Use neural network algorithms to securely detect HTTP traffic

Qiugen PEI, Dawen HUANG

1Guangdong Power Grid Co., Ltd, Guangzhou Guangdong, 510600, China
2Zhaoqing Power Supply Bureau of Guangdong Power Grid Co., Ltd., Zhaoqing, Guangdong, 526060, China
1586278294@qq.com

Abstract. HTTP traffic security detection is inseparable from the support of algorithms. However, in the traditional security detection methods, the classification ability of the algorithm is weak, resulting in detection results lower than expected. Therefore, based on neural network algorithm, a new HTTP traffic security detection method is studied. This research designs a collection module to collect HTTP traffic data. Structural features and statistical features are extracted to build a sensitive thesaurus. Based on neural network algorithm, HTTP traffic security is detected. The experimental results show that compared with the traditional detection methods, the detection method studied has better classification effect and the abnormal traffic data obtained is more accurate. It can be seen that under the application of neural network algorithm, the detection effect of the detection method has been further improved.

1. Introduction

Under the background of the rapid development of information technology, the development of Internet application is advancing by leaps and bounds. The scale and complexity of the network have shown an unprecedented expansion trend. According to the survey report, up to now, the total number of Internet users in China has exceeded 940 million. The number of Internet websites has also exceeded 4.97 million. It can be seen that the Internet has brought more convenient services to the majority of Internet users [1]. However, the statistical results show that when the Internet brings convenience to people’s daily life, there are also serious network security problems. The literature [2] method proposed a security detection method based on classification algorithm, but when this method performs security detection on HTTP traffic, the detection results obtained are not up to expectations. Therefore, based on neural network algorithm, a new HTTP traffic security detection method is proposed. Neural network algorithm has the characteristics of large-scale parallel structure and distributed storage, so it has excellent self-organization, adaptability and fault tolerance. This research makes full use of the characteristics of neural network algorithm to provide a more reliable detection technology for the national Internet HTTP traffic security.

2. HTTP traffic security detection based on neural network algorithm

2.1. Collection of HTTP traffic data
In the high bandwidth network environment, when the HTTP traffic fluctuates abnormally, it is necessary to detect this type of HTTP traffic[3]. Therefore, the first step in detecting HTTP traffic is to collect HTTP traffic data. Therefore, a traffic collection module is designed based on DPDK. The overall framework of DPDK is shown in Figure 1 below.

![Figure 1 DPDK overall frame structure](image)

Unlike other processing tools, most of the modules shown in Figure 1 are concentrated in user space, while a few code runs in kernel space [4]. As shown in Figure 1, DPDK provides the necessary processing tool library for high-speed packets of users, including 6 core components, rte_eal, rte_malloc, rte_ring, rte_mempool, rte_mbuf and rte_timer. They are sorted according to A1-A6 for convenience of recording. Among them, the environment abstraction layer EAL provides a general API interface for users to use space, and helps users shield the details of the underlying environment. It has three basic scheduling functions, access, initialization, and execution of resource allocation. Component A2 provides a management interface for large page memory to improve the access efficiency. Component A3 provides the task of supporting single or multiple producers. It also provides an in and out queue lock free processing interface for single or multiple consumers to ensure data synchronization and reduce system overhead. Component A4 provides a memory pool interface for other components to allocate a specified number of memory pools with unique names. Component A5 is a buffer management component that provides the basic functions of creating memory allocation and destroying data buffers for the acquisition module. And A6 is used as a timer component, providing timing function for HTTP traffic data collection [5]. According to the above components to build the flow data acquisition module. The rte_eal is used to initialize the environment abstraction layer. EAL is used to initialize the NIC. API is used to create the memory pool queue and call the data buffer mbuf. Then the packet port information is captured. Finally rte_eth_rx_burst() interface is called to receive cyclic data packets and realize the collection of HTTP traffic data.

2.2. Construction of sensitive thesaurus

HTTP request is a kind of semi-structured data, which has a standard structural form. At the same time, due to different request methods, the HTTP request format will be different. Therefore, according to the traffic data collection results in the previous section, the HTTP request is analyzed. The HTTP data is processed into a structured data, and finally expressed in a fixed format [6]. Through the above process, the HTTP data structure characteristic attributes are defined directly from the HTTP request. The results are shown in Table 1 below.

| No. | Characteristic property name   | No.     | Characteristic property name        |
|-----|-------------------------------|---------|-------------------------------------|
| 1   | Method                        | 2       | Protocol-Version                    |
| 3   | Accept                        | 4       | Accept-Charset                      |
| 5   | Accept-Language               | 6       | Accept-Encoding                     |
| 7   | Cache-Control                 | 8       | Connection                           |
| 9   | Content-Type                  | 10      | Pragma                              |
| 11  | User-Agent                    |         |                                     |
Some structural feature attributes in Table 1 can be valued according to simple enumerable strings. Some more complex feature attributes are composed of string sets, such as attribute 6, whose values can be in the form of “x-gzip” and “deflate”. When taking values for these attributes, these strings are regarded as collections to realize the extraction of structural feature attributes. Therefore, a large number of these features can be included in the statistics. Therefore, assuming that the collected content contains key value pairs with the number of \( n \), and the length of attribute value in each key value pair can be set to \( d_1, d_2, \ldots, d_n \). Then the mean value and variance of each attribute value can be calculated by using the following two groups of formulas:

\[
\text{mean} = \frac{1}{n} \sum_{i=1}^{n} d_i \\
\text{variant} = \frac{1}{n} \sum_{i=1}^{n} (d_i - \lambda)^2
\]

In the above calculation steps, formula (1) represents the mean value of the attribute value. Formula (2) represents the variance of the attribute value. \( d_i \) represents property value length with the order of \( i \), \( \lambda \) represents the deviation data. According to the above process, the structural features and statistical features are extracted. The sensitive thesaurus is constructed according to the combination of sample data, the normal text and abnormal text combined by word segmentation, the feature words after cleaning the segmentation, the abnormal text after traversing the segmentation and TF-IDF sorting [7]. The TF-IDF sorting process needs to calculate the weight, and the calculation of the value is inseparable from the vector composed of feature words. The formula is as follows,

\[
r = \left( (c_1, \omega_1), (c_2, \omega_2), \ldots, (c_s, \omega_s) \right)
\]

In the formula, \( c_i \in c_r \) represents the feature word. \( \omega_j \in \omega_a \) represents the corresponding weight of the feature word. According to the weight evaluation function of feature words, the value \( TF \) and value \( IDF \) are calculated. The formula is as follows,

\[
TF_{c_i} = \sum \frac{N_{c_i}}{N_{c_i}, k \in r}
\]

\[
IDF_{c_i} = \log \frac{|R|}{1 + \sum_{j \in r \cap c_i} |R_j|}
\]

In the formula, \( N_{c_i} \) represents the number of \( c_i \) occurrences in the text \( r \). \( \sum N_{c_i} \) represents the total number of occurrences of all feature words in the text. \( |R| \) represents the total number of texts in the data set. \( r_j \) represents all texts containing feature words \( c_i \). \( |r_j \in R : c_i \in r_j| \) represents the total number of texts containing feature words \( c_i \). According to the above process, sensitive thesaurus is constructed.

2.3. Detection of HTTP traffic security based on neural network algorithm

Based on the constructed sensitive database, the security of HTTP traffic is detected by neural network algorithm. The neural network algorithm is used to initialize the network. Assuming that the input and output of the neural network constitute a sequence of \( (X, Y) \), the input layer, the output layer and the hidden layer of the network can be determined. The numbers of the nodes are respectively \( a \), \( b \) and
The connection weights and thresholds are initialized. The connection weight between the input layer and the hidden layer is set as \( u_{ij} \). The neuron connection weight between the output layer and the hidden layer is set as \( u_{jk} \). The thresholds of the input layer and the output layer are \( p \) and \( q \) [8].

Then the neural network neuron excitation function is set as \( f(x) \) and the learning rate as \( v \). The second stage calculates the output of the hidden layer. If the network input variable is known as \( X \), the output of the hidden layer is,

\[
H_j = f \left( \sum_{i=1}^{n} u_{ij} x_i - p_j \right)
\]  

(6)

In the formula, \( j \in m \). There are many ways to express the excitation function. This research adjusts the excitation function as follows,

\[
f(x) = \frac{1}{1 + e^x}
\]  

(7)

According to the previous step, the output of the output layer is calculated,

\[
O_k = \sum_{j=1}^{n} H_j u_{jk} - q_k
\]  

(8)

Then, according to the expected output of the network and the calculation results of the above formula, the prediction error of the network is calculated as follows,

\[
e_k = O^\prime - O_k
\]  

(9)

In the formula, \( O^\prime \) represents the expected output of the network [9]. Then, combined with the preset learning rate of neural network, the weights are updated as follows,

\[
\begin{align*}
    u_{ij}' &= u_{ij} + v H_j (1 - H_j) x(i) \sum_{k=1}^{n} e_k u_{jk} \\
    u_{jk}' &= u_{jk} + v H_j e_k
\end{align*}
\]  

(10)

After the above calculation is updated, the threshold is recalculate according to the connection weight between the three network layers[10].

\[
\begin{align*}
    p_j' &= p_j + v H_j (1 - H_j) x(i) \sum_{k=1}^{n} e_k u_{jk} \\
    q_k' &= q_k + e_k
\end{align*}
\]  

(11)

According to the calculation results, whether the iteration end condition of the algorithm is met can be judged. If not, it will return to recalculation. If it is, the result will be output directly. The detection process of neural network algorithm is shown in Figure 2.
According to the above process, the security detection of HTTP traffic is completed by using neural network algorithm.

3. Experimental study
A comparative test is proposed. The proposed safety detection method is taken as the test object of the experimental group, and the traditional safety detection method is taken as the test object of the control group. Through the establishment of different test environment, the difference of security detection between the two test groups is compared.

3.1. Experimental preparation
In order to ensure the reliability of the test results, three different types of data sets are selected as the basic test environment. Table 2 shows the basic parameters of the three test data sets.

| Data set | Attribute number | Sample number | Category number |
|----------|-----------------|---------------|----------------|
| balance  | 5               | 626           | 3              |
| pima     | 8               | 768           | 2              |
| wdbc     | 30              | 569           | 2              |

According to the experience and experimental test, the number of particles is set as $s$ and $s = 50$. The inertia weight is 0.9. Then the output layer weight of neural network is obtained between $[-1, 1]$. According to the calculation, the target error is $e = 10^{-5}$. Therefore, under the above conditions, the maximum iteration steps of the two test groups are set to 3000. Three fold cross validation method is used to train and test the samples of experimental group and control group. At the same time, before the detection and classification, the SNR method is used to reduce the attributes of the dimension data set. The relationship curve between the dimension and the classification error rate under the application of the experimental group and the control group is obtained, as shown in Figure 3 below.
According to the curve in Figure 3, the error rate of the experimental group is the lowest in most dimensions compared with that of the control group. Therefore, the feature dimension of the experimental group is reduced to 8 dimensions, while that of the control group is reduced to 9 dimensions.

3.2. Results and analysis

According to the above preparation content, the first group of data sets is taken as an example to compare the ability of the algorithm to capture threat data when detecting HTTP traffic in the experimental group and the control group. The results are shown in Figure 4 below.

According to the comparison results in Figure 4, in the face of the threat data in the data set, the capture efficiency of the experimental group algorithm is higher. While the control group can also capture the threat data. But in general, its classification and detection ability is slightly weaker than
that of the experimental group. The second group and the third group of data sets are used as the test environment, and the same test is carried out. Table 3 shows the comparison results of classification accuracy of different detection methods in three experimental background environments.

| Experimental environment | Experimental group | Control group |
|--------------------------|--------------------|---------------|
| balance                  | 89.13%             | 85.46%        |
| pima                     | 97.82%             | 95.17%        |
| wdbc                     | 84.25%             | 75.43%        |

According to the calculation results in Table 3, the test results of the experimental group are better than those of the control group in different test environments. It can be seen that the neural network algorithm has better performance for HTTP traffic security detection.

4. Conclusion
This research draws on the core of traditional detection methods, uses neural network algorithm to replace the original algorithm, and achieves good research results. It provides a more reliable detection technology for the future network HTTP traffic security.

However, according to the experimental results, the detection method of this study still has a certain degree of room for improvement. Therefore, the future research work can shift the focus of work to the optimization algorithm, and further strengthen the classification effect by optimizing the neural network algorithm.

References
[1] LI J, YUN XC, LI SH, et al. HTTP malicious traffic detection method based on hybrid structure deep neural network[J]. Journal on Communications, 2019,40(01):24-33.
[2] LAI Q, ZENG HW. Simulation of Security Detection of Network Data Abnormal Information Traffic Transmission[J]. Computer Simulation, 2018,35(03):293-296.
[3] WANG L, ZHOU QH, WANG L, et al. Improved convolutional neural network algorithm for real-time recognition and location of mechanical parts[J]. Intelligent Computer and Applications, 2019,9(01):36-41+46.
[4] Wu JL, Zhang XH. Wsn data acquisition algorithm based on time varying evolutionary game mechanism[J]. Computer Applications and Software, 2020,37(02):91-98.
[5] GUO Q, ZHU YF, XIE YY, et al. Research on Key Technologies of Real-time Data Collection and Retrieval for Very Large Scale Network Flow[J]. Journal of Chinese Computer Systems, 2020,41(06):1314-1320.
[6] LIN XF, XIA YY, GUO JL, et al. Sensitive File Detection Method Based on CNN[J]. Computer and Modernization, 2018(07):28-32.
[7] Wusiman YBLY, GUO WQ, YU K. Research on Filtering Algorithm for Sensitive Information in Multi-form Uyghur[J]. Computer Engineering and Applications, 2020,56(10):127-133.
[8] HUANG XB, HU XW, ZHU YC, et al. Fault diagnosis of high-voltage circuit breaker based on convolution neural network[J]. Electric Power Automation Equipment, 2018,38(05):136-140+147.
[9] LIU MY, MAO JL. Optimization of Convolutional Neural Network Algorithm Based on Gray Relational Analysis[J]. Electronic Science and Technology, 2018,31(06):84-88+95.
[10] WANG J, YANG LL, YANG Min. Multitier ensemble classifiers for malicious network traffic detection[J]. Journal on Communications, 2018,39(10):155-165.