Computers & Security 89 (2020) 101681

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Model of the intrusion detection system based on the integration of spatial-temporal features

Jianwu Zhang[a], Yu Ling[a], Xingbing Fu[b,c,*], Xiongkun Yang[d], Gang Xiong[d], Rui Zhang[e]

[a] School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, PR China
[b] School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, PR China
[c] Science and Technology on Communication Networks Laboratory, Hebei, PR China
[d] National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu 610041, PR China
[e] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, PR China

ARTICLE INFO

Article history:
Received 14 August 2019
Revised 15 October 2019
Accepted 24 November 2019
Available online 25 November 2019

Keywords:
Intrusion detection system
Long short-term memory
Multiscale convolutional neural network
Spatial-temporal features
UNSW-NB15

ABSTRACT

The intrusion detection system can distinguish normal traffic from attack traffic by analyzing the characteristics of network traffic. Recently, neural networks have advanced in the fields of natural language processing, computer vision, intrusion detection and so on. In this paper, we propose a unified model combining Multiscale Convolutional Neural Network with Long Short-Term Memory (MSCNN-LSTM). The model first employs Multiscale Convolutional Neural Network(MSCNN) to analyze the spatial features of the dataset, and then employs Long Short-Term Memory (LSTM) Network to process the temporal features. Finally, the model employs the spatial-temporal features to perform the classification. In the experiment, the public intrusion detection dataset, UNSW-NB15 was employed as experimental training set and test set. Compared with the model based on the conventional neural networks, the MSCNN-LSTM model has better accuracy, false alarm rate and false negative rate.

1. Introduction

In modern society, the flow of information in cyberspace is growing at an alarming rate every year. In all aspects of economics, scientific research, military affairs and even people's daily life, network information security issues have been paid more attention in recent years. At present, in this research field, Intrusion Detection System (IDS) has become one of the hottest research topics because of its reliability, expandability and autonomous learning. IDS can create an effective defense system for us to defend against various network attacks, perform dynamic analysis during defensive attacks, collect key node data, and constantly improve itself to protect our cyberspace. The intrusion detection model based on machine learning is currently the most mainstream research direction such as support vector machine, Bayesian network, clustering (Miller et al., 2014)and deep learning neural network. Among them, support vector machine, Bayesian network and clustering belong to the conventional machine learning algorithms. Many researchers have applied these algorithms to the field of network information security. Nagaraja et al. (2010) constructed a model using random walk clustering to extract network traffic to detect P2P botnets; Antonakakis et al. (2012) used graph clustering to detect DGA-based malware families; Zhang et al. (2013) proposed that people perform research and analysis on bot query traffic, and use hierarchical clustering algorithm to detect whether there exists the attack behavior. Chen et al. (2011) built LS-SVM model and used the optimized SVM to classify the traffic of botnets. These supervised or unsupervised conventional machine learning algorithms can obtain better classification performance when dealing with datasets with smaller data volumes and lower dimensions (Kuang et al., 2014), which are all shallow learning algorithms. However, in the actual network environment, there are a lot of high-dimensional, label-free and non-linear data, which require us to establish new intrusion detection model.

In the field of deep learning, CNN (Danciu, 2015) and LSTM (Sundermeyer et al., 2015) have been applied to natural language processing, computer vision, speech recognition, etc. These two deep learning frameworks have their own unique network structure. In our work, the unified model combining Multiscale Convolutional Neural Network with Long Short-Term Memory (MSCNN-LSTM) are constructed, which is an integration detection model based on multi-scale spatial-temporal features.

The rest of this paper is organized as follows: in Section 2, we describe the related work; in Section 3, we describe the detailed

* Corresponding author at: School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, PR China.
E-mail address: fuxbuestc@126.com (X. Fu).

https://doi.org/10.1016/j.cose.2019.101681
0167-4048/

design; in Section 4, we explain the specific workflow of the intrusion detection model; in Section 5, we evaluate the experimental results of the model; in section 6, we make the conclusions and specify the future work.

2. Related work

2.1. Intrusion detection system

More than 30 years ago, people began to gradually pay attention to the network information security. Many current IDS systems are mainly divided into signature-based detection systems and anomaly-based detection systems. The signature-based detection system extracts its traffic signatures by analyzing known attack methods, and then compares these signatures with the signatures extracted by subsequent detection systems to discover subsequent attack traffic and issue warnings to users. The advantage of the signature-based detection system is that it has a high accuracy rate, but the detection means is limited, and it is impossible to analyze unknown attack patterns, such as 0-DAY vulnerability attack and APT (Advanced Persistent Threat) attack.

Anomaly-based detection systems, also known as network behavior-based detection systems, rely primarily on conventional machine learning algorithms and deep learning algorithms. In this method, part of the traffic features will be extracted in advance. We use a supervised or unsupervised learning method to build a learning framework based on these features. A network behavior-based detection system can detect both normal and malicious network traffic. The benefit of this approach is that it can detect unknown attacks. Therefore, this intrusion detection system has attracted more and more attention from the research community.

However, the network behavior-based intrusion detection system still has some problems in practical applications. The most difficult is to design a set of representative feature codes to detect network traffic and train the model. The design of the set of feature codes is important to the whole system, which has a great influence on false alarm rate(FAR) and false negative rate(FNR). The selection of different feature codes often has a great difference in detection performance.

2.2. Deep neural networks

The CNN and the recurrent neural network(RNN) are the two most widely used deep learning algorithms. Each of them can efficiently extract the spatial and temporal features of the datasets. In the neural network, the value of each neural node in each hidden layer is the sum of the weights of the nodes in the upper layer, and is transmitted to the corresponding next layer node through an activation function. The CNN is mainly used to extract the spatial features of datasets, and has made remarkable achievements in many computer vision fields. Based on the general neural network, the RNN adds a self-connected weighting value to each neural network node as a memory unit, which can memorize the former state of the neural network. Among them, Long Short-Term Memory (LSTM) network is developed since the RNN adds a Forget gate unit to the structure. Therefore, the LSTM network can effectively extract temporal features from long sequences, which has been used to solve many natural language processing problems such as machine translation. Spatial-temporal features are the two most common detection features in intrusion detection systems. For example, Zeng et al. (2014) used CNN to study HAR problems for the first time. By analyzing the series transmitted back by the sensors, they were converted into "image information" and then human behaviors are recognized. Wang et al. (2017) attempted to use the CNN to detect the malware attack, and transformed the network traffic into a matrix form as the training set of the CNN.

On the other hand, Torres et al. (2016) first employed the LSTM network to analyze the temporal features of linear data stream, and improved the detection accuracy of malware attack. In recent years, researchers have adopted CNN and RNN to construct network intrusion detection models based on spatial features or temporal features. All these research methods have the common property: they all apply the CNN or the RNN to construct the single model, which is obviously not comprehensive.

Based on different network protocols, multiple network traffic bytes are combined into one network packet, and then multiple data packets are combined into a data stream. The bytes, packets, and data streams of these network traffic are very similar to characters, sentences, and paragraphs in natural language processing. Classifying network traffic is similar to dividing a paragraph in natural language into positive samples and negative samples (Farhadloo, 2015). In the field of natural language processing, in recent years researchers have begun to try to integrate detection methods (Goldberg, 2016; Marie-Sainte et al., 2019); while the model learns the temporal and spatial characteristics of natural language, it has good performance. Therefore, we can also learn something from its research work and combine two deep neural networks to learn the spatial-temporal features of the original network data stream.

3. Design of MSCNN-LSTM

The MSCNN-LSTM detection model proposed in this paper is to automatically employ the CNN and LSTM to extract the spatial-temporal features of the target dataset and effectively improve the accuracy of the intrusion detection system. The learning process of the model is shown in Fig. 1.

The localized details of the MSCNN-LSTM model are shown in Fig. 2.

3.1. Selection of datasets

In this research field, the researchers commonly use KDDCUP99 and NSL-KDD which improved on the former. KDDCUP99 has been used in this field for nearly 20 years since its birth, but it has serious defects. For example, data samples are extremely unbalanced, some feature values are missing, and so on. NSL-KDD was obtained in 2009 by Tavallaee M et al.., based on KDDCUP99 (Tavallaee et al., 2009) after adding a part of the new sample data. Despite the improvement measures, NSL-KDD and KDDCUP99 still have many problems. In recent years, some researchers have observed that the defects of these two datasets will have a negative impact on the classification performance of intrusion detection systems. The main reasons are as follows (Moustafa and Slay, 2015): first, there is a lack of low-level attack samples which are common in today's networks. Low-level attacks will gradually weaken their characteristics over time and become normal traffic, which is a kind of stealth attack. Second, the test sets of these two datasets contain a part of new data, which contributes to different distribution of training sets and test sets, and it leads to false positives. Third, as the network environments change over time, there are a lot of redundant data in the two datasets that seem very unreasonable. In summary, although these two datasets have made significant contributions in the development of intrusion detection, they are not enough to reflect the real network environments now.

Therefore, in this work, the public dataset UNSW-NB15 will be used, which was created by the Australian Cyber Security Center (ACCS) laboratory (Moustafa and Slay, 2015; Moustafa and Slay, 2016). The abnormal behaviors in UNSW-NB15 are divided into nine major categories, and each abnormal behavior class will also be divided into specific attack behaviors. Compared with the four

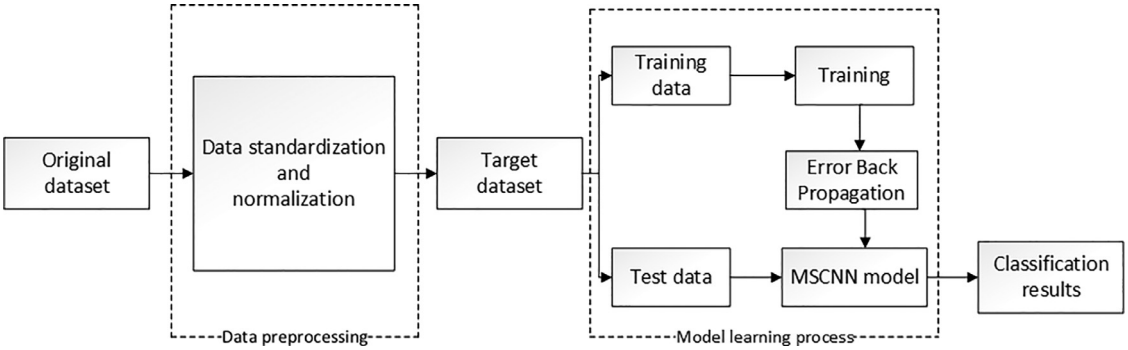

Fig. 1. MSCNN-LSTM model learning process.

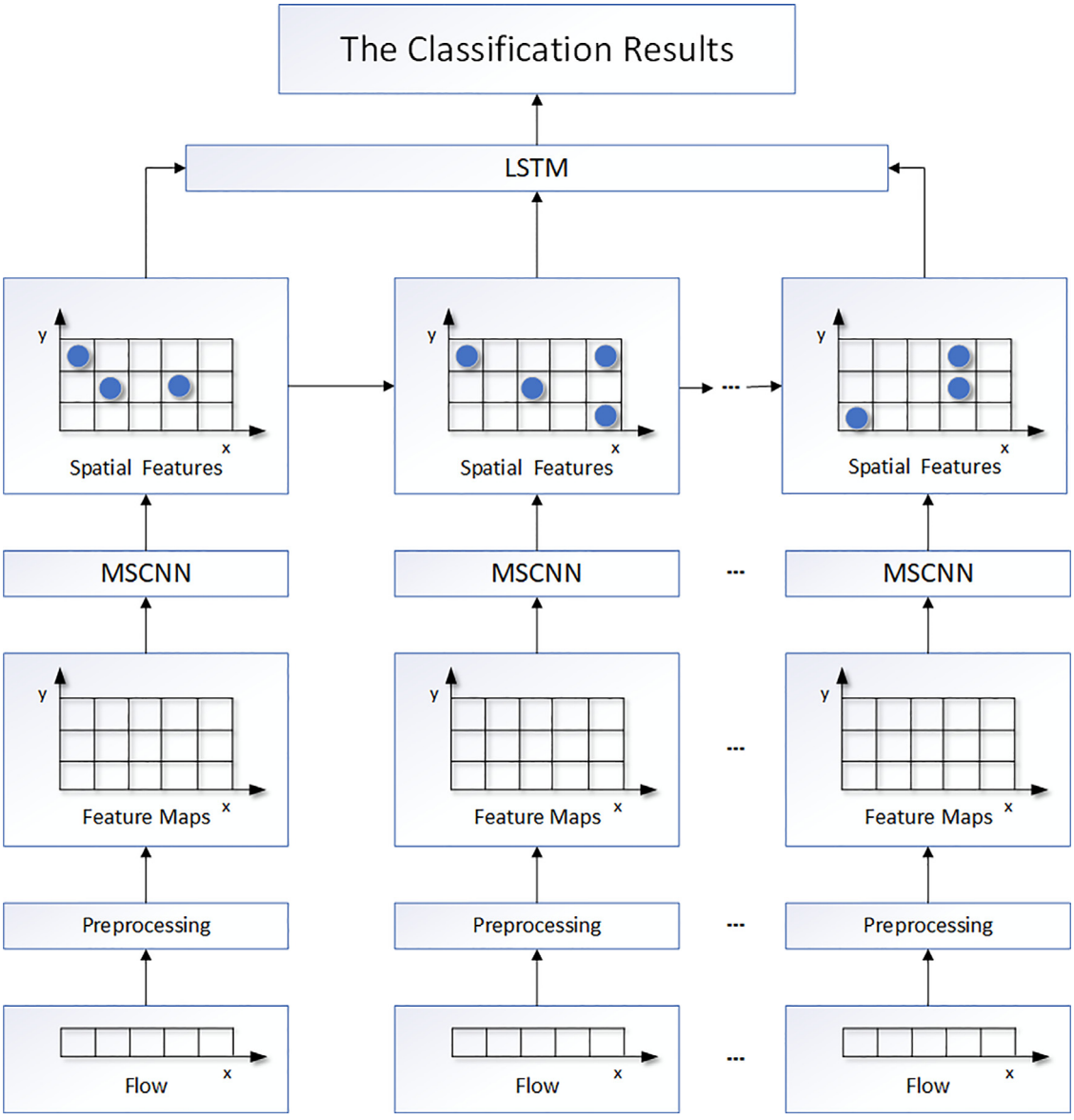

Fig. 2. Integration detection model structure.

categories of KDDCUP99, the number of the specific attack behavior of UNSW-NB15 is nearly five times. Because the abnormal behavior of the UNSW-NB15 dataset is novel, balanced, and reasonable in proportion, it is more suitable for relevant researchers to use in intrusion detection research.

3.2. Data preprocessing

This part includes data standardization and data format conversion in the model learning process. It intends to quantify and normalize the data. In the raw dataset, for example, some items are string attribute, include the 49th item “label”. In the “label”, there

Table 1
Feature subset.

proto	service	state	spkts	dpkts
sbytes	dbytes	dttl	dloss	sinpkt
djit	swin	tcprtt	smean	dmean
trans_depth	response_body_len	ct_srv_src	ct_dst_sport_ltm	is_sm_ips_ports

are only two attribute values, namely “attack” or “normal”. In order to be input as a data stream into the neural network model, we must convert the string attribute to a numeric attribute.

The dataset has nine major types of anomalous behaviors, namely Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. Each sample contains 49 features and 1 label, where the label is identified. The feature of the dataset can be specifically classified into the following types: traffic characteristics, basic features, content features, temporal features, additional generated features, and label features. A part of the redundancy obviously exists in the 49 features of UNSW-NB15. These redundant features will greatly affect the classification performance of the classifier. Because the UNSW-NB15 was established for a relatively short time, the research on feature selection of this dataset is currently concentrated on only a few documents. Therefore, in this work, the main part of the feature decides to use the feature subset evaluated by Khammassi and Krichen (2017). These feature subsets are obtained by using genetic algorithm as feature search strategy and logical regression as classifier, as shown in the Table 1.

A. Data Standardization
In the feature, the attribute values of ′ proto ′, ′ state ′, ′ service ′, and ′ attack_act ′ are symbolic, and we need to convert them to numerical type. For example, in the ′ proto ′ attribute, we map the three most important values ′ tcp ′, ′ udp ′, and ′ icmp ′ in network traffic to 1, 2, and 3, and map the remaining values to 4. After being digitized, feature values are relatively easier to handle.

B. Data Normalization
For each dimension of the feature, the values in their respective dimensions are inconsistent, and the range of values is quite different. This is particularly prominent in the UNSW-NB15 dataset. The normalization of the dataset is necessary. The high-level data has a high weight, which makes the data of low magnitude having little effect on the result, and loses some information hidden in the original dataset. The preprocessing method is shown in formula (2.1), and the feature values will be normalized to [0, 1] by linear transformation as follows.

$$f(x) = \begin{cases} \dfrac{x - x_{\min}}{x_{\max} - x_{\min}}, & x_{\max} \neq x_{\min} \\ 0, & x_{\max} = x_{\min} \end{cases} \quad (1)$$

3.3. Spatial feature learning

Most of time, CNN is used to learn the spatial features of the two-dimensional images. In this paper, the spatial features of the entire flow image are learned from a single $p \times q$ image, as shown in Fig. 2, and then, the output of the MSCNN structure is a single flow vector.

3.4. Temporal feature learning

The LSTM is used to learn the temporal features between multiple traffic vectors. In this paper, the LSTM learns the temporal relations between multiple flow vector, as shown in Fig. 2. The output is a single flow vector that represents the spatial-temporal features of the network flow. It will be classified according to the extracted features.

Table 2
MSCNN model hierarchy.

Layer	Type	Kenelsize	Stride	Padding	ActFunc
L1	MSConv	1*1/2*2/3*3	1	same	ReLu
L2	Conv	2*2	1	same	ReLu
L3	MSConv	1*1/2*2/3*3	1	same	ReLu
L4	Conv	2*2	1	same	ReLu
L5	MSConv	1*1/2*2/3*3	1	same	ReLu
L6	Conv	2*2	1	same	ReLu
L7	AvePool	2*2	2	same	ReLu
L8	FC				Sigmoid+Drop
L9	FC				Sigmoid+Drop
L10	FC				Sigmoid+Drop

3.5. Multiscale convolution

This work adapted the CNN structure. The CNN architecture was used to process image, and achieved good research performance in the image processing field. However, during the image processing, the CNN focuses on some local features of the image such as edge information. The identification of network traffic can not only rely on some discrete local features, but also need to combine multiple local features to perfrom the classification. Therefore, the CNN is adjusted and transformed into MSCNN to accomplish this task. When a human visual perception system maps an image in the brain, it will first form a complete set of images from far to near, and from fuzzy to clear. Therefore, the MSCNN simulates different projections of objects at different distances on the retina during human eye recognition. Network traffic is a high-dimensional dataset that cannot be identified by only a few discrete features. In the MSCNN, this paper uses multiple convolution kernels of different sizes to extract feature maps and combine them to obtain multiple sets of local features to achieve accurate identification. The MSCNN structure will be based on three original multi-scale convolutional layers, three convolutional layers, one pooling layer and three fully connected layers. The network structure parameters are shown in the Table 2.

The MS convolution layer will extract features of the dataset using 1*1, 2*2, and 3*3 convolution kernels. Because 2*2,3*3 convolution kernel is time consuming to do convolution operation on convolution layer, this paper refers to the Increment network structure (Szegedy et al., 2016), before 2*2,3*3 convolution, add a convolution process of 1*1 to reduce the dimension of features. The MS Convolution layer structure is shown in the Fig. 3.

3.6. LSTM networks

The LSTM network is specifically designed to learn time-series data. The biggest difference between RNN and LSTM is that there is a “cell” structure inside to determine whether the input information is useful. It consists of an input gate, a forget gate, and an output gate. When the information is useful, it will be left by the algorithm, and the useless information will be discarded.

This work uses a two-way LSTM network to scan and reversely scan the entire sequence. The application of LSTM networks in nat-

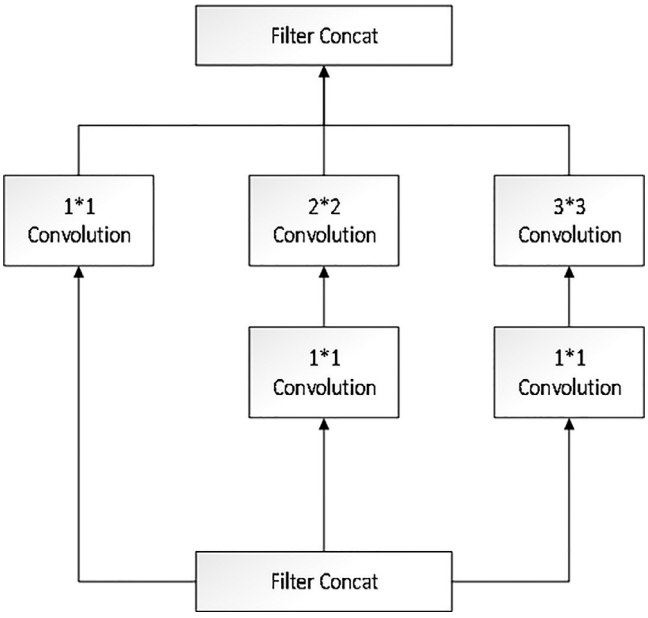

Fig. 3. MS convolution layer structure.

ural language processing shows that bidirectional scanning can extract temporal features more accurately (Li et al., 2016).

LSTM network inputs information through a different "cell" structure, which use three "gates" to achieve selecting information, each gate through a neural layer containing sigmoid function and a point-by-point multiplication operation to complete the respective tasks.

1) Forget Gate. The forget gate provides a forgetting factor f_t by reading the input layer value x_t and the output h_{t-1} of the previous ″cell″. The coefficient is between 0 and 1, which determines whether the information is discarded or not. The formula is:

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f), \tag{2}$$

where W_f is the weight matrix of the forget gate, and b_f is the bias term.

2) Input Gate. The input gate directly determines which information can be updated. The output of the input gate is multiplied by the value of the input node and a new cell state C_t is generated. The formula is:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i), \tag{3}$$

where W_i is the weight matrix of the forget gate, and b_i is the bias term.

3) Output Gate. Eventually we decide what value will be output, this output will be based on the state of our current "cell" structure, and the state of the hidden layer h_t is determined by the internal state C_t and the output o_t. The formula is:

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{4}$$

$$h_t = o_t * \tanh(C_t), \tag{5}$$

where W_o is the weight matrix of the forget gate, and b_o is the bias term.

3.7. Pooling layer

Pooling is a basic operation often used in convolutional neural networks. When pooling acts on areas of the image that do not coincide, as shown in the following figure, we define the size of the pooled window as SizeX in Fig. 4, which is the side length of the red square in the figure below, and defines the horizontal displacement of two adjacent pooled windows as Stride. When the pooled window is not repeated, then SizeX is equal to Stride. The general pooling method is mainly divided into two types: max-pooling and mean-pooling. The primary role of the pooling layer is to perform down-sampling. The design of the pooling layer is intended to reduce the parameter dimension and speed up the network training. When window sliding convolution is done, there is a large amount of overlap between the sliding windows, the resulting value will have a large amount of redundancy, and the pooling operation can minimize redundancy. When the pooling layer reduces redundant information, the local information is lost and the salient features are preserved. The reduction of redundant information can also significantly prevent over-fitting and increase the generalization ability of the model. In recent years, the classification models of the mainstream are max-pooling, and the mean-pooling is rarely considered. Generally speaking, max-pooling works better. Although max-pooling and mean-pooling subsample the data, max-pooling mainly selects features with better classification and identification. The error of feature extraction mainly comes from two aspects: The variance of the estimated value due to the limited neighborhood size increases and the error of convolution layer parameter causes the deviation of the estimated mean value. In this case, mean-pooling can effectively reduce the first type of error and retain more background feature information. In contrast, max-pooling can reduce the second error and retain more texture feature information. Therefore, mean-pooling emphasizes the subsampling of the overall feature information, and more contributions are reflected in the complete transfer of information. Even if the data is normalized and mapped to the [0,1] interval, the data that tends to end 0 and 1 in the interval still exist, if using max-pooling method will make some feature information missing.

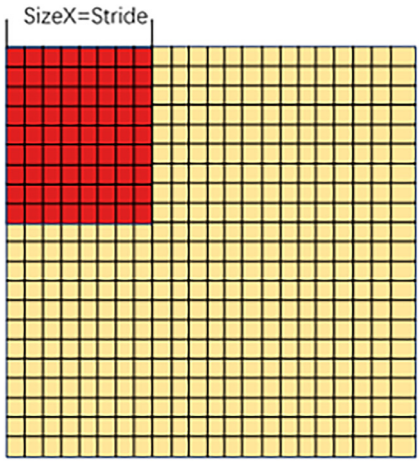

Fig. 4. Pooling layer.

Therefore, this paper uses the mean-pooling method to reduce the dimension and transfer information to the next module for feature extraction.

$$\hat{c} = average\{c\} \tag{6}$$

3.8. Model feedback

The model uses back propagation algorithm (BP) in the CNN. The BP algorithm uses the chain rule in the gradient descent method. The whole algorithm is divided into two parts, namely the forward propagation process and the back-propagation process. In the forward propagation process of the network, input information

is processed layer by layer through the input layer and transmitted to the output layer. If the output layer can not get the desired value, then the sum of the square of the output and the expected error is taken as the objective function, and then it is transferred to the back propagation process. At each layer, the partial derivative of the objective function to the weights of each neuron is obtained, which constitutes the gradient of the objective function to the weight vector, and is employed to modify the weights. Finally, the parameters of the whole model are modified and the detection accuracy of the model is improved.

According to the law of weight update, the forward conduction of the neural network can calculate the values of the neurons in the hidden layer and the output layer according to W_{ij}^1 and b_{ij}^1, and the corresponding activation values, and finally get the output, whereW_{ij}^1 is the first layer (i, j) node weight, and b_{ij}^1 is the (i, j) node offset of the first layer. If there is an error between the output and the model target, which is called feedback error, then the model needs to reverse this error to the previous conduction process, that is, W_{ij}^1 and b_{ij}^1 ; the model needs to know each neuron. How much error is brought about, and this degree of influence is expressed by the concept of " Residual ".

With the residual and objective function, the model can solve the parameters W_{ij}^1 and b_{ij}^1 using the stochastic gradient descent method. The update of W_{ij}^1 is determined by the residual, the activation function of the current neuron, and the input value. Repeat the iterations until it converges. The formula for feedback error is:

$$\delta_i^l = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) * f(z_i^{(l)}) \tag{7}$$

$$\delta_i^{(n_l)} = -\left(y_i - a_i^{(n_l)} \right) * f(z_i^{(n_l)}), \tag{8}$$

where n_l is the final output layer, δ_i^l is the residual of the i-node in first layer, z_i^l is the value input of the i-node in first layer and $a_i^{n_l}$ is the activation value of the i-node in n_l layer.

4. Experimental simulation

4.1. Experimental material

The whole process of the experiment was carried out in the Ubuntu 16.04 LTS environment, using the Keras 2.0 structure library and Tensorflow 1.1.0 as the back-end calculation.

The training and test subsets used in this work are based on the UNSW-NB15 dataset. A portion of the data has been extracted from the UNSW-NB15 dataset as a training set and test set, with 175,341 pieces of data in the training set and 82,332 pieces of data in the test set. The sampling needs to follow the principle of large sample with average sampling and small sample with full sampling. The test set obtained by sampling is Test_Set_A. The anomalous behavior categories and distributions in each dataset are shown in the following Table 3.

To further test the generalization capabilities of the model, a new test set Test_Set_B is employed in this work. Observing the pre-set training set and test set, we can find that the normal behavior of network traffic accounts for roughly one-third of the total traffic, and the rare attack categories “Worms” and “Shellcode” only account for a very small part, so the proportion of “Worms” and “Shellcode” types has increased in the new test set to test the system′s specific generalization ability. The specific distribution of the test set Test_Set_B is shown in the Table 4.

Table 3
Abnormal behavior distribution in training set and test set.

Category	Training Set	Test Set	Test_Set_A
Normal	56,000	37,000	2485
Reconnaissance	10,491	3496	457
Backdoor	1746	583	233
Worms	130	44	44
Analysis	2000	677	301
Shellcode	1133	378	255
Generic	40,000	18,871	457
Fuzzers	18,184	6062	457
Dos	12,264	4089	457
Exploits	33,393	11,132	457
Total	175,341	82,332	5576

Table 4
Distribution of the Test_Set_B.

Test_Set_B Distribution			
Category	Attack_act	Label	Test_Set_B
Normal	0	0	342
Reconnaissance	1	1	128
Backdoor	2	1	64
Worms	3	1	127
Analysis	4	1	85
Shellcode	5	1	343
Generic	6	1	128
Fuzzers	7	1	128
Dos	8	1	128
Exploits	9	1	128
Total			1601

Table 5
Comparisons of the detection performance of the four models.

	Acc	FAR	FNR	Time(s)
Lenet-5	63.9%	57.4%	70.3%	320s
MSCNN	91.4%	15.5%	7.7%	571s
HAST	85.7%	24.9%	18.0%	989s
MSCNN-LSTM	95.6%	9.8%	1.6%	1060s

4.2. Simulation process

The MSCNN-LSTM model uses three different sizes of kernels in the convolutional layer. The padding method is same padding, and the error loss is solved by the Categorical_crossentropy function. The optimizer uses AdamOptimizer, and the initial weights and offset values of each layer are taken with Gaussian initialization of 0 mean.

There are many ways to choose in the gradient descent of model training process.If each gradient descent is a calculated average gradient for all training data, this gradient descent method is called full-batch gradient descent method. When the amount of training data is of the magnitude of million, one iteration needs to wait for a long time, which greatly slows the training speed. In contrast, the batch size of random gradient descent is 1, and each training data need to update the weights. With small learning rates, the noise is much smaller, but the gradient descent is also slowed down. To solve the above problems, the model adopts mini-batch gradient descent method, and selects a batch size data quantity between 1 and the maximum training data quantity to train, and only a small part of data is trained each time, which ensures that the convergence speed is not too slow, and avoids completely falling into the local optimal solution.

In order to improve the generalization ability of the whole network, the dropout method is used in the full connection layer to avoid over-fitting. The connection probability p of the dropout

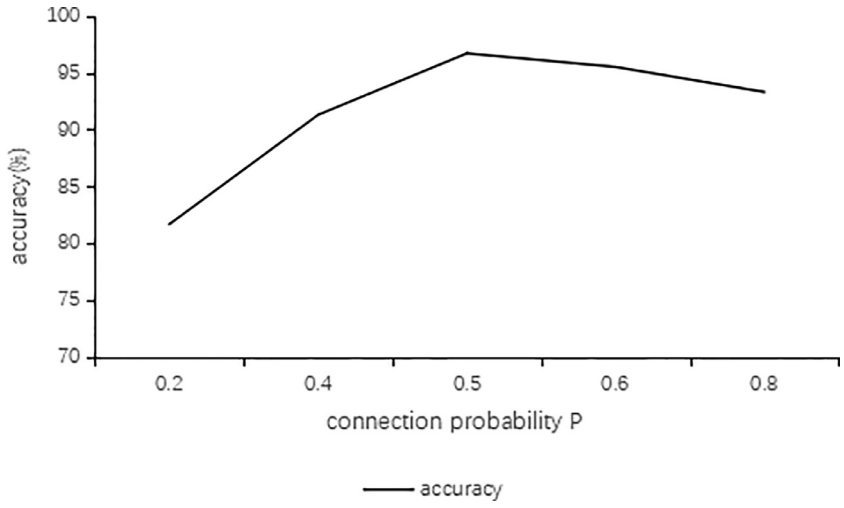

Fig. 5. Dropoutlayer connection probability.

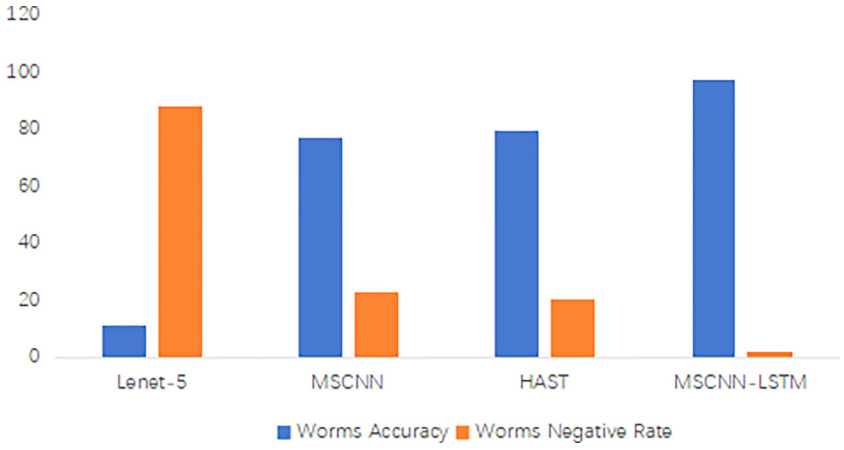

Fig. 6. Comparisons of detection rates of four models for rare attack "Worms".

layer is experimentally explored. The result is shown in Fig. 5 below. The model takes the connection probability p as 0.5.

4.3. Experimental metrics

The experiment is evaluated in terms of model training and test speed, Accuracy (ACC), False Alarm Rate (FAR) and False Negative Rate (FNR). Acc shows the overall effectiveness of an algorithm. FAR is the number of normal instances classified as an attack divided by the total number of normal instances in the test set. FNR shows the number of attack instances that are unable to be detected by the total number of normal instances in the test set.

$$Accuary\ (\mathrm{ACC}) = \frac{TP + TN}{TP + FP + FN + TN} \tag{9}$$

$$False\ Alarm\ Rate\ (FAR) = \frac{FP}{FP + TP} \tag{10}$$

$$False\ Negative\ Rate\ (FNR) = \frac{FN}{FN + TN} \tag{11}$$

For an excellent intrusion detection system, the pursuit of high accuracy, low false positive rate and false positive rate and short running time are inevitable.

4.4. Experimental results

Three sets of comparative experiments were designed, and the data were fed into the classical network Lenet-5 (Lecun et al., 1998), MSCNN, and the HAST network proposed by Wang et al. (2018) for comparisons. The MSCNN model compares the single feature and spatial-temporal features of the MSCNN-LSTM detection model proposed in this work to improve the detection accuracy. The convolutional kernels of each layer of the Lenet-5 and HAST networks use the same scale and will be used as an advantage to compare multi-scale convolutions in intrusion detection models.

The verification results are used to measure the performance of the model in terms of accuracy, false positive rate and false negative rate in order to ensure the reliability of the experiment. The final experimental results are shown in the table below:

The fusion detection model MSCNN-LSTM is superior to other models in terms of accuracy, false positive rate and false negative rate, and is only slightly inferior in computation time. The experimental time recorded in the table in this paper has included both the model training and the classification process. After the training of the model, the recognition time of the test set is very short, which can meet the requirements of real time detection.

Fig. 6 below compares the detection capabilities of the four models in the case of rare attacks, because rare attacks are often concealed and have long latency and greater destructive power.

Table 6
Final detection rate of four models for Testing_Set_B.

	Acc	FAR	FNR	Worms Acc	Worms FNR
Lenet-5	72.7%	55.3%	30.3%	79.1%	20.9%
MSCNN	85.6%	55.3%	12.1%	97.2%	2.8%
HAST	81.4%	59.9%	20.6%	89.5%	10.5%
MSCNN-LSTM	89.8%	47.4%	8.6%	99.1%	0.9%

Therefore, the focus of rare attacks is to reduce their FNR. In Fig. 6, it is shown that the integrated detection model is superior to other models in detecting accuracy and false negative rate for rare attacks.

In the last set of comparison experiments, this work tests the generalization ability of the model using Testing_Set_B as the final input test set of the model, and gets the Table 6.

The intrusion detection systems based on conventional neural network do not analyze the UNSW-NB15 dataset. In this work, we propose a new integrated detection model based on the existing model. Through the analysis of three experiments, we find that the integrated detection model is more suitable for intrusion detection system in terms of accuracy, false alarm rate, false negative rate, and the generalization ability of rare attacks.

5. Conclusions and future work

As a result, this method shows its powerful ability in the face of high-dimensional and high-complexity datasets. The method does not require any of the engineering techniques used in conventional intrusion detection system. The experimental results show that the model efficiently improves the accuracy compared to other existing methods. In addition, the FAR of many current intrusion detection methods is generally high. Our experimental results show that the MSCNN-LSTM effectively reduces the FAR because it automatically learns the spatial-temporal features, which improve the overall performance of the IDS.

In the future work, the feature selection part of the model will be further improved, and the model will have a good detection performance on imbalanced datasets. In the real world, the amount of malware traffic is small compared to the amount of normal traffic, and the proportions of different classes of malware traffic often differ greatly. Of course, improving the generalization ability of the model is also one of the research directions.

Declaration of Competing Interest

The authors declare that this paper has no conflict of interest.

Acknowledgements

This work was supported by the National Science Foundation of China No. 61772162 and No. U1866209, Science and Technology on Communication Networks Laboratory under grant 6142104180413, the Zhejiang Province Natural Science Foundation of China under grant No. LY19F020045 and LY16F020016, Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201718), and Key Research Project of Zhejiang Province No. 2017C01062.

References

Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., Dagon, D., 2012. From throw-away traffic to bots: Detecting the rise of DGA-based malware. In: Proceedings of the 21st USENIX Conference on Security Symposium. USENIX Association, Berkeley, CA, USA, p. 24.

Chen, F., Ranjan, S., Tan, P., 2011. Detecting bots via incremental LS-SVM learning with dynamic feature adaptation. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21–24, 2011, pp. 386–394. doi:10.1145/2020408.2020471.

Danciu, D., 2015. A cnn-based approach for a class of non-standard hyperbolic partial differential equations modeling distributed parameters (nonlinear) control systems. Neurocomputing 164, 56–70. doi:10.1016/j.neucom.2014.12.092.

Farhadloo, M., 2015. Statistical Models for Aspect-Level Sentiment Analysis. University of California, Merced, USA Ph.D. thesis.

Goldberg, Y., 2016. A primer on neural network models for natural language processing. J. Artif. Intell. Res. 57, 345–420. doi:10.1613/jair.4992.

Khammassi, C., Krichen, S., 2017. A GA-LR wrapper approach for feature selection in network intrusion detection. Comput. Secur. 70, 255–277. doi:10.1016/j.cose.2017.06.005.

Kuang, F., Xu, W., Zhang, S., 2014. A novel hybrid KPCA and SVM with GA model for intrusion detection. Appl. Soft Comput. 18, 178–184. doi:10.1016/j.asoc.2014.01.028.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324. doi:10.1109/5.726791.

Li, J., Xu, H., He, X., Deng, J., Sun, X., 2016. Tweet modeling with LSTM recurrent neural networks for hashtag recommendation. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24–29, 2016, pp. 1570–1577. doi:10.1109/IJCNN.2016.7727385.

Marie-Sainte, S.L., Alalyani, N., Alotaibi, S., Ghouzali, S., Abunadi, I., 2019. Arabic natural language processing and machine learning-based systems. IEEE Access 7, 7011–7020. doi:10.1109/ACCESS.2018.2890076.

Miller, Z., Dickinson, B., Deitrick, W., Hu, W., Hai Wang, A., 2014. Twitter spammer detection using data stream clustering. Inf. Sci. 260, 64â73. doi:10.1016/j.ins.2013.11.016.

Moustafa, N., Slay, J., 2015. Creating novel features to anomaly network detection using DARPA-2009 data set. In: Abouzakhar, N (Ed.), Proceedings of the 14th European Conference on Cyber Warfare and SecuritY (ECCWS-2015), pp. 204–212. 14th European Conference on Cyber Warfare and Security (ECCWS), Univ Hertfordshire, Hatfield, ENGLAND, JUL 02-03, 2015

Moustafa, N., Slay, J., 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Proceedings of the Military Communications and Information Systems Conference (MilCIS), pp. 1–6. doi:10.1109/MilCIS.2015.7348942.

Moustafa, N., Slay, J., 2016. The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf. Secur. J. A Global Perspect. 25 (1-3), 18–31. doi:10.1080/19393555.2015.1125974.

Nagaraja, S., Mittal, P., Hong, C.-Y., Caesar, M., Borisov, N., 2010. Botgrep: finding p2p bots with structured graph analysis. In: Proceedings of the 19th USENIX Conference on Security. USENIX Association, Berkeley, CA, USA, p. 7.

Sundermeyer, M., Ney, H., Schlüter, R., 2015. From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Trans. Audio, Speech Lang. Process. 23 (3), 517–529. doi:10.1109/TASLP.2015.2400218.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, pp. 2818–2826. doi:10.1109/CVPR.2016.308.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. A detailed analysis of the KDD CUP 99 data set. In: Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, Ottawa, Canada, July 8–10, 2009, pp. 1–6. doi:10.1109/CISDA.2009.5356528.

Torres, P., Catania, C., Garcia, S., Garino, C.G., 2016. An analysis of recurrent neural networks for botnet detection behavior. In: Proceedings of the IEEE Biennial Congress of Argentina (ARGENCON), pp. 1–6. doi:10.1109/ARGENCON.2016.7585247.

Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., Zhu, M., 2018. Hast-ids: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access 6, 1792–1806. doi:10.1109/ACCESS.2017.2780250.

Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y., 2017. Malware traffic classification using convolutional neural network for representation learning. In: Proceedings of the International Conference on Information Networking, ICOIN 2017, Da Nang, Vietnam, January 11–13, 2017, pp. 712–717. doi:10.1109/ICOIN.2017.7899588.

Zeng, M., Nguyen, L.T., Yu, B., Mengshoel, O.J., Zhu, J., Wu, P., Zhang, J., 2014. Convolutional neural networks for human activity recognition using mobile sensors. In: Proceedings of the 6th International Conference on Mobile Computing, Applications and Services, MobiCASE 2014, Austin, TX, USA, November 6–7, 2014, pp. 197–205. doi:10.4108/icst.mobicase.2014.257786.

Zhang, J., Xie, Y., Yu, F., Soukal, D., Lee, W., 2013. Intention and origination: An inside look at large-scale bot queries. In: Proceedings of the 20th Annual Network and Distributed System Security Symposium, NDSS 2013, San Diego, California, USA, February 24–27, 2013.

Jianwu Zhang is a professor at Hangzhou Dianzi University, and he received a Ph.D. from Zhejiang University in 1999. His research interests include mobile communication and image processing.

Yu Ling is currently pursuing the master degree in Electronics and communication engineering, Hangzhou Dianzi University. His research interests is artificial intelligence and pattern recognition.

Xingbing Fu is a lecturer, and he received the Ph.D. degree from University of Electronic Science and Technology of China (UESTC) in 2016. His research interests include cloud computing and cryptography.

Gang Xiong is an assistant professor at key laboratory of science and technology on blind signal processing. His research interests include information security and artificial intelligence.

Xiongkun Yang is currently pursuing the master degree in key laboratory of science and technology on blind signal processing. His research interests include network security and deep learning.

Rui Zhang received his B.E. degree from Tsinghua University, and M.S./Ph.D. degrees from the University of Tokyo, respectively. He was a JSPS research fellow before he joined AIST, Japan as a research scientist. Now he is with Institute of Information Engineering (IIE), Chinese Academy of Sciences (CAS) as a research professor and a professor with Univerisity of Chinese Academy of Sciences (UCAS). His research interests include applied cryptography, network security and information theory.