Research Article

CNID: Research of Network Intrusion Detection Based on Convolutional Neural Network

Guojie Liu^{1,2} and Jianbiao Zhang^{1,2}

¹Beijing University of Technology, Beijing 100124, China

²Beijing Key Laboratory of Trusted Computing, Beijing 100124, China

Correspondence should be addressed to Jianbiao Zhang; zjb@bjut.edu.cn

Received 20 December 2019; Revised 5 April 2020; Accepted 29 April 2020; Published 21 May 2020

Academic Editor: Luca Pancioni

Copyright © 2020 Guojie Liu and Jianbiao Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network intrusion detection system can effectively detect network attack behaviour, which is very important to network security. In this paper, a multiclassification network intrusion detection model based on convolutional neural network is proposed, and the algorithm is optimized. First, the data is preprocessed, the original one-dimensional network intrusion data is converted into two-dimensional data, and then the effective features are learned using optimized convolutional neural networks, and, finally, the final test results are produced in conjunction with the Softmax classifier. In this paper, KDD-CUP 99 and NSL-KDD standard network intrusion detection dataset were used to carry out the multiclassification network intrusion detection experiment; the experimental results show that the multiclassification network intrusion detection model proposed in this paper improves the accuracy and check rate, reduces the false positive rate, and also obtains better test results for the detection of unknown attacks.

1. Introduction

Network security is one of the most important security issues facing cloud computing, with frequent cyber attacks and cyber intrusions, such as a DDoS attack by a botnet controlled by the malware Mirai in October 2016 which caused widespread outages on the East Coast of the United States. The ransomware software WannaCry, which broke out in May 2017, exploited system vulnerabilities to poison the computers of hundreds of thousands of users in several countries around the world. In China, the annual losses caused by digital crimes such as pseudo-base-stations and malware extortion amount to tens of billions of yuan. The above examples show that network security not only affects the development of national economy but also affects social stability and national security [1].

Deep learning for network intrusion detection is one of the hot spots in recent academic research. With the enhancement of hardware computing power and the rapid

growth of data volume, the development of deep learning has been promoted, so that the practicality and popularity of deep learning have greatly improved [2]. Deep learning is a machine learning technique designed to enable artificial intelligence through experience and data to improve computer systems. Deep learning uses multiple nonlinear feature transformations, that is, processing layers formed by multilayer perception mechanisms, to characterize data learning [3]. Deep learning has been applied to computer vision [4], speech recognition [5], natural language processing [6], biomedicine [7], and malicious code detection [8], as well as many other fields. Since 2015, the research applied to deep learning in network security has gradually emerged, which has attracted wide attention from the academic circles. At present, deep learning is mainly used in the two major areas of network security for malware detection and network intrusion detection, and, compared with traditional machine learning, deep learning improves detection efficiency and reduces false positives. In addition, deep learning algorithms

get rid of the reliance on feature engineering and are able to intelligently identify attack features, helping to identify potential security threats.

Convolutional neural network algorithm (CNN) [9] is an effective algorithm of deep learning; convolutional neural network is designed to process multidimensional array data, and its greatest advantage is to be able to accurately extract the local correlation of features and improve the accuracy of feature extraction. Using convolutional neural network algorithm, combined with mainstream deep learning technology such as Dropout and ADAM and Softmax classifiers, this paper proposes a multiclassification network intrusion detection model based on convolutional neural network and implements the code based on TensorFlow. Finally, the model established in this paper is applied to the standard network intrusion detection dataset such as KDD-CUP 99 and NSL-KDD [10].

The main contributions of this article are as follows:

- (i) A multiclass network intrusion detection model based on convolutional neural networks is proposed. This model can automatically and intelligently learn and identify attack features, which is helpful to find potential security threats.
- (ii) Multiclass network intrusion detection experiments were performed using KDD-CUP 99 and NSL-KDD standard network intrusion detection datasets. The experimental results show that the network intrusion detection model proposed in this paper improves the accuracy and recall and reduces the false positive rate. The detection of unknown attacks has also achieved better detection results.
- (iii) Compared with the common deep learning models such as DNN, LSTM-RNN, GRU-RNN, and DBN, the experimental results show that the network intrusion detection model proposed in this paper has higher accuracy and check rate and lower false positive rate.

The rest of the paper is arranged as follows: Section 2 describes the relevant work, Section 3 introduces the proposed network intrusion detection model, Section 4 discusses the experiments and results, and Section 5 summarizes the paper.

2. Related Works

Network intrusion detection is one of the important security defence means to protect computer systems and networks. Deep learning for network intrusion detection is a hot topic of recent academic research, and many literatures have proposed the successful application of deep learning technology in solving network intrusion detection problems [11, 12]. At present, the experimental results of network intrusion detection using deep learning are mostly distinguished between normal and attack, and there is no distinction between the types of attack. The next focus is on several commonly used deep learning models for multiclassification network intrusion detection: deep neural

networks, recursive neural networks, and deep belief networks.

Network intrusion detection is one of the important security defence means to protect computer systems and networks. Deep learning for network intrusion detection is a hot topic of recent academic research, and many literatures have proposed the successful application of deep learning technology in solving network intrusion detection problems [11, 12]. At present, the experimental results of network intrusion detection using deep learning are mostly distinguished between normal and attack, and there is no distinction between the types of attack. The next focus is on several commonly used deep learning models for multiclassification network intrusion detection: deep neural networks, recursive neural networks, and deep belief networks.

2.1. Deep Neural Networks. Deep neural network (DNN) [13] is a neural network model of deep structure, which is widely used in the field of network intrusion detection. Deep neural networks typically consist of input layers, multiple hidden layers, and output layers, as shown in Figure 1. Kim et al. [14] proposed refined data for the KDD-CUP 99 dataset using a deep neural network model (DR = 99%, FAR = 0.08%). As a method for network attack detection, an accelerated deep neural network model is used together with AEs and Softmax layers for fine-tuning of supervised learning [15]. Evaluate their accelerated deep neural network models using the NSL-KDD dataset, where DR is 97.5% and FAR is 3.5%.

2.2. Recurrent Neural Networks. Recursive neural network (RNN) is another deep structural model widely used in network traffic anomaly detection in recent years. Recursive neural networks mainly include LSTM-RNN [16] and GRU-RNN [17]. Figure 2(a) shows the structure of the LSTM-RNN storage unit. Figure 2(b) shows the structure of the GRU-RNN unit. Ponkarthika and Saraswathy [18] explored the network intrusion detection system based on the LSTM-RNN architecture model. They trained and tested their models on the KDD-CUP 99 dataset with an accuracy of 83%. Kim et al. [19] introduced a long-term-short-term memory recursive neural network (LSTM-RNN) classifier for network intrusion detection on the KDD-CUP 99 dataset, with DR being 98.88% and FAR being 10.04%. Yin et al. [20] proposed a network intrusion detection system based on recursive neural network and applied it to the NSL-KDD dataset (DR = 72.95% percent, FAR = 3.44%). In Kim et al.' [21] study, an integrated method based on LSTM-RNN was proposed, and an ADFA dataset was evaluated, resulting in DR being 90% and FAR being 16%.

2.3. Deep Belief Networks. Deep belief network (DBN) [22] is a layered structure of layer-to-layer restricted Boltzmann machine (RBM). As a well-known deep learning model, it has been widely used in network intrusion detection tasks. Figure 3 describes the typical structure of a DBN. Fiore et al.

FIGURE 1: Deep neural networks.

FIGURE 2: Recurrent neural networks. (a) LSTM-RNN. (b) GRU-RNN.

FIGURE 3: Deep belief networks.

[23] used an RBM-based discriminant model to detect anomalies on 10% of the KDD-CUP 99 dataset. Gao et al. [24] proposed a DBN-based network intrusion detection model and performed experiments on the KDD-CUP 99 dataset (DR = 92.33%, FAR = 0.76%). Alom et al. [25] explored the DBN model, 40% of NSL-KDD ability to detect the abnormal data sets, to obtain a 97.5% detection accuracy. In Liu and Zhang's research [26], the extreme learning machine (ELM) was applied to the learning process of the DBN model and then evaluated using the NSL-KDD dataset (DR = 91.8%). Alrawashdeh and Purdy [27] proposed based on RBM and DBN deep learning method for of KDD-CUP 99 in 10% of the intrusion detection system abnormality, where the DR is 97.9% and FAR is 2.47%.

The comparison of the detection results of the above three deep models is shown in Table 1, which is helpful for researchers to compare the detection results of different deep learning models. We can see from the table that, using the same method, the detection results of the KDD-99 dataset are better than those of the NSL-KDD dataset. This is because the KDD-99 dataset contains a large number of identical data records, while the NSL-KDD dataset removes a large number of duplicate records.

Although the above studies have improved the recognition ability and performance of network intrusion detection samples, there are shortcomings such as overfitting and poor generalization ability in network training, and the detection accuracy and detection efficiency need to be improved. In order to avoid network trained to be merged to enhance the generalization ability, we use convolution neural network combined with the structural characteristics of cross-layer aggregation design concept proposed based on convolutional neural network of multiclassification network intrusion detection model.

3. The Proposed Model

The functional composition of the network intrusion detection model based on convolutional neural network is shown in Figure 4, which is composed of three functional modules: the data preprocessing module, the feature self-learning module, and the classifier module. Based on convolutional neural networks, the model is trained by preprocessed original sample datasets and optimized by circular feature extraction and iteration, so that the model can achieve good convergence effect.

3.1. Convolutional Neural Networks. Compared with other machine learning methods, network intrusion detection methods using convolutional neural networks significantly improve the accuracy of classification. As a semisupervised neural network, convolutional neural networks have the ability to abstractly represent low-level intrusion traffic data features as high-level features and outstanding feature learning capabilities, so they have been gradually applied to the field of network intrusion detection in recent years.

Convolutional neural networks are neural networks that use convolution operations in place of ordinary matrix multiplication operations in at least one layer of the network [28], as shown in Figure 5. Convolution is a special linear operation, such as image recognition tasks; each convolution corresponds to the different characteristics of the image; the network's lower-level convolution tends to learn the simple properties of the image, including the edge of the space frequency and colour [29].

The proposed convolutional neural network effectively solves the problem of the explosion of neural network parameters and also ensures the accuracy of classification. The three important core concepts in convolutional neural networks are local perception, parameter sharing, and pooling. Local perception means that neurons in the hidden

TABLE 1: Summary of test results for different depth models.

Deep learning model	Reference	Dataset	Results		
			Accuracy	DR	Far
DNN	[14]	KDD-99	—	99.00%	0.08%
	[15]	NSL-KDD	—	97.50%	3.50%
LSTM-RNN	[18]	KDD-99	83.00%	—	—
	[19]	KDD-99	—	98.88%	10.04%
	[20]	NSL-KDD	—	72.95%	3.44%
	[21]	ADFA	—	90.00%	16.00%
DBN	[24]	KDD-99	—	92.33%	0.76%
	[25]	NSL-KDD	97.50%	—	—
	[26]	NSL-KDD	—	91.80%	—
	[27]	KDD-99	—	97.90%	2.47%

layer do not need to be connected to all the input pixels, and different hidden layer neurons need only to be connected to a specific area of the input pixel. In convolutional neural networks, local perception is realized by convolutional computations of the convolutional layers, which are realized on input data by convolution nucleus.

3.2. Data Preprocessing. The data preprocessing module characterizes the data, including the numericalization of text features and the standardization of numerical features, and the original intrusion data is usually one-dimensional vector data, which needs to be converted into two-dimensional data similar to the image, so that the convolutional neural network can process it. Using a data-based transformation algorithm, based on retaining all the information of the original sample, the sample is extended with features and normal data is used to fill the extended features, which is to expand the original data sample, thus preserving all the useful information in the original data sample. The expanded features increase the information capacity of the data sample, increase the distance between different categories of data in the sample space, and improve the accuracy of detection to a certain extent.

This data needs to be processed during the data preprocessing phase because each characteristic value of the intrusion detection data has a different range of values and is very different. In this paper, the numerical characteristics of the intrusion data are standardized by using the mainstream z -score standardized method, as shown in formula.

$$x'_i = \frac{x_i - \bar{x}}{\nu}. \quad (1)$$

In the formula, $\bar{x} = (1/n) \sum_{i=1}^n x_i$, $\nu = \sqrt{(1/n-1) \sum_{i=1}^n (x_i - \bar{x})^2}$, n is the total number of samples, x_i is the characteristic value of a dimension of the sample data before standardization, and x'_i is the characteristic value of the dimension corresponding to the sample data after standardization.

FIGURE 4: CNN-based network intrusion detection architecture.

FIGURE 5: Convolutional neural networks.

3.3. Feature Self-Learning. The main function of the feature self-learning module is to use convolutional neural networks to automatically learn and extract useful features from the original data samples and to learn, map, and generate new features from the original data samples. Lecun et al. [30] systematically expounded convolutional neural networks. The basic structure of the feature self-learning module based on convolutional neural network designed in this paper is shown in Figure 6. The deep learning technology used in the feature self-learning module mainly includes convolution operations and pooling operations, dropout, activation functions, and ADAM optimization algorithms [31].

Convolutional neural network effectively solves the problem of neural network parameter explosion. The three important core concepts of convolutional neural networks are local perception, parameter sharing, and pooling. The local perception of convolutional neural networks is realized through convolutional operations. Convolution is as shown in formula (2) [3]. In the formula, s is the output data also called feature map, x is the input sample data, w is the weight value of the kernel function, b is the offset value, and f is the activation function.

$$s = f(x \times w + b). \quad (2)$$

Convolutional neural networks introduce parameter sharing to further reduce the parameters of the neural network, the essence is that all hidden neurons share a set of weight parameters and bias parameters, and the statistical characteristics based on different parts of the image are usually the same [3]. A set of weight parameters and bias parameters generate a feature map, and the representation capability of a feature map is limited. Therefore, in practical applications, a convolution layer will generate multiple feature maps. The pooling process is mainly to reduce the dimensions of features. The pooling operation generally calculates the average or maximum value of multiple features in a local area. Therefore, the pooling operation in

convolutional neural networks is divided into maximum pooling and average pooling. The model proposed in this paper uses the average pooling operation.

The common activation functions of convolutional neural networks are sigmoid, tanh, ReLU [32], and so forth, where tanh is also known as the double-curving function, and the tanh function will have a good effect when the characteristics differ significantly and will expand the feature effect in the course of the cycle. Therefore, tanh is used as the activation function of the convolutional neural network.

The common method of preventing overfitting include regularization, early stopping, increasing the sample size, dropout, and batch normalization. This paper uses the method of inserting a dropout layer between the feature self-learning module and the classifier to prevent overfitting. The implementation process of dropout is as follows: during model training, some neurons in the neural network are randomly dropped according to the probability p ; and, during the test phase, all neurons are online, which can be mitigated by preventing the synergy of certain features overfitting [33, 34]. Using dropout later, each subnetwork is trained neural network of the original, thus, for containing n neural network hidden nodes, $2n$ models can be obtained. When making predictions, the prediction results of all sub-models are averaged to improve the model's capacity and generalization ability. Srivastava et al. [33] pointed out that when $p = 0.05$, Dropout has the best effect, and the network structure generated at this time is the richest.

The ADAM algorithm has been the most widely used first-order optimization algorithm in the field of deep learning in recent years. Kingma and Ba [31] pointed out that the ADAM algorithm includes the advantages of both adaptive gradient algorithms and root mean square propagation algorithms and has designed different adaptive learning rates for different parameters, so it can converge faster. Network intrusion detection data usually has problems of noise and sparseness. Therefore, this paper chooses

FIGURE 6: Feature self-learning module.

the ADAM algorithm as the optimization algorithm of the convolutional neural network model.

3.4. Classifier Module. The classifier module gives the final test results based on the characteristics learned by the self-learning module. This article uses the Softmax classifier as a classification module for convolutional neural networks. The Softmax classifier is shown in formula (3). j in the formula is the j weight vector, and $x^{(i)}$ is the i data sample.

$$y_j = \frac{e^{\theta_j x^{(i)}}}{\sum_k \theta_k x^{(i)}}. \quad (3)$$

Common loss functions are mean square error (MSE) and cross entropy error (CEE). The equal square error loss function is mostly used for linear regression and is suitable for predicting values, that is, regression problem model. The cross-entropy error loss function is mostly used for logical regression and is suitable for prediction probability, that is, classification problem. Therefore, the cross-entropy error loss function is used as the loss function of the convolutional neural network model.

4. Experiments and Evaluation

4.1. Experiment Setting. The computer configuration used in the experiment in this paper is as follows: CPU i7-3920XM, 32 Gb of memory, 1 Tb SSD, installed Ubuntu 16.04 operating system with Docker 19.03.5 container virtualization environment, using TensorFlow 1.12.0 as a deep learning framework and Python 3.7 as the programming language.

This paper conducts multiple types of network intrusion classification tests, in which each dataset has a normal (negative) and a mixture of various attack (positive) samples. As shown in Table 2, the number of classes marked in each dataset is different. Therefore, when each model is applied to a specific dataset, a multiclass combined matrix is created to visualize the performance of the model [35]. This confusion

TABLE 2: KDD-CUP 99 data details.

Data type	Training set	Test set
Normal	97278	60593
Attack	DoS	223298
	Probe	2377
	R2L	5993
	U2R	39
Total	494021	292300

matrix maintains information about actual and predictive classes. Four main results can be extracted from the confusion matrix, namely, true positive (TPs), true negative (TNs), false positive (FPs), and false negative (FNs).

Unlike the two classification schemes, these four results have slightly different meanings in multiclass classification tasks. First, TN is the correct predictor of a normal sample. FP can be calculated by formula (4), where N is the number of attack classes and FP_i is misclassified as the normal number of samples of the i attack class. TP is the sum of all attack samples that are actually marked as their appropriate attack category using formula (5), where TP_i is the exact predictor of the i attack category. Finally, FN is the sum of all attack samples that are misclassified into normal classes. FN can be calculated according to formula (6), where FN_i is the number of samples of the attack class misjudged as normal; that is,

$$FP = \sum_{i=1}^N FP_i, \quad (4)$$

$$TP = \sum_{i=1}^N TP_i, \quad (5)$$

$$FN = \sum_{i=1}^N FN_i. \quad (6)$$

These four results are then used to calculate five evaluation indicators, allowing us to evaluate the performance of the model on the dataset. In order to adapt to the terminology definition of the multiclass NIDS system described earlier, some equations have been adjusted. The evaluation indicator definition used and its corresponding equation are shown below.

- (i) Accuracy shows the true prediction rate for all test sets; that is,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (7)$$

- (i) Precision is the accuracy of the classifier, that is, the rate at which the attack is correctly marked from all samples classified as an attack from the test set; that is,

$$\text{Precision} = \frac{TP}{TP + FP}. \quad (8)$$

- (i) Recall is the integrity of the classifier, that is, the correct labelled attack rate for all attack samples in the test set. It is also called true positive rate (TPR), detection rate (DR), or sensitivity; that is,

$$\text{Recall} = \frac{TP}{TP + FN}. \quad (9)$$

- (i) F-Score can be viewed as the harmonic mean of the precision (P) and recall (R) indicators; that is,

$$F - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (10)$$

- (i) The error alert rate (FAR) shows that all normal samples in the test set are misclassified as normal sample rates for any attack category. It is also known as false positive rate (FPR); that is,

$$\text{FAR} = \frac{FP}{FP + TN}. \quad (11)$$

4.2. Experiment Datasets. This paper uses the commonly used network intrusion datasets KDD-CUP 99 and NSL-KDD as experimental datasets, which can verify the effect of the network intrusion detection model proposed in this paper. KDD-CUP 99 and NSL-KDD are standard datasets in the field of network intrusion detection and are used by a large number of security research works [36]. In this paper, two experiments are designed for these two datasets.

4.2.1. KDD-CUP 99 Dataset. The KDD-CUP 99 dataset is widely used in the field of network intrusion detection and can be downloaded on the official website [37]. The complete dataset includes approximately 5 million records in the training set and approximately 2 million records in the test set. In fact, only 10% of the KDD-CUP 99 dataset is used for training and testing. There were 494,021 samples in the training data and 292,300 samples in the test data. Each sample is marked as normal or attack recorded. In 10% of the

dataset, there are 38 types of attacks. In order to evaluate the effectiveness of the test model testing new attacks that did not appear in the training set, only a sample of 24 types of attacks appeared in the training set. In addition, similar attacks are grouped into one category, forming four main attack categories, namely, DoS, Probe, R2L, and U2R. Details of the KDD-CUP 99 dataset are shown in Table 2.

4.2.2. NSL-KDD Dataset. Tavallae et al. [38] improved and simplified the 10% KDD-CUP 99 dataset in 2009 to form the NSL-KDD dataset. They solved the disadvantage of 10% KDD-CUP 99 in two ways. First, they removed all the extra records from the training and test ingress. Second, they divided the records into different difficulty levels and then selected records from each difficulty level which were inversely proportional to the 10% record percentage in the original KDD-CUP 99 dataset. As a result, NSL-KDD has a reasonable number of records in the training and test set, enabling it to run experiments on a complete set. Although it is no longer a good representation of the real network, it is still considered a benchmark and is widely used in network intrusion detection research. In addition, the NSL-KDD dataset is public on the Internet [39]. Each record in the NSL-KDD dataset consists of 41 characteristics that represent a network connection. The data in the dataset is marked as normal and attacked, and the attack types are divided into four broad categories, with a total of 39 attack types. Twenty-two attacks appear in training and test sets, and 17 attacks appear only in test sets. Details of the NSL-KDD dataset are shown in Table 3.

4.3. Experiment Results

4.3.1. KDD-CUP 99 Experiment. In the data preprocessing stage, the three text features in the dataset are first digitized, each text feature is converted to a corresponding integer value, and then the data sample of 41 features is expanded to 42 dimensions, and the extended feature is transformed using a data transformation algorithm. Fill it and convert it into two-dimensional data. Take 75% of the training data as the training set and 25% as the validation set. A total of 30 iterative trainings were performed. After the training was completed, the test set was used for testing. The experimental results are shown in Table 4. The first 4 columns of the data in Table 4 are the average of the experimental results of the pretrained (w/) and nonpretrained (w/o) stages in [40]; the 5th and 6th columns of the data are the experimental results in [41, 42]. Figure 7 can intuitively compare the evaluation indexes of all models. Experimental results show that the model proposed in this paper obtains 98.02% accuracy and 0.02% false positive rate and has good generalization ability and also has good detection ability for unknown attack types. The accuracy of the detection results is higher than the best detection result in the literature [40–42], 95.00%, and the false alarm rate is lower than the best detection result in the literature [40–42], 0.97%. Figure 8(a) intuitively describes the detection rate of each type in KDD-CUP 99. It can be seen from Figure 8(a) that if the amount of data of the attack type

TABLE 3: NSL-KDD data details.

Data type		Training set	Test set
Normal		67343	9711
Attack	DoS	45927	7458
	Probe	11656	2421
	R2L	995	2887
	U2R	52	67
Total		125973	22544

TABLE 4: KDD-CUP 99 experiment result.

Metric	DNN [40]	LSTM-RNN [40]	GRU-RNN [40]	DBN [40]	KNN [41]	CNNA [42]	CNID (this paper) (%)
Accuracy	91.43	93.28	92.41	95.00	94.35%	92.18%	98.02
Precision	97.60	97.55	97.29	97.42	93.55%	90.95%	99.98
Recall	91.43	93.87	93.02	96.24	—	—	99.81
F1-Score	94.41	95.68	95.10	96.82	—	—	99.89
FAR	8.61	6.99	7.92	3.24	2.34%	0.97%	0.02

FIGURE 7: KDD-CUP 99 experiment result.

FIGURE 8: Detection rate for each class in the dataset. (a) KDD-CUP 99. (b) NSL-KDD.

TABLE 5: NSL-KDD experiment result.

Metric	DNN [40] (%)	LSTM-RNN [40] (%)	GRU-RNN [40] (%)	DBN [40] (%)	ICNN [43]	CNID (this paper) (%)
Accuracy	85.74	90.39	89.58	92.66	91.79%	97.09
Precision	96.73	97.52	97.02	97.43	93.65%	99.98
Recall	77.19	83.70	82.95	89.56	—	97.14
F1-Score	86.05	90.99	89.79	93.33	—	98.49
FAR	2.75	2.03	2.58	1.74	2.32%	0.87

FIGURE 9: NSL-KDD experiment result.

training set is large, the detection rate is correspondingly higher. In the experiment, it is also found that the attack type data unknown to the training set will also be correctly classified. For example, the upper class of mscan is PROBE, which only appears in the test set. It is an unknown threat to the training set and will be recognized as PROBE in the experiment.

4.3.2. KDD-NSL Experiment. The NSL-KDD dataset was processed using the same processing method as KDD-CUP 99, and a total of 30 iterative trainings were performed. The test results are shown in Table 5. The first 4 columns of the data in Table 5 are the average of the experimental results of the pretraining stage (w/) and the nonpretraining stage (w/o) in [40]; the fifth column of the data represents the experimental results of [43]. Figure 9 can intuitively compare the various evaluation indexes of all models. From the experimental results in Table 5, it can be seen that using the network intrusion detection model proposed in this paper has an accuracy rate of 97.09% and a false alarm rate of 0.87%. The accuracy rate in the detection results is higher than the best detection result in the literature [40, 43], 92.66%, and the false alarm rate is lower than the best detection result in the literature [40, 43], 1.74%. Figure 8(b) intuitively describes the detection rate of each type in NSL-KDD. Similar to the KDD-CUP 99 data and the experimental results, it can be seen from Figure 8(b) that if the amount of data in the attack type training set is large, the detection rate is correspondingly higher. In the experiment, it is also found that the attack type data unknown to the training set will also be correctly classified.

4.3.3. Comparison with Other Related Works. The effectiveness of the model for detecting network intrusion depends on the reasonable setting of its evaluation indicators. The higher the accuracy, accuracy, recall, and F-Score, the lower the FAR value, indicating that the classifier is effective. The accuracy and recall of an ideal classifier reach 1, and the FAR value reaches 0. The experimental results on the KDD-CUP 99 dataset and NSL-KDD dataset are compared with the four deep learning models in the latest literature [40–42] on the same datasets. On KDD-CUP 99 data, the accuracy rate of 98.02% in this paper is better than the accuracy rate of 95.00% in the literature [40–42]. On NSL-KDD data, the accuracy rate of 97.09% in this article is better than the accuracy rate of 92.66% in the literature [40, 43]. Literature [44] proposed a new network intrusion detection model using convolutional neural network (CNN), using CNN to automatically select traffic features from the original data set, and set the cost function weight coefficient according to the number of categories to solve the problem of balance. This model is used for large-scale network intrusion detection, using NSL-KDD dataset, and its accuracy is lower than the model proposed in this paper. Literature [45] proposed a network intrusion detection system based on the convolutional neural network model Lenet-5 and introduced OHE coding and normalization method to process the feature matrix; using KDD-CUP99 dataset, its accuracy is lower than this article's proposed model. Similar literatures [46–48] use convolutional neural networks for intrusion detection in different research areas. The method proposed in this paper has better generalization ability, good detection ability for unknown attack types, and good detection performance in distinguishing normal data and attack data, but

there is room for further improvement in distinguishing different attack types.

5. Conclusions

Network intrusion detection is very important in the field of network security. In recent years, although there has been a lot of research on network intrusion detection, there is very little in-depth research on this issue, especially in multiclass network intrusion detection. In this paper, a multiclass network intrusion detection model based on convolutional neural network is proposed, and the algorithm is optimized. It was tested on a computer configured with 32 Gb of memory, 1 Tb of solid-state drive, and Ubuntu 16.04 operating system and Docker 19.03.5 container virtualization environment. The experiment uses KDD-CUP99 dataset and NSL-KDD dataset and compares the experimental results with the deep learning models of DNN, LSTM-RNN, GRU-RNN, DBN, KNN, ICNN, and so on. The experimental results show that the network intrusion detection model proposed in this paper improves the accuracy and recall, reduces the false positive rate, and obtains better detection results for the detection of unknown attacks.

The accuracy of the model proposed in this paper in multiclass experiments needs to be improved; in particular, the classification results of unknown different attack types still have room for improvement, which needs to be explored in future work. The dataset used in the experiment in this paper is a dataset that has been manually processed and optimized. In the future work, the following datasets will be studied: the newly emerging dataset for network intrusion detection will extract the corresponding data from real network traffic features to verify the method proposed in this article.

Data Availability

The basic data used in this article was downloaded from the Internet. There are two-part datasets: The KDD-CUP 99 is a public dataset that can be downloaded from <http://kdd.ics.uci.edu/databases/kddcup99/>. The NSL-KDD is a public dataset that can be downloaded from https://github.com/defcom17/NSL_KDD.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was sponsored by the Opening-up Project of National Defense Science and Technology Laboratory of Information Security (No. 2015XXAQ08).

References

- [1] J.-Z. Luo, M. Yang, L. Zhen et al., "Architecture and key technologies of cyberspace security," *SCIENTIA SINICA Information*, vol. 46, no. 8, pp. 939–968, 2016.
- [2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.
- [4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 2, pp. 2012–2025, 2012.
- [5] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.
- [6] X. F. Xi and G. D. Ghou, "A survey on deep learning for natural language processing," *Acta Automatica Sinica*, vol. 42, no. 10, pp. 1445–1465, 2016.
- [7] A. Cruz-Roa, J. E. A. Ovalle, A. Madabhushi et al., "A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection," in *Proceedings of the 9th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 403–410, Springer, Nagoya, Japan, September 2013.
- [8] W. Huang and J. W. Stokes, "MtNet: a multi-task neural network for dynamic malware classification," in *Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 399–418, Springer, Como, Italy, July 2016.
- [9] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: a comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.
- [10] M. Tavallaei, Bagherie, W. Lu et al., *NSL-KDD [Z/OL]*, University of New Brunswick, Fredericton, Canada, 2018, <http://www.unb.ca/cic/datasets/nsl.html>.
- [11] E. Aminanto and K. Kim, "Deep learning in intrusion detection system: an overview," in *Proceedings of the 2016 International Research Conference on Engineering and Technology (2016 IRCET)*, Higher Education Forum, Kuta, Indonesia, January 2016.
- [12] R. Vani, "Towards efficient intrusion detection using deep learning techniques: a review," *International Journal of Advanced Research in Computer Science and Electronics Engineering*, vol. 6, no. 10, pp. 375–384, 2017.
- [13] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [14] J. Kim, N. Shin, S. Y. Jo et al., "Method of intrusion detection using deep neural network," in *Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (Big Comp)*, Jeju, South Korea, February 2017.
- [15] S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced intrusion detection system," in *Proceedings of the 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, September 2016, Berlin, Germany.
- [16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [17] K. Cho, B. van Merriënboer, C. Gulcehre et al., "Learning phrase representations using rnn encoder–decoder for statistical machine translation," 2014, <https://arxiv.org/abs/1406.1078>.
- [18] M. Ponkarthika and V. R. Saraswathy, "Network intrusion detection using deep neural networks," *Asian Journal of Science and Technology*, vol. 2, no. 2, pp. 665–673, 2018.
- [19] J. Kim, J. Kim, H. L. T. Thu et al., "Long short term memory recurrent neural network classifier for intrusion detection," in

- Proceedings of the 2016 International Conference on Platform Technology and Service (Plat Con)*, Jeju, South Korea, January 2016.
- [20] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
 - [21] G. Kim, H. Yi, J. Lee et al., "LSTM-based system-call language modeling and robust ensemble method for designing host-based intrusion detection systems," 2016, <https://arxiv.org/abs/1611.01726>.
 - [22] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
 - [23] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted Boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.
 - [24] N. Gao, L. Gao, Q. Gao et al., "An intrusion detection model based on deep belief networks," in *Proceedings of the 2014 Second International Conference on Advanced Cloud and Big Data (CBD)*, Huangshan, China, November 2014.
 - [25] Z. Alom, V. R. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *Proceedings of the 2015 National Aerospace and Electronics Conference (NAECON)*, Dayton, OH, USA, June 2015.
 - [26] Y. Liu and X. Zhang, "Intrusion detection based on IDBM," in *Proceedings of the 2016 IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, 14th Intl Conference on Pervasive Intelligence and Computing, 2nd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, Auckland, New Zealand, 2016.
 - [27] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Anaheim, CA, USA, December 2016.
 - [28] L. Deng and Y. Dong, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2014.
 - [29] RSIP, *Deep Learning and Convolutional Neural Networks: RSIP Vision Blogs [EB/OL]*, RSIP, Jerusalem, Israel, 2016, <http://www.rsipvision.com/exploring-deep-learning/>.
 - [30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
 - [31] D. Kingma and J. Ba, "ADAM: a method for stochastic optimization," 2017, <https://arxiv.org/abs/1412.6980v9>.
 - [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing System*, Curran Associates, Lake Tahoe, CA, USA, pp. 1097–1105, December 2012.
 - [33] N. Srivastava, G. Hinton, A. Krizhevsky et al., "Dropout: a simple way to prevent neural networks from over fitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
 - [34] E. Hintong, N. Srivastava, A. Krizhevsky et al., "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.
 - [35] J. Swets, "Measuring the accuracy of diagnostic systems," *Science*, vol. 240, no. 4857, pp. 1285–1293, 1988.
 - [36] J. McHugh, "Testing Intrusion detection systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, pp. 262–294, 2000.
 - [37] University of California, *KDD CUP 1999 Data Set*, University of California, Irvine, CA, USA, 1999, <http://kdd.ics.uci.edu/databases/kddcup99/>.
 - [38] M. Tavallaei, E. Bagheri, W. Lu et al., "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, Canada, July 2009.
 - [39] NSL-KDD dataset, https://github.com/defcom17/NSL_KDD.
 - [40] W. Elmasry, A. Akbulut, and A. H. Zaim, "Empirical study on multiclass classification-based network intrusion detection," *Computational Intelligence*, vol. 35, no. 4, pp. 915–954, 2019.
 - [41] J. Ren, X. Liu, Q. Wang et al., "An multi-level intrusion detection method based on KNN outlier detection and random forests," *Journal of Computer Research and Development*, vol. 56, no. 3, pp. 566–575, 2019.
 - [42] Y. Liu, S. Liu, X. Zhao et al., "Intrusion detection algorithm based on convolutional neural network," *Transactions of Beijing Institute of Technology*, vol. 37, no. 12, pp. 1271–1275, 2018.
 - [43] H. Yang and F. Wang, "Network intrusion detection model based on improved convolutional neural network," *Journal of Computer Applications*, vol. 39, no. 9, pp. 2604–2610, 2019.
 - [44] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018.
 - [45] P. Liu, "An intrusion detection system based on convolutional neural network," in *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering ICCAE*, pp. 62–67, Perth, Australia, February 2019.
 - [46] H. Yang and F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," *IEEE Access*, vol. 7, pp. 64366–64374, 2019.
 - [47] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, Article ID 100198, 2020.
 - [48] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42210–42219, 2019.