Front. Electr. Electron. Eng. China (2006) 4: 475–479
DOI 10.1007/s11460-006-0091-3

RESEARCH ARTICLE

ZHU You-chan, WANG Jian, SHANG Li-biao

Design of a new type of integrated classifier for network intrusion detection systems

Abstract Based on the analysis of the network intrusion detection model, a new design scheme for the integrated classifier is proposed. The attribute reduction algorithm of the discernibility matrix is used for the optimization design of reducing nodes of input and hidden layers. The experimental test result shows that this design is valid.

Keywords intrusion detection system, rough sets, discernibility matrix, integrated classifier

1 Introduction

In recent years, with more intrusion detection products appearing, new intrusion detection methods being introduced, and especially with the introduction of neural network theory into intrusion detection systems, the realm for the research of intrusion detection system has been expanding. The intrusion detection system based on neural networks has the following advantages:

1) The neural network can acquire knowledge and obtain the ability of predicting by training using a great deal of data. It can be a process of abstract calculation, and need not emphasize the assumption for data partitioning nor explain the details of the knowledge of the neural network.

2) Retraining neural networks with new attack samples can make it remember this sample, so that the system can possess the ability to self adapt.

Translated from *Journal of North China Electric Power University*, 2005, 32(6): 37–41 (in Chinese)

ZHU You-chan, SHANG Li-biao(✉)
Center of Information and Network Management,
North China Electric Power University, Baoding 071003, China
E-mail: zyc_hd@ncepubd.edu.cn

WANG Jian
Jiangsu Electric Power Design Institute, Nanjing 210024, China

3) After a neural network masters the normal work mode of the system, it can respond to the deviating affairs, and discover some new attack affairs.

4) The trained neural network will transfer the matching and judgment of the network to numerical calculation, thus improving its speed, which makes it suitable for real-time systems.

However, this intrusion detection system also has a lot of shortcomings. When the structure of the network is too complex, the burden of network calculation will be aggravated, the speed of network convergence will become slower, and the function of the whole system will be affected. According to the shortcomings mentioned above, a new intrusion detection model based on neural networks is introduced, and has been tested on our campus net.

2 The structure of the intrusion detection model

The main idea of the model is structured as follows: firstly, this system collects the IP packets from the network and then analyzes them and extracts their features. Secondly, the integrated classifier is trained by the information gained from the step above in order to obtain knowledge. After that, knowledge, as a hidden form, is stored in the integrated classifier. This allows the classifier to tell whether a current connection belongs to an aggressive one or not and make further responses. Finally, the information that has been saved into the database will be used to train the classifier.

According to the idea mentioned above, the system is divided into five parts: the data acquisition module, pre-processing module, integrated classifier module, database module and response module. The system work flow diagram is shown in Fig. 1.

2.1 Data acquisition module

The data acquisition module gathers packets from the network, and sends these packets into the pre-processing

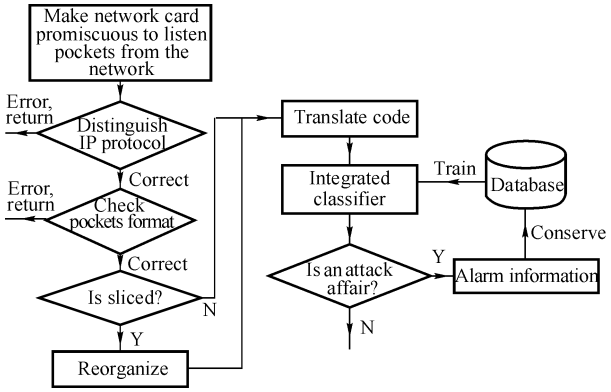

Fig. 1 The system work flow diagram

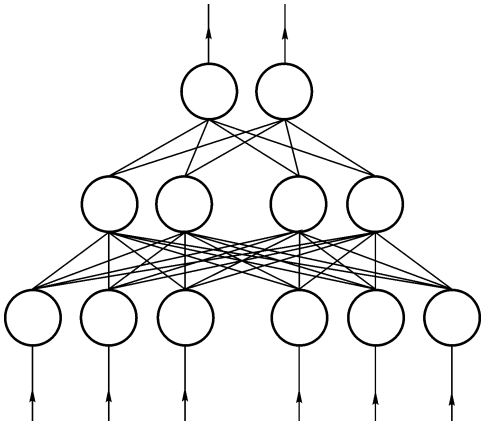

Fig. 2 Base classifier

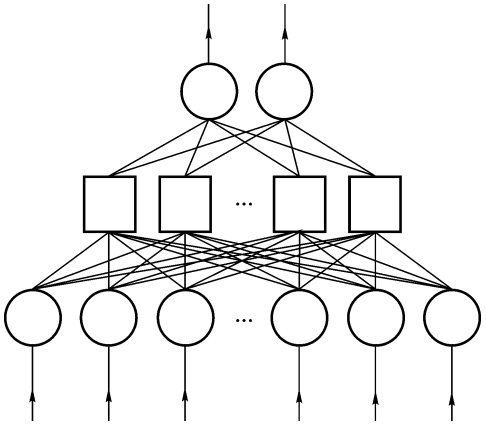

Fig. 3 Integrated classifier

module. The system firstly makes the network card promiscuous to listen to the data on the whole network. This module primarily filters packets according to the following rules. First, it judges whether the version of the IP protocol is IPV4. If not, it discards the packet. Then it carries on the format check for the IP packets. If sliced, the module reorganizes them. Finally, packets are sent into the pre-processing module for further processing.

2.2 Pre-processing module

This module adopts the method of similar-numerical to encode the information of the IP packet required (includes: IP address, port, the length of packet, etc.), and produces the input of the neural network. The similar-numerical string uses numbers to denote the information of packets. The method of similar-numerical encodes the information of the IP packets, which include non-numerical information and numerical information, to a numerical format. Non-numerical information includes IP address, etc.. Numerical information includes the length of the packet, etc.. According to this method, the module translates the information gained from the data acquisition module into the feature vectors that have many sub-vectors. Then it sends these to the integrated classifier module.

2.3 Integrated classifier module

The integrated classifier module is the core of the whole model. It includes some smaller networks that have certain functions, which is shown in Fig. 2. Called the base classifier, these smaller networks can run independently, which is used to examine attacks for a network service. Therefore, we can divide the complicated checking task into many simple attacks, which can be achieved by a base classifier. The integrated classifier includes many base classifiers that aim at different network services, including DNS, Telnet, ftp, SMTP, Http, etc.. The squares in Fig. 3 represents the base classifier.

2.4 Database module

The database module stores the alarm information and experts' knowledge. The system uses oracle 8i database. It adopts a compression-storage method to save the identification rules of packets. The method divides the input of the neural network into many blocks according to physical meaning. This module translates each block into hexadecimal numeral and saves the statement of the corresponding characters.

2.5 Response module

The response module deals with the output of the neural network. When the result of intrusion detection is abnormal, the module sends the report to the user and saves the alarm information into the database module. The purpose is to use the information for analyzing and training neural networks. The general alarm information includes: invader name, intrusion time, path of intrusion, intrusion analysis, detection time and detection scope, etc. When the result is normal, the

module makes no response and continues to process the next information.

3 The design of the structure of integrated classifier

The key of the intrusion detection model mentioned is the design of the classifier, which includes the base classifier and the integrated classifier. The design of the classifier has the following features:

1) It was proved by Robert in 1980 that the neural network, which only has one hidden layer, can approximate all continuous function. Therefore, the base classifier and integrated classifier in the intrusion detection model have three layers, which are *p*-*q*-*r*, and *P*-*Q*-*R*, respectively.

2) The transfer function of the classifiers is the sigmoid function: $f = 1/(1+e^{-x})$

3) We can see from Fig. 2 that the base classifier and integrated classifier have the same node amount on the input layer: p=P.

4) The node amount on the output layer is a constant. We can know the number of the base classifier denoted as r, and determined by the attack behavior acting on this classifier. We can also confirm the number of the integrated classifier, which is denoted as R, and determined by the whole attack behavior.

5) According to the node of the hidden layer, choosing the moderate number is always the key and nodus. The moderate node amount on the hidden layer can decrease the training time of the network, accelerate the speed of convergence, and improve the ability of generalization. In this intrusion detection model, the node of the hidden layer in the integrated classifier has certain meanings. It represents the base classifier, therefore, Q is a constant. We only need to confirm the node number of the hidden layer in the base classifier. First, we let $q = p-1$, then reduce the node by training the network to a moderate number. This number is q.

6) According to the design for the weight matrix between the input layer and the hidden layer, we let its value be a random number much less than 1. The weight matrix between the hidden layer and the output layer，half are +1 and others are −1. The purpose is to allow all nodes to make the output of the transfer function remarkable. This method can accelerate the speed of neural network training.

When the node amount on the input and hidden layer is too much, the structure of the network becomes complex, network training slows down dramatically, and the network may obtain the yawp of samples, inosculate samples and excessively lower the ability of generalization. So, after designing the preliminary structure of the neural network, we need to optimize the structure. Rough set theory can extract the feature from a great deal of data, so we use this theory to reduce the nodes of the input layer and the hidden layer. The reason is that under the condition of guaranteeing the accuracy, we set up the neural network with fewer nodes.

4 Optimize the structure of the classifier based on rough sets

4.1 Rough set theory [1, 2]

Rough set theory [3] was put forward at the earliest in 1982 by Polish scholar Z. Pawlak. In recent years, rough sets have become a new focus in the field of artificial intelligence, and have been researched and applied extensively in machine learning, knowledge acquisition, knowledge discovery and decision-making analysis, etc. The main characteristic of this theory is that knowledge and classification are related, and redundant information is reduced by knowledge reduction in the condition of maintaining the classification ability of the decision system.

Definition 1 $S = (U, A, V, F)$ is denoted as an information system. $U = \{X_1, X_2, \cdots, X_n\}$, a nonempty, finite set of objects; A, a nonempty, finite set of attributes; $V = \bigcup_{P\in A} V_P$, the value set of A; $F_P : U \times A \to F_P$, an information function, $F_P(x_i, A_j) \in V_j$. When A is composed of condition attribute set C and decision attribute set D, and $C \cap D = \phi$, $S = (U, C \cup D, V, F)$ is denoted as a decision system,

Definition 2 In a decision system, every condition attribute associates with each other to some extent. The attribute reduction means that it can express the association most simply between the decision attribute set and the condition attribute set, under the condition of no loss of information.

Definition 3 For a decision system $S = (U, C \cup D, V, F)$, the discernibility matrix $\boldsymbol{M} = (m_{ij})$ improved by Hu is defined as follows: $m_{ij} = \{a \in C : a(x_i) \neq a(x_j)\}$, $D(x_i) \neq D(x_j)$; $m_{ij} = \phi$, $D(x_i) = D(x_j)$. Where $a(x)$ is the value of object x in the condition attribute a, and $D(x)$ is the value of object x in the decision attribute D.

Definition 4 For a decision system $S = (U, C \cup D, V, F)$, attribute a_i has k different values, $V_I = \{V_{1I}, V_{2I}, \cdots, V_{KI}\}$, and the weightiness function for attribute a_i, $f(a_i)$, is defined as follows:

$$f(a_i) = V_{1I}\ V_{2I} + V_{1I}\ V_{3I} + \cdots + V_{k-2i}\ V_{K-1I} + V_{K-1I}\ V_{KI}\ .$$

4.2 Attribute reduction algorithm based on discernibility matrix [3−6]

Attribute reduction is an important research topic in rough set theory. People always wish to gain the optimum result, but attribute reduction is an NP problem and can only be settled through heuristic information. The attribute reduction algorithm has many categories. In this model, we use the attribute reduction algorithm based on the discernibility matrix [1, 4, 7].

This algorithm has three steps. Firstly, we calculate the discernibility matrix then calculate the core. It consists of

elements that have one attribute in the matrix. Finally, we calculate the result of the reduction. According to the non-core attribute, it is sorted by the importance of attribute, which is determined by the function $f(a_i)$. The most important attribute is added into the core set, and this attribute is deleted from the matrix. This process is repeated until the matrix is empty, which results in the core set. This algorithm is as follows:

Input: decision system $S = (U, C \cup D, V, F)$;

Output: reduction system $S' = (U, R \cup D, V', F')$, R is the reduction of C according to D.

Step 1 According to Definition 3, calculate the discernibility matrix $\boldsymbol{M}$.

Step 2 Calculate the core of C according to D, $\mathrm{CORE}_D(C)$.

Let $\mathrm{CORE}_D(C) = \phi$, the element in matrix $\boldsymbol{M}$ is sorted by the number of the attribute. If the number of the attribute is single, add this attribute into $\mathrm{CORE}_D(C)$ and delete it from the matrix. Do this until the matrix has no element that has a single attribute.

Step 3 Let $R = \mathrm{CORE}_D(C)$. According to Definition 4, calculate the weightiness function $f(a_i)$. Then add the attribute whose functional value best fits into R set, and delete this attribute from the matrix. Then do this until the matrix is empty. The R set is the result.

Step 4 $S' = (U, R \cup D, V', F')$ is a decision system after reduction, R is the reduction of C according to D, V', F' are obtained from V, F by getting rid of the redundant attribute.

4.3 Optimize the structure of the classifier

The steps of optimizing the structure of the classifier are as follows:

1) Optimize the node of the input layer. The information table S_1 is built. Its elements are gained from the database module. The columns in the table represent attribute, rows represent the value of the attribute. table S_1 is reduced by the attribute reduction algorithm mentioned earlier, then the number of the remaining columns is the node amount of the input layer p.

2) Optimize the node of the hidden layer. The structure of the neural network is p-q-r and let q=p−1. Take a few samples to train this network by using the BP algorithm. When the training is finished, use the input of the hidden layer to build table S_2, and use weights between the hidden layer and the output layer to build table S_3. The output of the hidden layer is associated with the weight between the input layer and the hidden layer. So the node amount of the hidden layer can be obtained by reducing table S_2 and table S_3. After reducing the two tables by the reduction algorithm mentioned, the node amount of hidden layer q = max {the number of columns of table S_2, the number of columns of table S_3}.

5 Learning of classifier

After optimizing the structure of the classifier, it is trained by data taken from the database module. The steps of the training are as follows:

1) Training the base classifier – different base classifiers are trained respectively. A quantity of samples is selected to train the neural network. To each sample, the output of the node corresponding with this sample is 1, and the output of other nodes is 0. BP algorithm is usually adopted. Weights are saved after learning.

2) Training the integrated classifier – the inner structure of the base classifier is preserved and not changed after its training. Then, the integrated classifier, which combines all base classifiers, should be trained. During the course of learning, the connection weights are adjusted between the output layer of the base classifier and that of the integrated classifier. BP algorithm is usually adopted. Weights are saved after learning.

3) A comprehensive assessment function for the structure of the network is put forward. This function is defined as follows: $f = aA+\beta B+\Gamma c$, a, β, Γ are weights, A is the learning precision of the network, A = {(1–0.7) good, (0.3–0.7) moderate, (0–0.3) bad}, B is convergence time, B = {(1–0.7) quick，(0.3–0.7) moderate, (0–0.3) slow}, C is the node amount of the whole network, C = {(1–0.7) more, (0.3–0.7) moderate，(0–0.3) less}.

4) When the training of the classifier is finished, we use the assessment function to assess this classifier. If this classifier cannot satisfy the demand, we rebuild table S_1, S_2, S_3 and reduce them again, and finally build a new model. We do this until the model meets the demand.

6 Experimental results

The experiments are done using a PentiumIII 500, 200 M main memory and Windows2000 OS. Intrusion detection to DNS and Telnet service is the main task. The process of the experiments is concretely shown as follows.

6.1 Optimize the structure of the classifier

1) Optimize input nodes. 400 samples are selected individually from the database module to build an information table named S_1. The time that is spent in reducing input nodes base on table S_1 is 14 minutes 35 seconds and 15 minutes 46 seconds, respectively.

2) Optimize hidden nodes. 200 samples are selected to train the neural network preliminarily. The time that is spent is 12 minutes 20 seconds and 13 minutes 30 seconds, respectively. After the last step is accomplished, the information table and weight table, which are named S_2 and S_3 correspondingly, are picked up and reduced. The time spent is 4 minutes 25 seconds and 5 minutes 51 seconds,

respectively.

The optimization results are displayed in Table 1. As shown in the table, the structure of the integrated classifier is 78–2–9.

Table 1 The optimization result

	Before reduction	After reduction
DNS classifier	160–159–6	64–15–6
Telnet classifier	160–159–4	78–19–4

6.2 Learning of classifier

After the structure optimization of the classifier is finished, the DNS and Telnet classifier are trained respectively following the steps mentioned above.

There were 5 kinds and 3 kinds of 500 samples selected to train. The weights are conserved after training. The time spent in training is 40 minutes 42 seconds and 34 minutes 52 seconds, respectively.

The next step is to collect a total of 8 kinds of 1 000 samples to train the integrated classifier and saving the weights. The time spent in training is 15 minutes 23 seconds.

6.3 Results

After training, 2 000 samples are selected to test the intrusion detection model. Results of the classifier are shown in Tables 2 and 3:

Table 2 Results of DNS classifier

Attack	Lquery	Tsign LSD	Tsign OWN	Infoleak	Tsign lucy
Identifying rate	91.36 %	93.85 %	91.67 %	95.78 %	96.78 %
Misreporting rate	0.38 %	0.23 %	0.46 %	0.23 %	0.25 %

Table 3 Results of Telnet classifier

Attack	Overflow	Guess	Warezclient
Identifying rate	93.58 %	96.85 %	98.95 %
Misreporting rate	0.39 %	0.22 %	0.15 %

The average identifying rate is 94.85 %, and the average misreporting rate is 0.29 %. The results of the comparison of the traditional neural network and the improved classifier, are shown in Table 4.

Table 4 Comparing of results

	The net structure	Learning time	Average identifying rate/%	Average misreporting rate/%
Traditional classifier	160–159–9	4 hours 34 minutes 58 seconds	94.31	0.28
Improved classifier	78–2–9	1 hours 58 minutes 45 seconds	94.85	0.29

7 Conclusions

In our model, an integrated classifier is composed of several base classifiers, with a single function and simple structure. The reason is that a complex attack should be divided into several simple attacks aimed at different services. At the same time, in the process of designing the structure of the classifier, the structure is optimized through introduction of a reduction algorithm. The purpose is to wipe out redundant information of samples and decrease net nodes in the condition of information integrality. Our tests show that under the condition of maintaining precision, the model has the attributes of quicker learning speed and stronger fault-tolerance, etc..

During the reduction of the hidden layer, training may go on before reduction because reduction should be based on the outputs of the hidden layer and weights between the hidden layer and the output layer. If there are too many samples, values of the model will be lost, because the time that the neural network spends in learning after reduction is longer than what is before reduction; whereas, if the information between the hidden nodes is not described well, reduction distortion will occur easily. Choosing the proper proportion between the attribute reduction and learning of the neural network in process of the hidden layer structure optimization will be our future study.

References

1. Liu Qing, Rough set and rough inference, Beijing: Science Press, 2001 (in Chinese)
2. Zhang Wen-xiu, Wu Wei-zhi et al. Rough set theory and methods, Beijing: Science Press, 2001 (in Chinese)
3. Wang Qing-yi, Fang Yan, Cai Qing-sheng, Research reduction of knowledge, Mini-Micro Systems, 2000, 21(6): 623627 (in Chinese)
4. Dai Jian-hua, Li Yuan-xiang, An algorithm for reduction of attributes in decision system based on rough set, Mini-Micro Systems, 2003, 24(3): 523−526 (in Chinese)
5. Miao Duo-qian, Hu Gui-rong, A heuristic algorithm for reduction of knowledge, Journal of Computer Research and Development, 1999, 36(6): 681−684 (in Chinese)
6. Jelonek J., Rough set reduction of attributes and their domains for neural networks, Computational Intelligence, 1995, 11(2): 339347
7. Pawlak Z., Rough sets: theoretical aspects of reasoning about data, Dordrecht The Northland: Kluwer Academic Publishers, 1991
8. Shang Li-biao, Research and application of intelligent intrusion detection system, Baoding: North China Electric Power University, 2004 (in Chinese)
9. Dou Bing-lin, Zhu You-chan, Shang Li-biao el al., Research on SNMP network management framework based on mobile agent, Journal of North China Electric Power University, 2004, 31(3): 100−103 (in Chinese)
10. Dai Qing-xin, Song Yu, Li Fu-liang, Application of fuzzy expert system in electric heated boilers controlling, Electric Power Science and Engineering, 2003, (2): 55−58 (in Chinese)