An Immune-Based Model for Computer Virus Detection

Tao Li[1], Xiaojie Liu[1], and Hongbin Li[2]

[1] Department of Computer Science,
Sichuan University, Chengdu 610065, China
litao@scu.edu.cn

[2] Department of Electrical and Computer Engineering,
Stevens Institute of Technology,Hoboken, NJ07030, USA
hli@stevens.edu

Abstract. Inspired by biological immune systems, a new immune-based model for computer virus detection is proposed in this paper. Quantitative description of the model is given. A dynamic evolution model for self/nonself description is presented, which reduces the size of self set. Furthermore, an evolutive gene library is introduced to improve the generating efficiency of mature detectors, reducing the system time spending, false-negative and false-positive rates. Experiments show that this model has better time efficiency and detecting ability than the classical model ARTIS.

1 Introduction

As the fast development of Internet, the generating and spreading speed of new computer viruses is getting higher and higher. Then, computer viruses and worms are becoming an increasing problem in the world [1,2]. Therefore, it is necessary to detect and eliminate computer viruses, especially the unknown viruses, in real-time. However, it is very difficult for traditional preventing methods [3-5] to solve this problem effectively. In recent years, researchers have taken some researches on the computer network topologies and the spreading mechanism of computer viruses [6-8], then presented some methods to restrain virus spreading [9-11]. These methods can reduce the speed of virus spreading, however, they can not prevent virus spreading [11]. Especially, the problem for unknown virus detection is still not solved.

The problems found in computer security systems are quite similar to the ones encountered in Biological Immune Systems (BIS). BIS has successfully solved the problem of unknown virus detection [12]. Therefore, Artificial Immune System (AIS) [13-15] is considered as a new way to defeat fast-proliferating computer viruses. In 1994, Forrest presented a method of computer virus detection based on the negative selection algorithm[16], which is the first time to use immune mechanism for virus detecting and has greatly promoted the research of computer virus immune system (CVIS). The most important works should be the general

Y.G. Desmedt et al. (Eds.): CANS 2005, LNCS 3810, pp. 59–71, 2005.

framework ARTIS for AIS and the computer virus immune model proposed, respectively, by Hofmeyr [17,18] and Kephart [19,20]. In ARTIS, the concepts and mechanisms of BIS, including self, nonself, self tolerance, immune cell (detectors), memory cell (memory detectors), and costimulation were well simulated. Many CVISs are mainly derived from ARTIS. For example, the computer virus detection system proposed by Okamoto and Ishida [21], the agent based computer virus immune architecture proposed by Harmer [22], and the HMM [23] based computer immune model proposed by Jensen [24]. Different from ARTIS, the computer virus immune model [19,20] proposed by IBM laboratory uses only partial immune mechanisms, however, some other techniques such as automatic extraction of computer virus signatures [19], virus trap [20], etc. have also been adopted.

There are three major defects in the present CVISs: The first is that the self set is very large in size. For example, during the experiments of LISYS [25], a famous application of CVIS based on ARTIS, Hofmeyr and his colleagues collected over 2.3 million self elements in 50 days.The cost for mature detector training is exponentially related to the size of self set [22], making it impossible to directly collect self data from the network for the self tolerance of immature detectors. LISYS has to aim at the detection of 7 kinds of network intrusions, where the services provided by the network, as well as the normal network activities, were simplified in order to decrease the size of self set. After laborious and complicated classification, Hofmeyr finally selected over 3900 elements as self for the tolerance process of the detectors, reducing the training cost for the tolerance of detectors. However, the computation cost is still high.

The second deficiency is that the definitions of self and nonself in the system are described in a static way with almost no changes. However, it is very difficult to use a fixed definition for self and nonself in most practical applications. Furthermore, the roles of self and nonself may exchange at times, e.g., the legal network behaviors today may be dangerous tomorrow, and vice versa. Therefore, it is necessary to update the definitions of self and nonself from time to time. The static description model for self/nonself lacks the adaptability, and thus cannot cater for the network monitoring in the real network environment.

The third, the absence of rigorous quantitative descriptions in most presented CVIS models results in the randomicity of CVIS implementation. Therefore, it is not convenient to put these models into practical applications.

The above three problems have become the major obstacles to CVIS applications. Inspired by biological immune systems, a new immune-based model for computer virus detection is proposed in this paper. Quantitative description of the model is given. A dynamic evolution model for self/nonself description is presented, which reduces the size of self set. Furthermore, an evolutive gene library is introduced to improve the generating efficiency of mature detectors, reducing the system time spending, false-negative and false-positive rates. Experiments show that this model has better time efficiency and detecting ability than the classic model ARTIS.

2 Proposed Theoretical Models

Given problem domain Ω, where $\Omega = \{0,1\}^l$, l is a natural number. Antigens[1] (Ag,$Ag \subset \Omega$) are defined as binary strings composed of program characteristics, and is divided into two set: *Self* and *Nonself*, such that $Self \cup Nonself = Ag, Self \cap Nonself = \Phi$, where *Self* is the normal program characteristic set, and *Nonself* is the program characteristic set infected by virus, respectively. The task of a virus detection system is to classify an input pattern $x \in Ag$ as either *Self* or *Nonself*. This detection methodology can generate two types of errors: false-positive error and false-negative error. A false-positive error occurs when a member of *Self* set is incorrectly classified as malicious. Conversely, a false-negative error is the classification of a member of *Nonself* set as benign. Given detector set $B = \{< a, age, count > |a \in \{0,1\}^l \wedge age, count \in Z^+ \wedge age \leq max\_age\}$, where a is antibody, l is the length of antibody a, age is the detector age, $count$ is the detector affinity, and max_age is the upper limit of the detector age. B is divided into immature, mature and memory detectors. Immature detectors are newly generated ones given by $I = \{x|x \in B \wedge x.age < \lambda \wedge x.count = 0\}$, where λ is tolerance period. Mature detectors are the ones that are tolerant to *Self* but not activated by antigens, and given by $T = \{x|x \in B \wedge \lambda \leq x.age < max\_age \wedge x.count < \varepsilon \wedge \forall y \in Self\ (f_{match}(x.a, y) = 0)\}$, where the lifecycle of mature detector is from λ to max_age, ε is the activation threshold, f_{match} is the matching function based on the affinity between the detector and an antigen: if the affinity is greater than a specified threshold, then 1 is returned, otherwise, 0 is returned. Memory detectors evolve from mature ones that accumulate enough affinity in their lifecycle, and given by $M = \{x|x \in B \wedge x.age = max\_age \wedge \forall y \in Self\ (f_{match}(x.a, y) = 0)\}$. Given antibody gene library $G = \{0,1\}^{[l/4]}$, where l is the antibody length of detectors.

Fig. 1 illustrates the framework of our proposed model. Antigens (Ag) are binary strings, having the program characteristics in a computer system. This model serves to classify an input set (Ag) into self (Ag_{Self}) and nonself ($Ag_{Nonself}$) by mature and memory detectors.

The new immature detectors, which are generated from antibody gene library through some evolutionary strategies (e.g., gene edit, genetic operator, etc.), have to experience a self tolerance period: the detector will be eliminated if it matches any self antigens (negative selection). The immature detectors that survived in self tolerance period will evolve into mature ones, there the mature detectors have a fixed lifecycle: the detectors will be eliminated if they do not accumulate enough affinity in their lifecycle; they will be activated if they get enough affinity, i.e., viruses are found. However, the activated detectors will be eliminated if they do not receive co-stimulation, i.e., false positive error, there the detected antigens are self elements. Meanwhile, the acti-

[1] The classification method of antigens used in this paper is the one in the academic immunology, which means antigens are classified into self antigen and nonself antigen, called self and nonself for short.

vated detectors will evolve into memory ones with the help of co-stimulation, there the detected antigens are sure nonself elements. The memory detectors have an infinite lifecycle, and will be activated as soon as they match an antigen.

When a detector (e.g., a memory detector, or a mature one) detects a virus, it will also clone itself and create a lot of similar detectors to protect the system against similar virus infection. In each step, our proposed model will delete the mutated self antigens from *Self* set in time through the dynamic description of self. The tolerance of immature detectors to mutated self antigens is thus prevented. Therefore, the false-negative error rate is reduced. Furthermore, the false-positive error rate is also reduced by adding new self antigens into *Self.* As the self set is dynamically defined, the immune tolerance in our model is also called dynamic tolerance.

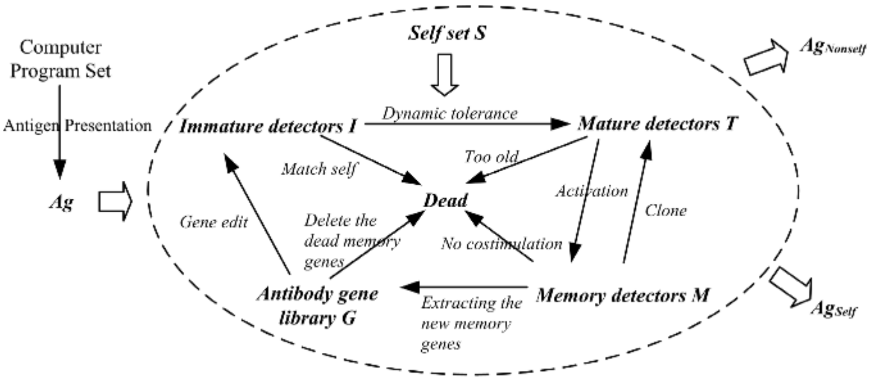

Fig. 1. The framework of our proposed model

In the following sections, the self set, antibody gene library, immature detector set, mature detector set, memory detector set, and antigen set, are, respectively, described in a quantitative way of set algebra.

2.1 The Evolution of Self

$$S(t) = \begin{cases} S_{first}, & t = 0 \\ f_{s\_lim}((S(t-1) \cup S_{del}(t)) \cup S_{new}(t)), & t > 0 \end{cases} \tag{1}$$

where $S(t), S(t-1) \subset Self$ are, respectively, indicate the self-set at time t and t-1. S_{first} is the initial self set. f_{s_lim} is a function used to limit the number of self set: if the number of self set is larger than a given value *max_s_size*, the least resent used self antigen is selected and discarded, and this procedure continues until the size of self set is equal to *max_s_size*. $S_{del}(t)$ are the mutated self antigens discarded at time t, which includes three parts: 1) the unloaded software; 2) the elements recognized by new memory detectors; 3) the elements infected by viruses. $S_{new}(t)$ are the new self antigens (e.g., loading new software) added into self set at time t.

2.2 The Evolution of Antibody Gene Library

$$G(t) = \begin{cases} G_{first}, & t = 0 \\ (G(t) - G_{dead}(t)) \cup G_{new}(t), & t > 0 \end{cases} \tag{2}$$

where $G(t), G(t-1) \subset G$ are, respectively, the antibody gene library at time t and t-1. G_{first} is the initial gene-library, $G_{dead}(t) = \bigcup_{x \in M_{dead}(t)} \{f_{g\_ext}(x)\}$ are the genes eliminated at time t, where $M_{dead}(t)$ are the dead memory detectors which cause false-positive error. $G_{new}(t) = \bigcup_{x \in T_{cloned}(t)} \{f_{g\_ext}(x)\}$ are some excellent genes added into the gene-library at time t, where $T_{cloned}(t)$ are the activated mature detectors[2] at time t, $f_{g\_ext}(x)(x \in B)$ is a function used to extract genes from a given detector x. The antibody gene-library is used to generate immature detectors more efficiently, since the new immature detectors, which are generated from antibody gene library through some evolutionary strategies (e.g., gene edit, genetic operator, etc.), have a higher probability to go through the self-tolerance than those generated randomly.

2.3 The Evolution of Immature Detectors

$$I(t) = \begin{cases} \Phi, & t = 0 \\ (f_{age\_crt}(I(t-1)) - (I_{untolerance}(t) \cup I_{matured}(t))) \cup I_{new}(t), & t > 0 \end{cases} \tag{3}$$

$$I_{untolerance}(t) = \{x | x \in f_{age\_crt}(I(t-1)) \wedge \exists y \in S(t-1)(f_{match}(x.a, y) = 1)\} \tag{4}$$

$$I_{matured}(t) = \{x | x \in f_{age\_crt}(I(t-1) - I_{untolerance}(t)) \wedge x.age > \lambda\} \tag{5}$$

Equation (3) simulates the lymphocytes growth in the marrow, where the immature detectors have to pass through the negative selection (see Eq.(4)), and undergo $\lambda(\geq 1)$ tolerance period steps of tolerance to evolve into mature ones. $I(t), I(t-1) \subset I$ are, respectively, the immature detector set at time t and t-1. $f_{age\_crt}(I(t-1))$ is to increase the age of each detector in $I(t$-1$)$ by 1. $I_{untolerance}(t)$ are the immature detectors which do not tolerate the self antigens. $I_{matured}(t)$ are the new mature detectors. $I_{new}(t)$ are the newly generated immature detectors. The generation of $I_{new}(t)$ is based on the antibody gene library G, where the key step is to generate the antibodies of detectors. The newly generated antibodies of immature detectors are usually composed of two parts: some antibodies are generated randomly, the others are derived from G, while the deriving methods include gene edit, genetic algorithm, etc.

[2] The viruses detected by mature and memory detectors are, respectively, new viruses and known ones.

2.4 The Evolution of Mature Detectors

$$T(t) = \begin{cases} \Phi, & t = 0 \\ (T_{prod}(t) - (T_{dead}(t) \cup T_{cloned}(t))) \cup I_{matured}(t) \cup T_{permutation}(t), & t > 0 \end{cases} \quad (6)$$

$$T_{prod}(t) = f_{age\_crt}(f_{count\_crt}(T(t-1), Ag(t-1))) \quad (7)$$

$$T_{dead}(t) = \{x | x \in T_{prod}(t) \land x.age = max\_age \land x.count < \varepsilon\} \quad (8)$$

$$T_{cloned}(t) = \{x | x \in (T_{prod}(t) - T_{dead}(t)) \land x.count \geq \varepsilon\} \quad (9)$$

$$T_{permutated}(t) = f_{clone\_mutation}(T_{cloned}(t) \cup M_{cloned}(t)) \quad (10)$$

where $T(t), T(t-1) \subset T$ are, respectively, the mature detector set at time t and t-1. $T_{prod}(t)$ refers to the detectors evolving into the next generation detectors, there the age and affinity of the detectors are increased. $T_{dead}(t)$ is the set of mature detectors that have not accumulate enough affinity ($\varepsilon > 0$) in their lifecycle or are activated with no co-stimulation at time t. $T_{cloned}(t)$ is the set of mature detectors activated by antigens. $T_{matured}(t)$ is the set of newly matured detectors. $T_{permutated}(t)$ is the set of clone detectors generated by the cloning of activated detectors. $f_{count\_crt}(X, Y)(X \subset B, Y \subset Nonself)$ is used to accumulate the affinity of each detector in X, where the affinity of detector $x \in X$ is increased by $|\{y | y \in Y \land f_{match}(x.a, y) = 1\}|$. $f_{clone\_mutation}(A)(A \subset B)$ is a clone and mutation function, where each element $x \in A$ will clone $\lceil \theta * x.count \rceil$ ($\theta > 0$) new detectors, and, the new clone detectors will undergo a process of mutation, there the mutation operation is to reedit the gene of the detector. $M_{cloned}(t)$ refers to equation (13). The evolution of mature detectors simulates the primary response in BIS, whereas the clone selection mechanism and the gene edit give the proposed model the learning ability.

2.5 The Evolution of Memory Detectors

$$M(t) = \begin{cases} M_{first}, & t = 0 \\ (M(t-1) - M_{dead}(t)) \cup f_{age\_set}(T_{cloned}(t)), & t > 0 \end{cases} \quad (11)$$

$$M_{dead}(t) = \{x | x \in M(t-1) \land \exists y \in S(t-1)\ f_{match}(x.a, y) = 1\} \quad (12)$$

$$M_{cloned}(t) = \{x | x \in M(t-1) \land \exists y \in Ag(t-1)\ f_{match}(x.a, y) = 1\} - M_{dead}(t) \quad (13)$$

where $M(t), M(t-1) \subset M$ are, respectively, the memory detector set at time t and t-1. M_{first} is the initial memory detector set. $M_{dead}(t)$ are the memory detectors which recognize self antigens (false-positive error) and need to be eliminated. f_{age_set} is used to set the age of new memory detectors ($T_{cloned}(t)$) to max_age. $M_{cloned}(t)$ are the activated memory detectors at time t.

Similar to BIS, our model has two types of immune response for antigens: the primary response and secondary response, which are, respectively, performed by mature detectors and memory ones. The primary response performed by mature detectors requires a relatively long period of time for learning: firstly, some time is needed to generate suitable immature detectors; secondly, these detectors have to undergo λ steps of tolerance period for evolving into mature detectors; thirdly, they will not be activated until they accumulate adequate affinity. Therefore, the primary response has a lower efficiency. During this learning process, those detectors, which play no effective function in classifying antigens, will be killed. However, those superior detectors that have a good effective function in classifying antigens will be reserved and evolve into memory ones. Therefore, similar antigens will be detected quickly when they intrude the system again. The secondary response, issued by memory detectors, is prompt, robust, and needs no learning process, i.e. a memory detector will be activated immediately once it matches with an antigen.

2.6 Antigen Detection

$$Ag(t) = \begin{cases} Ag_{first}, & t = 0 \\ (Ag(t-1) - Ag_{checked}(t)) \cup Ag_{new}, & t > 0 \end{cases} \tag{14}$$

$$Ag_{Nonself}(t) = \{x | x \in Ag_{checked}(t) \wedge \exists y \in (T_{cloned}(t) \cup M_{cloned}(t)) \\ (f_{match}(y.a, x) = 1)\} \tag{15}$$

$$Ag_{Self}(t) = \{x | x \in Ag_{checked}(t) \wedge \forall y \in (M(t) \cup T(t))(f_{match}(y.a, x) = 0)\} \tag{16}$$

Where $Ag(t), Ag(t-1) \subset Ag$ are, respectively, the antigen set at time t and t-1. Ag_{new} are the new antigens collected at time t. Ag_{first} is the initial antigen set. $Ag_{checked}(t)$ are the antigens detected by mature or memory detectors at time t, where $Ag_{Nonself}(t)$ and $Ag_{Self}(t)$ are, respectively, detected as self and nonself antigens.

3 Performance Analysis

Suppose the program number in a computer system is N_p, the average proportion of nonself antigens in the system is $\rho_N (0 < \rho_N < 1)$, the size of self set is $|S|$, the size of mature detector set is $|T|$, the size of memory detector set is $|M|$, the active threshold is ε , the probability of a detector matching an antigen is P_m, and $P(A)$ is the probability of event A.

Theorem 1. *Given P_n the probability that a detector matching a self antigen which is not listed in the self definition, such that $P_n = (1 - P_m)^{|S|} \bullet [1 - (1 - P_m)^{\lceil N_p \bullet (1-\rho_N) \rceil - |S|}]$.*

Proof. Suppose A is the event that a detector does not match any self antigen, B is the event that a detector matches at least one self antigen which is not listed in the self definition. From (3), (4), and (5), we have $P_n = P(AB)$. As events A and B are independent each other, so $P(AB) = P(A)P(B)$. Suppose X is the number of a detector matching an antigen in event A, from [30] we have $X \sim b(n, p)$, where $n = |S|, p = P_m$. Therefore, $P(A) = P(X = 0) = (P_m)^0 (1 - P_m)^{|S|} = (1 - P_m)^{|S|}$. Furthermore, suppose Y is the number of a detector matching an antigen in event B, $Y \sim b(n,p)$, where $n = N_p(1-\rho_n) - |S|, \rho = P_m$. Then, $P(B) = 1 - P(Y=0) = 1 - (1-P_m)^{\lceil N_p \bullet (1-\rho_N) \rceil - |S|}$], so $P_n = (1-P_m)^{|S|} \bullet [1-(1-P_m)^{\lceil N_p \bullet (1-\rho_N) \rceil - |S|}]$. □

Theorem 2. *Given a randomly selected nonself antigen x, the probability of which is correctly recognized is* $P_r = 1-(1-P_m)^{[|M|+|T|(1/\varepsilon)](1-P_n)} \approx 1 - e^{-P_m[|M|+|T|(1/\varepsilon)](1-P_n)}$.

Proof. Suppose A is the event that x matches the detectors, including memory detectors and mature ones. From (15), we have $P_r = P(A)$. Let X be the number of a detector matching an antigen in event A, from [30] we have $X \sim b(n,p)$, where n is number of the really used detectors for detecting the nonself antigens. Suppose the stimulate level of mature detectors is between 0 and $\varepsilon - 1$ [12], then the number of really used mature detectors is $|T|/\varepsilon$. As the detectors which recognize self antigens are not considered, so the total number of the really used detectors for detecting the nonself antigens is $n = (|M| + |T|/\varepsilon)(1-P_n)$, where P_n is shown in Theorem 1, and $p = P_m$. Therefore, $P_r = P(A) = 1 - P(X = 0) = 1-(1-P_m)^{[|M|+|T|(1/\varepsilon)](1-P_n)}$. According to Poisson theorem [30], $P_r \approx 1 - e^{-P_m(|M|+|T|(1/\varepsilon))(1-P_n)}$, when P_m is very small and $(|M| + |T|/\varepsilon)(1-P_n)$ is very big. □

Theorem 3. *Given a randomly selected nonself antigen x, the probability of which is classified as a self antigen by mistake* $P_{neg} = (1-P_m)^{(|M|+|T|)(1-P_n)} \approx e^{-P_m(|M|+|T|)(1-P_n)}$ *. Given a randomly selected self antigen y, the probability of which is classified as a nonself antigen by mistake* $P_{pos} = 1-(1-P_m)^{(|M|+|T|(1/\varepsilon))P_n} \approx 1 - e^{-P_m(|M|+|T|(1/\varepsilon))P_n}$.

Proof. Suppose A is the event that x does match any memory and mature detectors, B is event that y matches the memory detectors or mature ones. From (15) and (16), $P_{neg} = P(A), P_{pos} = P(B)$. Let X be the number of a detector matching a nonself antigen in event A, from [30] we have $X \sim b(n,p)$, where $n = (|M| + |T|)(1-P_n)$ is number of detectors which recognize nonself antigens, and $p = P_m$. Then, $P_{neg} = P(A) = P(X=0) = (1-P_m)^{(|M|+|T|)(1-P_n)}$. According to Poisson Theorem [30], $P_{neg} \approx e^{-P_m(|M|+|T|)(1-P_n)}$, when P_m is very small and $(|M|+|T|)(1-P_n)$ is very big. Furthermore, suppose Y is the number of a detector matching a self antigen in event B, where $Y \sim b(n,p), n = (|M|+|T|/\varepsilon)P_n$ is the number of detectors which recognize self antigens, $p = P_m$. From the same way of P_{neg}, we have $P_{pos} = 1-(1-P_m)^{(|M|+|T|(1/\varepsilon))P_n} \approx 1 - e^{-P_m(|M|+|T|(1/\varepsilon))P_n}$. □

Theorem 4. *The number of self set is less than a constant, and the description of self is macroscopically complete.*

Proof. From equation (1), we have that the number of self set is always less than a constant *max_s_size*. Although a few of self elements are collected by the dynamic model for self description (i.e., section 2.1) in each step, however, $\bigcup_{t=0}^{\infty} S(t)$ will cover the whole self space as time goes on. In other words, we have that the description of self is macroscopically complete. □

4 Simulations and Experimental Results

A fixed length binary string (l=128) is used as the pattern characteristics of software. IBM lab shows that the characteristic code with 128bit long is enough [19], furthermore, the 128bit long characteristic code is become an industry standard [22]. The length of antibody is also 128bit. The self set is defined as 200 important system files. The experiment aims at the detection of 100 computer viruses, and the antigen set is formed by 200 self files and 200 files infected by experimental viruses. In the experiments, the parameter λ and max_age are, respectively, set to 5 and 15. And the matching function is defined by

$$f_{match}(x,y) = \begin{cases} 1, f_{h\_dis}(x,y)/min(l_x,l_y) \geq \beta \\ 0, otherwise \end{cases} \tag{17}$$

where $\beta > 0$ is threshold, $f_{h\_dis}(x,y)$ is the Hamming distance [15] and given by

$$f_{h\_dis}(x,y) = \sum_{i=1}^{l} \delta_i \tag{18}$$

where $\delta_i = \begin{cases} 1, & y_i = x_i \\ 0, & otherwise \end{cases}, \ 1 \leq i \leq l.$

Fig.2 and Fig.3 show how parameter β affects the performance of the model, where $|M| = 50, |S| = 100$. The results show that the smaller the β, the stronger the recognition ability of the model, and the lower the false-negative rate, however, the higher the false-positive rate. The results fit to Theorem 2 and 3.

The false-negative rate is mainly caused by the size of initial memory detector-set (please refer to Fig.4). Although the β increasing will result in the increasing of false-negative rate, however, it will not exceed 50%. According to Fig.2 and Fig.3, we set β=0.8.

Fig.4 and Fig.5 show how parameter $|M|$ affects the performance of the model, where $\beta = 0.8, |S| = 100$. The false-negative rate of the model is nearly 100% when $|M|$ equals 0, and the recognition ability of the model is weak. However, with the increasing of the size of memory detector-set, the recognition ability is

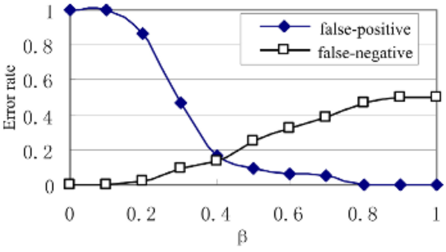

Fig. 2. The effect of matching threshold β to the error rate

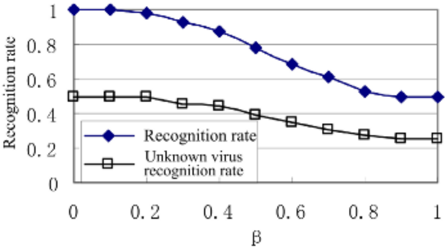

Fig. 3. The effect of matching threshold β to the recognition ability

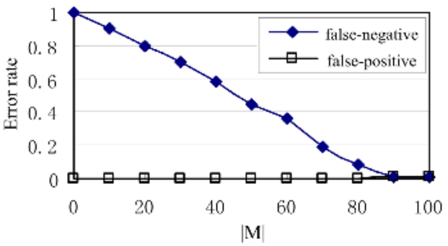

Fig. 4. The effect of the memory detector size to the error rate

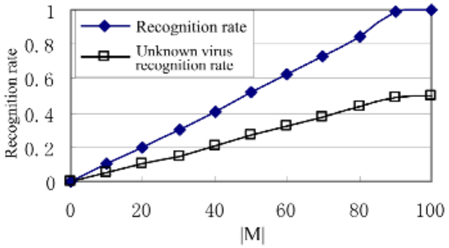

Fig. 5. The effect of the memory detector size to the recognition ability

improved rapidly. When $|M|$ equals to 90, the model can recognize almost all 200 computer viruses (100 original viruses, 100 are their variations). The model can detect almost all the variation viruses and new viruses (10% of all viruses). This indicates that this model has a strong ability of self-learning. It also indicates that the detection ability will be improved while increasing the size of memory detector-set.

To test the performance of our model, the corresponding comparison experiments were undertaken, with ARTIS [17, 18], proposed by Hofmeyr and Forrest et al, selected as the opponent. ARTIS is a typical model in traditional CVIS, which has significant impact on the design of CVIS.

Fig.6 shows the situation that the number of the needed immature detectors for generating a fixed number (here is 20) of the mature detectors, where $\beta = 0.8, |M| = 50$. The result shows that our proposed model has a higher efficiency than ARTIS. The number of candidate immature detectors is exponentially related to the size of self-set in ARTIS, however, it is linear in our model. This indicates that the time needed in self tolerance is much reduced when the candidate immature detectors are generated through the antibody gene-library.

Fig.7 shows how the size of memory detector-set affects the performance for both ARTIS and our model, where $\beta = 0.8, |S| = 100$. The result shows that our model is better than ARTIS. Since the antibody genes are extracted from memory detectors, thus, the larger the memory detector-set, the more excellent genes, therefore, better candidate immature detectors can be generated from antibody gene library. In the experiments, we found that the mature detectors generated from antibody gene library are distributed around the memory detectors, thus, they will find the variation viruses or similar ones, however, it is difficult for them to find the viruses that are much different from the known ones (i.e., new viruses). In the experiments, we also found that this problem can be solved by randomly generating immature detectors. Thus, a good idea for generating new immature detectors is to adopt two strategies: some detectors are derived from antibody gene library, but the others are randomly generated.

Fig.8 and Fig.9 show how the evolutive self-set affects the performance of ARTIS and our proposed model, where some viruses are put into self-set, and $\beta = 0.8, |M| = 50$. In the experiments we found that: 1) the size of self-set will little affect the false-positive rate; 2) the size of memory detector set will affect

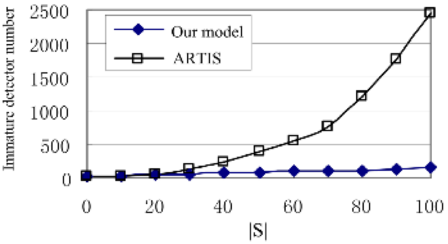

Fig. 6. The comparison experiment for the mature detector generating efficiency of ARTIS and our model

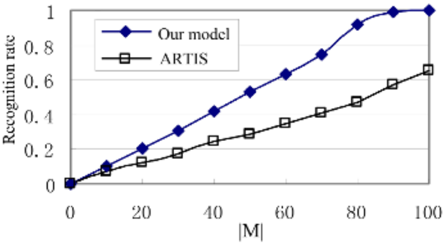

Fig. 7. The comparison experiment for the recognition ability of ARTIS and our model

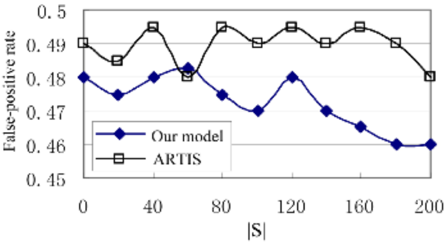

Fig. 8. The comparison experiment for the false positive rate of ARTIS and our model under different size of self set

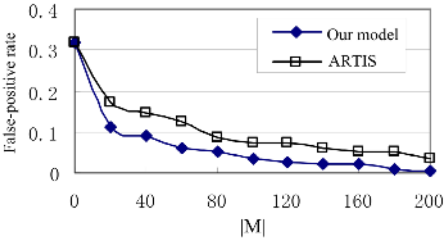

Fig. 9. The comparison experiment for the false positive rate of ARTIS and our model under different size of memory detector set

the false-positive rate, the reason is that the description of self is not completed; 3) the evolutive self-set can effectively reduce the false-positive rate, the reason is that the nonself elements in self-set will be eliminated by the evolutive self model through the feedback ability of memory detectors and the costimulation from a outside system. The experimental results show that our proposed model has a lower false-positive rate than ARTIS.

5 Conclusion

The previous models or methods, such as ARTIS, lack the ability of self-adaptation, have a higher false-positive and false-negative rate, therefore, have limited applications. In this paper, a quantitatively depiction for dynamic evolutions of *self-set, antibody gene-library, immature detector-set, mature detector-set* and *memory detector-set* are presented. Then, an immune-based dynamic model for computer virus is thus built. This model can efficiently reduce both the false-positive rate and false-negative rate, and enhance the ability of self-adaptation and diversity.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant No.60373110, the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No.20030610003, the New Century Excellent Expert Program of Ministry of Education of China under Grant No. NCET-04-0870, and the Innovation Foundation of Sichuan University under Grant No.2004CF10.

References

1. F-Secure Corporation's Data Security Summary for 2004. F-Secure Corporation. Available: http://www.f-secure.com/2004/. April 2005
2. Staniford, S., Paxson, V., Weaver, N.: How to own the internet in your spare time. In Proc. of the USENIX Security Symposium, San Francisco Marriott (2002)
3. Cohen, F.: Computer viruses: theory and experiments. Computers and Security, vol. 6 (1987) 22-35
4. Spafford, E. H.: Computer Viruses—A Form of Artificial Life? Technical Report, Purdue University (1994)
5. Swimmer, M.: Dynamic detection and classification of computer viruses using general behavior patterns. In Proc. of the Fifth International Virus Bulletin Conference, Boston (1995)
6. Albert, R., Jeong, H., Barabasi, A. L.: Diameter of the world wide web. Nature, vol. 401 (1999) 130-131
7. Lloyd, A. L., May, R. M.: How viruses spread among computers and people. Science, vol. 292 (2002) 1316-1317
8. Newman, M. E. J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Phys. Rev. E, vol. 66(035101) (2002)
9. Albert, R., Jeong, H., Barabasi, A. L.: Attack and error tolerance of complex networks. Nature, vol. 406 (2002) 378-382
10. Callaway, D. S., Newman, M. E. J., Strogatz, S. H., Watts, D. J.: Network robustness and fragility: percolation on random graphs. Phys. Rev. Lett., vol. 85 (2002) 5468-5471
11. Balthrop, J., Forrest, S., Newman, M. E. J., Williamson, M. M.: Technological networks and the spread of computer viruses. Science, vol. 304 (2004) 527-529
12. Perelson, A. S., Weisbuch, G.: Immunology for physicists. Review of Modern Physics, vol. 69(4) (1997) 1219-1263
13. De Castro, L. N, Timmis, J. I.: Artificial immune systems as a novel soft computing paradigm. Soft Computing journal, vol. 7(8) (2003) 526-544
14. Li, T.: An Introduction to Computer Network Security. Publishing House of Electronics Industry, Beijing (2004)
15. Li, T.: Computer Immunology. Publishing House of Electronics Industry, Beijing (2004)
16. Forrest, S., Perelson, A. S.: Self-nonself discrimination in a computer. in Proc. of IEEE Symposium on Security and Privacy, Oakland (1994) 202-213
17. Hofmeyr, S.: An Immunological Model of Distributed Detection and its Application to Computer Security. Ph.D. dissertation, Univ. New Mexico (1999)
18. Hofmeyr, S., Forrest, S.: Architecture for an artificial immune system. Evolutionary Computation, vol. 8(4) (2000) 443-473

19. Kephart, J. O., Arnold, W. C.: Automatic extraction of computer virus signatures. In Proc. of the Fourth International Virus Bulletin Conference, St. Helier, Jersey, UK (1994)
20. Kephart, J. O., Sorkin, G. B., Swimmer, M., White, S. R.: Blueprint for a computer immune system. In Proc. of the 1997 International Virus Bulletin Conference, San Francisco, California (1997)
21. Okamoto, T., Ishida, Y.: A distributed approach against computer viruses inspired by the immune system. IEICE Trans. on Communication, E83-B(5) (2000) 908-915
22. Harmer, P. K., Williams, P. D., Gunsch, G. H., Lamont, G. B.: An artificial immune system architecture for computer security applications. IEEE Transactions on Evolutionary Computation, vol. 6(3) (2002) 252-280
23. Rabiner, L.: A tutorial on Hidden Markov Models and selected applications in speech recognition. In Proc. of the IEEE, 77(2) (1989) 257-286
24. Jensen, R. S.: Immune system for virus detection and elimination. Master's Thesis, Technical University of Denmark, DTU (2002)
25. LISYS. Available: http://www.cs.unm.edu/ forrest/software/lisys/, April 2005
26. Li, T.: An immunity based network security risk estimation. Science in China Ser. F Information Sciences, vol. 48(5) (2005) 798-816
27. Li, T.: An immune based dynamic intrusion detection model. Chinese Science Bulletin, vol. 50(17) (2005)
28. Li, T.: A new model of immune-based network surveillance and dynamic computer forensics. Lecture Notices in Computer Science, vol. 3611 (2005) 799-808
29. Xu, C., Li, T.: A weather forecast system based on artificial immune system. Lecture Notices in Computer Science, vol. 3611 (2005) 795-798
30. Shen, J., Xie, S.: Probability and Statistics. Higher Education Press, Beijing (1989)