,*] and Qinyong Wang[1]

[1]*Zhejiang College of Security Technology, Wenzhou, Zhejiang Province, China 325016*
[2]*Wenzhou University of Technology, Wenzhou, Zhejiang Province, China 325035*
E-mail: Mengfr@163.com
[*]*Corresponding Author*

Received 10 September 2024; Accepted 02 December 2024

Abstract

Frequent hacker attacks and network viruses pose a serious threat to network security, and traditional static and passive defense technologies can no longer meet the high security demands of networks. This paper proposes a computer network security intrusion detection algorithm based on the optimization of convolutional neural networks through transfer learning. Initially, a one-dimensional convolutional neural network (1D-CNN(T)) is trained to form a stable model. Subsequently, in the transfer learning phase, a target function is constructed, experimental data is acquired, and 1D-CNN(E) is trained. The network parameters are fine-tuned using a co-gradient algorithm, and features are extracted from the experimental data, with connections being dropped during the process to reduce overfitting. Based on this algorithm, an intrusion detection system is designed, which is modularly constructed to provide a complete intrusion detection solution. The paper also provides a detailed analysis of modules such as data acquisition, preprocessing, feature

Journal of Cyber Security and Mobility, Vol. 14_1, 75–100.
doi: 10.13052/jcsm2245-1439.1414

extraction, and neural network classifiers. Finally, the performance of the proposed convolutional neural network-based intrusion detection algorithm, which leverages transfer learning, is validated through a neural network model experiment. This algorithm is compared with other classical classification algorithms commonly used in intrusion detection, including RF, AlexNet, LeNet-5, CNN, and BiLSTM, each serving as a classifier. The experimental results demonstrate that the proposed classification algorithm achieves an accuracy of 83.58% and a recall rate of 84.49%. Compared to the RF algorithm, the proposed method exhibits improvements of 8.87% in accuracy and 9% in recall rate. When benchmarked against the AlexNet model, the accuracy and F1-Measure of the proposed algorithm are enhanced by 6.56% and 7.26%, respectively. The classifier in TLCNN-IKNN achieves the highest classification accuracy using weighted Euclidean distance, with an accuracy rate of 99.07% for detecting COMBO attacks in the Bashlite botnet, 97.02% for detecting Junk attacks, 97.71% for detecting Scan attacks, 98.08% for detecting TCP attacks, and 100.00% for detecting UDP attacks. These findings underscore the effectiveness of the transfer learning-based intrusion detection algorithm in boosting the detection rate of intrusion detection systems and reducing false positive rates, thereby establishing its high practical application value.

Keywords: Intrusion detection, Network security, Transfer learning, Convolutional neural network, Cogradient algorithm.

1 Introduction

With the rapid advancements in computer technology and network communication, coupled with the booming information industry, information technology and networks have become deeply ingrained in various core sectors of today's society, including scientific and educational progress, economic development, cultural heritage, and e-commerce, playing a pivotal role [1]. However, as society's dependence on computer networks continues to intensify, the consequences of a breach in computer network security are unthinkable. Not only would it severely disrupt social order, but it would also incur substantial economic losses and even pose a threat to national security [2]. Given the severity and urgency of computer network security issues, it has emerged as a major focus in the field of information science research today. Network intrusion detection technology, as an indispensable part of the network security technology system, holds paramount significance

in preventing cyber-attacks and safeguarding network security. Therefore, delving deeply into the research and exploration of network intrusion detection technology is not only a necessary measure to address current network security challenges but also an urgent requirement to align with China's current national conditions and promote the development of the information security industry.

Post the 1970s, the burgeoning development of large-scale and very large-scale integrated circuits ushered in an era of enhanced computer performance and compact size, facilitating their pervasive application across various facets of society. This evolution amplified the demand for computer security [3]. As traditional firewalls proved inadequate against sophisticated network hacker intrusions, the stage was set for the advent of intrusion detection technology [4]. The technology has traversed four distinct research phases: initial explorations, host-based intrusion detection system research, network-based intrusion detection system research, and intelligent network intrusion detection system research. China's exploration into network intrusion detection, however, has lagged behind its international counterparts. It wasn't until the mid-to-late 1980s that the Computer Security Professional Committee of the China Computer Society, under the leadership of Miao Daoji, commenced its operations [5]. This was followed by a concerted effort from numerous domestic experts and scholars who engaged in the research and development of network intrusion detection technology [6]. The state's significant investment in human and material resources for intrusion detection research catalyzed the advancement of computer network security. In alignment with these national efforts, domestic computer network security enterprises have embarked on their own research and development initiatives. For instance, Beijing Zhongke Wangwei Information Technology Co., Ltd. developed the "Tianye" Network Intrusion Detection System, a real-time network-based intrusion detection solution [7]. Additionally, Qiming Star Information Technology Co., Ltd. crafted the "Tiantian" Hacker Intrusion Detection System, a dynamic intrusion detection and response system, contributing to the robust landscape of computer network security in China.

Intrusion detection systems commonly employ a variety of detection methods, including feature detection, anomaly detection, state detection, protocol analysis, and more. Each of these methods has inherent shortcomings. For instance, anomaly detection often relies on statistical methods, and the determination of the threshold value in these methods can be challenging; a threshold set too low can result in a high number of false positives, while a threshold set too high can lead to a high number of missed detections.

This paper harnesses transfer learning technology to optimize convolutional neural networks, addressing the issues of slow convergence and local minima in traditional neural networks. The aim is to develop an intrusion detection algorithm that is well-suited to the security needs of computer networks. The basic model 1D-CNN(T) is formed by training one-dimensional convolutional neural networks on training data sets. Then, 1D-CNN(E) is further trained under the transfer learning framework by constructing the objective function and using the experimental data set. In this process, the common gradient algorithm is used to optimize the network parameters and reduce the overfitting of the network by dropping some connections, so that the parameters can be fine-tuned in the feature extraction stage. Finally, based on the proposed intrusion detection algorithm, a modular intrusion detection system is designed, and the data acquisition, preprocessing, feature extraction, neural network classifier, and response module of the system are analyzed in detail.

2 Convolutional Neural Network Intrusion Detection Algorithm Based on Transfer Learning

2.1 Data Preprocessing

The data preprocessing module includes three operations: symbol data conversion, data normalization, and conversion to the corresponding gray-level map. The purpose is to input the characteristic data of intrusion detection data set into the network model through normalization and conversion to the corresponding gray level map.

The features in the KDDCUP99 data set are continuous and discrete and can be subdivided into numerical features (32 features), binary features (6 features), and symbolic features (3 features) according to the attributes of the data. In the process of data processing, it is necessary to convert all 41-dimensional features of each data in the KDDCUP99 data set into higher-dimensional binary data [8].

In the numerical attributes of the KDDCUP99 dataset, the value ranges of continuous numerical features are different, and the features with larger value ranges have a greater influence on decision-making. The data set contains 32 numerical features, so all the continuous data are normalized:

$$xi = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \quad 1 \leq i \leq n \tag{1}$$

After data normalization, each numerical feature represents the number between 0 and 1, with an interval of 0.1, and is converted into a binary number, the numerical feature can be represented by a 10-dimensional binary, and the 32 numerical features can be represented by a 320-dimensional binary.

In the KDDCUP99 data set, each traffic data has 41 bits of valid features, and the 42nd bit of feature is a label. The 42nd bit of the tag represents 4 types of attacks and 37 types of small attacks. When normal traffic features are added, a total of 38 bits of attribute value are converted into 38-dimensional binary data [9]. After data processing, the 41-dimensional feature data is expanded into 84-dimensional symbolic features, 12-dimensional binary features, 320-dimensional numerical features, and 38-dimensional attack-type features, totaling 454-dimensional data.

After the 41-dimensional features of the dataset are extended to 454-dimensional data by data preprocessing, dimensionality reduction operations are needed to process them into n*n image data format as the input of the model. Here, the variance coefficient is used as the basis for dimensionality reduction screening, and the function is defined as:

$$\mathrm{CV} = \frac{\sigma}{\mu} \tag{2}$$

Where σ represents the standard deviation and u represents the mean. A larger variance coefficient indicates a more concentrated feature distribution. After comparison, dimensions with smaller variance coefficients are removed.

2.2 Intrusion Detection Optimization of Convolutional Neural Networks

CNNS (Convolutional Neural Network) were originally used as deep learning neural networks for 2D visual image processing and classification, automatically extracting complex features from images through filters (cores). In network security intrusion detection, the matrix formed by the time series of feature vectors can be regarded as an image, and one-dimensional data can be processed through CNN, that is, the convolution axis corresponds to the time, and the feature vector is connected with the continuous time series to represent the relevant local features.

Transfer learning represents a sophisticated machine learning approach that leverages pre-existing knowledge for application in distinct yet related

domains, facilitating problem-solving through the strategic transfer of knowledge. This methodology enables the transference of parameters from a pre-trained model to a nascent one, thereby expediting the model's optimization and enhancing the training process [10]. Investigating the application of transfer learning in Convolutional Neural Networks (TLCNN) within the context of network security detection models addresses the dual challenges faced by traditional Convolutional Neural Networks: the scarcity of adequate training samples in small data scenarios and the time-consuming nature of network training. Consequently, TLCNN proves effective in extracting intricate features of network attacks and accurately detecting and classifying such intrusions.

In this section, it is proposed that the convolutional neural network (TLCNN) based on transfer learning is used for network security detection and classification. The data items in the network intrusion data set are usually from network packets, which are input in vector form [11]. The TLCNN network architecture consists of two one-dimensional CNNS (1D-CNNs) combined in series and a fully connected network, where 1D-CNN(T) is used for training data and another 1D-CNN(E) is used for experimental data. Each 1D-CNN contains multiple layers, each made up of three basic computations: convolution, pooling, and discarding. The network structure of a convolutional neural network (TLCNN) based on transfer learning is shown in Figure 1.

This section proposes a convolutional neural network based on transfer learning for classification and identification of network attacks in network security situation awareness. The basic idea is to first train one-dimensional CNN networks based on training data sets and form stable 1D-CNN(T). Then, this paper proceed to the stage of transfer learning. The introduction of transfer learning aims to leverage the knowledge acquired from previous tasks to accelerate the learning process of new tasks and enhance the model's generalization ability. To this end, this paper constructs an objective function that guides the model to adaptively learn on the new experimental dataset. In this process, 1D-CNN(T) is utilized as a pre-trained model, upon which a new 1D-CNN model, denoted as 1D-CNN(E), is built. Through further training on the new dataset, 1D-CNN(E) is able to learn feature representations that are more relevant to the current task. To further improve the model's performance, this paper employs the conjugate gradient algorithm to finely tune the network parameters. This step not only helps the model better adapt to the new dataset but also reduces the risk of overfitting to some extent. During the fine-tuning process, feature extraction is simultaneously conducted, which

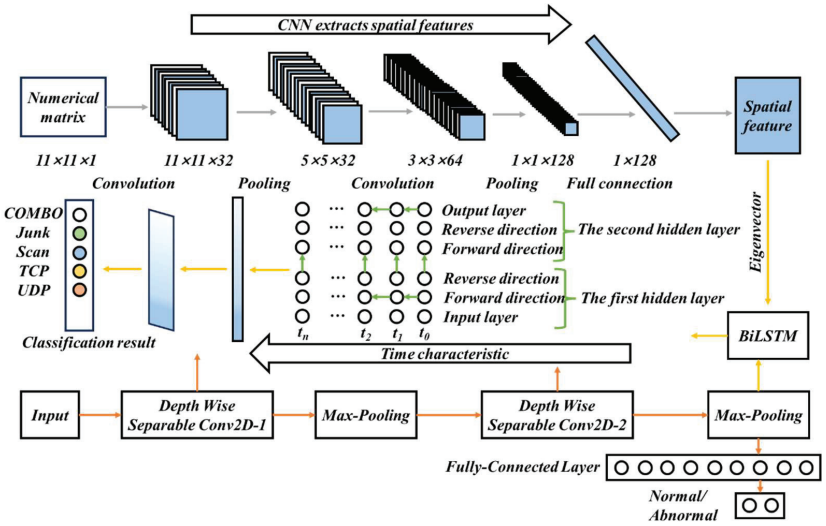

Figure 1 Convolutional neural network model structure based on transfer learning.

involves strategically dropping certain connections to reduce the complexity of the network, thereby further enhancing the model's generalization ability [12]. Finally, the output results of the convolutional neural network (TLCNN) based on transfer learning are input to the KNN (K-Nearest Neighbors) classifier based on similarity distance optimization through the fully connected layer, and the weighted Euclidean distance between the judgment data and the sample of the experimental dataset is calculated. The K points with the smallest distance are selected, and the classification categories of discriminant data are obtained to obtain the final decision.

The steps of the classification algorithm based on TLCNN-IKNN proposed in this section are as follows:

Step 1: Train 1D-CNN(T) on the training data set.

(1) Initialize ownership values and biases randomly.

(2) Forward propagation algorithm.

The calculation process of the convolution layer and pooling layer is as follows:

$$\begin{cases} x_j^l = f\left(\sum_i x'_{j-1} w_{ij} + b_j^l\right) \\ x'_j = g(\beta'_j \mathrm{down}(x_j^{l-1}) + b'_j) \end{cases} \tag{3}$$

Where 1 is the current layer, b is the bias of the current layer, and β is the multiplicative parameter of the current layer. g(x) is the pooling function. There are two types of pooling methods: max pooling and average pooling. This section adopts the max pooling method.

(3) Back propagation algorithm.

Based on the cross-entropy cost function and the introduction of regular terms, the overall cost function C is constructed as follows:

$$C = -\frac{1}{N}\sum_{x_j}[y_j \log a_j^l] + \frac{\lambda}{2N}\sum_{w} w^2 \tag{4}$$

Calculate the bias derivation of the weight and bias and adjust the weight and bias:

$$\begin{cases} w_i^l \to w_i^l - \eta\dfrac{\partial C}{\partial w_i^l} \\ b^l \to b^l - \eta\dfrac{\partial C}{\partial b^l} \end{cases} \tag{5}$$

(4) All network parameters are stable.

Step 2: Train 1D-CNN(E) on an experimental data set.

(1) The initialization uses 1D-CNN(T) network parameters.
(2) Forward propagation algorithm, same as 1D-CNN(T).
(3) Construct the objective function.

The conjugate gradient algorithm is used to optimize the parameter values in the convolutional neural network so that the overall cost function of 1D-CNN on the experimental data set reaches a small value [13].

$$\min f(x) \stackrel{\text{def}}{=} C(w,b), \quad x \in R^n \tag{6}$$

$$g_j = \nabla f(x^{(j)}) \tag{7}$$

(4) Back propagation algorithm.

With 1D-CNN(T), the conjugate gradient algorithm is used to optimize 1D-CNN(E), and the network parameters are fine-tuned. The conjugate gradient algorithm is as follows:

Given the initial point, allowed error, set:

$$y^{(1)} = x^{(1)}, d^{(1)} = -\nabla f(y^{(1)}), \quad k = j = 1 \tag{8}$$

$$f(y^{(j)} + \lambda_j d^{(j)}) = \min_{\lambda \geq 0} f(y^{(j)} + \lambda d^{(j)}) \tag{9}$$

make:

$$y^{(j+1)} = y^{(j)} + \lambda_j d^{(j)} \tag{10}$$

$$\beta_j = \frac{\| \nabla f(y^{(j+1)}) \|^2}{\| \nabla f(y^{(j)}) \|^2} \tag{11}$$

$$\beta_j = \frac{g_{j+1}^T (g_{j+1} - g_j)}{g_j^T g_j} \tag{12}$$

Step 3: Determine the sample weight of the experimental data set.

To effectively solve the problem of category imbalance, different weights can be assigned to samples of different categories [14]. The calculation method of sample weight of the experimental data set is:

$$r_i = \frac{n_i}{n} \tag{13}$$

Where: n indicates the total number of samples. r_i represents the weight of the i-th category of samples, and n_i represents the number of samples in the y_i-th category.

Step 4: Calculate the weighted Euclidean distance between the judgment data and the training sample.

The following distance formula is used to calculate the distance of each training sample Xj in X and S [15]:

$$d(X_i, X_j) = \left(\sum_{d=1}^{p} r_i^2 \lambda_i^2 |(x_{i,d} - x_{j,d})^2| \right)^{\frac{1}{2}}$$

$$s.t. \sum_{i=1}^{N} r_i = 1, r_i \geq 0$$

$$s.t. \sum_{i=1}^{N} \lambda_i = 1, \lambda_{ii} \geq 0 \tag{14}$$

$$\lambda_i = \frac{\left[\frac{1}{d(X_i, X_j)^l} \right]}{\sum_{i=1}^{n} \left[\frac{1}{d(X_i, X_j)^l} \right]} \tag{15}$$

Step 5: Select the K points with the smallest distance.

Step 6: Decide to get category categories.

Finally, determine the sample category:

$$\hat{y} = \arg\max_{c} \sum_{i \in KNN(X), y_i = c} w_j \tag{16}$$

2.3 Experiment

The dataset employed in this research is KDD Cup 1999, which was compiled by the Lincoln Laboratory of the Massachusetts Institute of Technology from the network data of the United States Air Force. Widely recognized for evaluating and testing intrusion detection task methodologies, the KDDCUP99 dataset encompasses four major categories of attacks and 39 subcategories. These are categorized as denial-of-service attacks (DoS), unauthorized access from remote hosts (R2L), unauthorized local superuser privileged access (U2R), and port monitoring or scanning (PROBE). Each record in the dataset consists of 41 feature vectors and a corresponding label, indicating normal or attack activity. For this study, a subset of the KDDCUP99 dataset was utilized for model training. This subset includes 41,168 training samples and 24,883 test samples, with the distribution of various intrusive behaviors detailed in Table 1.

The small-batch training method was used to make the samples fully trained. In this paper, a total of 50 iterations were set during the training process, and 20% of the training set was observed as the verification set.

The performance of network attack classification recognition is finally measured by the following indicators:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{17}$$

Table 1 Distribution of 10% KDDCUP99 data sets

Attack Type	Number of Training Sets	Number of Experiment Sets
COMBO	8134	4864
Junk	32561	17980
Scan	97	328
TCP	6	1694
UDP	346	8

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{19}$$

$$FNR = \frac{FN}{TP + FN} \times 100\% \tag{20}$$

$$FRP = \frac{FP}{FP + TN} \times 100\% \tag{21}$$

Where TP denotes a true positive case, TN is a true negative case, FP is a false positive case, and FN is a false negative case.

The classifier in TLCNN-IKNN adopts an N algorithm based on inverse square distance weighted Euclidean distance optimization. The effects of different distance functions on the classification of COMBO attacks are shown in Figure 2. Among them, the classification accuracy rate of weighted European distance is the highest. The accuracy rate of detecting COMBO attacks in the Bashlite botnet is 99.07%, that of detecting Junk attacks is

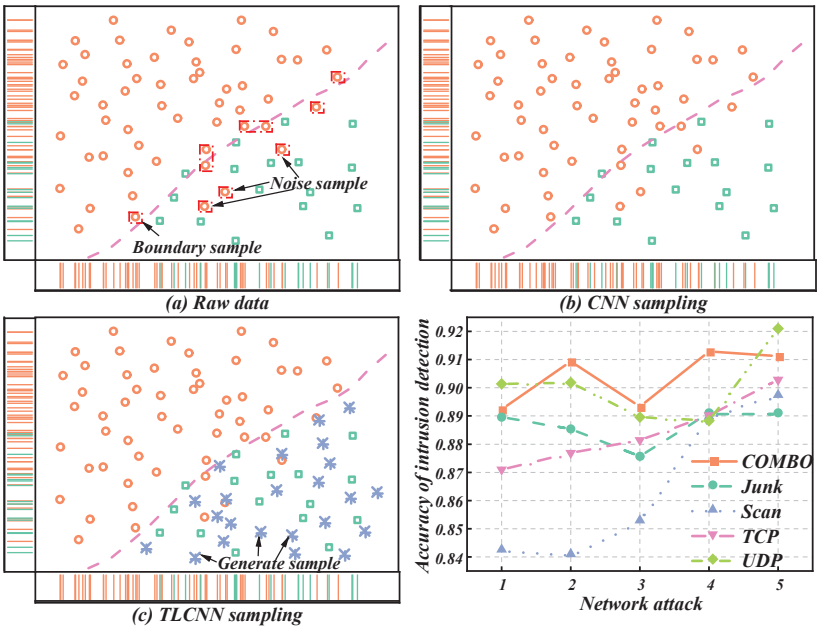

Figure 2 Intrusion detection by TLCNN-IKNN algorithm.

97.02%, and that of detecting Scan attacks is 97.71%. The accuracy of TCP (Transmission Control Protocol) attack detection is 98.08%, and the accuracy of UDP attack detection is 100.00%.

3 Network Security Intrusion Detection System

3.1 Data Preprocessing Module

To process the collected data preliminarily, a data preprocessing module is designed. The captured network data in the data acquisition module is binary data, and the analysis of the network data means decoding the captured binary data flow according to the network protocol structure and extracting the information of interest to the intrusion detection system [16]. A large number of packets are processed into TCP connection records reflecting session information and then complete data filtering, format conversion, window-based information statistics, and other work so that the data format can be accepted by the detection engine for further processing. Each record contains the complete feature set of the network connection record, which includes 41 features.

(1) Parsing the intercepted raw network data packets into session-based connection records involves two steps. The first step is to convert the binary, unprocessed network data packets obtained from the acquisition module into American Standard Code for Information Interchange (ASCII) records. From the processed output, it can be seen that each line represents the necessary information of a network data packet (or packet) [17]. Each packet contains a time field, and packets are arranged according to the order of the timestamp.

In the second step, the format records in the table cannot reflect the essential characteristics of the data well and need to be analyzed by the protocol hierarchy model. Then the analyzed data is converted into the format shown in Table 2, so that the connection record is relatively clear, and the complete process of the session is reflected in each record.

(2) Perform data filtering, leaving only the records of the target positive address belonging to the VLAN, and filtering out other records.

Table 2 Link format

Serial number, timestamp flag, source IP address, destination IP address, source PORT, Duration, End flag, data volume, emergency flag, resending flag, resending condition.
16, Mar21 18:15:43.048120, 210.44.64.135, 210.44.67.9, 2926, 8000, 0:00:01.191029, 1 716, 0, 0, 0.

Table 3 Records the format after the conversion

Serial number, timestamp flag, source IP address, destination IP address, source PORT, Duration, End flag, data volume, emergency flag, resending flag, resending condition.
16, 181543.048, 2104464135, 21044679, 2926.00, 8000.00, 1.19, 716.00, .00, .00, .00.

When an intrusion occurs, the intruder is always the one who initiates the connection request. That is to say, the intruder's host plays the role of the source in this connection, while the victim plays the role of the destination. The responsibility and goal of the intrusion detection system is to ensure the security of the internal network information, so if the purpose of the data P does not belong to the connection record to the VAN, then there is no need to do further processing, and it is just filtered out directly.

(3) Data format conversion of related records.

The attributes in the table are not only numeric, but also time-type attributes and date-type data, and also contain a variety of formats such as character data. This mixed data type cannot be processed, and the prerequisite for the processor to be able to accept the processing is that the data in these different formats need to be converted into numeric data types, as shown in Table 3.

(4) The converted data in the numerical standard format will be reprocessed according to the corresponding time-size window, and the statistical value will be further calculated.

TCP connection records are the result of a series of processing steps applied to the binary network data captured by the network data acquisition module. Firstly, the collected binary network data packets are parsed into session-based connection records, which involves converting the binary data packets into American Standard Code for Information Interchange (ASCII) records. These records are then analyzed using a protocol layer model to transform them into a specific format. Subsequently, data filtering (retaining only records where the destination IP address belongs to the local VLAN), data format conversion (converting various data types into numerical data), and additional processing within time-based windows to calculate statistical values are performed, ultimately yielding the TCP connection records.

Although the TCP connection record after initial processing contains some inherent characteristics, it can also reflect the internal information of a TCP connection. However, such as scanning attacks and denial of service attacks, these characteristics are often not only reflected in a single record but also reflected in the relationship between connected records [18].

(5) Divide the statistical data according to the P group in the auxiliary decision module, divide the data of the target IP as the object, and store the data in different files.

3.2 Neural Network Classifier Module

Apart from the input and output layers, a neural network typically comprises one or more intermediary hidden layers, characterized by an absence of connections between nodes of the same layer, while forming a fully connected network between adjacent layers of neurons. Owing to the lack of coupling between nodes within the same layer, the neurons of each layer exclusively receive inputs from the preceding layer's neurons, and, correspondingly, the output of each layer's neurons solely impacts the output of the subsequent layer's neurons. This architecture ensures a sequential data flow from input to output, facilitating the network's ability to learn complex representations.

The error backpropagation learning algorithm for neural networks is a generalized form of the least mean squares algorithm. There exists a mean squared error between the actual output and the desired output of the neural network, and the error backpropagation algorithm uses the gradient descent method to recursively solve for the weights of the network and the thresholds of each node according to the criterion of minimizing the cost function [19]. At the inception of network training, weights, and node thresholds are initialized with random values, while the expected output values for nodes are predetermined. Following the input of training data, the network's cost function becomes calculable [20]. Through the neural network, errors are propagated backward, layer by layer, towards the input layer. This process enables the network to continuously and adaptively adjust its weights and node thresholds, thereby reducing the cost function value until it attains an acceptable minimum value, denoted as ε, or ceases to decrease further. The learning sequence for a neural network unfolds as follows:

(1) Initialization. Assign random values within the interval $(-1, 1)$.
(2) Randomly select a set of inputs and target samples to provide to the network.
(3) Calculate S and B.
(4) Calculate L and C.
(5) Calculate the generalized error of each unit of the output layer.

$$d_t^k = (y_t^k - c_t)c_t(1 - c_t) \quad t = 1, 2, \ldots, q \tag{22}$$

(6) Calculating the generalized error of each element of the hidden layer [21].

$$e_j^k = \left[\sum_{t=1}^{q} d_t \cdot v_{jt}\right] b_j(1-b_j) \tag{23}$$

(7) Amended.

$$\begin{aligned} &v_{jt}(N+1) = v_{jt}(N) + \alpha \cdot d_t^k \cdot b_j \\ &\gamma_t(N+1) = \gamma_t(N) + \alpha \cdot d_t^k \\ &(t = 1, \quad 2, \ldots, q; j = 1, 2, \ldots, p; \quad 0 < \alpha < 1) \end{aligned} \tag{24}$$

(8) Amended.

$$\begin{aligned} &w_{ij}(N+1) = w_{ij}(N) + \beta \cdot e_j^k \cdot a_i \\ &\theta_j(N+1) = \theta_j(N) + \beta \cdot e_j^k \\ &(i = 1, \quad 2, \ldots, n; j = 1, \quad 2, \ldots, p; \quad 0 < \beta < 1) \end{aligned} \tag{25}$$

(9) Select the next learning sample randomly to provide to the network, and return to step (3) until m training samples have been trained.

(10) Re-select a set of input and target samples randomly from the m learning samples, and return to step (3) until the global network error E is less than a predetermined minimum ε, indicating that the network has converged. If the number of learning iterations exceeds the current value, the network will not converge [22].

As can be seen from Figure 3, as the decomposition frequency decreases, the predicted value of the low-frequency component has a higher and higher coincidence with the actual value, and the actual value of the component has a smaller deviation from the predicted value. The prediction error of

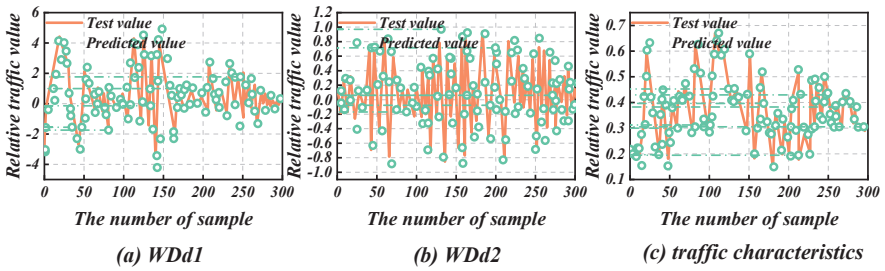

Figure 3 Comparison of test data with predicted values.

the high-frequency component increases, and there is a deviation between the predicted value and the actual value at a few time series data points. The main reason is that the non-stationarity of the traffic characteristic data leads to the peak error of the high-frequency data. From the overall prediction effect of the model, the prediction accuracy is high on most of the time series traffic characteristics data, and the prediction model has a good fitting trend. In short, the TLCNN-IKNN prediction model predicts network intrusion by monitoring traffic anomalies in real-time, and can accurately predict short-term network security situations. However, such methods cannot identify the types of intrusion attacks, so other intrusion characteristic data need to be integrated for analysis. When experienced attackers take some measures to ensure that the overall characteristics of network traffic do not change significantly to launch attacks, for example, when an attacker chooses to launch attacks when worms are causing abnormal network traffic, the prediction error of TLCNN-IKNN prediction model will increase. At this time, the GNNMARIMA model, which can predict the trend of network traffic over a long period, should be integrated for analysis [23]. Therefore, the convolutional neural network prediction model, as a supplement to the prediction method based on network intrusion events, has a high accuracy in short-term network security situation prediction.

4 Improved Intrusion Detection Experiment of Neural Network

4.1 Experiment Preparation

This section presents training and detection examples, using actual network data packets as the data source, to obtain the experimental results of NNIDM in simulation detection, and to calculate the accuracy, false positive rate, and false negative rate. The data packet attributes used include source IP, destination IP, source port, destination port, packet length, protocol number, Seq (sequence) number, Ack (acknowledgment) segment, Ack bit, URG bit, etc.

Convolutional neural network parameter setting:

(1) Number of network input layer nodes

There are 164 nodes in the input layer, and 164 is the binary digit corresponding to the data packet attribute value determined by the experiment. For the specific meaning, see Table 4. The number of digits can also be flexibly increased or decreased according to needs.

Table 4 Mapping between output values and attack behaviors

Expected Output	Aggressive Behavior	Description	Agreement	Owning Layer
000	Normal	Normal behavior		
001	Land attack	The source IP address is the same as the destination IP address	IP	Network layer
010	Scan null	Both seq and ack are 0	TCP	Transport layer
011	U2R	The packet length is greater than 65536 bytes	IP	Network layer
100	Probe	The source IP address is unclear and the Ack bit does not respond. Procedure	TCP	Transport layer
101	R2L	The destination port is 139 and the URG bit is OOB	TCP	Transport layer
110	Smurf attack	The destination address is a broadcast address	ICMP	Network layer
111		Inactive		

(2) The number of nodes in the hidden layer of the network

The number of hidden nodes is generally obtained from experience and experiment, and 20, 40, 60, 80, 100, and 120 are used for testing. According to the experiment, the detection effect is best when the hidden node is 80.

(3) Number of network output layer nodes

Select 3 nodes, each node has two outputs of 0 and 1, and there are 8 types of group sum, which can also be increased according to needs during detection.

(4) Determination of initial weights

The initial weight is a set of values that should not be exactly equal, producing a set of random numbers from -1 to 1 in the program as the initial weight of the network.

(5) Determination of the initial threshold

The initial threshold is also a set of values that should not be exactly equal, producing a small set of random numbers in the program that serve as the initial threshold of the network.

(6) Training rate

Provided that the learning rate does not induce oscillation, a larger value is preferable. In this study, the initial value is set to 0.6. The program will

automatically adjust the learning rate based on the training error to reduce oscillation and achieve optimal performance.

(7) Allowable error

The specified error threshold is informed by empirical judgment. In pursuit of minimizing both the false positive rate and misclassification, different allowable error levels are assigned to various stages. In the context of this study, the allowable error threshold ranges between 0.000001 and 0.025.

4.2 Comparison of Intrusion Algorithms

The classical classification algorithms typically employed in intrusion detection systems have been contrasted with the novel algorithm presented in this study. Comparative methodologies encompass Random Forest (RF), AlexNet, LeNet-5, Convolutional Neural Networks (CNN), and Bidirectional Long Short-Term Memory (BiLSTM), each serving as a distinct classifier. Our findings reveal that, in comparison to alternative classifiers, the proposed algorithm achieves a precision of 83.58% and a recall of 84.49%. This represents enhancements of 8.87% and 9% in accuracy and recall, respectively, over the RF algorithm. Furthermore, juxtaposed with the AlexNet model, our algorithm exhibits a 6.56% and 7.26% augmentation in accuracy and F1 measure, respectively. A graphical representation of these comparative metrics for the various methods is depicted in Figure 4.

The F1-Measure represents the balance between precision and recall. It can be viewed as the harmonic mean of precision and recall. Therefore, Figure 4 summarizes the F1-Measure values for different categories and compares them with other methods. Specifically, for the category labeled as Normal, the F1-Measure reaches 92.15%, which is 13.92% higher than the RF metric. It is more intuitively evident that for the U2R category with extremely limited data, the classification performance also shows a significant improvement compared to other methods.

Across the board, the classification outcomes for U2R and R2L categories lag behind those of other classifications. This discrepancy can primarily be attributed to the paucity of samples in these categories within the training set. Consequently, the classifier garners fewer pertinent features pertinent to these specific attack types during the learning phase. The method delineated herein, however, has notably escalated the detection efficacy for U2R, affirming its capability to mitigate the low detection rate stemming from data imbalance, albeit partially. Hence, our proposed methodology addresses the challenge of

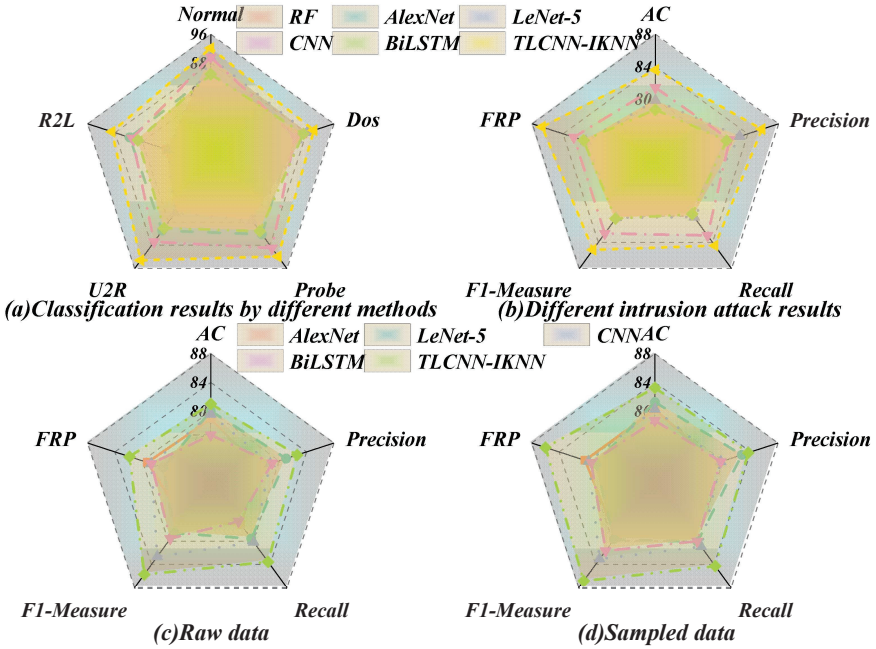

Figure 4 Comparison of intrusion algorithms.

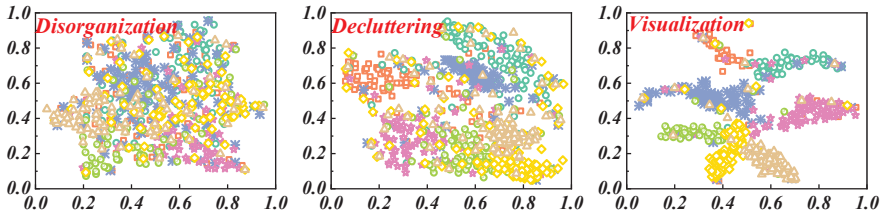

Figure 5 Convolutional neural network visualization.

category imbalance in Intrusion Detection Systems (IDS) implementation to a considerable degree.

The model's output is visually depicted in Figure 5, with (a) representing the output of the perceptron model, (b) the output of the convolutional neural network, and (c) the output of the model proposed in this paper. When the model parameters are kept constant, the convolutional model exhibits an accuracy improvement of 19.8% over the multi-layer perceptron. The recall rate and F1-score for accuracy are also significantly enhanced, with

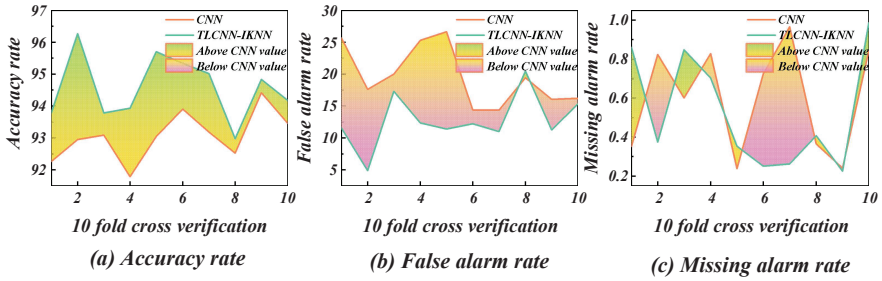

Figure 6 Crossover analysis of convolutional neural networks.

the experimental results largely aligning with theoretical expectations. The perceptron model tends to focus on the features of individual nodes while neglecting topological structure features, and its parameters lack constraints, which can easily lead to overfitting. In this experiment, increasing the training batch to 500 rounds resulted in a substantial decrease in the perceptron's accuracy, and its convergence rate was slower than that of the convolutional model's perceptron. This is because more parameters need to be updated for each backpropagation, and it relies heavily on the optimizer to jump out of the local optimal value, which will fall into the local optimal value when the perceptron adopts stochastic gradient optimization.

The basic idea of the convolution model is to combine the attribute features of nodes with the structural features of nodes so that the global features of nodes can be captured. In the convolution model, the convolution kernel shares parameters, and the convolution model converges faster when the model parameters are the same. After the Chebyshev convolution kernel is adopted, it is unnecessary to calculate the feature decomposition of the Laplace matrix. The operation efficiency is also greatly improved.

As is evident from Figure 6, in terms of intrusion detection accuracy, the optimized convolutional neural network algorithm demonstrates an effective improvement over traditional neural network models. Post-optimization, there is a slight reduction in the algorithm's false positive rate, which is now on par with that of the algorithm.

In Figure 7, a comparison is made between different classes of convolutional neural networks and the transfer learning-based convolutional neural networks presented in this paper. The intrusion detection algorithm proposed in this paper exhibits clear advantages in terms of accuracy, low false positive rate, and false negative rate, surpassing the performance of traditional convolutional neural networks. The performance of the optimized

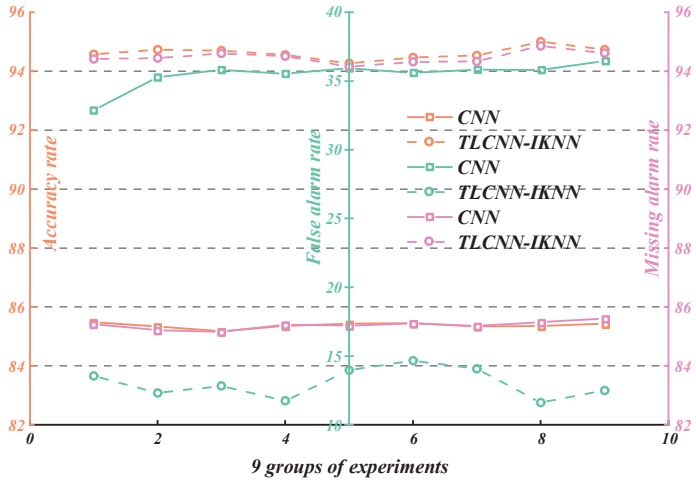

Figure 7 Multi-group experimental analysis of convolutional neural networks.

convolutional neural network is evaluated from various perspectives. It should be pointed out that because different experimental environments and different parameter Settings have an impact on the effect of neural network algorithms, there is no fixed norm for the parameter Settings of neural network methods at present. Therefore, in different experiments, the results of the same algorithm are also different. In this experiment, multiple comparison experiments should be conducted as far as possible to pay attention to the average value of algorithms. The results of various experiments show that the proposed idea of using transfer learning to optimize the convolutional neural network algorithm is effective and feasible, and can improve the detection accuracy rate. This optimized algorithm can be applied not only in intrusion detection but also in other fields.

5 Conclusion

To address the issues of slow convergence and getting trapped in local minima prevalent in current network intrusion detection algorithms, this paper introduces a convolutional neural network-based intrusion detection algorithm utilizing transfer learning, aimed at enhancing network robustness and completeness. The basic model 1D-CNN(T) is formed by training one-dimensional convolutional neural networks on training data sets. Then, 1D-CNN(E) is further trained under the transfer learning framework by

constructing the objective function and using the experimental data set. Specific conclusions are as follows:

1. A convolutional neural network (TLCNN) based on transfer learning is constructed. The introduction of transfer learning effectively solves the problem of knowledge acquisition and training efficiency of feature extraction of high-dimensional complex data in a big data environment where CNN is used. The conjugate gradient descent algorithm is used to optimize the performance of the CNN network based on transfer learning, and the problem of low classification accuracy caused by unbalanced data categories in the KNN classification algorithm is also improved and solved. The efficiency of classification calculation and accuracy of classification detection are improved.
2. The transfer learning-based convolutional neural network optimization proposed in this paper demonstrates significant performance enhancements. Specifically, the algorithm achieves an accuracy of 83.58% and a recall rate of 84.49%. Compared to the RF algorithm, the proposed algorithm exhibits improvements of 8.87% in accuracy and 9% in recall rate. Moreover, when compared with the AlexNet model, the proposed algorithm achieves substantial increases of 6.56% in accuracy and 7.26% in the F1-Measure, affirming the superiority of the proposed algorithm in performance.
3. The intrusion detection algorithm presented in this paper has markedly improved the detection rate for U2R attacks, thereby validating the method's capability to address the issue of low detection rates due to data imbalance to a certain extent. Consequently, the proposed method mitigates the problem of imbalanced data categories in the implementation of intrusion detection systems.

The network security situation assessment method studied in this paper conducts situational analysis and evaluation of network vulnerabilities, malicious programs, attack events, etc., in the big data environment. However, it still lacks analysis and evaluation of network asset operational status and network security business management. Additionally, the designed network security situation assessment model and method based on the improved evidence theory, while meeting system requirements, still need further enhancement in computational efficiency. These areas have room for improvement. Establishing a unified and standardized evaluation index system for network security situations is more conducive to objectively, comprehensively, and accurately portraying network security situations in the big data environment,

facilitating the reference and integration of network security situation assessment and prediction models in the big data context. Therefore, researching unified metrics for network security situations in the big data environment remains a key focus for future studies.

References

[1] Gu Z, Nazir S, Hong C, et al. Convolution Neural Network-Based Higher Accurate Intrusion Identification System for the Network Security and Communication[J]. Security and Communication Networks, 2020, 2020(1): 8830903.

[2] Nguyen M T, Kim K. Genetic convolutional neural network for intrusion detection systems[J]. Future Generation Computer Systems, 2020, 113: 418–427.

[3] Xiao Y, Xing C, Zhang T, et al. An intrusion detection model based on feature reduction and convolutional neural networks[J]. IEEE Access, 2019, 7: 42210–42219.

[4] Kim J, Shin Y, Choi E. An intrusion detection model based on a convolutional neural network[J]. Journal of Multimedia Information System, 2019, 6(4): 165–172.

[5] Nicolosi L, Tetzlaff R, Abt F, et al. Cellular Neural Network (CNN) based control algorithms for omnidirectional laser welding processes: Experimental results[C]//2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010). IEEE, 2010: 1–6.

[6] Li Z, Liu F, Yang W, et al. A survey of convolutional neural networks: analysis, applications, and prospects[J]. IEEE transactions on neural networks and learning systems, 2021, 33(12): 6999–7019.

[7] Orfila A, Carbó J, Ribagorda A. Autonomous decision on intrusion detection with trained BDI agents[J]. Computer Communications, 2008, 31(9): 1803–1813.

[8] Wu P, Guo H. LuNET: a deep neural network for network intrusion detection[C]//2019 IEEE symposium series on computational intelligence (SSCI). IEEE, 2019: 617–624.

[9] Yadav P, Menon N, Ravi V, et al. EfficientNet convolutional neural networks-based Android malware detection[J]. Computers & Security, 2022, 115: 102622.

[10] Ahmad S, Mehfuz S, Beg J. An efficient and secure key management with the extended convolutional neural network for intrusion

detection in cloud storage[J]. Concurrency and Computation: Practice and Experience, 2023, 35(23): e7806.

[11] Baluta T, Shen S, Shinde S, et al. Quantitative verification of neural networks and its security applications[C]//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. 2019: 1249–1264.

[12] Nie L, Ning Z, Wang X, et al. Data-driven intrusion detection for intelligent internet of vehicles: A deep convolutional neural network-based method[J]. IEEE Transactions on Network Science and Engineering, 2020, 7(4): 2219–2230.

[13] Lin G, Wen S, Han Q L, et al. Software vulnerability detection using deep neural networks: a survey[J]. Proceedings of the IEEE, 2020, 108(10): 1825–1848.

[14] Boopathy A, Weng T W, Chen P Y, et al. Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 3240–3247.

[15] Kan X, Fan Y, Fang Z, et al. A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network[J]. Information Sciences, 2021, 568: 147–162.

[16] Song H, Montenegro-Marin C E. Secure prediction and assessment of sports injuries using deep learning based convolutional neural network[J]. Journal of Ambient Intelligence and Humanized Computing, 2021, 12(3): 3399–3410.

[17] George A, Marcel S. Learning one class representations for face presentation attack detection using multi-channel convolutional neural networks[J]. IEEE Transactions on Information Forensics and Security, 2020, 16: 361–375.

[18] Drewek-Ossowicka A, Pietrołaj M, Rumiński J. A survey of neural networks usage for intrusion detection systems[J]. Journal of Ambient Intelligence and Humanized Computing, 2021, 12(1): 497–514.

[19] Derhab A, Aldweesh A, Emam A Z, et al. Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering[J]. Wireless Communications and Mobile Computing, 2020, 2020(1): 6689134.

[20] Lee J, Kim J, Kim I, et al. Cyber threat detection based on artificial neural networks using event profiles[J]. Ieee Access, 2019, 7: 165607–165626.

[21] Wu Y, Wei D, Feng J. Network attacks detection methods based on deep learning techniques: a survey[J]. Security and Communication Networks, 2020, 2020(1): 8872923.
[22] Chouhan N, Khan A. Network anomaly detection using channel boosted and residual learning based deep convolutional neural network[J]. Applied Soft Computing, 2019, 83: 105612.
[23] Sharma K, Aggarwal A, Singhania T, et al. Hiding data in images using cryptography and deep neural network[J]. arXiv preprint arXiv:1912.10413, 2019.

Biographies

Huizhong Zhang received the Master's Degree in Education from Northeast Normal University, Changchun, China, in 2007. He is currently an associate research fellow at Zhejiang College of Security Technology, China. His research interests include Development Strategy of Higher vocational colleges, computer educational technology, and computer-aided education.

Fanrong Meng received her Master's Degree in Human Geography from Northeast Normal University, Changchun, China, in 2003. In September 2003, she joined Wenzhou University, China. Her research interests include Enterprise Resource Economy and Strategic Management, and computer-aided education.

Qinyong Wang received an MS degree in plasma physics from Donghua University in 2010. He has published multiple research articles as first author and co-author in several internationally peer-reviewed journals. He has obtained several patents and led multiple research projects. His main research interests include business intelligence and recommendation systems.