A Network Intrusion Detection Method Based on Improved ACBM

Chen Shan*, Diao Hong-Bin
Qingdao Hotel Management College, Qingdao 266100, China
*Corresponding author, e-mail: cs_ssc@yeah.net

Abstract

In order to solve the problem which includes the difficulties of network intrusion detection, redundancy of network intrusion character, difficulties for feature matching, a network intrusion detection method based on improved ACBM algorithm is proposed in this article. The improved ACBM algorithm is used to achieve matching module, which add the filtering function for module improvement. The distribution characteristics of stability and the use of the bifurcated binary tree model is used to complete the feature classification. In the stable feature, the inter-class distance is used as the classification method of support vector machine, which has strong ability of generalization as well as high identification accuracy, and has the application foreground detection. Finally, the superiority of the proposed method is proved through the data in KDDCup99 database. It shows that the proposed network intrusion detection method by experiment, which combines the ACBM algorithm and the classification mechanism, has a better accuracy than LSSVM and SVM, and it is proved that this method is very suitable for network intrusion detection under complex features environment.

Keywords: *ACBM algorithm, support vector machine, network intrusion, detection*

1. Introduction

With the rapid development of computer network, the network attacks which have diversity and concealment characteristics [1], acauses network attack types diversified and the detection rate is relative lower. The recent popular artificial neural network detection can only detect about 70% attacks, which can't meet the network security requirements. How to improve the network information security has become an important research topic [2, 3].

In order to overcome the disadvantage of tradition method low rate in the network security intrusion detection, some scholars proposed SVM (Support Vector Machine) network incursion methods, such as Dai Tianhong with four kinds of support vector machines to build the network intrusion detection system, which can successfully detect four normal incursion types; Wang Tao states the network incursion models and develops the network incursion system based on SVM [4-6]. However, the accuracy of the detection is lower and can't guarantee the network security. LS-SVM (Least Squares Support Vector Machine) is an evolutional SVM model, which constructs new quadratic loss function to convert the original support vector machines quadratic programming problem into solving linear equations ,which increases the speed and accuracy of the SVM solutions. The training parameters in the LS-SVM are particularly important for the classification performance [7]. Thus, in order to improve the accuracy of network incursion detection, It is first proposed to the matching modules based on ACBM algorithm, which adds the filtering functions to optimize the modules. This method has high detection accuracy and can be widely applied in network incursion detection [8-10].

2. The principle of Network Incursion Detection

Through analysis and research of the network incursion, the BM algorithm is relative better in the efficiency of incursion characteristics matching algorithm. The BM algorithm is signal pattern matching which can't be used to multi-pattern matching, However,the AC algorithm is poor efficiency, which fails to use heuristic strategies to leap [11]. The ACBM algorithm is a multi-pattern matching algorithm which actually uses accurate matching BM algorithm for multi-pattern, and allowing different rules in one rule tree to search and match

Received August 5, 2013; Revised October 22, 2013; Accepted November 18, 2013

simultaneously. This algorithm combines the advantages of the previous two algorithms and the efficiency is better than AC and BM algorithms. Finally, ACBM is applied to system intrusion detection [12-14].

The ACBM algorithm forms mode tree with the prefixes of the different modes in order to match the texts. It is moved from right to left while the mode tree is matching. The comparison of the characters is from the root node characters to leaf nodes layer by layer. ACBM also uses good-suffix and bad-character to move the matching modes. The difference is that BM algorithm moves mode cluster while ACBM moves mode tree.

The ACBM algorithm is parallel character string matching algorithm based on BM, which allows different rules in one tree named keyword tree or mode tree. All these rules need to be indexed by contents, and then the BM algorithm is used to search the tree. The keyword tree moves from the right of the data packet workload to left. Once the keyword tree locates proper position, the character will begin comparison from left to right. The algorithm relies on the derivative functions from the same heuristic functions of the standard BM algorithm. The difference is that the standard BM algorithm moves one mode once, but the ACBM algorithm moves a model tree,usees bad-suffix and good-suffix at the same time. Bad character shift is to make improvements to the first heuristic function of BM algorithm. The details are as follows. If the pattern can’t be matched, the mode tree moves intending to match other keyword in the tree with current texts. If in current depth, the characters don’t appear any keyword, the offset of the mode tree is the length of the shortest mode. When the mode tree is moving, the offset of the mode tree can’t be larger than the length of the shortest mode, which ensures it will not miss the pattern matching modes close to the heuristic function prediction.

2.1. The Steady State Distribution of the Characteristics

Research on network distribution characteristics ,it is assumed that the intrusion feature θ has a stable distribution, if there is the parameter: $0<\alpha\le 2,-1\le\beta\le 1,\sigma\ge 0,-\infty\le\mu\le\infty$, which has the characteristic function (CF) in the following form:

$$\phi(\theta)=\begin{cases}\exp\left\{-\sigma^{\alpha}\ |\theta|^{\alpha}\ (1-i\beta(sign\theta)\tan\frac{\pi\alpha}{2}+i\mu\theta)\right\}, & \alpha\ne 1\\ \exp\left\{-\sigma\,|\theta|\,(1+i\beta\frac{2}{\pi}(sign\theta)\ln|\theta|+i\mu\theta)\right\}, & \alpha=1\end{cases} \tag{1}$$

Wherein, the symbol function meets:

$$sign\theta=\begin{cases}1, & \theta>0\\ 0, & \theta=0\\ -1, & \theta<0\end{cases}$$

The parameters of intrusion feature distribution can be described as follows: Feature factor α determines the the degree of tailing of the distribution. Skew parameter β determines the asymmetry of the distribution feature. Scale parameter σ determines the the scale (ie, the degree of dispersion) of the distribution. Location parameter μ determines the location of the distribution. The basic features of the distribution are as follows:

(1) Different from the Gaussian distribution with exponential attenuation, the tailing of the distribution attenuates in square law, the smaller the α is, the slower the rate of attenuation is, and the distribution has more pronounced tailing, as shown in Figure 2.

(2) When $\beta=0$, the distribution is symmetric, otherwise the distribution is non-symmetrical. The comparison of the specific features are shown in Figure 1 and Figure 2.

(3) There are three exceptions in the distribution: If $\alpha=2,\beta=0$, it is a Gaussian distribution, and the corresponding characteristic function is $\phi(\theta)=\exp\left\{i\mu\theta-\sigma^2|\theta|^2\right\}$. If $\alpha=1,\beta=0$, it is a Cauchy distribution. If $\alpha=0.5,\beta=1$, it is a Levi distribution.

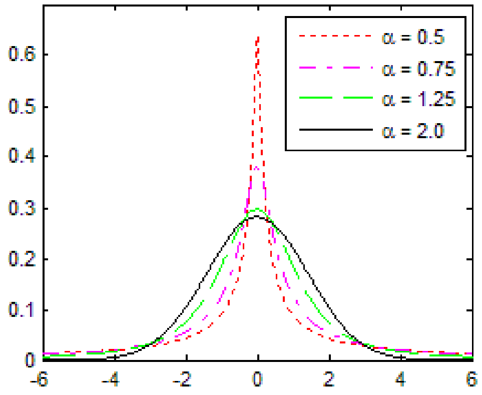

Figure 1. Symmetry Intrusion Feature Stable Distribution

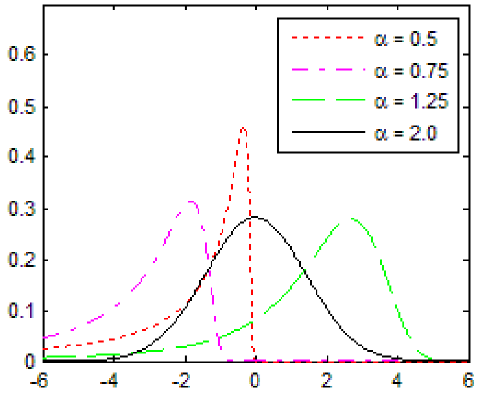

Figure 2. Asymmetric Intrusion Feature Stable Distribution

2.2. The Intrusion Detection Process of the ACBM Algorithm

The pattern matching algorithm is based on a tree K of the data structure.

(1) A intrusion feature is assigned for each edge e of the tree as a flag.

(2) It is different for network flags which are linked with the same node edge;

(3) For each mode p∈P, there is a node v to make L(v)=p, in which L(v) represents the tags set in the path root to v;

(4) For each leaf node v', there is one mode p∈P to ensure L(v')=p.

The basic thought of the finite automotive machine is to determine the next state and output according to the input and current state. In the multi-pattern matching of the intrusion feature, the automotive machine can output searched mode string and the locations of the mode string in the target strings after inputting the targeted intrusion feature. The construction of finite automotive machine is to transform mode strings sets into the finite automotive machines, which is consisted of the directional functions, loss function and output functions. The processing of pattern matching for intrusion features is then transformed to state transferred process which begins with the "start" state. Thus the process of searching modes in the main string is transferred to the searching process in the mode tree. In order to search for a string T, it should be started from the root node of the pattern tree along with the path, which takes the feature in (2) as the tag to go down: If the automatic machines can reach the final state r, then there exists the mode L(v) in T. If not so, there is no mode in T.

The automotive machine (AC) is constructed by AC matching algorithm. The detailed construction process is divided into two stages as follows.

(1) First stage: to construct the mode tree with the similar attack mode P={p1,…, ps} in the attack characteristics mode library.Beginnging with the sole root node 0, pi is added one by one. Even along the path marked by the characters in pi, if pi end in node v, v should be marked as the identifier; if all the characters in pi end after running out of all the previous paths, the flags of the remaining characters in the edge of pi will participate to nodes to finish the directional function of G. For the tags of the edges from different nodes, g(0, a)=0.

(2) Second stage: complete the loss function F

This function is attained by width priority method in Trie tree. Assuming the f function of one node which is more close to the root node is got, the node r and u=g(r, a) are considered. According to the definition, the node r is the farther node of u and L(u)=L(r)a. f(u) is the node which has the longest (deepest) distance with root node and it uses the longest true-suffix as the tag. In the construction of finite automotive machine, the characters of each mode strings are added in the order from front to behind. While matching, the input of target string, which is also called matching, is also following the order from front to behind.

Following is the moving of intrusion feature of the ACBM algorithm.

Assuming the mode set is"time, tired, tiring, tinted, tinsel" and text is "timeisonmysideo", the keyword tree and the corresponding location of the text is shown in Figure 3 left. First the shortest mode "time" in the tree is right-aligned with the text to check whether the characters match from left to right. The position of initial check start is shown as “*” in the figure, which is the character “s”. The next “s” appears in the mode tinsel. According to the adaptable bad-character rules, the offset is 3. After moving, the keyword tree and the corresponding text locations are as shown in Figure 3 right.

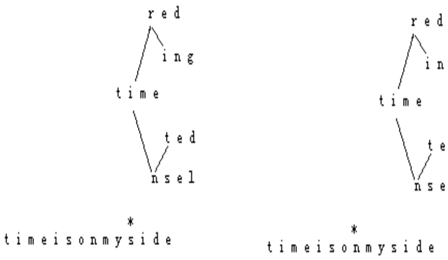

Figure 3. Bad Character Moving in ACBM Algorithm

The following example is about the good-suffix moving in the ACBM algorithm

Assuming the mode set is” time, tired, tiring, tomato, tornado” and the text is” tomatone”, the keyword tree and the corresponding location of the text is shown in Figure 4 left. First, the shortest mode “tree” in the tree is right-aligned with the text to check whether the characters match or not from left to right. The position of initial check start is shown as “*” in the Figure 4. When it failed at matching character “n” , the successful matching character “to” moves 4 characters, which make the last two characters at right-aligned with the “t0” in the text. After moving the keyword tree and the corresponding location is shown in Figure 4 right.

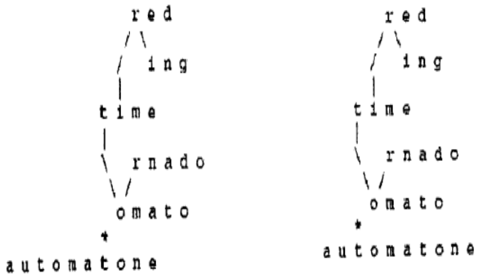

Figure 4. Good-suffix Moving in ACBM Algorithm

2.3. Data Interference Filtering

The traditional ACBM algorithm should be added classification filtering function due to the exterior interference is serious. This paper applies binary tree models to complete the classification. There are two types of binary tree classification models. Assuming there are four types of samples needed to be classified. They are named partial binary tree classification and complete binary tree classification structure and other binary trees are between two structures which are called approximate complete binary tree structure.

The process of building partial binary tree is as follows. k class training sample can train k-1 support vector machine. The first support vector machine uses the first class sample as the positive training samples and sets $2,3,\mathrm{L},K$ classes as the negative training samples to train SVM1. ith support vector machine uses ith sample as positive training samples and $i+1,i+2,\cdots,k$ class samples as negative training samples to train SVMi until k-1th uses k-1 class sample as the positive samples and k class sample as the negative samples to train SVM(k-1). For k class questions, k-1 classifiers are necessary.

How to construct a good tree structure is researched field which can be solved by binary tree algorithm forming by the distances among the classes. The basic idea is to separate the classes whose distance with other classes are longer firstly.

Definition 1: the shortest distance

The distance between the two closet samples in the class A_i and A_j is named as the distance between class A_i and A_j .

$$d_{ij} = \min\{\|x_a - x_b\| \ , x_a \in A_i, x_b \in A_j\} \tag{2}$$

The algorithm is described as follows.

Step 1: the distance $d_{ij}\ (i, j = 1,2\mathrm{L}k, i \neq j)$ is computed according to the equation (2);

Step 2: For each class, the distances among all other k-1 are computed and the distance of each class is sequenced by descending order and renumbered, such as the descending order of the distances $d_{ij}(i, j = 1,2\cdots k, i \neq j)$ between ith class and other classes is $l_i^1 \le l_i^2 \le \cdots \le l_i^{k-1}$.

Step 3: First the values of $l_i^1\ (i = 1,2,\mathrm{L}, k)$ are descending. If there are two or more than two classes have the same l_i^1 , l_i^2 should be compared and so on so forth. If their l_i^k are totally the same, the class with smaller tag number is put in front. For example, if l_i^1 < l_j^1 after comparison of ith and jth class, jth class should be in front of ith. If l_i^1 > l_j^1 , ith class should be in front of jth. If the values are the same, l_i^2 and l_j^2 should be compared and so on. Finally the sequence $n_1, n_2, \cdots n_k$ of all of the classes is attained. The binary tree clustering model is formed by $n_1, n_2, \mathrm{L}\, n_k \in \{1,2,\mathrm{L}\, k\}$.

3. Experiment and Simulation

3.1. The Selection of Experiment Data

In order to prove the proposed algorithm, KDDCup99 dataset is chosen as the experiment data. There are four types of attack in the dataset-Denial of Service (DoS) attack, User to Root (U2R) attack, Remote to Loca-tion (R2L) attack and Probing attack. The amount of the normal data in the dataset is 2000. The amount of DoS attack is 1000. The amount of U2R attack is 70. The amount of R2L attack is 20. The amount of Probe attack is 100. The training data in the normal data is 2000, DoS is 800, U2R data is 70, R2L is 80 and Probe attack is 100. The test data of the normal data is 1000, the DoS is 400, U2R is 70, R2L is 40 and Probe is 50.

3.2. Data Extraction and Pretreatment

CUP 99 released by the MIT Lincoln Laboratory is not only a model for IDS integrated test system, but also the currently most influential and credible intrusion detection data set in academic circles. MIT LL specifically offers a more practical 10% data sets for network security audit researchers, including the training set (with 494,021) and test sets (with 311,029), and containing five major kind of network attacks, namely Normal, Probe, DoS, U2R and R2L. Each record in the dataset is consisted of 41 feature attributes and one attack type token.

The data pre-process contains two steps. First step maps the character attribute to digital value; second is to describe the data. Finally all of the characteristics are transformed linearly to the values among the rang [0.0, 1.0]. The litter attributes in the integer range are duration [0, 58329], wrong_fragment [0, 3], urgent [0, 14], hot [0, 101], num_failed_logins [0, 5], num_compromised [0, 9], su_attempted [0, 2], num_root [0, 7468], num_file_creations [0, 100], num_shells [0, 5], num_access_files [0, 9], count [0, 511], srv_count [0, 511], dst_host_count [0, 255], and dst_host_srv_count [0, 255] which linearly transformed to the values in the rang [0.0, 1.0].

In order to contrast the probability distributions in horizontal between different types of attacks more intuitively and ensure the relative stability of the internal characteristics for each type of attack (if only one or two records are extracted for a certain type of attack, the condition is prone to appear that the eigenvalue is higher or lower), the extraction method for experimental data is as follows: The sample sizes of both the training sets and test sets are 600, and their samples are formed from randomly taking the largeer proportion of the 6 type of attacks, each for 100, from the 10% data set, namely normal, satan, back, smurf, apache2 and mailbomb. The original records need three steps of pretreatment: a. Translate the feature attributes with character style (such as the protocol type) into numeric style; b. Replace the attack type tags with numeric values. For example, 1 means the normal, 2 denotes the satan, and the rest may be deduced by analogy; c. Normalize the 41 characteristic values in the Matlab program, and make them fall on the interval [-1,1], as follows:

0, tcp, http, SF, 181, 5450, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 8, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 9, 9, 1.00, 0.00, 0.11, 0.00, 0.00, 0.00, 0.00, 0.00, normal.

0, 1, 22, 10, 181, 5450, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 8, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 9, 9, 1.00, 0.00, 0.11, 0.00, 0.00, 0.00, 0.00, 0.00, 1.

3.3. Experimental Results and Analysis

It shows the network intrusion stable distribution parameter values obtained by using the sample characteristic function method on table 1. It can be seen from Table 1 that, α is not equal to 2, which indicates that the intrusion detection data does not obey the Gaussian distribution, with the non-Gaussian feature; β is not equal to 0, which further validates the asymmetric distribution of the data.

Table 1. The Feature Stable Distribution PDF Parameter Estimation

Attack style/Class number	α	β	σ
normal/1	1.32	0.07	0.86
satan/2	1.13	-0.93	0.62
back/3	1.72	-1.00	1.23
smurf/4	0.84	-0.59	0.49
apache2/5	1.21	-1.00	0.65
mailbomb/6	1.36	-1.00	0.78

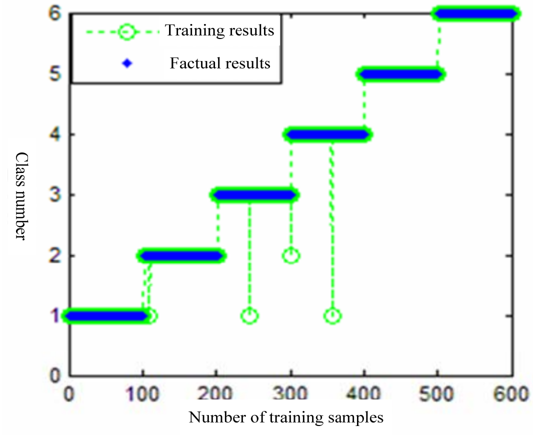

Figure 5. Intrusion Feature Training Effect Diagram

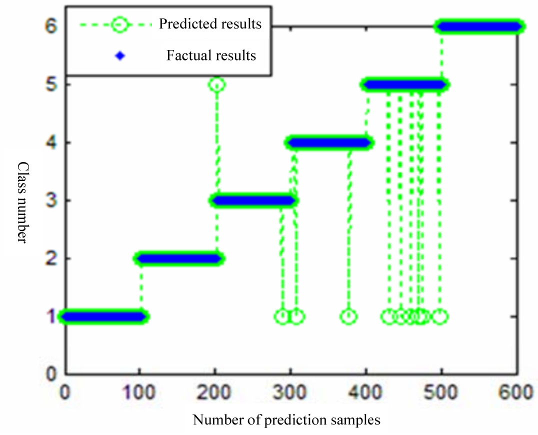

Figure 6. Feature Distribution Prediction Effect Diagram

Figure 5 is the classification result obtained by performing the substitution calculation on the training samples, and it reflects the effect of the training data. The abscissa represents the sample number, and the vertical axis denotes the classification number of the attack type. If the open circle does not overlap with the solid dot, it means the misclassification. There are 4 classification errors in Figure 5, the correct rate is up to 99.3%, and the training effect is more ideal. Figure 6 is the prediction sample renderings. There are 10 predicted sample classification errors, most of which mistake the attack behavior as the normal behavior, the correct rate of the prediction is up to 98.3%, and the rate of missing report is 1.5%, the classification ability is strong. To further validate the effectiveness of the proposed method, it is compared with the PNN and BP neural network. The results are shown in Table 2 and Table 3 respectively. The PNN smoothing factor is $\sigma = 1.5$, the maximum training time of the BP neural network is $H = 100$, and the corresponding minimum mean square error is $\varepsilon = 0.001$.

Table 2 shows the differences of the three algorithms in training effect. Compared with PNN and BP neural network, the results of the proposed training method increased 0.2% and 0.8% respectively, and the overall training effect is slightly better than PNN and BP neural network.

Table 2. Comparison of the Training Effect

Type of attack	The proposed method	PNN	BP
normal	100	100	100
satan	100	98	97
back	99	99	96
smurf	98	99	100
apache2	100	100	99
mailbomb	100	100	100
Correct rate (%)	99.5	99.3	98.7

Table 3 reflects the differences of the prediction effect of the three algorithms. The error distribution of the three algorithms is mainly concentrated in the back, smurf and apache2, and the erroneous results of both back and front smurf are in line with expectations, which is roughly the same with the training effect error distribution. However, the error of apache2 is obvious, and has a large contrast with the training effect. This is mainly because that the apache2 attack belongs to the Apache HTTP service attacks, the audit flow of which is similar with normal user traffic flow and is easy to be mistaken as normal behavior. It can be seen from this table that due to taking the feature stable distribution PDF as the basis for classification, the proposed method has some advantages compared with PNN in terms of detection rate and false negative rate, while ensuring the stability of the false alarm rate. The BP neural network has a lower recognition rate for the back attack, and it is found in the experiment that the BP neural network is easy to mistake the back attack as the smurf attack, with which the behavior feature is similar. Experimental results show that, compared with BP neural network, the proposed method has more obvious advantages, and it does not need to re-train the network when adding or deleting the samples.

Table 3. Comparison of the Prediction Effect

Type of attack	The proposed method	PNN	BP
normal	100	100	99
satan	100	91	91
back	98	90	36
smurf	98	94	98
apache2	94	89	57
mailbomb	100	100	100
Detection rate(%)	98.3	94.0	80.2
False alarm rate(%)	0.0	0.0	1.0
False negative rate(%)	1.5	3.2	10.7

Note:
Detection rate = number of records correctly identified / total number of records in the sample set;
False alarm rate = number of records of mistaking the normal behavior as aggressive behavior / total number of records of the sample set;
False negative rate = number of records of mistaking the aggressive behavior as normal behavior / total number of records of the sample set.

4. Conclusion

It is first proposed to the matching modules based on ACBM algorithm, which adds the filtering functions to optimize the modules. This method has high detection accuracy and can be widely applied in network incursion detection. Finally, the data of the KDDCup99 dataset is used to prove the superiority of the proposed method by experiment. It shows that the accuracy of the network intrusion detection method, which combines the ACBM algorithm and support vector machine, is higher than the LSSVM and SVM. It is obvious that the proposed method is quite suitable for network intrusion detection.

References

[1] Sharief MA Oteafy, Hossam S Hassanein. Resource Re-use in Wireless Sensor Networks: Realizing a Synergetic Internet of Things. *Journal of Communications.* 2012; 7(7): 484-493.

[2] Weifa Liang. Constrained Resource Optimization in Large-Scale Wireless Sensor Networks with Mobile Sinks. *Journal of Communications*. 2012; 7(7): 494-499.

[3] Dilip Krishnaswamy, Danlu Zhang, Dirceu Cavendish, Weiyan Ge, Samir S. Soliman, Bibhu Mohanty, Srinivasa Eravelli. COBA: Concurrent Bandwidth Aggregation - A Case Study in Parallel Wireless Communications. *Journal of Communications*. 2012; 7(7): 524-537.

[4] Katsuya Suto, Hiroki Nishiyama, Xuemin Shen, Nei Kato. Designing P2P Networks Tolerant to Attacks and Faults Based on Bimodal Degree Distribution. *Journal of Communications*. 2012; 7(8): 587-595.

[5] Zhongtian Jia, Lixiang Li, Zhuoran Yu, Shudong Li, Yixian Yang. A Secure Message Transaction Protocol for Delay Tolerant Networks. *Journal of Communications*. 2012; 7(8): 622-633.

[6] Luca Becchetti, Luca Filipponi, Andrea Vitaletti. Privacy support in people-centric sensing. *Journal of Communications*. 2012; 7(8): 606-621.

[7] Xiaodong Lin, Joel José PC Rodrigues, Xu Li. Guest Editorial. *Journal of Communications*. 2012; 7(8): 575-576.

[8] Subir Biswas, Jelena Mišic. Prioritized WAVE-based Parking Assistance with Security and User Anonymity. *Journal of Communications*. 2012; 7(8): 577-586.

[9] Gong Juan, Duan Shuhua. Application of Neural Network Based on Particle Swarm Algorithm for Intrusion Detection. *Computer Measurement & Control*. 2010; 18(8): 1924-1927.

[10] Zhou Lu-ping, Li Bing-rong. The study on the Intrusion Detection Algorithm Analysis using the Improved Genetic Optimized Neural Network. *JCIT*. 2013; 8(5): 571-577.

[11] Chen xiao-ming. Research on the Network Intrusion Detection Model with neural network parameters optimized by ant-colony algorithm. *JCIT*. 2013; 8(5): 619-627.

[12] Liu Jun. Research on the Network Intrusion Detection based on the Improved Immune Algorithm. *JCIT*. 2013; 8(5): 754-761.

[13] Saeid Asgari Taghanaki, Behzad Zamani Dehkordi, Ahmad Hatam, Behzad Bahraminejad. Synthetic Feature Transformation with RBF neural network to improve the Intrusion Detection System Accuracy and Decrease Computational Costs. *International Journal of Electrical and Computer Engineering*. 2012; 1(1): 28-36.

[14] Wen Yangdong, Song Yang, Wang Ying. Fault Diagnosis Method for Power Transformer Based on Fuzzy Neural Network. *Computer Measurement & Control*. 2013; 21(1): 39-41.