A Computer Virus Detecting Model based on Artificial Immune and Key Code

Zhang Li[1,*], Xie Bin[1], Lou Fang[1], He Z. Qiang[2] and Dong Z. Xin[1]

[1]Institute of Computer Application, China; [2]Academy of Engineering Physics, Sichuan, 621900 China

Abstract: Existing antivirus technology depends on extracting signatures. They are inefficient on detecting diverse forms of computer viruses, especially new variants and unknown viruses. Inspired by biological immune system, a virus detection model based on artificial immune and key-signatures extraction is proposed. This model adopt TF-IDF Algorithm to extract virus ODNS from virus DNA parts on code level, and on gene level these virus ODNs are matched by slither window to form virus candidate gene library and normal candidate gene library; then distinguish these gene through negative selection algorithm to generate a detecting virus gene library; Last on the testing procedure level, use a cosine similarity algorithm to estimate the testing procedure relevant to virus. To identify most of new variants and camouflage viruses**,** virus polymorphism is considered. Different unsteady length genes compose a virus, and a r-adjustable match rule based on RCB r-chunks is adopted to extract virus detecting library, which can mostly present virus signatures. In order to make full use of effective information and fully taking the advantages of relevance between virus genes, in procedure phase, suspicious programs are analyzed in contrast to the detecting gene matching technique, which leads to a fairly level false and positive rate.

Keywords: Artificial immune, cosine similarity algorithm, feature extraction, successive matching, TF-IDF algorithm.

1. INTRODUTION

Traditional computer anti-virus technology is based on virus feature detection, which has effective recognition to known and existing viruses. However, this kind of detection has limited recognition to unknown and mutated viruses.

Artificial immune system [1] based on Biological immune system is an important branch in the field of artificial intelligence, as well as neural network and genetic algorithm is also important method of intelligent information processing and is widely researched by increasing number of e experts. This system can distinguish “self” and “non-self” to defend against external invasion. This system in computer field can filter useless and harmful information, which is similar with the recognition of malicious code [2]. Based on Biological immune system, many represents active research results such as negative selection algorithm, clonal selection algorithm, and immune genetic algorithm are adopted in artificial immune model to detect computer virus [3-5]. Negative selection algorithm [6] can detect infinite abnormal sample programs with limited normal sample programs, but without prior knowledge and becomes the principal method. However, this algorithm in detector generation efficiency, detector length selection, generalization ability, and complexity of algorithm need lots of resources, and it also does not take full use of the correlation between multiple detectors.

Research on computer virus feature extraction, citing Deng *et al.*, the description of the characteristics of the virus:

1). Virus feature should take actual unsteady length;

2). Multiple features need to identify a virus, rather than use a virus feature;

3). Features of a virus have relevance with each other.

Considering the characteristics of the virus, this paper proposes a feature extraction method of key code based on artificial immune system which makes full use of the key characteristics of relevance. In code layer, a key code extraction model can be established to match the feature generation and storage. In gene layer, key code correlation analysis can be established to improve training set detection efficiency.

2. PREPARE KNOWLEDGE

Kephart *et al.* [7] propose virus feature extraction method, a few known virus which can infect large number of files are extracted steadily with 12 to 36 bytes areas. In these areas most distinguish virus will be selected as virus feature. Although this method does not need the help of experts and can quickly extract the features of the virus, but author believes that this method cannot apply to polymorphism of the virus.

Other effective detection method also attempts to use win32 DLL called, ASCII string or sequence of bytes as characteristics. Due to the specific training set to produce a fitting feature of the phenomenon. Henchiri and Japkowicz propose a virus feature extraction and evaluation model based on data mining [8]. This model focus on the feature of

1874-110X/14 2014 Bentham Open

different family virus classification. Data mining is adopted to distinguish internal and external family to establish detailed index of virus features.

Inspired by negative selection mechanism, Forrest proposes detection algorithm for abnormal challenge. This algorithm without non-self information is especially suitable for the unknown time-varying environment of the fault diagnosis and computer security monitoring. But, features in collection are exponential based on the "self" information. In addition, the algorithm adopts without wizard randomly generated characteristics will have a lot of useless operations, which will cause detector redundancy.

Adopting fixed length string to identify the individual problem is also irrational. Therefore, a variable fuzzy matching negative selection algorithm is proposed [9], in which a low complexity and detector selection generation algorithm is given to make the characteristic number and "self" information number into a linear relationship. This method greatly reduces the number of features, but not fundamentally solve the problem of feature.

From above point of view, virus detection model based on artificial immune system is proposed to detect unknown viruses [10]. The model uses the prior knowledge to generate the virus features regularly, overcome random selection and is only suitable for stabilization system. However, the validity of the model is influenced by feature extraction field, and unit size. Unknown factors in practice are not available while a small translation will cause failure.

For these reasons, Karnik *et al.* proposed a cosine similarity method to measure program features [11]. This method can identify form of polymorphism of file and detect viruses. This method has high measuring efficiency for unknown viruses and virus variation.

In addition, Chen qi *et al.* [12] proposed a filtering feature of the selected algorithm based on TF*IDF and also provides good ideas for virus detection.

3. VIRUS FEATURE EXTRACTION

3.1. Tendency Selection Algorithm

Although many experts, based on the negative selection algorithm proposed linear method, greedy method, template method, evolutionary method, the negative database, and r-variable improved algorithm. However, these models do not fully consider the operational features of the virus itself, and lack correlation between the viral genes. The overall non-self detection rate is not high, the rate of false positives for self has also a higher level.

To solve these problems, this paper attempts to run the mechanism of the virus itself. According to the feature of the virus in training sets, the virus code is extracted through a key extraction algorithm, which is composed of different kinds of virus gene pool, and then through the selection of threshold as control, uses the prior knowledge mining information available to the greatest extent.

Inspired by biological immune system, some academic terms are redefined in this paper:

-code DNA: Hex string in program;

-Gene: Hex string represent virus feature, as comparative item;

-Nucleotides: each two bytes Hex string represents a Nucleotides, as ODN. A quantity of Nucleotides compose a gene.

A small amount of effective key code forms the virus genes and these genes are composed of several ODNs orderly. This paper adopts the sliding window for ODN counts.

With gray pigeons virus code (text) as an example, the code of virus is as follow:

14 EB 57 FF 75 14 6A 66 56 F8 24

ODN code:

14EB EB57 57FF FF75 7514 146A 6A66 6656 56F8 F824

In order to locate and extract the key code accurately, this paper intends to introduce TF-IDF keywords localization algorithm and combines the concentration ODN of virus, to extract the key virus code from training sets.

- I_S : the number of ODN of normal program in training sets.

- I_V : the number of ODN of virus program in training sets.

- I_S^i : the number of ODNi of normal program in training sets.

- I_V^i : the number of ODNi of virus program in training sets.

- N_S : the number of normal programs in training sets.

- N_V : the number of virus programs in training sets.

- N_S^i : the number of normal programs which contain ODNi in training sets.

- N_V^i : the number of virus programs which contain ODNi in training sets.

- ODN_S^i : ODNi in normal program in training set.

- ODN_V^i : ODNi in virus program in training set.

This paper adopt TF-IDF algorithm [13] to calculate ODN_S^i and ODN_V^i, and generate TF-IDF formulas (3-1-1):

$$TF-IDF(ODNi)=\begin{cases}\dfrac{I_s^i}{\max\limits_i\left\{I_s^i\right\}}*\log\left(\dfrac{N_S}{N_S^i+1}\right), ODN\mathrm{i}\in normal-program\\ \dfrac{I_v^{\mathrm{i}}}{\max\limits_i\left\{I_v^{\mathrm{i}}\right\}}*\log\left(\dfrac{N_V}{N_v^i+1}\right), ODN\mathrm{i}\in virus-program\end{cases} \qquad (3\text{-}1\text{-}1)$$

$$concentration(\mathrm{ODNi}) = \begin{cases} W_S^{\mathrm{i}} = \dfrac{I_S^i * N_{\mathrm{S}}^{\mathrm{i}}}{\left(\sum\limits_{\mathrm{i}} I_S^i\right) * N_{\mathrm{S}}} \quad \pounds - ODN\mathrm{i} \in normal - program \\ W_V^{\mathrm{i}} = \dfrac{I_{\mathrm{V}}^i * N_{\mathrm{V}}^{\mathrm{i}}}{\left(\sum\limits_{\mathrm{i}} I_{\mathrm{V}}^i\right) * N_{\mathrm{V}}} \pounds - ODN\mathrm{i} \in virus - program \end{cases} \tag{3-1-2}$$

$$T(ODNi) = \begin{cases} TF - IDF(ODNi)^S * W_S^{\mathrm{i}} \pounds - ODN\mathrm{i} \in normal - program \\ TF - IDF(ODNi)^V * W_V^{\mathrm{i}} \pounds - ODN\mathrm{i} \in virus - program \end{cases} \tag{3-1-3}$$

Considering the discrete distribution of normal program and virus program in training sets, concentration is introduced to balance randomness of training sets, as shown in formula (3-1-2).

Tendency selection algorithm of ODNi is as shown in formulas (3-1-3), (3-1-4).

$$\begin{cases} T(ODNi)^{\mathrm{S}} > T(ODNi)^V, ONDi \in normal - program \\ T(ODNi)^{\mathrm{S}} < T(ODNi)^V, ONDi \in virus - program \end{cases} \tag{3-1-4}$$

Due to formulas (3-1-3), (3-1-4), Process of Tendency selection algorithm is as follow:

Step 1: Initialization:1 set I_S, I_V, I_S^i, N_S^i, N_V^i, N_V, N_S, I_V^i, $temp[j++]$ to zero;

Step 2: select normal program, initialize array flag[i]=0;

Step 3: Using sliding window length as 2 bytes, read ODN_S^i, calculate the value of the i;

Step 4: I_S^i++, I_S++;

Step 5: If flag[i]=0, N_S^i++, flag[i]=1, denote ODN_S^i as it appears one time in this program;

Step 6: sliding window, return to step 3; until this program is over;

Step 7: Return to step 2, until all of normal programs in training sets are over;

Step 8: Repeat above steps, instead of virus program(replace subscript s to v); until all of virus programs in training sets are over;

Step 9: $\forall i$, calculate $T(ODNi)$, if

$T(ODNi)^s = T(ODNi)^v$

then, temp[j++]=i; until $T(ODNi)$ is calculated.

3.2. Virus Detection Model

Compared to randomly detector generation, the advantages of virus ODN detector generation are as follow: First, ODN detector generation, filtered from the virus program and normal program with Tendency selection algorithm, has a higher efficiency than that of random detector; Second, from the algorithm that is given in section 3.1, ODN library generation complexity is low;

3.3. Match Rule

3.3.1. Virus Candidate Gene Generation Rule

This article adopted the candidate genes generation rule which is based on RCB rules [14], Virus ODN model uses continuous matching to match the program and then generates candidate genes of the virus. RCB matching rules refers to the two strings x and y from the same location to match by sliding window, until no longer matches. When checking the matching process, it contains many virus ODN in the ODN library, if the number of the matching is greater than T, then, denote that the matching of the virus code contains enough information, and use it as a virus candidate genes, otherwise this code does not contain enough information, and cannot be used as a candidate gene of the virus. This way of continuous matching has two bytes in ODN fault-tolerant ability.

3.3.2. Virus Detecting Genes Generation Rule

This scheme is proposed based on RCB r-chunks rules [15, 16] variable r matches rules to train virus candidate gene and generate detecting gene.

In this paper, the value of r is considered as variable, and r is related to the length of genes which can match each other. When two of genes begin to match, the length is satisfied as in formula():

$$r \geq \max\left\{3, \left(\text{larger length-1}\right) \cdot \frac{2}{3}\right\} \tag{3-3-1}$$

3.3.3. Virus Similarity Model

This paper argues that after the matching rules, virus detection gene pool contains enough virus features information. However, all of these matching activities occurred in training and optimization of individual genes lack a rationality model for the recognition of the program.

Cosine similarity of conception [17] is adopted to establish model for gene matching among virus programs. The results of similarity of virus are used as a basis to distinguish whether a program is a virus or not.

In this model, denote $\left\{G_{11}, G_{12}, ..., G_{1m}\right\}$ as virus-suspicious genes which are generated by detection programs and Denote $\left\{V_{i1}, V_{i2}, ..., V_{in_j}\right\}_{j \in [1, N_V]}^{i \in [1, N_V]}$ as virus detection genes. m is the number of genes in detecting gene, *n* is the number of genes in virus detecting gene.

Denote S_{ij} as the number of consecutive match ODNs of G_{1i} and V_{i1}. In order to give accurate measurement to calculate the similarity of two detecting programs, similarity matrix of detecting gene and virus detection are as follows:

$$\text{similarity}\quad\text{maxtrix}\ \frac{X}{T}=\begin{array}{c|cccc|} & V_{i1} & V_{i2} & & V_{in_j} \\ G_{11} & S_{1,i1}/S'_{i1} & S_{1,i2}/S'_{i2} & \cdots & S_{1,in_j}/S'_{in_j} \\ G_{12} & S_{2,i1}/S'_{i1} & S_{2,i2}/S'_{i2} & \cdots & S_{2,in_j}/S'_{in_j} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ G_{1m} & S_{m,i1}/S'_{i1} & S_{m,i2}/S'_{i2} & \cdots & S_{m,in_j}/S'_{in_j} \end{array}$$

S'_{1i} is the number of ODN in gene V_{i1}. Calculation procedures are as follow:

Step 1: Calculate

$$A_{1i}=\max\left\{S_{i,i1}/S'_{i1},S_{i,i2}/S'_{i2},\cdots,S_{i,in_j}/S'_{in_j}\right\},$$

$$A_{1i}=\max\left\{S_{i,i1}/S'_{i1},S_{i,i2}/S'_{i2},\cdots,S_{i,in_j}/S'_{in_j}\right\}.$$

Step 2: $\forall V^{i}_{n_j}\in\left\{V^{i}_{n_j}\right\}^{i\in[1,Nv]}_{j\in[1,N_v]}$, Calculate genes similarity of virus $V^{i}_{n_j}$ and virus detecting genes, as formula (3-3-2).

$$COS\theta=\frac{\sqrt{\sum_{i=1}^{m}A_{1i}\times A'_{1i}}}{\sqrt{\sum\limits_{i=1}^{m}(A_{1i})^2}\times\sqrt{\sum\limits_{i=1}^{m}(A'_{1i})^2}}\qquad(3\text{-}3\text{-}2)$$

Step 3: Denote $\max\limits_{N_v}\left\{\cos\theta_1,\cos\theta_2,...,\cos\theta_{N_v}\right\}$ as similarity of detecting programs and virus programs, compare it with S, if

$$\max_{N_v}\left\{\cos\theta_1,\cos\theta_2,...,\cos\theta_{N_v}\right\}>s$$

decide whether this program is a virus program or not. Here, the value of S is 0.5.

4. ALGORITHM ANALYSIS

4.1. Match the Time Complexity and Space Complexity

4.1.1. Complexity of Trend Selection Algorithm

Set $L_S=\max\limits_{i}Length\left\{N_{S_i}\right\}$, $L_V=\max\limits_{j}Length\left\{N_{V_j}\right\}$, Let $Length\ \ N_{S_i}$ be the length of the code of the normal program S_i. Let $Length\ \ N_{V_j}$ be its counterpart of the virus program, set T as the algorithm complexity to calculate all normal programs and virus programs, ODN generated and trend selection, the algorithm complexity meets the formula (4-1):

$$T\sim O\left(\left(L_S-1\right)\cdot N_S\right)+O\left(\left(L_V-1\right)\cdot N_V\right)\qquad(4\text{-}1)$$

4.1.2. Probability of Matching of Variable r

In this paper, we give the matching probability of candidate genes and legal virus genes:

$A=\left\{\text{normal-program-ODN}\right\}^{I_S}$

$B=\left\{\text{virus-program-ODN}\right\}^{I_V}$, $C=\left\{\text{virus-}ODN-\text{pool}\right\}^{KL}$

$D=\left\{ODNi\in\left\{A\cap B\right\}_{i=1}^{i=M}\right\}$, $E=\left\{D\cap C\right\}^{M1}$. After the trend selection, $ODN_j\in C$, Let KL be the number of ODN in the virus ODN pool, M is the number of common ODN between normal program and virus program. So anything between candidate genes and normal suspicious virus genes $\left\{A-A\cap B\|\text{C}\|\text{A}\cap\text{B-C}\right\}$, it can be seen that each of genes consists of three parts:

1). Its own ODN cluster;

2). Virus ODN cluster;

3). Remove common ODN cluster of virus ODN cluster.

Thus it is concluded that $\left\{B-A\cap B\|\text{C}\|\text{A}\cap\text{B-C}\right\}$, satisfying the candidate genes and legal procedures virus gene matching probability P1 of variable r, matching rules meets the formula (4-2):

$$\frac{C^{r}_{M1}}{C^{r}_{KL}}\leq P1\leq\sum\nolimits_{l=r}^{\min\{G_V-1,G_S-1\}}\frac{C^{l}_{M1}}{C^{r}_{KL}}\qquad(4\text{-}2)$$

4.1.3. Analysis of Similarity and Detection Rate

Similarity Matrix model is given in the paper, by comparing the virus gene pool and each gene in the virus detection to be detected in the gene pool similarity value, to determine whether the program to be detected is a virus.

Obviously, the similarity value is proportional to each element value in the similarity matrix and is inversely proportional to the length of the gene in the virus detection gene pool. By the formula (3-3-1), the similarity value S satisfies the inequality (4-3);

$$\frac{\sqrt{\sum_{i=1}^{m}A_{1i}{}^{T}\times A'_{1i}}}{\sqrt{\sum\limits_{i=1}^{m}(A_{1i}{}^{T})^2}\times\sqrt{\sum\limits_{i=1}^{m}(A'_{1i})^2}}\leq S\leq 1\qquad(4\text{-}3)$$

where, $A_{1i}{}^{T}=r/S'_{i1}+r/S'_{i2}+\cdots r/S'_{in_j}$ 。

Set P2 as the detection rate of the virus program. Taking in account that a program to be detected is misidentified as legal, the procedure is as follows: the program to be detected and the virus ODN pool generated class virus gene pool and all virus detection of virus samples and not any one gene in the gene pool match, namely $\left\{\cos\theta_j\right\}^{j\in[1,N_V]}<S$, calculate

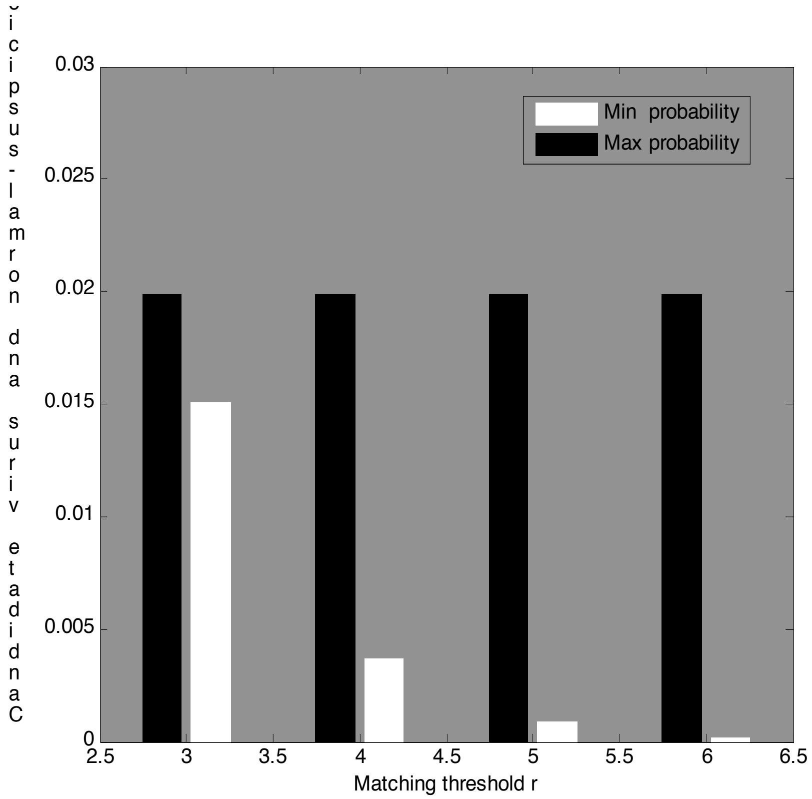

Fig. (1). Candidate virus genes with normal-suspicious virus gene matching probability.

the probability of the event P3, by the formula $\{G_{1i}\}^{i\in[1,m]} \in \{V_{i,n_j}\}_{j\in[1,N_V]}^{i\in[1,N_V]}$, formula (4-4) is established.

$$\begin{cases} P3 = P\left\{\{\cos\theta_j\}^{j\in[1,N_V]} < S\right\} \\ P2 = 1 - P3 = 1 - P\left\{\{\cos\theta_j\}^{j\in[1,N_V]} < S\right\} \end{cases} \tag{4-4}$$

This paper collected 800 viruses and 1200 legal procedures, according to their respective properties. They are divided into 245 kinds of viruses. Through the statistical analysis, the simulation results are presented:

Fig. (**1**) shows matching probability between the candidate gene pool and legal virus genes, it can be seen that after virus ODN generated by a candidate gene pool and legal virus gene pool there still exists certain intersection.

Fig. (**2**) shows the relationship between the matching threshold r and similarity threshold S. As the threshold r increases, the similarity threshold S also will increase. The similarity model given in the conclusion conforms to the virus.

Fig. (**3**) shows the intersection between virus program ODN and legal program ODN, the intersection part is related to the samples space concentrated in the training set.

Fig. (**4**) shows the correct detection rate of program, with the increasing of similarity threshold S, the detection rate of legal procedure increases, but the detection rate of virus program will decrease. The main reason is that the similarity threshold S is inversely proportional to the value r and matching value, but r is proportional to the virus detection rate. Therefore, with the increasing of S, virus detection rate falls.

The method was applied to this scenario, compared with the algorithm of literature [18].

1). Literature [18] ignores the gene frequency of the same virus, it is easy to cause the variations of the virus to identify the effectiveness of the shortage, thus to calculate the similarity values; This scenario uses gene comprehensive similarity values and calculates the program of a particular gene to be detected and the sum of all genes similarity to each type of virus as the genetic similarity, well balanced frequency problem;

2). As well as using the maximum as the judge whether the program has viruses, because its core code is crucial. There is only a few and often hidden in a large number of data. There is plenty of virus variation and the unknown virus detection is very difficult. Maximum value evaluation method can maximize the ability to detect and mine programs that contain virus features.

The matching similarity threshold value S of the last section is in chapter 3.4.3. Considering this is related to the coverage of virus sample program and the number of detection sets, in this case, to balance the comprehensiveness of detection accuracy, misjudgment rate and false negative rate, it needs to constantly add new virus samples from training set.

Give a more appropriate threshold and update the virus detection gene pool to improve S value.

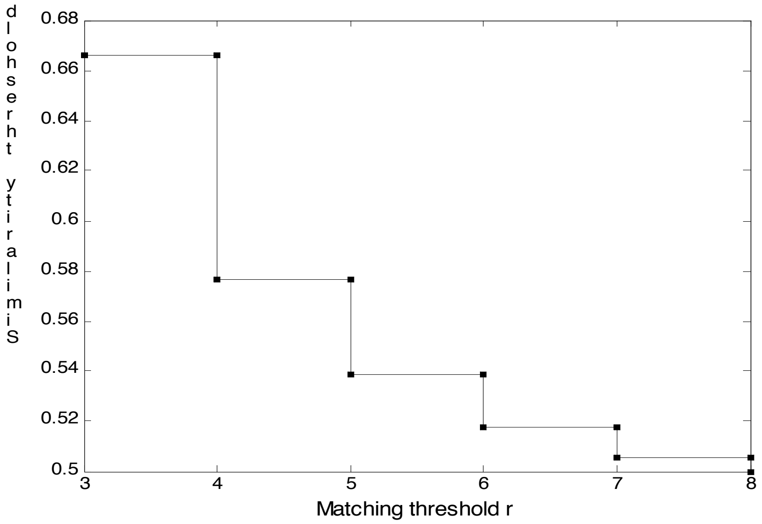

Fig. (2). Matching threshold r and similarity threshold S.

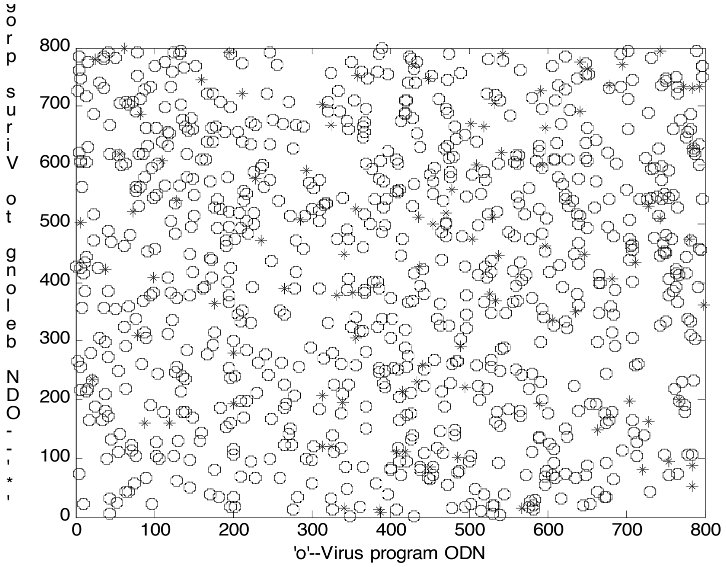

Fig. (3). Virus program ODN and normal ODN distribution.

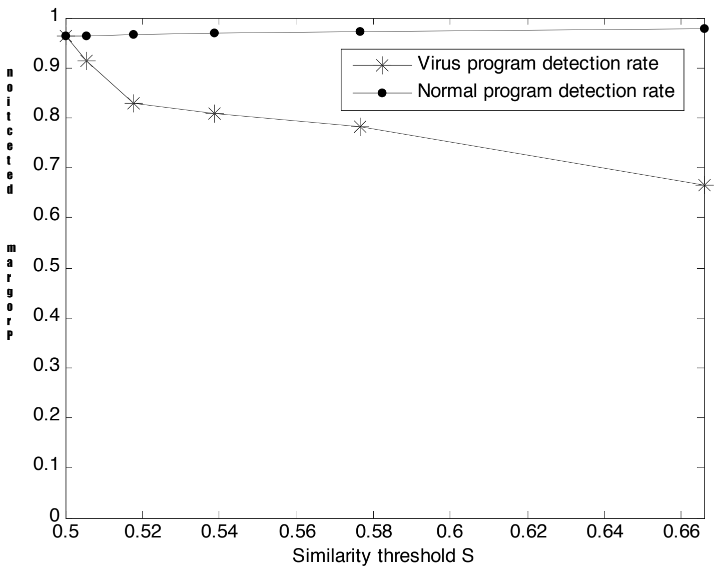

Fig. (4). Program detection rate.

CONCLUSION

On the basis of previous studies, this paper mainly discusses the deficiency of existing virus detection technology under the new situation, and to efficiently extract the virus characteristic code, thus improving the efficiency of the recognition of virus mutation and the unknown. In this paper, based on the deficiency of the prior knowledge of positing detection technology in the identification code to replace, insert redundant code, crossover operation and mutation and unknown virus code, a hierarchical code extraction and detection methods is proposed. Through the establishment of ODN trend selection model generation gene pool implementing code layer, and by gene layer continuous implementation the distinction on the gene layer is achieved. At last implementing the distinction on the sample procedure layer through the similarity model and the entire process is to effectively control the number of gene pool to avoid the redundant gene pool and improve the efficiency of the recognition of virus mutation and the unknown.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

We wish to acknowledge China Academy of Engineering Physics, expansion funds (2012A0403021).

REFERENCES

[1] H.W. Mo, *The Principles and Applications of Artificial Immune System*, Harbin Institute of Technology Press: Harbin, 2002.

[2] T. Li, *Computer Immunology*. Publishing House of Electronics Industry: Beijing, 2004, (in Chinese).

[3] C.M. Ou, "Host-based intrusion detection systems adapted from agent-based artificial immune systems", *Neuro Computing*, vol. 88, no. 1, pp.1-9, 2011.

[4] J. Zeng, X.J. Liu, T. Li, G. Li, H. Li, and J. Zeng, "A novel intrusion detection approach learned from the change of antibody concentration in biological immune response", *Springer Applied Intelligence*, vol. 35, no. 1, pp. 41-62, 2011.

[5] Y.Z. Li, C.W. Jing, and J. Xu, *A New Distributed Intrusion Detection Method Based on Immune Mobile Agent*, Springer, Berlin: pp. 233-243, 2010.

[6] S. Forrest, A.S. Perelson, and L. Allen, "Self-nonself discrimination in a computer," In: *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy IEEE*. CA: Los Alamitos, pp. 221-231, 1994.

[7] Z.Z. Jin, M.H. Liao, and G. Xiao, "Survey of negative selection algorithms", *Journal on Communications*, vol. 34, no. 1, pp. 159-170, 2013.

[8] O. Henchiri, and N. Japkowicz, "A feature selection and evaluation scheme for computer virus detection," In: *Proceedings of the 6th International Conference on Data Mining (ICDM06)*, Hong Kong, China, 2006: pp. 891-895.

[9] S. Forrest, and S.A. Hofmeyr, *Immunology as information processing, Design Principles for the Immune System and Other Distributed Autonomous Systems*. USA: Oxford University Press, USA, pp. 361-387, 2000.

[10] H. Lee, W. Kim and M. Hong, "Artificial immune system against viral attack," In: *Proceedings of the ICCS 2004. Lecture Notes in Computer Science 3037*. Krakow, Poland, pp. 499-506, 2004.

[11] J. Greensmith, J. Twycross, and I.U. Aickel, "Dendritic cells for anomaly detection," In: *IEEE Congress on Evolutionary Comp utation (CEC2006)*. Vancouver, Canada, 2006.664-671.

[12] Q. Chen, Z.S. Wu, and F. Yao, "Improved feature selection algorithm in spam filtering based on TF*IDF", *Application Research of Computers*, vol. 26, no. 6, pp. 2165-2168, 2009.

[13] P. Deepak, S. Parameswaran, "Spam filtering using spam mail communities," In: *Proceedings of IEEE SAINT# 05. [S. .l]*, IEEE Press, 2005, 377383.

[14] Hou H Y, and Dozier G. "An evaluation of negative selection algorithm with constraint based detector". *ACM Southeast Regional Conference 2006*, Melbourne, Florida, USA, 2006, pp. 134-139.

[15] J. Balthrop, F. Esponda, S. Forrrest, and M. Glickman, "Coverage and generalization in an aritificial immune system," In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, New York, USA, pp. 3-10, 2002.

[16] C. Xiao, Z.X. Cai, and Y. Wang, "Application of evolutionary strategy to negative selection algorithm", *Journal of Chinese Computer Systems*, vol. 29, no. 11, pp. 2091-2094, 2008.

[17] K. Anna, "A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes", *Data Mining and Knowledge Discovery*, vol. 20, no. 2, pp. 259-289, 2010.

[18] W. Wang, P.-T. Zhang and Y. Tan, "A feature extraction method of computer viruses based on artificial immune and code relevance", *Chinese Journal of Computers*, vol. 34, no. 2, pp. 204-215, 2011.

Received: November 01, 2014 Revised: December 15, 2014 Accepted: December 16, 2014