Cloud edge cooperative attack recognition based on CNN

Qingwen Ran¹, Xiaoming Ju¹*, Xibo Zhang² and Yan Zhang²

¹School Computer Science and Software Engineering, East China Normal University, Shanghai, 200062, China
²Ningbo Xinsheng Middle Electric Appliance Company, Zhejiang, 315032, China

*Corresponding author’s e-mail: xmju@sei.ecnu.edu.cn

Abstract. In recent years, the concept of cloud edge collaboration has been proposed, which uses some operations of edge and cloud to complete all kinds of collaboration, and has produced various applications in various fields. CNN is one of the hot spots of scientific research, and has excellent performance in the field of image voice. In view of these advantages, combined with the cooperation of CNN and cloud edge, this paper proposes cloud edge collaborative attack recognition based on CNN. The attack recognition model is trained by using CNN and open source data set, and cloud edge collaboration is realized by data and model transmission between edge and cloud. This method realizes the security collaboration between the cloud and the edge, and reduces the resource pressure of the cloud. At the same time, the cooperation of multiple edge ends increases the security identification ability of the cloud and improves the security coordination between the cloud and the edge.

1. Introduction

The emergence of cloud computing makes the network scale grow rapidly, and people's life is more convenient. However, with the increase of data, the centralized data processing mode of cloud computing has brought great challenges and pressure. The increase of hardware and edge devices has brought many challenges to cloud computing. The real-time performance of cloud computing can not meet the concept of the Internet of things. The increase of data increases the pressure of cloud computing, and it is difficult to maintain accurate real-time performance. Secondly, there are some hidden dangers in data security and privacy of cloud computing.

The edge of the cloud can not only cooperate with the cloud, but also reduce the pressure on the edge of the cloud. Edge computing terminal refers to the layer between the network edge terminal and the cloud, which ensures the communication connection between the terminal and the cloud, and stores the data uploaded by the terminal. Both cloud and edge computing provide services for customer terminals. The difference is that the edge computing end is closer to the user and has a faster response. The edge computing end provides services to users, and does not need to provide services through cloud search, which ensures the response speed. At the same time, it also brings some security risks. Malicious attackers can attack the edge computing end to obtain private information.

Attack identification is initially used to be out of various types of attacks, in order to effectively defend against all kinds of network attacks, and protect the normal operation of the network system, analysis of network node traffic to identify malicious attacks. In the past few years, the development of machine learning has made great breakthroughs in object detection, speech recognition and so on. More and more people apply it to the research of various industries. However, many studies have shown that machine learning shows excellent processing ability in anomaly detection, classification and prediction.
In the literature [2], another multi-level hybrid attack type recognition model is proposed, which uses support vector machine and limit learning to improve the efficiency of known and unknown attacks. K-means is used to construct the training data set to improve the performance of the classifier. In the literature [3], we use the modified density peak clustering algorithm and the deep belief network intrusion detection. We use the improved density peak fuzzy clustering algorithm to reduce the size of the training set, and use the deep belief self network to classify the data, which solves the problem of sample imbalance. The literature [4] uses the network-based intrusion detection model based on the hidden Markov of counter countermeasures. The concept of pattern entropy is defined in this paper. The dynamic window and threshold technology are used to improve the adaptive, anti competitive and online learning capabilities of the system.

In this paper, a cloud edge collaborative attack recognition method based on deep learning is proposed. The attack is identified by training open source attack data sets. The training is distributed in each edge computing end. The edge computing end has certain resource training ability and information collection ability. Finally, the training set with high recognition accuracy is transmitted to the cloud to update the training set.

2. Related information

2.1 Cloud edge collaboration model

Cloud edge collaboration refers to data interaction and collaborative work between the edge and the cloud center. Edge computing is directly introduced into the edge and cloud to help users achieve smoother and lower delay service delivery from cloud to edge terminal. Therefore, the cloud edge architecture models at this stage are generally three-tier "cloud edge end". The specific model is shown in Figure 1. The cloud is provided by the cloud server, which manages all the edge computing terminals. It is the data center and service center of the whole model, and provides technical service support for the edge computing end. The edge computing end is distributed in the server near the user terminal, providing data support for users, faster response and improving user experience. The hardware equipment of the edge computing terminal is worse than that of the cloud, and the data it can store is limited. Therefore, it is necessary to transmit data to the cloud periodically to sort out the local data. The edge end is also the user end, which controls the terminal for the user.

2.2 Convolution neural network and attack recognition

The purpose of attack identification is to identify the types of various malicious attacks. It does not provide information support for the next defense. It is a favorable strategy to protect computer networks and systems. Attack identification is a way of intrusion detection. Using the method of machine learning, this paper extracts and classifies the network traffic data in the computer network, so as to judge whether there is attack behavior in the network, which is an active defense way. It has the characteristics of initiative, real-time and dynamic, which can make up for the deficiency of other static defense tools. The common network intrusion detection and identification model is shown in Figure 2. By monitoring
the network nodes and obtaining packets, analyzing the characteristics of the packets to determine whether there is an attack.

Convolutional neural network has become one of the hot spots in the field of science. Its excellent performance in the field of image and speech makes more researchers invest in it. In recent years, some researchers have used the advantages of deep learning to complete some attack type recognition and detection tasks. CNN generally includes five parts: input layer, convolution layer, activation function layer, pooling layer and full connection layer. The input layer is mainly to preprocess the original input data. The convolution layer uses a filter to set a series of parameters to extract the features of the input data at different levels. The output of convolution layer will be mapped by activation function layer. Pool layer is to compress the parameters of the network and reduce over fitting. The neurons in the final total connection layer are connected to all the neurons in the upper layer, and the probability of belonging to the category is calculated.

3. Design of cloud edge cooperative attack recognition model based on CNN

3.1 Model design
This model is based on the cloud edge collaborative attack recognition model. Attack recognition training is carried out at each edge computing end, and the new data learned is updated to the current training set. The overall framework of the attack recognition model is shown in Figure 3, which is divided into five modules: data acquisition module, data preprocessing module, feature selection module, CNN attack identification module, and data transmission update module. The data acquisition module is mainly to collect malicious attack data from the client. After cleaning, add the data set to complete the self-learning function of the model. The data preprocessing module can process the data digitization, standardization and normalization, which is convenient for later training. Feature selection module mainly selects feature data to improve the training accuracy in the later stage. CNN attack recognition module is mainly used to image the input vector data first, then the training set data is used for training, and finally the test data set is used for detection. Data transfer and update function: the initial data set of each edge computing terminal is the same, and the current edge computing end data set is updated by collecting malicious data of the managed client in the later stage, so as to achieve self-learning function. In a fixed period of time, the edge computing end with high recognition rate will transmit its own data set to the cloud. The cloud will update the data set and carry out training, and the better model will be distributed to each edge computing terminal.

![Figure 3. Cloud edge cooperative attack identification architecture based on CNN](image)

3.2 Feature module design
Although CNN performs well in the field of image processing, it also has some success in the field of language processing. It is suitable for training multidimensional data and data with strong local
correlation. In CNN training, a large number of intermediate data will be produced, which results in the decrease of training speed. Reducing the number of CNN intermediate parameters is the key method to improve the training speed of the model. The dimension of the input vector data is an important factor that directly affects the amount of intermediate parameters. Therefore, it is very important to select the best data features. Purity is usually used to select features. Generally, information gain and information gain rate are used to measure purity. Since the number of attributes of different data features in text selected dataset is quite different, information gain rate is used for feature selection. The information gain rate is defined as:

\[
GR(D, a) = \frac{IG(D, a)}{IV(a)}
\]

(1)

\[
IV(a) = -\sum_{i} \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}
\]

(2)

Where IG (D, a) is the information gain, D is the sample set, and IV (a) is the fixed value of attribute a. The more types of attribute a, the greater the value of IV (a).

This model selects features according to the information gain rate of different feature data.

3.3 design of attack recognition based on CNN

(1) Classification output process. In essence, the process of classification output is to use fully connected neural network to classify and learn the one-dimensional feature vector output from the feature extraction layer:

\[
c^{(l+1)} = W^{(l+1)} + a^{(l)} + b^{(l+1)}
\]

(3)

\[
a^{(l+1)} = f(c^{(l+1)})
\]

(4)

Where \( l = 1, 2, 3 \ldots L \). L is the total number of output layer neural networks, \( a^{(l+1)} \) is the output of \( l + 1 \) layer, \( c^{(l+1)} \) is the input weighted sum vector of \( l + 1 \) layer, \( f(x) \) is the excitation function, \( W(L + 1) \) is the weight of \( L + 1 \) layer, \( B(L + 1) \) is the bias of \( L + 1 \) layer.

The essence of neural network parameter updating is error back propagation, and the error function is assumed to be \( J(W, b) \). In the fully connected neural network, the vector expression of parameter updating is as follows:

\[
w^{(l)} = w^{(l)} - \alpha \frac{\partial J}{\partial w^{(l)}}
\]

(5)

\[
b^{(l)} = b^{(l)} - \alpha \frac{\partial J}{\partial b^{(l)}}
\]

(6)

Where \( \alpha \) is the learning rate and is derived from the chain derivation rule.

\[
\frac{\partial J}{\partial w^{(l)}} = \delta^{(l+1)} (a^{(l)})^T
\]

(7)

\[
\frac{\partial J}{\partial b^{(l)}} = \delta^{(l+1)}
\]

(8)

\[
\delta^{(l+1)} = \frac{\partial J}{\partial a^{(l+1)}} f'(c^{(l+1)}) \quad l=1,2,\ldots,L-1
\]

(9)

Where: \( \delta() \) is called energy function. According to formula (5) - equation (9), all parameters of output layer can be updated after \( l \) times of iterative calculation. (2) Feature extraction process. In the forward transfer stage of feature extraction, the input data are convoluted with different convolution kernels to obtain the feature matrix. Because in each layer of convolution operation, the input characteristic graph will do convolution operation with different convolution kernels, so the convolution process can be expressed as follows:

\[
x^{(l+1)}_{j+1} = f(\sum_{k \in M} W_{j,k}^{(l)} \otimes X^{(l)}_j + b^{(l)}_j)
\]

(10)

Where: \( W_{j,k}^{(l)} \) denotes the \( i \)-th convolution layer, and the \( k \)-th input eigenvector corresponds to the \( j \)-
5th convolution kernel parameter; $X_{i}^{(j)}$ denotes the input eigenvector of the i-th convolution neural network $X_{j+1}^{(i+1)}$ represents the output corresponding to the j convolution kernel in the i-th convolution layer; $b_{j}^{(i)}$ is the offset vector corresponding to the j convolution kernel in the i-th convolution layer; $f$ is the excitation function; $m$ is the set of all input eigenvectors. In each convolution operation, the convolution kernel performs convolution operation repeatedly with the input eigenvector to generate new features.

The sampling layer performs down sampling based on local correlation to reduce the amount of data while retaining useful information. Generally speaking, the pooling layer appears after the convolution layer, and the output of the i-th sampling layer is assumed to be $X_{j+1}^{(i+1)}$, the input is $X_{j}^{(i)}$, represents the output of the i-th convolution layer, and the sampling process can be expressed as follows:

$$X_{j+1}^{(i+1)} = \text{subsampling}(X_{j}^{(i)})$$

Where: subsampling $(x)$ is the sampling function, which usually takes the maximum or average value of the window area. In the sampling process, the multiplicative bias is 1 and the additive offset is 0.

After several convolution and sampling operations, a one-dimensional eigenvector is finally generated.

In the reverse propagation stage, the neuron parameters are updated by the inverse derivation of the error function. Compared to the feature extraction stage, the difference is that the existence of the sampling layer reduces the matrix dimension. Therefore, delta $\delta^{(i+1)}$ needs to be sampled as the matrix dimension of the volume layer, and the upper sampling function $\text{up}()$ is introduced.

$$\delta^{(i)} = f''(e^{(i+1)}) \text{up}(\delta^{(i+1)})$$

Where: $\circ$ is the multiplication of each element. Similar to the classification output process, the parameter update formula can be obtained as follows:

$$W_{jk}^{(i)} = W_{jk}^{(i)} - \alpha (X_{j-1}^{(i-1)}) \Theta(\delta^{(i)})^T$$

### 3.4 Cloud edge data transmission design

Data transfer and update function: the initial data set of each edge computing terminal is the same, and the current edge computing end data set is updated by collecting malicious data of the managed client in the later stage, so as to achieve self-learning function. In a fixed period of time, the edge computing end with high recognition rate transmits its own data set to the cloud. After the cloud updates the data set and carries out training, the better model is distributed to each edge computing end.

### 4. Experiment

#### 4.1 Data preprocessing

The data set used in this paper is the kdd-cup99 data set, which is a classic data set in the field of intrusion detection by collecting 90000 network connections and system audit data. The data set consists of 39 attack samples and normal samples, including 17 unknown abnormal samples. These abnormal samples can be divided into four categories: DOS (denial of service attack); U2L (unauthorized access from remote host); U2R (unauthorized local super user privilege access); Probe (scan attack). Each data in the dataset has 41 dimensions, 38 dimensions are digital features, and 3 dimensions are symbolic features. It is necessary to preprocess each feature attribute to eliminate the differences between features and make them comparable. Therefore, it is necessary to standardize the features of continuous data and then normalize the discrete features.

1. The digitization of symbol features requires that the digitized features include three character features and markers. Since the input value of the algorithm is required to be a digital matrix, it is necessary to convert three symbolic features: protocol. Type feature, flag feature and service feature are converted into digital form. protocol. The type feature includes three symbols: TCP, UDP and ICMP, which are represented by numbers 1, 2 and 3. Similarly, the service feature has 70 symbols, so it is
represented by positive integers from 1 to 70. There are 11 symbols in flag feature, which can be expressed by positive integers from 1 to 11. 23 kinds of marker sites were divided into five categories: normal, DOS, prob, u2l, u2r. They are represented by 0, 1, 2, 3 and 4 respectively.

(2) Among the 41 dimensional features of the continuous data standardized data set, 22 are continuous features, and the range and method of these 22 different continuous attributes are different. At the same time, \( \tanh \) function is used as activation function, and its distribution is around the zero point. It can be used for continuous attributes. In order to standardize the range of data set value, the data value is near the zero point. Moreover, because the absolute value of the ramp rate near the zero point of the activation function value is large, the convergence rate of the convolution neural network model can be accelerated. The standardized formula is as follows:

\[
x^* = \frac{x - \mu}{\sigma}
\]

Where: \( X \) is the original data; \( \mu \) is the mean value of the sample; \( \sigma \) is the standard deviation of the sample; \( X^* \) is the normalized data, which follows the normal distribution with the mean value of 0 and the variance of 1.

(3) In order to improve the efficiency of the algorithm, all discrete data features are normalized in the feature selection process. Similarly, in order to improve the convergence speed of convolutional neural network and eliminate the huge difference between different features caused by dimensionality, it is necessary to normalize the other 19 discrete features and adopt the maximum minimum normalization method. The normalization formula is as follows:

\[
x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}
\]

Where: \( X \) is the original data of the feature, \( x_{\min} \) and \( x_{\max} \) are the minimum and maximum values of each feature respectively, and \( X^* \) is the normalized data. Finally, the values of each feature are in \([0,1]\).

4.2 experimental environment and evaluation method

In this paper, we use tensorflow deep learning framework in Linux environment, and use Python language to write the algorithm. The evaluation criteria are accuracy rate (AC), false alarm rate (FP) and recall rate (recall). TN is the number of normal behavior samples correctly classified; TP is the number of abnormal behavior samples correctly classified; FP is the number of normal behavior samples that are wrongly classified; FN is the number of abnormal behavior samples that are wrongly classified.

\[
AC = \frac{TP + TN}{TP + TN + FP + FN}
\]

\[
FP = \frac{FN}{TP + FN}
\]

\[
Recall = \frac{TP}{TP + FN}
\]

4.3 experimental results and analysis

In this paper, 10% kdd-cup99 data set is used as the experimental data set. Firstly, the gain rate of different features in the data set is calculated to filter the features, and the 20 features with the highest gain rate are intercepted. The is_hot_login feature is discarded, and the characteristic value of this term is 0. As shown in Table 1.

| features                | Gain rate | features                | Gain rate |
|-------------------------|-----------|-------------------------|-----------|
| Wrong_fragment          | 0.972     | logged_in               | 0.722     |
| num_comromised          | 0.928     | Service                 | 0.715     |
| Land                    | 0.915     | dst_host_srv_serror_rate| 0.689     |
| protocol_type           | 0.851     | dst_host_serror_rate    | 0.675     |
| flag                    | 0.813     | diff_srv_rate           | 0.613     |
| num_failed_logins       | 0.807     | dst_host_same_src_port_rate | 0.582   |
In this paper, the input data is selected according to the gain rate of different characteristics of the input data, and the corresponding accuracy rates of four different attack modes are obtained by training the training data set, as shown in Table 2. It can be seen that when the dimension of the input data is greater than or equal to 25 dimensions, the accuracy rates of the four attacks do not change much, among which, the DoS attack accuracy rate is 92.1% - 95.8%, and the Probe attack accuracy rate is 87.9% - 90.7%. The accuracy of U2R attack was 90.1% - 92.7%, and that of U2L was 92.8% - 93.8%. Therefore, the dimension of the input vector can be selected as 25 dimensions.

Table 2. accuracy of four attacks in different dimensions

| dimension | 5  | 10 | 15 | 20 | 25 | 30 | 35 | 41 |
|-----------|----|----|----|----|----|----|----|----|
| Dos (%)   | 78.2 | 83.2 | 89.7 | 91.5 | 92.1 | 93.5 | 94.7 | 95.8 |
| Probe (%) | 74.5 | 84.1 | 85.4 | 87.4 | 87.9 | 89.1 | 89.5 | 90.7 |
| U2R (%)   | 68.8 | 81.2 | 88.3 | 89.6 | 90.1 | 91.4 | 91.6 | 92.7 |
| U2L (%)   | 65.1 | 78.5 | 84.7 | 91.1 | 92.8 | 93.5 | 93.7 | 93.8 |

After adding real DoS attack data, the training will be conducted. DOS attack accuracy decreased by 21.3%. Before the data preprocessing, it is necessary to clean up the real data, remove the unnecessary data, clean the dirty data to meet the experimental data preprocessing standards, and then use the data preprocessing data digitization, data standardization, data normalization methods for processing. Table 3 shows the accuracy rates under different dimensions after adding real DoS attacks.

Table 3. Accuracy rates under different dimensions after adding real DoS attacks

| dimension | 5  | 10 | 15 | 20 | 25 | 30 | 35 | 41 |
|-----------|----|----|----|----|----|----|----|----|
| Dos (%)   | 58.2 | 63.2 | 69.7 | 71.5 | 72.1 | 73.5 | 74.3 | 74.5 |

Compared with the untreated convolutional neural network intrusion detection scheme used in reference [10], because the scheme only studies the two classification problem, the four attacks are replaced by abnormal behavior, and the multi classification is transformed into the binary classification problem. Table 4 shows the average training time of the model under different input data dimensions. Compared with the method in reference [10], the method adopted in this paper can reduce the training time by 198.04s and reduce 75%.

Table 4. average training time of the model

| dimension | 5  | 15 | 25 | 35 | 41 |
|-----------|----|----|----|----|----|
| time/s    | 26.75 | 36.04 | 63.27 | 124.38 | 263.23 |

5. Conclusion
This paper proposes a cloud edge collaborative attack recognition model based on CNN, and conducts attack recognition training at each edge computing end, and updates the learned new data to the current training set. The model is divided into five modules: data acquisition module, data preprocessing module, feature selection module, CNN attack recognition module, data transmission update module. In order to increase the speed of feature extraction, the gain rate is used to select features. Experiments show that the average training time of this method is improved. In this paper, the self-learning ability of the whole model is added. The data obtained from the edge end is added to the training set after data cleaning, which makes the model more suitable for real scenes. However, due to the complexity of the real data, the accuracy of the model decreases after adding the real data, which needs to be improved in the later stage.
References:

[1] Zheng Weifa. Research on Intrusion Detection Algorithm Based on cnn-lstm hybrid model [J]. Network security technology and application, 2020 (05): 61-64

[2] Li Yong, Zhang Bo. An intrusion detection algorithm based on depth CNN [J]. Computer applications and software, 2020,37 (04): 324-328

[3] Ioffe S, Szegedy C, Batch normalization: accelerating deep network training by reducing internal covariate shift[C]//International Conference on International Conference on Machine Learning. JMLR.org, 2015.

[4] Stolfo S J, Fan W, Lee W, et al. Cost-based modeling for fraud and intrusion detection: Results from the JAM project[C]//Proceedings DAPA Information Survivability Conference and Exposition. DISCEX'00, IEEE, 2000.

[5] Chand N, Mishra P, Krishna C, et al. A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection[C]//2016 International Conference on Advances in Computing, Communication & Automation (ICACCA). IEEE, 2016.

[6] Sultana N, Chilamkurti N, Peng W, et al. Survey on SDN based network intrusion detection system using machine learning approaches[J]. Peer-to-Peer Networking and Applications, 2019, 12(2): 493 – 501.

[7] Hodo E K, Bellekens X, Hamilton A, et al. Threat analysis of IoT networks using artificial neural network intrusion detection system[C]//2016 International Symposium on Networks, Computers and Communications ( ISNCC). IEEE, 2016.

[8] Kim K, Aminanto M E. Deep learning in intrusion detection perspective: overview and further challenges [C]//2017 International Workshop on Big Data and Information Security (IWBIS). IEEE, 2018.

[9] Ren Jiadong, Liu Xinqian, Wang Qian, et al. Multilayer intrusion detection method based on KNN outlier detection and random forest [J]. Computer research and development, 2019, 56 (3): 566-575.

[10] Tao P, Sun Z, Sun Z. An improved intrusion detection algorithm based on GA and SVM[J]. IEEE Access, 2018, 6: 13624-13631.