Journal of Quantum Computing
DOI:10.32604/jqc.2021.016857
Article

Tech Science Press

A Hybrid Intrusion Detection Model Based on Spatiotemporal Features

Linbei Wang[1], Zaoyu Tao[1], Lina Wang[2,*] and Yongjun Ren[3]

[1]Changwang School of Honors, Nanjing University of Information Science and Technology, Nanjing, China
[2]School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, China
[3]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China
[*]Corresponding Author: Lina Wang. Email: wangllna@163.com
Received: 16 June 2021; Accepted: 30 August 2021

Abstract: With the accelerating process of social informatization, our personal information security and Internet sites, etc., have been facing a series of threats and challenges. Recently, well-developed neural network has seen great advancement in natural language processing and computer vision, which is also adopted in intrusion detection. In this research, a hybrid model integrating Multi-Scale Convolutional Neural Network and Long Short-term Memory Network (MSCNN-LSTM) is designed to conduct the intrusion detection. Multi-Scale Convolutional Neural Network (MSCNN) is used to extract the spatial characteristics of data sets. And Long Short-term Memory Network (LSTM) is responsible for processing the temporal characteristics. The data set used in this experiment is KDDCUP99 with different probability distributions in the training set and test set involving some newly emerging attack types, making the data more realistic. As a result, this type of data set is widely applied in the simulation experiment of intrusion detection. In this experiment, the assessment indices such as the accuracy rate, recall rate and *F1* score are introduced to check the performance of this model.

Keywords: Intrusion detection; deep learning; Multi-Scale Convolutional Neural Network; Long Short-Term Memory Network

1 Introduction

1.1 Research Background and Its Significance

Nowadays, both the high-speed information spreading and the resources sharing become a reality with the development of the Internet. Consequently, the problems of the security of computer system and network system are inevitably appearing. How to resolve this issue is a big challenge in the big data era.

There has been a sharp rise in the threats to the security of network, website and other organization due to the sheer scale of the data. To cope with it, various Intrusion Detection System (IDS) based on artificial intelligence or machine learning had been introduced for different types of network attacks. However, few of systems are capable of recognizing the different attack types and giving real time response. It's deduced that the single detection system cannot effectively monitor and process the intrusion embedded in the complicated data flow. The new way we initiate to resolve this issue is adopting deep learning technology into the intrusion detection as its excellent performance in the large size data analysis.

1.2 Current Research Status

Generally speaking, Network Intrusion Detection System (NIDS) can be divided into two types: feature-based detection and anomaly-based detection. The biggest problem with feature-based NIDS is that it cannot identify unknown attacks, which is obviously not suitable for today's complex and heterogeneous network environment. The network intrusion detection model based on machine learning (ML) or deep

learning (DL) can realize the classification and prediction of unknown network behaviors through learning the existing normal and abnormal network behaviors. At present, some researchers have successfully applied the ML/DL method into the field of network intrusion detection. CANN algorithm is proposed, six dimensional data features of KDDCUP99 data set are selected for training, and four attacks of PROB, DOS, R2L and U2R are tested. The intrusion detection algorithm has high accuracy. A hybrid intrusion detection algorithm based on bacterial foraging optimization algorithm (BFOA) was proposed to optimize the clustering instability in K-means algorithm. KDDCUP99 data set was used to verify the intrusion detection accuracy which is up to 98.33%. In addition, there is a so-called Second Training Intrusion Detection Model applying Decision Tree, naive Bayes and K-Nearest Neighbor algorithm (KNN) with better performance using KDDCUP99 data set. The intrusion detection algorithm based on Artificial Neural Network (ANN) is proposed and the DDoS/ DoS attack test on the model is carried out with highest accuracy of 99.4%. In addition, there is a deep learning hybrid model DBN-ELM, which combines Deep Belief Network (DBN) and Extreme Learning Machine (ELM). KDDCUP99 data set is selected as the original data set of the experiment, and the accuracy of this intrusion detection model can reach 97%. Compared with the traditional algorithms, Convolutional Neural Network (CNN) based intrusion detection algorithm has higher accuracy.

In recent years, it has become a trend to process big data based on deep learning technology. At present, Convolutional Neural Network (CNN) and Long Short-term Memory Network (LSTM) are widely used in data processing. CNN technology is mainly applied in the field of image processing for feature extraction with more accurate resolution than DBN. LSTM network is a Recursive Neural Networks (RNN), which can learn the dependencies between features better than other types of RNN. Another variant of the LSTM is referring to the Weight Decline of the Long Short-term Memory (WDLSTM) network. The regularized RNN uses descent join technique for recursive implicit weight matrix in LSTM to maintain the long-term correlation between extracted features and prevent over fitting of recursive joins. However, there are few research on intrusion detection applying the deep learning algorithm in the big data environment. Faker et al. [1] experimented with various deep learning technologies on a big data platform to improve the performance of intrusion detection system. Three classifiers are used to classify the attacks in second and multi-class patterns: (1) Deep Neural Network, (2) Random Forest, (3) Gradient Enhanced Tree. Lippmann et al. [2] proposed a Deep Neural Network (DNN) model to detect and classify unpredictable network attacks in intrusion detection systems.

At present, IDS has become one of the research hotspots in this field because of its reliability, scalability and self-learning ability. IDS models based on machine learning are the most mainstream research direction at present, such as Support Vector Machine (SVM), Bayesian Network and Deep Learning Neural Network.

Many researchers applied the traditional algorithms such as SVM into the field of network information security. Lippmann et al. [2] constructed a model to extract network flow using random walk clustering to detect P2P botnets. Ilgun [3] used graph clustering to detect DGA malware. Feng et al. [4] conducted the research and analysis on bot query network flow, and used the hierarchical clustering algorithm to detect whether there was aggressive behavior. Liu et al. [5] established the LS-SVM model and used the optimized SVM to classify the network flow.

In the field of deep learning, CNN and LSTM have been applied in natural language processing, computer vision, speech recognition and other fields [6–13]. From the perspective of their development, they have unique network structures. The application of the two to intrusion detection will further improve the accuracy of the detection system.

As shown in Fig. 1, MSCNN-LSTM model is adopted to extract the temporal and spatial characteristics of KDDCUP99 data set, so as to detect various attacks on the data set. In addition, a comparative experiment was established. In other words, the CNN-LSTM model was also used to detect the data set KDDCUP99, and the accuracy of the two models was compared to show the advantages of MSCNN-LSTM model. The main operational steps will be introduced as follows:

(1) The pre-processing of KDDCUP99 data set is conducted, namely processing the non-numeric data in the data set. The specific processing methods will be described in detail in the following part.

(2) MSCNN-LSTM model is established to detect the processed KDDCUP99 data set. CNN-LSTM model is performed in the comparative experiment.

(3) Two established models are trained, and the pictures reflecting accuracy and loss changes in the training set and test set during iteration are depicted.

(4) The accuracy, precision, recall rate and *F1* score derived from the two trained model tests are the indexes in the intrusion detection experiments as the criterion for testing the model.

(5) The above mentioned four indexes are compared in the two models, revealing the respective characteristics of two models with different advantages and disadvantages.

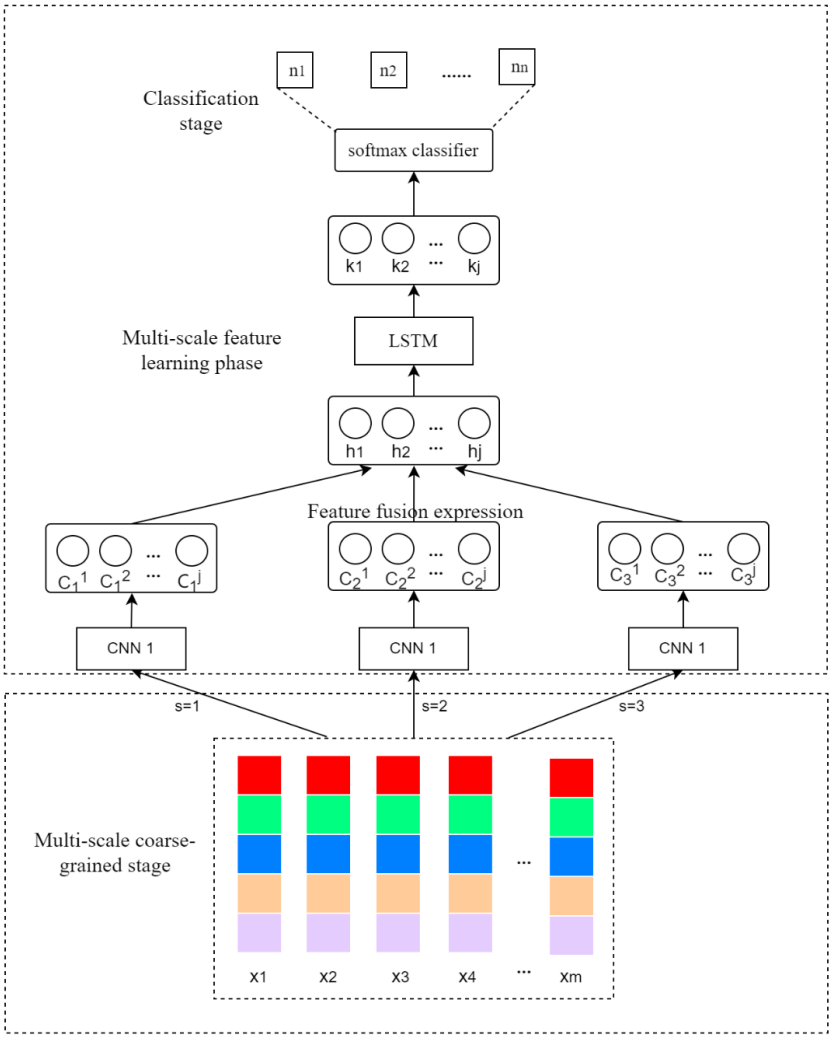

Figure 1: Learning process of MSCNN-LSTM model

2 Data Pre-Processing

2.1 Introduction to Data Sets

The KDDCUP99 data set used in the experiment is collected from an emulated local area network of the US Air Force. The reason for its wide application in the intrusion detection experiments is that the test

data and training data have different probability distributions and what is more prominent is there existed the attacks only in the test set, making the model experiment more realistic and simulating the operational status of model.

There are 41 attributes with fixed features, 9 of which are discrete/symbolic and others are continuous. KDDCUP99 set contains five types of attacks: DoS, Normal, Probe, R2L and U2R shown in Table 1.

Table 1: Identification of KDDCUP99 intrusion detection experimental data types

Identification type	Implication	Specific classification identification
Normal	Normal record	nNormal
Dos	Denial of service attack	back, land, neptune, pod, smurf, teardrop
Probe	Surveillance and other detection activities	ipsweep, nmap, portsweep,satan
R2L	Illegal access from remote machine	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster
U2R	Illegal access of ordinary users to local super user privileges	buffer_overflow, loadmodule, perl, root kit

KDDCUP99 data set consists of 5 million records, 10% training and test data are extracted providing the researchers the experimental data. 10% training subset and test subset is adopted in this experiment.

2.2 Data Pre-Processing

Numerical data is more applied to calculate the distance in data clustering method, characterized by fixed connection attribute. There are two types of recorded values: discrete attribute and continuous attribute. It is necessary for sample data to go through normalization process as the great differences among the attributes of the records in the experimental data sets, so as to prevent the decimal number from being rounded by large numbers. The preprocessing of data is divided into two steps, numerical standardization and numerical normalization respectively.

In the experiment, the two main data preprocessing techniques were adopted into KDDCUP99 set: data conversion and data normalization. Data conversion is to convert the characteristics of data flow into numerical format, ensuring all data digital. Data normalization is to reduce the variance of features to a certain value range. In addition, these null values were removed in the normalization process to prevent errors occurring during the training phase. In order to standardize large values and prevent the phenomenon "large numbers" eating "decimal numbers", the minimum-maximum scaling method was adopted to control the values fluctuating between 0 and 1, as is described according to the following equation:

$$f_{i,j} = \frac{f_{i,j} - min(f_{i,j})}{max(f_{i,j}) - min(f_{i,j})} \tag{1}$$

$f_{\mathrm{i,j}}$ represents the eigenvalue in the *i*th row and the *j*th column of the data set matrix.

The concrete structure of CNN is to process the transformed data sets in a centralized way by Softmax classifier and identify the most remarkable features of these four attacks. Five different attack states are in KDDCUP99 data set, including Dos, Probe, R2L, U2R and Normal. What we intend to do is to classify these five kinds of data.

The next step is to digitize the symbolic features of KDDCUP99 data. In this part, we use the method of attribute mapping. For example, attribute 2 is the protocol type, which has three values, namely tcp, udp and icmp. We transform them into binary values which will be accessed and recognized by computers.

3 Introduction to Model

3.1 Multi-Scale Convolutional Neural Network (MSCNN)

CNN is originally used to process images. However, in the image classification, more emphasis is placed on the extraction some important local features in images to complete the image classification, such as face recognition. For the network traffic, it will need to splice the features of several parts rather than single local feature. Taking the KDDCUP99 data set used in this experiment as an example, the data features consist of 41 attributes of 39 different types. Therefore, the final accuracy rate is bound to be very lower if the data classification is conducted using single local feature.

In view of the above problems existed in CNN, MSCNN replacing CNN in this experiment promotes the extraction of data sets' spatial feature. Compared with CNN, MSCNN extracts more features with different sizes from multiple convolution kernels and combines with global response to achieve accurate classification of data. So, MSCNN is more suitable for classification of network traffic than CNN.

The difference between MSCNN and CNN is that MSCNN model is similar to FCNT tracking method, getting the unused features of data sets based on different convolutional neural network layers. Taking the data set in this experiment as an example, there are three different types of protocol in KDDCUP99 data set. Different standards are set at different convolution layers to extract multi-layer features. Feature extraction is set for tcp in conv-3 and udp in conv-5, which is multi-scale extraction. According to the different convolution layers, we have different scale standards to conduct the multi-scale detection. Framework of the model as shown in Fig. 2.

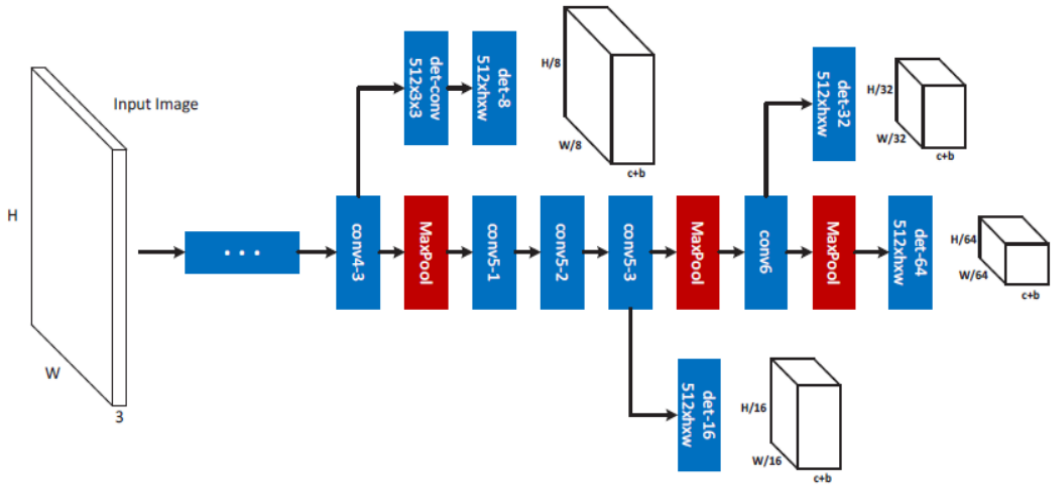

Figure 2: Framework of MSCNN model

In the model designed for this experiment, 3 multi-scale convolution layers (MS Convolution), 2 convolution layers, 1 average pooling layer and 3 full connection layers are used in the MSCNN part, finally connecting to softmax classifier. At the convolutional layer, BN algorithm is applied to speed up the learning rate of network. Relu function is selected as activation function at each layer. At the full connection layers, Dropout is designed to resist over-fitting and adopt the activation function sigmoid. In the CNN part embedded in the CNN-LSTM model, 3 convolutive layers, 1 maxpooling layer and 3 full connection layers are designed, connecting to softmax classifier.

The parameters of convolution layer in the MSCNN model are shown in Table 2.

Table 2: MSCNN model junction

Layer	Type	Convolution kernels	Step length	T fill	Activation function
L1	MSConv	1 * 1, 3 * 3, 5 * 5	1	same	ReLu
L2	Conv	3 * 3	1	same	ReLu

L3	MSConv	1 * 1, 3 * 3, 5 * 5	1	same	ReLu
L4	Conv	3 * 3	1	same	ReLu
L5	MSConv	1 * 1, 3 * 3, 5 * 5	1	same	ReLu
L6	Conv	3 * 3	1	same	ReLu
L7	Avepool	2	2	same	Sigmoid
L8	FC				Sigmoid+Drop

3.1.1 Pooling Layer

Pooling is a basic operation commonly used in convolutional neural networks. The main pooling methods are divided into two types: maximum pooling and average pooling. The purpose of designing pooling layer is to reduce parameter dimension and speed up network training. There is a large amount of overlap between the sliding windows, resulting in the excessive redundancy. Pooling operations will minimize the redundancy. As the pooling layer reduces the redundant message, the local information will be lost and the salient features will be retained. The reduction of redundant information will be beneficial to the prevention of over-fitting, making the model widely used.

In recent years, the mainstream classification model is maximum pooling, while the average pooling is rarely considered. The max-pooling is mainly used to select, classify and identify the features. In most models of neural network, the errors occurring in the feature extraction are originated from following reasons. One is the increasing variance of estimated value with the expanding neighborhood size, and the other one is the error of convolution layer parameters resulting in the deviation of the estimated mean. The average pooling can effectively reduce error occurring in the first case and retain more background characteristics information. In contrast, maximum pooling reduces the second error occurance and retains more texture feature information. Therefore, average pooling focuses on sub-sampling from the overall characteristic information, contributing to the complete transmission of information. In this research, the method of average pooling is to reduce dimension, and transmit the information to the next module for feature extraction.

3.1.2 Connection Layer

The full connection layer of convolutional neural network is equivalent to the feedforward neural network of the conventional hidden layer. The last layer is the fully connected part of the convolutional neural network's hidden layer.

The convolutional layer and pooling layer in the neural network play the role in extracting the targeted features from the input data. The function of the full connection layer is to make nonlinear combination of extracted features, so as to obtain the output results required through the experiment. The full connection layer tries to achieve the learning goal based on the existed higher-order features. In some convolution neural network, the function layer can be fully connected to the global average collection (global average pooling).

3.2 Long Short-Term Memory Network (LSTM)

Long short-term memory network (LSTM) is a special RNN model. The main function of LSTM is to resolve the problem of gradient disappearance and explosion occurring in the process of long sequence training, which is an important method of extracting the time characteristics of data sets [14]. The LSTM performs better in longer sequences than the conventional circulating neural network (RNN). The biggest difference between LSTM and RNN is that LSTM has an internal "cell" structure that determines the usefulness of the input information. This "cell" unit consists of an input gate, a forgetting gate, and an output gate, as shown in Fig. 3. If the input information is useful, it will be retained controlled by the algorithm. And the useless information will be discarded. The LSTM inputs information through different "cell" structures and selects the reasonable data through the above mentioned gates. Each gate is equipped with a neural layer and a point-by-point multiplication operation.

As is shown in Fig. 3, there are three main stages in the LSTM:

1. Forgetting Stage. This stage is mainly to selectively forget the input from the previous node. In short, unimportant results will be forgotten and the relatively important numercial features in the data set will be retained, thus saving memory space and improving the processing efficiency.
2. Memory Selecting Stage. Input information will be selectively entered into "memory" at this stage. The main input options will be retained whatever the key focus has been recorded before. The current input data is obtained through the above calculation process. Information is selected by the gate controlling signal. The results will be obtained by transferring the above two steps to the next stage.
3. Output Stage. This phase determines what will be output in accordance with the current state. This part is controlled by the output gate, which adopts the function 'tanh' to adjust the result obtained from the previous step.

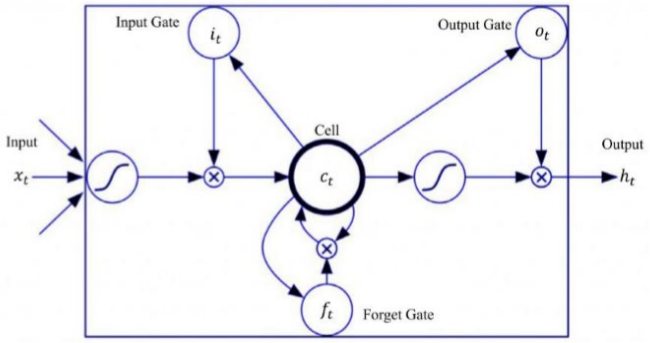

Figure 3: Basic structure of LSTM

In a word, adding LSTM module into the intrusion detection model helps to effectively extract the temporal characteristics of the data set, especially for the long time sequence.

3.3 MSCNN-LSTM Model

The MSCNN-LSTM detection model proposed adopts CNN and LSTM to automatically extract the spatial and temporal characteristics of the target data set, effectively improving the accuracy of the intrusion detection system. The MSCNN-LSTM model uses three kernels with different sizes at the convolutional layer. The same filling method is used to solve the error loss by the classified cross entropy function. The optimizer chooses Adam Optimizer to take the initial weights and offsets of each layer, setting the average value as 0 at the Gauss initialization stage.

In most cases, CNN is applied to learn the spatial characteristics of two-dimensional images. In this paper, the spatial characteristics of the entire flow image are learned from a single $P \times Q$ image. Then, the output of the MSCNN structure is a single flow vector.

LSTM is used to learn the temporal characteristics among the multiple business vectors. In this paper, LSTM is to reveal the temporal relationship among multiple flow vectors, outputting a single vector. It will be classified according to the extracted features.

In addition, the MSCNN-LSTM model adopts the back propagation algorithm (BP) in CNN. BP algorithm [15] uses chain rule in gradient descent method. The whole forward propagation algorithm is divided into two parts. In the network propagation process, the input information is transported from input layer to output layer. The weight of factors at every layer will be adjusted according to the feedback of estimated error of the objective function at the output layer trough the backward propagation process. The parameters of the overall model will be modified to improve the detection accuracy of the model.

4 Experimental Analysis

4.1 Evaluation Index

In intrusion detection experiments, the indices are adopted to verify the performance of the intrusion detection model, including *Accuracy*, *Precision*, *Recall Rate* and *F1 score*. The calculation formula shown as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall\ Rate = \frac{TP}{TP + FN} \tag{4}$$

$$F1\,score = \frac{2 * Precision * Recall\ Rate}{Precision + Recall\ Rate} \tag{5}$$

TP represents the true positive rate of abnormal samples with positive detection (correct detection), TN represents the true negative rate of normal samples with negative detection (correct detection), FP represents the false positive rate of normal samples with positive detection (false detection), and FN represents the false negative rate of abnormal samples with negative detection (incorrect detection).

Accuracy implies the overall effectiveness of the algorithm, which represents the percentage of the correctly predicted results to all sample data [16–18]. The sample selected is not involving every data in the data sets with inhomogeneous distribution. The higher accuracy will not fully prove the performance of the detection model with sampled data. Therefore, it's necessary to design and integrate the other indexes with "Accuracy" to comprehensively evaluate the detection performance of the model.

Precision [19] represents the prediction correctness of positive sample results, which is different from the accuracy involving the positive and negative samples.

The Recall rate is different from the above indexes. It is only for the original sample, representing the probability of being predicted as a positive sample in the actually positive samples [19].

The F1 score (F1-score) shows the number of attack cases in the test set that cannot be detected by the total number of normal cases [19]. During the experiment, it is hoped that the precision rate and recall rate are both higher, but it is impossible as their being contradicted. Considering the indexes' contradiction status, the appropriate threshold points balancing the precision rate and recall rate will be produced described as *F1* score. F1 score is relatively making the precision rate and recall rate both higher in the range.

For an excellent intrusion detection system, it is necessary to pursue higher accuracy, lower false alarm rate and shorter running time. Therefore, the above introduce indexes will be beneficial to verification of the designed model, promoting the detection performance.

4.2 Experimental Results

In addition to the completion of MSCNN-LSTM intrusion detection on data sets, the CNN-LSTM model is also set as a control experiment is to compare the detection results of CNN-LSTM model and MSCNN-LSTM model on the same data KDDCUP99, so as to reveal the advantages of MSCNN-LSTM model.

The first is the result of CNN-LSTM model after detecting the data set, as shown in Fig. 4.

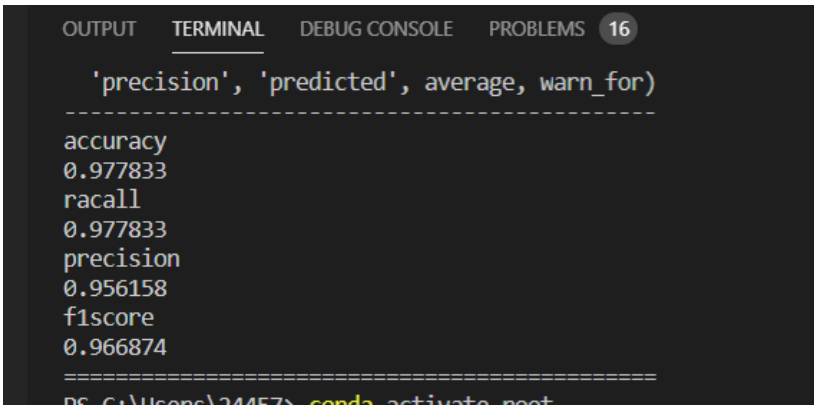

Figure 4: CNN-LSTM experimental results

The activation functions of Recall, Precision and F1-sore are also weighted with 6 decimal places reserved.

From the indexes results shown in Table 3, CNN-LSTM model is successful in detecting this data set and the verification index value are relatively higher.

Table 3: CNN-LSTM model results

Accuracy rate	Precision rate	Recall rate	F1 score
97.78%	95.62%	97.78%	0.9669

Fig. 5 shows the changes of Accuracy and Loss with iteration numbers. CNN-LSTM model has excellent performance in stable higher accuracy rate both at the training and test stages. Although there is rising error rate, it is acceptable.

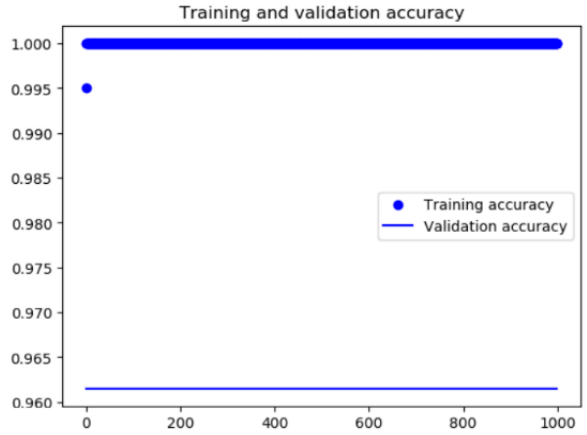

Figure 5: Change of accuracy during training

At the training stage, the CNN-LSTM model's winning rate is relatively stable. Except for the first iteration, the testing accuracy is higher in general. Although the error rate is rising during the test, it is prone to be stable eventually.

According to the four indexes obtained, CNN-LSTM model has a detection success on KDDCUP99 data set. However, CNN-LSTM model still has some shortcomings. As is mentioned above, its accuracy is not stable with the error rate abruptly increasing at the beginning of the test. The error rate is accordingly increasing as the growing number of data. However, due to the limitation of experimental equipment, it is hardly to test a great deal of data sets. Using CNN to precisely detect the intrusion embedded in the numerous real data seems impossible.

The results from MSCNN-LSTM model are shown in Fig. 6.

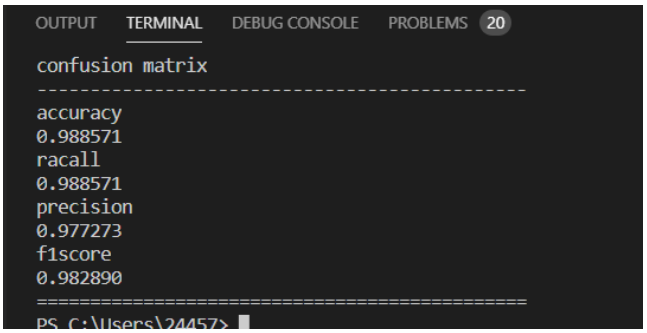

Figure 6: Experimental results of MSCNN-LSTM

According to the control variable method, the activation functions of Recall, Precision and F1 score for MSCNN-LSTM model are also weighted with 6 decimal places reserved.

From Table 4, it is shown that MSCNN-LSTM model has a better performance in the detection of KDDCUP99 data set. The accuracy rate and F1 score have been obviously increased. As the same settings of LSTM, it's revealed that MSCNN has better performance in the features extraction than CNN.

Table 4: Results of MSCNN-LSTM model

Accuracy rate	Precision rate	Recall rate	F1 score
98.86%	97.72%	98.86%	0.9829

The following Fig. 7 shows the change status of accuracy in the iterative process with MSCNN-LSTM model. The accuracy and recall rate of MSCNN-LSTM model are as same as those of CNN-LSTM model, which are stable with higher precision and lower error rate.

MSCNN-LSTM model has more optimized detection ability than CNN-LSTM model with KDDCUP99 data set both in the training process and test process as the advantage of multi-scale convolution layer. Compared with CNN's single convolution layer, the multi layers are corresponding with the different feature extraction, which is to efficiently identify the specific types of intrusion and improve the accuracy of identification.

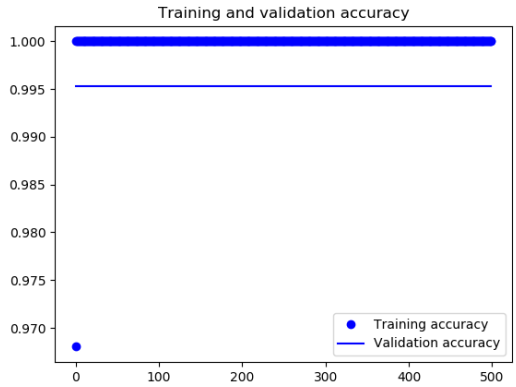

Figure 7: Changes of accuracy during iteration

In Table 5, it is shown that there is still difference of iteration time in MSCNN-LSTM model and CNN-LSTM model. With the scale of data set increasing, the iteration time in MSCNN-LSTM model is increasing with the value 5.785 s/step. As is introduced above, the MSCNN model with the multi-scale convolutional neural network extracts different features in different convolutional layers, leading to more

time consumption. Obtaining higher accuracy of detecting the different intrusion type in MSCNN-LSTM is at the cost of time consumption. Actually, this is the problem to be resolved in the future. Considering resisting invasion and guarding the network security, it is recommended to use MSCNN-LSTM model.

Table 5: Average iteration time

Model	Average time per iteration (s)
MSCNN-LSTM	5.785
CNN-LSTM	1.678

Moreover, integrating LSTM with models is beneficial to the detection performance. The accuracy of intrusion detection is about 94% in the two detection models with the convolutional neural network, which is greater than 94% both in MSCNN-LSTM and CNN-LSTM with LSTM module. Adopting the LSTM is useful to extract different feature characteristics and identify the specific intrusion type in the larger scale data sets, even though it will spend more time.

4.3 Evaluation and Comparison

In this research, the proposed models are verified in intrusion detection using holdout method. Holdout technology is a conventional technology used in the field of machine learning to evaluate the model. The data set used in the experiment is divided into two subsets, one of which is used as the training set of the model, and the other used as the test set to evaluate the performance of MSCNN-LSTM model. 10,000 lines of KDDCUP99-10precent-corrected data set [20] is used as the training set and 2,500 lines of data as the test set to evaluate the intrusion detection performance of MSCNN-LSTM model and CNN-LSTM model.

In the test process, the learning rate of CNN-LSTM model is set as 0.1, the output size of LSTM as 128, and the descending connection ratio as 1. In addition, the number of convolution filters is set to 64 at the first and second layer. It is setting the kernel size to 3 and the maximum pool length to 2. Ensuring the same data in the other experiment, we also set the learning rate to 0.1, the output size of LSTM to 70, and the descending connection ratio to 1 in MSCNN-LSTM model.

Compared with CNN-LSTM model, MSCNN-LSTM model has made improvement in data classification, intrusion recognition, recall rate and F1 score, especially in the detection and classification of abnormal data in data sets. With the increasing scale of real data set with more attack types, CNN model will be worse in feature extraction than its performance in this experiment.

Therefore, we adopt MSCNN model with different convolution layers corresponding with different feature extraction in improving the detection accuracy and precision of intrusion embedded in big data set.

5 Discussion and Conclusion

In this research, a hybrid model integrating Multi-Scale Convolutional Neural Network and Long Short-term Memory Network (MSCNN-LSTM) is designed to conduct the intrusion detection. Multi-Scale Convolutional Neural Network (MSCNN) is used to extract the spatial characteristics of data sets. And Long Short-term Memory Network (LSTM) is responsible for processing the temporal characteristics. The data set used in this experiment is KDDCUP99 with different probability distributions in the training set and test set involving some newly emerging attack types, making the data more realistic. As a result, this type of data set is widely applied in the simulation experiment of intrusion detection. In this experiment, the assessment indices such as the accuracy rate, recall rate and F1 score are introduced to check the performance of this model.

Acknowledgement: We are grateful to the peoples for the support and encouragement.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] O. Faker and E. Dogdu, "Intrusion detection using big data and deep learning techniques," in *Proc. of the 2019 ACM Southeast Conf.,* ACM, Kennesaw, GA, USA, pp. 86–93, 2019.

[2] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Computer Networks*, vol. 34, no. 4, pp. 579–595, 2000.

[3] K. Ilgun, "State transition analysis: A rule-based intrusion detection approach," in *Computer Security Applications Conf.*, IEEE, 1995.

[4] L. Feng, J. Li and F. Polytechnic, "Crowd counting using MS-CNN convolutional neural network," *Modern Computer*, vol. 1, no. 1, pp. 1–11, 2019.

[5] Y. F. Liu, C. Wang and Y. B. Zhang, "Multiscale convolutional CNN model for network intrusion detection," *Computer Engineering and Applications*, vol. 55, no. 3, pp. 90–95, 153, 2019.

[6] J. Liu, A. Shahroudy, X. Dong and W. Gang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *European Conf. on Computer Vision*, Springer, Cham, pp. 816–833, 2016.

[7] Z. R. Liang. "The model and architecture of network intrusion detection and its architecture," *Network Security Technology and Application*, no. 1, pp. 29–31, 2005.

[8] Y. H. Liu, D. X. Tian and X. G. Yu, "Large-scale network intrusion detection algorithm based on distributed learning: Algorithm for large-scale network intrusion detection based on distributed learning," *Journal of Software*, vol. 19, no. 4, pp. 993–1003, 2008.

[9] Y. Liu, S. Liu and X. Zhao, "Intrusion detection algorithm based on convolutional neural network based on convolutional neural network," *Journal of Beijing Institute of Technology*, vol. 37, no. 12, pp. 1271–1275, 2017.

[10] H. Y. Lei, H. B. Zou and H. C. Zhou. "An intrusion detection algorithm based on clustering support vector machine," *Radio Engineering*, vol. 39, no. 2, pp. 45–47, 2009.

[11] Y. Wu, M. Schuster, Z. Chen, Q. V. Le and M. Norouzi, "Google's neural machine translation system: Bridging the gap between human and machine translation," in *European Con. on Computer Vision*, vol. 1, no. 1, pp. 1–11, 2016.

[12] H. C. Shin, H. R. Roth and M. Gao, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[13] R. P. Lippmann and R. K. Cunningham, "Improving intrusion detection performance using keyword selection and neural networks," *Computer Networks*, vol. 34, no. 4, pp. 597–603, 2000.

[14] H. Q. Wang, Y. Du and Y. G. Pang, "Research on a survey of intrusion detection techniques," *Research in Computer Applications*, vol. 20, no. 10, pp. 90–94, 115, 2003.

[15] H. H. Li, J. Tian and J. Chang, "Intrusion detection system based on abuse detection and abnormal detection," *Computer Engineering*, vol. 29, no. 10, pp. 14–16, 2003.

[16] L. Yao and X. M. Wang, "Current state and trends of intrusion detection system," *Telecommunications Science*, vol. 18, no. 12, pp. 30–35, 2002.

[17] P. Chen and W. F. Lv, "Research on network-based intrusion detection approach: A survey," *Computer Engineering and Applications*, vol. 37, no. 19, pp. 44–48, 60, 2001.

[18] N. Moustafa and J. Slay. "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Military Communications & Information Systems Conf.*, IEEE, 2015.

[19] F. A. Khan and A. Gumaei, "A comparative study of machine learning classifiers for network intrusion detection," in *Int. Conf. on Artificial Intelligence and Security*, Springer, Cham, pp.75–86, 2019.

[20] M. Tavallaee, E. Bagheri and W. Lu, "A detailed analysis of the KDD CUP 99 data set," in *Int. Conf. on Computational Intelligence for Security & Defense Applications*, vol. 1, no. 1, pp. 1–10, 2009.