Application of Convolution Optimization Algorithm Based on Neural Network in Web Attack Test

Li Nan, Guan Sheng
Digital Technology College, Dalian University of Science and Technology
79734008@qq.com

Abstract: As society enters the process of information development, the demand for information services in various industries is increasing. Various types of software based on Web services have been widely used in various system platforms, and also bring greater security service requirements. All kinds of network hackers will cause serious damage to the attacks on port access information data. Based on the security of Web services, the paper starts with the characteristics of Web attacks, and studies the structure scheme of Web information feature extraction as a vector model, which is brought into the neural network to calculate by convolution optimization algorithm. The feature vector extraction proposed in the paper is a multi-dimensional training method for convolution input layer. It is suitable for information structure changes in different network environments. By setting the nodes of the Web network as elements of neurons, the connection of several neurons for the model matrix is selected to simulate the effect against web attacks. In the test process, the multi-dimensional space of the proposed algorithm is superior to other algorithms, and has the characteristics of adaptability and controllability. The result of the analysis proves the accurate effect of the research content.

1. Introduction
In the process of social information network modernization, web-based programs are becoming more and more mainstream trends. Many applications based on Web services can access people's electronic device information through various ports, which also creates security risks. All kinds of hackers' attacks on the Web will cause serious problems such as leakage of information resources and loss of property and economy. In the face of these problems, it is necessary to take security measures to prevent risks[1].

At present, hacker traffic attacks in complex network environments are dominated by positive rule matching and machine learning. Based on the characteristics of Web attack, the paper studies the structure scheme of Web information feature extraction as a vector model, and brings it into the neural network to calculate by convolution optimization algorithm. The feature vector extraction proposed in this paper is a multi-dimensional training method for convolution input layer. It is suitable for information structure changes in different network environments. Its multi-dimensional space is more flexible than other algorithms, and has the advantages of adaptability and controllability.

2. Convolutional neural network
The convolutional neural network is a network structure in which the network structure of the two-dimensional structure is expanded into a three-dimensional or multi-dimensional network neural structure based on the traditional neural network, and the elements based on the neuron organization can be interconnected in the structure. The calculation of the neuron elements is first weighted, and
then the function algorithm is selected by setting the domain definition, and the result is calculated. The Web network attack test process studied in this paper is to simulate the attack effect on the Web network by setting the nodes of the Web network as the elements of the neurons to select the connection of several neurons to the model matrix[2-3].

According to the structure of the convolutional neural network, the neuron node is defined as: Le represents a weighted term of the element; $\lambda_z$ is an activation function. The Sigmoid function suitable for the study of this paper is used as the calculation output, and the formula is expressed as:

$$\lambda_z = \frac{1}{1 + le^{-z}} \ (1)$$

The function output range in formula (1) is $(0, 1)$.

3. Web security detection based on convolutional neural network

According to the characteristics of convolutional neural networks, multi-dimensional network organization can be constructed. Each organization can be set as an input layer for output, and the results of each layer are mapped to the next layer for iteration. When a layer of results is regarded as a learning sample entering the next layer, it can be passed to the next layer for output after passing through the pooling layer function. As a standard sigmoid type function, the selection criteria of its nodes are activated as functions. The whole process is shown in Figure 1.

3.1 Convolutional layer

The convolutional transformation form is widely used to achieve multiplication and multiplication of multiple variables in a multidimensional range. In the study of this paper, the input setting of the convolution operation is set, the weight of the convolution is set to $h(m)$, and the convolution operation formula is[4]:

$$y(m) = \sum_{i=-\infty}^{\infty} x(i)h(m-i) \ (2)$$

The convolution operation based on the feature extraction process can enhance the initial features and eliminate the influence of noise, so the analysis of feature weights is more accurate. The convolution feature model has a very strong role from the primary model to the highest convolution kernel. After the convolutional feature model is brought in by the upper layer original model features, the current model feature can be obtained:
\[ x_i^l = \sum_{j \in N_i} y_i^{l-1} \otimes k_{ij}^l + b_j^l \]  

(3)

In formula 3, \( x_i^l \) represents the input of \( i \)-th feature map of the \( l \)-layer; \( y_i^{l-1} \) is the output of \( i \)-th feature model layer of the \( l-1 \)-layer; \( k_{ij}^l \) represents the corresponding weight; \( b_j^l \) represents the bias of the \( j \)-th feature model of the current layer Set[5-6].

3.2 Pooling layer

According to the output of the convolutional layer, it is brought into the pooling layer for calculation. The main function is to reduce the complexity of the program in the high-dimensional operation, constrain the original data neural layer and neurons, and achieve the effect of reducing the dimension and reducing the neurons. In the pooling layer, the input and output are equivalent quantities. When the input model is \( N \), the output will also have \( N \) models whose formula is expressed as:

\[ x_i^l = f[\text{down}(x_i^{l-1})] + b_j^l \]  

(4)

In formula 4, \( \text{down}(x_i^{l-1}) \) is the down-sampling of the \( j \)-th feature model of the \( l-1 \)-layer.

From formula 4, for the \( j \times j \) block summation of the feature model, the complexity of the 1/2 dimension can be reduced on the basis of the original 2D of the model output, and the data model is guaranteed not to receive the influence of the change. In the study of this paper, the maximum pooling and average pooling methods are adopted. Among them, the maximum eigenvector of the original data is used as the feature model, which is the maximum pooling method; while the feature point weighted average performance feature model region is extracted by the average pooling[7].

3.3 Full connection layer

The fully connected layer belongs to the convolutional neural network connection mode, which plays a vital role in the accuracy of each layer. The relationship of the neuron output is:

\[ h_{n,b}(x) = f(w^T x + b) \]  

(5)

In formula 5, the input neuron is set to \( x \); the output neuron is \( h_{n,b}(x) \); the input weight matrix is \( W \); and the offset vector is \( b \).

3.4 classification layer

The role of the classification layer in convolution is to classify the input data and determine the data range by class identification probability. In the paper, Softmax is used as a classifier, and the input neuron \( x \) setting function is used to divide the class probability of each vector data. The relational expression is:

\[ p = p(y = i|x) \]  

(6)

In Equation 6: When the input data is \( x \), \( p \) is the probability of the output category \( y = i \).

\[ p(y = i|x) = \frac{e^{t_x(i)}}{\sum_{i=1}^{k} e^{t_x(i)}} \]  

(7)

After undergoing at least 2 stages of training, a convolution-based neural network model is implemented. The size of the data model is determined by the different training phases and data set sizes. Therefore, the convolutional neural network model can be applied to different data sets, and the training level and data set size are also quite flexible. After satisfying the problem coverage area, the operation result can be obtained, and the calculation accuracy can be controlled[8-9].
4. Web attack test simulation experiment

4.1 Test method
The steps based on the simulated web attack test first obtain the traffic information by requesting the virtual terminal to grab the data packet of the local web service; the reverse flow data is sorted according to the data packet capture situation, and combined into the web traffic; the information format is converted according to the web request data that generate related files; finally, according to the convolutional neural network model, it is detected whether the relevant files are web traffic generated by web attacks.

4.2 Experimental environment
The experiment is carried out in the server virtual machine environment. The component platform includes that Lenovo server is based on CentOS1511 system, Xeon processor *4, 128GECC memory, 6TB virtual storage space. For the virtual environment, 255 virtual terminals are set. The virtual environment adopts a scale compression of 1000:1, the iteration layer is set to 15 layers, and the number of attack test trainings is 100 units, which are successively superimposed[10-11].

4.3 Experimental process
First, according to the main information of the Web attack test, the Web feature is built to build a feature model. In the Web information, the feature extraction value of Table 1 is taken as the most basic parameter.

Table 1. Web information characteristics

| No | Feature item                        | Weights | No | Feature item                                    | Weights |
|----|-------------------------------------|---------|----|------------------------------------------------|---------|
| 01 | Initial value                       | 0.25    | 17 | Access to the total amount of IP (repair)       | 0.43    |
| 02 | System resource                     | 0.34    | 18 | Access IP C segment (repair)                    | 0.58    |
| 03 | First visit time                    | 0.16    | 19 | Access UA amount (repair)                       | 0.69    |
| 04 | Visits                              | 0.55    | 20 | UA anomaly number                               | 0.74    |
| 05 | Latest access time                  | 0.33    | 21 | URI visit valid times                           | 0.45    |
| 06 | Total number of pages               | 0.23    | 22 | URI visit invalid number of times               | 0.64    |
| 07 | Page exposure                       | 0.45    | 23 | Total number of page visits                     | 0.45    |
| 08 | Page update count                   | 0.32    | 24 | Total page visit time                           | 0.34    |
| 09 | Page get request times              | 0.54    | 25 | Page visit time period                          | 0.34    |
| 10 | Page post request times             | 0.34    | 26 | Refer total                                     | 0.52    |
| 11 | Page corresponding maximum          | 0.64    | 27 | Refer is equal to the number of paths           | 0.68    |
| 12 | Page corresponding minimum          | 0.69    | 28 | Refer is equal to path ratio                    | 0.78    |
| 13 | Page corresponding average          | 0.55    | 29 | Page corresponding to the total number of sessionid | 0.79    |
| 14 | Different number of page corresponding | 0.63    | 30 | Sessionid exception number                      | 0.83    |
| 15 | Page corresponding proportion       | 0.42    | 31 | Sessionid exception percentage                  | 0.86    |
| 16 | Total page visit value              | 0.35    |     |                                                 |         |

The Web parameters in table 1 are set to the real data output of the website. The defined value size and the index weight index are based on the data provided by the CSIC2010 data set. The data set is tens of thousands of Web requests made by CSIC for Web attack testing. It simulates the network
application environment in different environments such as business class, and adopts ASCII encoding for data features. Therefore, the feature vector can be obtained by positive rule matching.

Then, the data packets based on the feature vector set are brought into the multi-layer convolutional neural network for training, and the training speed is performed by a ratio of 15:1 in the normal state, and the ratio of the normal flow to the attack flow of the push layer is set at 20:1.

Establish a startup operation:

1) C1 input layer: According to the neural network data set range, the matrix size is $32 \times 32$; the initial state convolution kernel $x_i$ that the size is $28 \times 28$, and after entering the convolution input layer, a $15 \times 15$ feature vector is generated and mapped to the input of the S2 layer.

2) S2 pooling layer: The set pooling window is $14 \times 14$ size, and the $14 \times 14$ feature vector generated by the C1 input layer is brought into the pooling function, as shown in Equation 4.

3) C3 layer: Convolution training is performed on the mapping vector of the S2 layer, and the generated mapping feature is $10 \times 10$, and the generated data is used as the mapping data of the S4 layer.

4) S4 layer: This layer will be pooled for all feature vectors, and the pooled vector map will generate $5 \times 5$ feature vectors.

5) Fully connected layer: Expanded into a vector space according to the model order, in which the model training and weight coefficient output are performed, as shown in Equation 5.

6) In the final classification layer, a data training model will be output for different types of network attack tests and results.

4.4 Test results and analysis
The training data set and the test data set obtained by the cross-training data set sample collection work are analyzed and studied.

$$\text{Accuracy} = \frac{\text{Correctly classify the number of samples}}{\text{Total number of samples}} \quad (8)$$

$$\text{Error rate} = \frac{\text{False positives for the number of normal samples of the invasion}}{\text{Normal sample size}} \quad (9)$$

The training test data in the paper is shown in Table 2.

| category | total sample/One | the proportion/% |
|----------|------------------|------------------|
| Normal   | 9000             | 15.3             |
| Dos      | 38000            | 65               |
| Probe    | 7000             | 12               |
| R2L      | 4000             | 0.07             |
| U2R      | 500              | 0.008            |

The test training result data is shown in Table 3.

| category | accuracy/% | false alarm rate/% |
|----------|------------|-------------------|
| Normal   | 95.7       | 4.1               |
| Dos      | 94.6       | 5.5               |
| Probe    | 94.2       | 5.8               |
| R2L      | 98.7       | 1.7               |
| U2R      | 99.11      | 0.65              |
According to the statistical analysis of the data of the test process, the accuracy and false positive rate of the research content under the optimization of the neural network-based convolution algorithm meet the test requirements. In the paper, the method of detecting the regular matching feature vector will have higher accuracy under the premise of prefabrication rule matching, and the more stringent rules will lead to the decrease of precision, which indicate that the convolution algorithm has greater controllability and suitable for adaptive training in different rule environments. The trend of accuracy changes is shown in Figure 2.

![Figure 2: Analysis of Web Attack Test Accuracy and False Positive Rate](image)

In the accuracy analysis of Figure 2, the range between accuracy and false positive rate is extremely large and adjustable, which proves the flexibility of the algorithm studied in this paper. Moreover, the advantage of the convolution algorithm is that more feature vectors participate in the neural network, which will form a larger amount of data information. The convolution algorithm can be used to set the current spatial dimension and convolution training level as the information changes to improve accuracy.

5. Conclusion

Based on the Web attack test, the paper uses the convolutional neural network algorithm to optimize the design. The Web data is used as the feature model index, and the feature vector is brought into the convolution input layer, and the calculation process is performed by convolution, pooling and classification. In the test process, the results of the analysis verify the accurate effect of the research content, and still need to optimize the convolution algorithm structure in the future work, reduce the false positive rate, and improve the security of Web services.

References:

[1] Design of Software to Search ASP Web Shell[J]. Xu Mingkun, Chen Xi, Hu Yan. Procedia Engineering. 2012

[2] Intrusion detection using reduced-size RNN based on feature grouping[J]. Mansour Sheikhan, Zahra Jadidi, Ali Farrokhi. Neural Computing and Applications. 2012 (6)

[3] HMMPayl: An intrusion detection system based on Hidden Markov Models[J]. Davide Ariu, Roberto Tronci, Giorgio Giacinto. Computers & Security. 2011 (4)

[4] McPAD: A multiple classifier system for accurate payload-based anomaly detection[J]. Roberto Perdisci, Davide Ariu, Prahlad Fogla, Giorgio Giacinto, Wenke Lee. Computer Networks. 2008 (6)

[5] Anomaly-based network intrusion detection: Techniques, systems and challenges[J]. P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez. Computers & Security.
2008 (1)
[6] The theory and practice in the evolution of trusted computing[J]. Dengguo Feng, Yu Qin, Wei Feng, Jianxiong Shao. Chinese Science Bulletin. 2014 (32)
[7] Small-world overlay P2P networks: Construction, management and handling of dynamic flash crowds[J]. Ken Y.K. Hui, John C.S. Lui, David K.Y. Yau. Computer Networks. 2005 (15)
[8] Global stability analysis of a general class of discontinuous neural networks with linear growth activation functions[J]. Huaiqin Wu. Information Sciences. 2009 (19)
[9] Elastic-net regularization in learning theory[J]. Christine De Mol, Ernesto De Vito, Lorenzo Rosasco. Journal of Complexity. 2009 (2)
[10] Recognition of human activities using SVM multi-class classifier[J]. Huimin Qian, Yaobin Mao, Wenbo Xiang, Zhiquan Wang. Pattern Recognition Letters. 2009 (2)
[11] A cross entropy approach to design of reliable networks[J]. Fulya Altiparmak, Berna Dengiz. European Journal of Operational Research. 2008 (2)