Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2013, Article ID 268639, 15 pages
http://dx.doi.org/10.1155/2013/268639

Research Article

A Real-Valued Negative Selection Algorithm Based on Grid for Anomaly Detection

Ruirui Zhang, Tao Li, and Xin Xiao

College of Computer Science, Sichuan University, Chengdu 610065, China

Correspondence should be addressed to Ruirui Zhang; zhangruiruisw@gmail.com

Received 15 March 2013; Accepted 13 May 2013

Academic Editor: Fuding Xie

Negative selection algorithm is one of the main algorithms of artificial immune systems. However, candidate detectors randomly generated by traditional negative selection algorithms need to conduct self-tolerance with all selves in the training set in order to eliminate the immunological reaction. The matching process is the main time cost, which results in low generation efficiencies of detectors and application limitations of immune algorithms. A novel algorithm is proposed, named GB-RNSA. The algorithm analyzes distributions of the self set in real space and regards the n-dimensional $[0, 1]$ space as the biggest grid. Then the biggest grid is divided into a finite number of sub grids, and selves are filled in the corresponding subgrids at the meantime. The randomly generated candidate detector only needs to match selves who are in the grid where the detector is and in its neighbor grids, instead of all selves, which reduces the time cost of distance calculations. And before adding the candidate detector into mature detector set, certain methods are adopted to reduce duplication coverage between detectors, which achieves fewer detectors covering the nonself space as much as possible. Theory analysis and experimental results demonstrate that GB-RNSA lowers the number of detectors, time complexity, and false alarm rate.

1. Introduction

In the past decade, the artificial immune systems have caused great concerns as a new method to solve complex computational problems. At present, there are four main areas in the studies of artificial immune systems [1]: the negative selection algorithm (NSA) [2], the artificial immune network (AINE) [3], the clonal selection algorithm (CLONALG) [4], the danger theory [5], and dendritic cell algorithms [6]. By simulating the immune tolerance in T-cell maturation process of biological systems, NSA removes self-reactive candidate detectors to effectively recognize nonself antigens, and is successfully applied to pattern recognition, anomaly detection, machine learning, fault diagnosis, and so forth [7, 8].

The negative selection algorithm is proposed by Forrest et al. [7]. This algorithm adopts strings or binary strings to encode the antigens (samples) and antibodies (detectors) and r-continuous-bit matching method to compute affinities between antigens and detectors, which is denoted SNSA [7]. The work in [9, 10] pointed out that the generation efficiency of detectors in SNSA is low. Candidate detectors become mature through negative selection. Given that N_s is the training set size, P' is the matching probability between random antigen and antibody, and P_f is the failure rate; then the number of candidate detectors $N = -\ln(P_f)/(P'(1 - P')^{N_s})$, which is exponential to N_s, and the time complexity of SNSA, is $O(N \cdot N_s)$.

Because many problems in practical applications are easy to be defined and studied in the real space, a real-valued negative selection algorithm (RNSA) is put forward in [11]. The algorithm adopts n-dimensional vectors in real space $[0, 1]^n$ to encode antigens and antibodies and Minkowski distance to calculate affinities. A real-valued negative selection algorithm with variable-sized detector (V-Detector) is proposed in [12, 13], resulting in better results. The algorithm dynamically determines the radius of a detector to generate mature ones, by computing the nearest distance between the center of the candidate detector and self-antigens. This algorithm also proposes a method for calculating detectors' coverage rate based on the probability. In the work of [14],

genetic-based negative selection algorithm is put forward, and in the work of [15], clonal optimization-based negative selection algorithm is put forward. Detectors of these two algorithms need to be processed by optimization algorithms, to gain greater coverage of nonself space. Superellipsoid detectors are introduced in [16] in the negative selection algorithm and superrectangular detectors in [17], to achieve the same coverage rate with less detectors compared with sphere ones. A self detector classification method is proposed in [18]. In this method, selves are viewed as self detectors with initial radius and the radius of selves is dynamically determined by the ROC analysis in the training stage, to increase the detection rate. A negative selection algorithm based on the hierarchical clustering of self set is put forward in [19]. This algorithm carries out the hierarchical clustering preprocess of self set to improve the generation efficiency of detectors.

Because of the low generation efficiency of mature detectors, the time cost of negative selection algorithms seriously limits their practical applications [18, 19]. A real-valued negative selection algorithm based on grid is proposed in this paper, denoted GB-RNSA. The algorithm analyzes distributions of the self set in the shape space and introduces the grid mechanism, in order to reduce the time cost of distance calculations and the duplication coverage between detectors. The remainder of this paper is organized as follows. The basic definitions of real-valued negative selection algorithms which are also the background of this paper are described in Section 2. The basic idea, implementation strategies, and analyses of GB-RNSA are described in Section 3. The effectiveness of GB-RNSA is verified using synthetic datasets and University of California Irvine (UCI) datasets in Section 4. Finally, the conclusion is given in the last section.

2. Basic Definitions of RNSA

The SNS (self/nonself) theory states that the body relies on antibodies (T cells and B cells) to recognize self antigens and nonself antigens, in order to exclude foreigners and maintain the balance and stability of the body [2, 8]. Inspired by this theory, antibodies are defined as detectors to identify nonself antigens in the artificial immune system, and their quality determines the accuracy and effectiveness of the detection system. However, randomly generated candidate detectors may identify self antigens and raise the immune self-reaction. According to the immune tolerance mechanism and mature process of immune cells in the biological immune system, Forrest put forward the negative selection algorithm to remove detectors which can recognize selves [7]. The algorithm discussed in this paper is based on real value. The basic concepts of RNSA are as follows.

Definition 1 (antigens). $Ag = \{ag \mid ag = \langle x_1, x_2, \ldots, x_n, r_s\rangle, x_i \in [0,1],\ 1 \le i \le n,\ r_s \in [0,1]\}$ are the total samples in the space of the problem. ag is an antigen in the set. n is the data dimension, x_i is the normalized value of the ith attribute of sample ag which represents the position in the real space, and r_s is the radius of ag which represents the variability threshold of ag.

Definition 2 (self set). $Self \subset Ag$ represents all the normal samples in the antigen set.

Definition 3 (nonself set). $Nonself \subset Ag$ represents all the abnormal samples in the antigen set. *Self*/*Nonself* have different meanings in various fields. For network intrusion detections, *Nonself* represents network attacks, and *Self* represents normal network access; for virus detections, *Nonself* represents virus codes, and *Self* represents legitimate codes.

$$Self \cap Nonself = \emptyset, \qquad Self \cup Nonself = Ag. \tag{1}$$

Definition 4 (training set). $Train \subset Self$ is a subset of *Self* and is the priori detection knowledge. N_s is the size of the training set.

Definition 5 (set of detectors). $D = \{d \mid d = \langle y_1, y_2, \ldots, y_n, r_d\rangle,\ y_j \in [0,1],\ 1 \le j \le n,\ r_d \in [0,1]\}$. d is a detector in the set. y_j is the jth attribute of detector d, r_d is the radius of the detector, and N_d is the size of the detector set.

Definition 6 (matching rule). $A(ag, d) = dis(ag, d)$, and $dis(ag, d)$ is the Euclidean distance between antigen ag and detector d. In the detectors' generation process, if $dis(ag, d) \le r_s + r_d$, the detector d arises the immune self-reaction and cannot become a mature detector. In the detectors' testing process, if $dis(ag, d) < r_d$, the detector d recognizes the antigen ag as a nonself.

Definition 7 (detection rate). *DR* means the proportion of non-self samples which are correctly identified by detectors in the total non-self samples and is represented by (2). *TP* is short for true positive, which means the number of nonselves which are correctly identified by detectors. *FN* is short for false negative, which means the number of non-selves which are wrongly identified:

$$DR = \frac{TP}{TP + FN}. \tag{2}$$

Definition 8 (false alarm rate). *FAR* means the proportion of self samples which are wrongly identified as non-selves in the total self samples and is represented by (3). *FP* is short for false positive, which means the number of selves which are wrongly identified by detectors, and *TN* is short for true negative, which means the number of selves which are correctly identified:

$$FAR = \frac{FP}{FP + TN}. \tag{3}$$

In general, the generation process of detectors which is the basic idea of RNSA is shown in Algorithm 1.

In the algorithm of RNSA, the randomly generated candidate detectors need to do the calculation $dis(d_{\text{new}}, ag)$ with all the elements in the training set. With the increase of the number of selves N_s, the execution time is in exponential growth, while the probability of coverage overlaps between detectors also raises, resulting in a large number of invalid detectors and low efficiency. The aforementioned problems greatly limit the practical applications of the negative selection algorithms.

RNSA($Train, r_d, maxNum, D$)
Input: the self training set $Train$, the radius of detectors r_d, the number of needed detectors $maxNum$
Output: the detector set D
Step 1. Initialize the self training set $Train$;
Step 2. Randomly generate a candidate detector d_{new}. Calculate the Euclidean distance between d_{new} and all the selves in $Train$. If $dis(d_{\text{new}}, ag) < r_d + r_s$ for at least one self antigen ag, execute Step 2; if not, execute Step 3.
Step 3. Add d_{new} into the detector set D;
Step 4. If the size of D satisfies $N_d > maxNum$, return D, and the process ends; if not, jump to Step 2.

Algorithm 1: The algorithm of RNSA.

3. Implementations of GB-RNSA

This section describes the implementation strategies of the proposed algorithm. The basic idea of the algorithm is described in Section 3.1. Sections 3.2, 3.3 and 3.4 are the detailed descriptions of the algorithm. The grid generation method is introduced in Section 3.2. Coverage calculation method of the non-self space is introduced in Section 3.3. And the filter method of candidate detectors is introduced in Section 3.4. Performance analysis of the algorithm is given in Section 3.5. Time complexity analysis of the algorithm is given in Section 3.6.

3.1. Basic Idea of the Algorithm. A real-valued negative selection algorithm based on grid GB-RNSA is proposed in this paper. The algorithm adopts variable-sized detectors and expected coverage of non-self space for detectors as the termination condition for detectors' generation. The algorithm analyzes distributions of the self set in the real space and regards $[0, 1]^n$ space as the biggest grid. Then, through divisions step-by-step until reaching the minimum diameter of the grid and adopting 2^n-tree to store grids, a finite number of subgrids are obtained, meanwhile self antigens are filled in corresponding sub grids. The randomly generated candidate detector only needs to match with selves who are in the grid where the detector is and in its neighbor grids instead of all selves, which reduces the time cost of distance calculations. When adding it into the mature detector set, the candidate detector will be matched with detectors within the grid where the detector is and neighbor grids, to judge whether the detector is in existing detectors' coverage area or its covered space totally contains other detector. This filter operation decreases the redundant coverage between detectors and achieves that fewer detectors cover the non-self space as much as possible. The main idea of GB-RNSA is as shown in Algorithm 2.

Iris dataset is one of the classic machine learning data sets published by the University of California Irvine [20], which are widely used in the fields of pattern recognition, data mining, anomaly detection, and so forth. We choose data records of category "setosa" in the dataset Iris as self antigens, choose "sepalL" and "sepalW" as antigen properties of first dimension and second dimension, and choose top 25 records of self antigens as the training set. Here, we use only two features of records, for that two-dimensional map is intuitive to illustrate the ideas, which does not affect comparison results. Figure 1 illustrates the ideas of GB-RNSA and the classical negative selection algorithms RNSA and V-Detector. RNSA generates detectors with fixed radius. V-Detector generates variable-sized detectors by dynamically determining the radius of detectors, through computing the nearest distance between the center of the candidate detector and self antigens. Detectors generated by the two algorithms need to conduct tolerance with all self antigens, which will lead to redundant coverage of non-self space between mature detectors with the increase of coverage rate. GB-RNSA first analyzes distributions of the self set in the space, and forms grids. Then, the randomly generated candidate detector only needs to perform tolerance with selves within the grid where the detector is and neighbor grids. Certain strategies are conducted for detectors which have passed tolerance, to avoid the duplication coverage and make sure that new detectors cover uncovered non-self space.

3.2. Grid Generation Method. In the process of grid generation, a top-down method is selected. First, the algorithm regards the n-dimensional $[0, 1]$ space as the biggest grid. If there are selves in this grid, divide each dimension into two parts and get 2^n sub grids. Then, continue to judge and divide each sub grid, until a grid does not contain any selves or the diameter of the grid reaches the minimum. Eventually, the grid structure of the space is obtained, and then the algorithm searches each grid to get neighbors in the structure. This process is shown in Algorithms 3 and 4.

Definition 9 (minimum diameter of grids). $r_{gs} = 4r_s + 4r_{ds}$, where r_s is the self radius and r_{ds} is the smallest radius of detectors. Suppose that the diameter of a grid is less than r_{gs}, then divide this grid; the diameter of sub grids is less than $2r_s + 2r_{ds}$. If there are selves in the sub grid, it is probably impossible to generate detectors in the sub grid. So, set the minimum diameter of grids $4r_s + 4r_{ds}$.

Definition 10 (neighbor grids). If two grids are adjacent at least in one dimension, these two grids are neighbors, which are called the basic neighbor grids. If selves of the neighbor grid are empty, add the basic neighbor grid of it in the same direction as the attached neighbor grid. The neighbors of a grid include the basic neighbor grids and the attached ones.

The filling process of neighbor grids is shown in Algorithm 5.

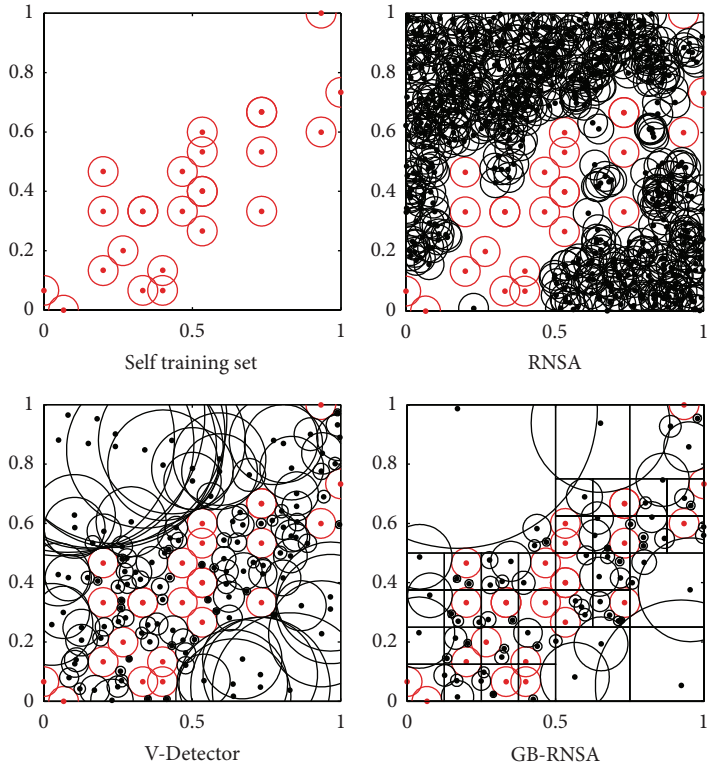

Figure 1: Comparison of RNSA, V-Detector, and GB-RNSA. (To reach the expected coverage $C_{exp} = 90\%$, three algorithms resp., need 561, 129, and 71 mature detectors, where the radius of self is 0.05, the radius of detector for RNSA is 0.05, and the smallest radius of detectors for V-Detector and GB-RNSA is 0.01).

GB-RNSA($Train, C_{exp}, D$)
Input: the self training set $Train$, expected coverage C_{exp}
Output: the detector set D
N_0: sampling times in non-self space, $N_0 > \max(5/C_{exp}, 5/(1 - C_{exp}))$
i: the number of non-self samples
x: the number of non-self samples covered by detectors
CD: the set of candidate detectors $CD = \{d \mid d =< y_1, y_2, \ldots, y_n, r_d >, y_j \in [0, 1], 1 \le j \le n, r_d \in [0, 1]\}$
Step 1. Initialize the self training set $Train$, $i = 0$, $x = 0$, $CD = \emptyset$, $N_0 = ceiling(\max(5/C_{exp}, 5/(1 - C_{exp})))$
Step 2. Call *GenerateGrid*($Train, TreeGrid, LineGrids$) to generate grid structure which contains selves, where $TreeGrid$ is the 2^n-tree storage of grids and $LineGrids$ is the line storage of grids;
Step 3. Randomly generate a candidate detector d_{new}. Call *FindGrid*($d_{new}, TreeGrid, TempGrid$) to find the grid $TempGrid$ where d_{new} is;
Step 4. Calculate the Euclidean distance between d_{new} and all the selves in $TempGrid$ and its neighbor grids. If d_{new} is identified by a self antigen, abandon it and execute Step 3; if not, increase i;
Step 5. Calculate the Euclidean distance between d_{new} and all the detectors in $TempGrid$ and its neighbor grids. If d_{new} is not identified by any detector, add it into the candidate detector set CD; if not, increase x, and judge whether it reaches the expected coverage C_{exp}, if so, return D and the algorithm ends;
Step 6. Judge whether i reaches sampling times N_0. If $i = N_0$, call *Filter*(CD) to implement the screening process of candidate detectors, and put candidate detectors which passed this process into D, reset i, x, CD; if not, return to Step 3.

Algorithm 2: The algorithm of GB-RNSA.

GenerateGrid(*Train*, *TreeGrid*, *LineGrids*)
Input: the self training set *Train*
Output: *TreeGrid* is the 2^n-tree storage of grids, *LineGrids* is the line storage of grids
Step 1. Generate the grid of *TreeGrid* with diameter 1, and set properties of the gird, including lower sub grids, neighbor grids, contained selves, and contained detectors;
Step 2. Call *DivideGrid*(*TreeGrid*, *LineGrids*)to divide grids;
Step 3. Call *FillNeighbours*(*LineGrids*) to find neighbors of each grid.

Algorithm 3: The process of grid generation.

DivideGrid(*grid*, *LineGrids*)
Input: *grid* the grid to divide
Output: *LineGrids* the line storage of grids
Step 1. If there are not any self or the diameter reaches r_{gs} of *grid*, don't divide, add *grid* into *LineGrids*, and return; if not, execute Step 2;
Step 2. Divide each dimension of *grid* into two parts, then get 2^n sub grids, and map selves of *grid* into the sub grids;
Step 3. For each sub grid, call *DivideGrid*(*grid.sub*, *LineGrids*).

Algorithm 4: The process of *DivideGrid*.

FillNeighbours(*LineGrids*)
Input: *LineGrids* the line storage of grids
Step 1. Obtain the basic neighbor grids for each grid in the structure *LineGrids*;
Step 2. For each basic neighbor of every grid, if selves of this neighbor are empty, complement the neighbor of this neighbor in the same direction as an attached neighbor for the grid;
Step 3. For each attached neighbor of every grid, if selves of this neighbor are empty, complement the neighbor of this neighbor in the same direction as an attached neighbor for the grid.

Algorithm 5: The filling process of neighbor grids.

Figure 2 describes the dividing process of grids. The self training set is also selected from records of category "setosa" of the Iris data set. Select "sepalL" and "sepalW" as antigen properties of first dimension and second dimension. As shown in Figure 2, the two-dimensional space is divided into four sub grids in the first division, and then continue to divide sub grids whose selves are not empty, until the subs cannot be divided.

Figure 3 is a schematic drawing of neighbor grids, and grids with slashes are the neighbors of grid $[0, 0.5, 0.5, 1]$ which positions in the up-left of the space.

3.3. Coverage Calculation Method of the Nonself Space. The non-self space coverage P is equal to the ratio of the volume V_{covered} covered by detectors and the total volume V_{nonself} of nonself space [12], as is shown in the following:

$$P = \frac{V_{\text{covered}}}{V_{\text{nonself}}} = \frac{\int_{\text{covered}} dx}{\int_{\text{nonself}} dx}. \tag{4}$$

Because there is redundant coverage between detectors, it is impossible to calculate (4) directly. In this paper, the probability estimation method is adopted to compute the detector coverage P. For detector set D, the probability of sampling in the non-self space covered by detectors obeys the binomial distribution $b(1, P)$ [13]. The probability of sampling m times obeys the binomial distribution $b(m, P)$.

Theorem 11. *When the number of non-self specimens of continuous sampling* $i \leq N_0$, *if* $(x/\sqrt{N_0P(1-P)}) - \sqrt{N_0P/(1-P)} > Z_\alpha$, *the non-self space coverage of detectors reaches* P. Z_α *is a percentile point of standard normal distribution,* x *is the number of non-self specimens of continuous sampling covered by detectors, and* N_0 *is the smallest positive integer which is greater than* $5/P$ *and* $5/(1-P)$.

Proof. Random variable $x \sim B(i, P)$. Set $z = x - N_0P/\sqrt{N_0P(1-P)} = (x/\sqrt{N_0P(1-P)}) - \sqrt{N_0P/(1-P)}$. We consider two cases.

(1) If the number of non-self specimens of continuous sampling $i = N_0$, known from De Moivre-Laplace theorem, when $N_0 > 5/P$ and $N_0 > 5/(1-P)$, $x \sim AN(N_0P, N_0P(1-P))$. That is, $x - N_0P/\sqrt{N_0P(1-P)} \sim AN(0, 1)$, $z \sim AN(0, 1)$. Do assumptions that H_0: the non-self space coverage of detectors $\leq P$; H_1: the non-self space coverage of detectors $> P$. Given significance level a,

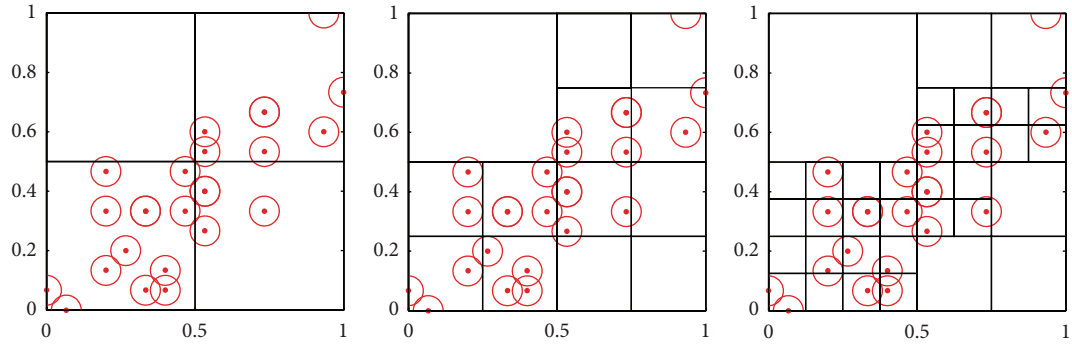

Figure 2: The process of grid division.

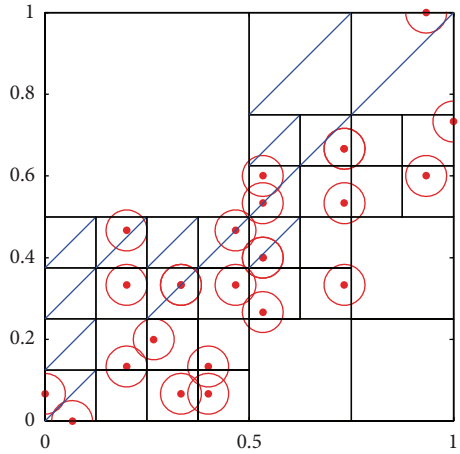

Figure 3: The neighbor grids.

$P\{z > Z_\alpha\} = a$. Then, the rejection region $W = \{(z_1, z_2, \ldots, z_n) : z > Z_\alpha\}$. So, when $(x/\sqrt{nP(1-P)}) - \sqrt{nP/(1-P)} > Z_\alpha$, z belongs to the rejection region, reject H_0, and accept H_1. That is, the non-self space coverage of detectors $> P$.

(2) If the number of non-self specimens of continuous sampling $i < N_0$, $i \cdot P$ is not too large, x approximately obeys the Poisson distribution with λ equaling $i \cdot P$. Then $P\{z > Z_\alpha\} < a$. When $(x/\sqrt{N_0P(1-P)}) - \sqrt{N_0P/(1-P)} > Z_\alpha$, the non-self space coverage of detectors $> P$. Proved.

□

From Theorem 11, in the process of detector generation, only the number of non-self specimens of continuous sampling i and the number of non-self specimens covered by detectors x need to be recorded. After sampling in the non-self space, determine whether the non-self specimen is covered by detectors of D. If not, generate a candidate detector with the position vector of this non-self specimen, and then add it into the candidate detector set CD. If so, compute whether $(x/\sqrt{N_0P(1-P)}) - \sqrt{N_0P/(1-P)}$ is larger than Z_α. If it is larger than Z_α, the non-self space coverage reaches the expected coverage P, and the sampling process stops. If not, increase i. When i is up to N_0, put candidate detectors of CD into the detector set D to change the non-self space coverage, and then set $i = 0$, $x = 0$ to restart a new round of sampling. With the continuous addition of candidate detectors, the size of the detector set D is growing, and the non-self space coverage gradually increases.

3.4. Filter Method of Candidate Detectors. When the number of sampling times in the non-self space reaches N_0, detectors of candidate detector set will be added into the detector set D. At this time, not all candidate detectors will join D, and the filtering operation will be performed for these detectors. The filtering operation consists of two parts.

The first part is to reduce the redundant coverage between candidate detectors. First, sort detectors in the candidate detector set in a descending order by the detector radius, and then judge whether the candidate detectors in the back of the sequence have been covered by the front ones. If so, this sampling of the non-self space is invalid, and the candidate detector generated from the position vector of this sampling should be deleted. There is no complete coverage between candidate detectors which have survived the first filtering operation.

The second part is to decrease the redundant coverage between mature detectors and candidate ones. The candidate detector will be matched with detectors within the grid where the detector is and neighbor grids when adding it into the detector set D, to judge whether it totally covers some mature detector. If so, the mature detector is redundant and should be removed. The filtering operations ensure that every mature detector will cover the uncovered non-self space.

The filtering process of candidate detectors is shown in Algorithm 6.

3.5. Performance Analysis. This section analyzes the performance of the algorithm from the probability theory. Assuming that the number of all the samples in the problem space is N_{Ag}, the number of antigens in the self set is N_{Self}, the number of antigens in the training set is N_s, and the number of detectors is N_d. The matching probability between a detector and an antigen is P', which is associated with

Filter(*CD*)
Input: the candidate detector set *CD*
Step 1. Sort *CD* in a descending order by the detector radius;
Step 2. Make sure that centers of detectors in the back of the sequence do not fall into the covered area of front detectors. That is to say, $dis(d_i, d_j) > r_{di}$, where $1 <= i < j <= N_{cd}$, r_{di} is the radius of detector d_i, and N_{cd} is the size of *CD*;
Step 3. Add candidate detectors into *D*, and ensure that they do not entirely cover any detector in *D*. That is to say, $dis(d_i, d_j) > r_{di}$ *or* $dis(d_i, d_j) <= r_{di}$ and $2r_{dj} > r_{di}$, where $1 <= i <= N_{cd}$, $1 <= j <= N_d$, r_{di} and r_{dj} are the radiuses of d_i and d_j respectively, and N_{cd} and N_d are the sizes of *CD and D* respectively.

Algorithm 6: The filtering process of candidate detectors.

the specific matching rule [7, 9]. $P(A)$ is defined as the probability of occurrence of event A [21].

Theorem 12. *The probability of matching an undescribed self antigen for a detector which is passed self-tolerance is* $P_d = (1 - P')^{N_s} \cdot (1 - (1 - P')^{N_{self} - N_s})$.

Proof. From the proposition, a given detector passing the self-tolerance indicates that this detector does not match any antigen in the self training set. Let event A be "the given detector does not match any antigen in the self set," event B "the given detector matches at least one antigen which is not described," then $P_d = P(A)P(B)$. In the event A, the number of times for a detector matching antigens in the self set X meets the binomial distribution, $X \sim b(N_s, P')$. Therefore, $P(A) = P(X = 0) = (1 - P')^{N_s}$. In the event B, the number of times for a detector matching undescribed self antigens Y meets the binomial distribution, $Y \sim b(N_{Self} - N_s, P')$. Then, $P(B) = 1 - P(Y = 0) = 1 - (1 - P')^{N_{Self} - N_s}$.

So, $P_d = P(A)P(B) = (1 - P')^{N_s} \cdot (1 - (1 - P')^{N_{self} - N_s})$. Proved. □

Theorem 13. *The probability of correct identification for a non-self antigen is* $P_{tp} = 1 - (1 - P')^{N_d \cdot (1 - P_d)}$, *and the probability of erroneous identification for a non-self antigen is* $P_{fn} = (1 - P')^{N_d \cdot (1 - P_d)}$. *The probability of correct identification for a self antigen is* $P_{tn} = (1 - P')^{N_d \cdot P_d}$, *and the probability of erroneous identification for a self antigen is* $P_{fp} = 1 - (1 - P')^{N_d \cdot P_d}$.

Proof. Let event A be "the given non-self antigen matches at least one detector in the detectors set." In the event A, the number of times for a non-self antigen matching detectors X meets the binomial distribution, $X \sim b(N_d \cdot (1 - P_d), P')$. Therefore, $P_{tp} = P(A) = 1 - P(X = 0) = 1 - (1 - P')^{N_d \cdot (1 - P_d)}$, and $P_{fn} = 1 - P_{tp} = (1 - P')^{N_d \cdot (1 - P_d)}$.

Let event B be "the given self antigen does not match any detector in the detectors set." In the event B, the number of times for a self antigen matching detectors Y meets the binomial distribution, $Y \sim b(N_d \cdot P_d, P')$. Therefore, $P_{tn} = P(B) = P(Y = 0) = (1 - P')^{N_d \cdot P_d}$, and $P_{fp} = 1 - P_{tn} = 1 - (1 - P')^{N_d \cdot P_d}$. Proved. □

P' is substantially constant for specific matching rules [7, 9]. Assuming that $P' = 0.005$ and $N_{Self} = 1000$, then Figure 4 shows variations of P_{tp}, P_{fn}, P_{fp}, and P_{tn} under the effects of N_d and N_s. As can be seen from the figure, when the number of selves in the training set N_s and the number of detectors N_d are larger, the probability of correct identification for an arbitrary given non-self antigen P_{tp} is greater, the probability of erroneous identification P_{fn} is small, and variation tendencies of P_{tp} and P_{fn} are not large while N_d and N_s change. Thus, when the coverage of non-self space for the detector set is certain, the detection rates of different algorithms are relatively close. When N_s and N_d are larger, the probability of correct identification for an arbitrary given self antigen P_{tn} is greater, the probability of erroneous identification P_{fp} is small, and variation tendencies of P_{tn} and P_{fp} are large while N_d and N_s change. So, when the coverage of non-self space for the detector set is certain, the false alarm rate of GB-RNSA is smaller for that the algorithm significantly reduces the number of detectors.

3.6. Time Complexity Analysis

Theorem 14. *The time complexity of detector generation process in GB-RNSA is* $O((|D|/(1 - P'))(N_s + |D|^2))$, *where* N_s *is the size of the training set,* $|D|$ *is the size of the detector set, and* P' *is the average self-reactive rate of detectors.*

Proof. For GB-RNSA, the main time cost of generating a new mature detector includes the time spending of calling *FindGrid* to find the grid, the time spending of self-tolerance for candidate detectors, and the time spending of call *Filter* to screen detectors.

Known from Section 3.2, the depth of 2^n-tree is $Ceil(\log_2(1/(4r_s + 4r_{ds})))$. So, for a new detector, the time complexity of finding the grid $grid'$ where the detector is $t1 = O(Ceil(\log_2(1/(4r_s + 4r_{ds})))^n)$. n is the space dimension, r_s is the radius of selves, and r_{ds} is the smallest radius of detectors. So, $t1$ is relatively constant.

Calculating the radius of the new detector needs to compute the nearest distance with selves in the grid where the detector is and neighbors. The time complexity is $t2 = O(N_{s'})$, where $N_{s'}$ is the number of selves in $grid'$ and neighbors.

The time complexity of calculating whether the new detector is covered by existing detectors is $t3 = O(D')$, where D' is the number of detectors in $grid'$ and neighbors.

The time complexity of calling *Filter* to screen detectors includes the time spending of sorting the candidate detectors

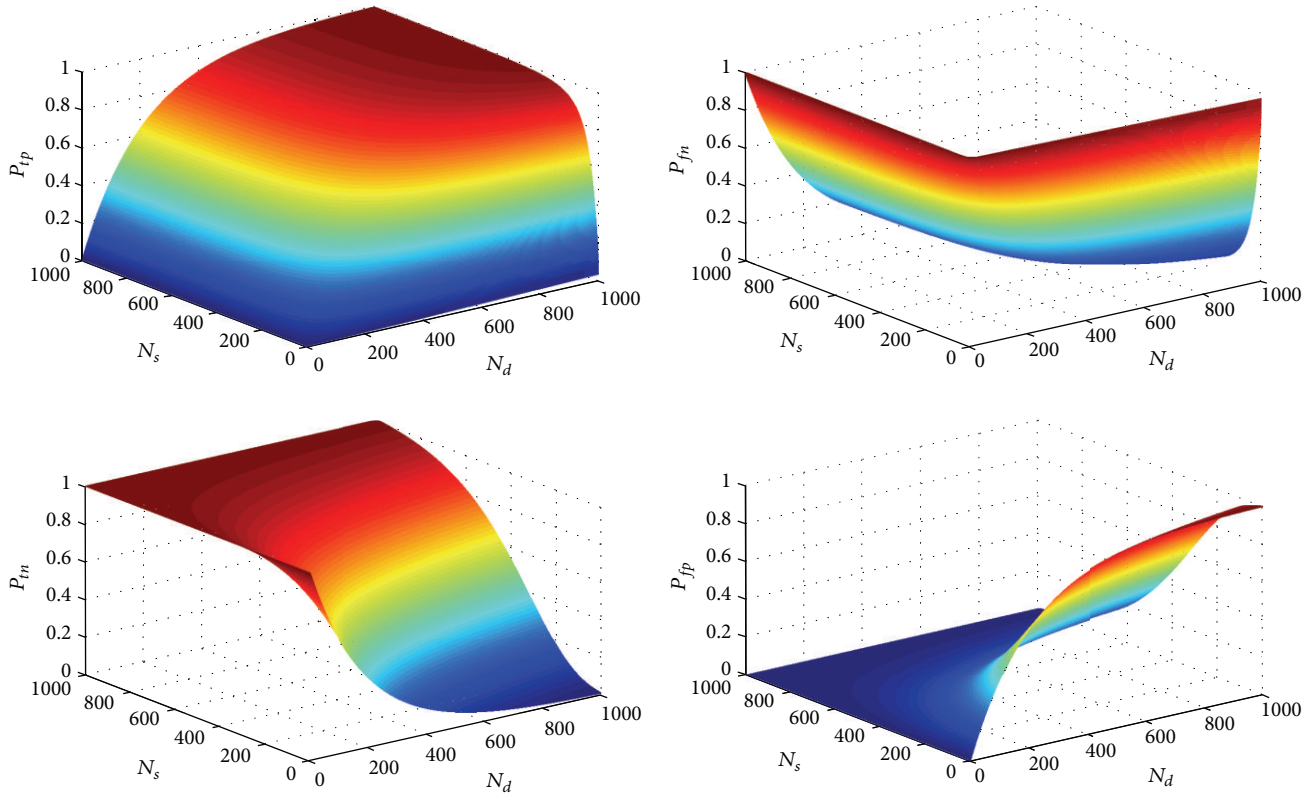

Figure 4: Simulations of Theorem 13.

and judging whether redundant coverage exists; that is, $t4 = O(N_0^2 + N_0 \cdot D')$.

Suppose N' is the number of candidate detectors to generate the detector set D, then the time complexity of sampling is $N' \cdot (t1 + t2) + N' \cdot (1 - P') \cdot t3 + (N'/N_0) \cdot t4$. And $N' \approx |D|/(1 - P')$, so, the time complexity of generating the detector set D is as follows:

$$O\left(\frac{|D|}{1-P'}\left(t1 + \sum N_{s'}\right) + |D|\left(\sum D'\right) + \frac{|D|\left(N_0 + \sum D'\right)}{(1-P')}\right)$$
$$= O\left(\frac{|D|}{1-P'}N_s + |D|^2 + \frac{|D|^2}{1-P'}\right)$$
$$= O\left(\frac{|D|}{1-P'}\left(N_s + |D|^2\right)\right). \quad (5)$$

So, the time complexity of detector generation process in GB-RNSA is $O((|D|/(1 - P'))(N_s + |D|^2))$. Proved. □

SNSA, RNSA, and V-Detector are the main detector generation algorithms and are widely used in the fields of artificial immune-based pattern recognition, anomaly detection, immune optimization, and so forth. Table 1 shows the comparisons of these negative selection algorithms and GB-RNSA. As seen from Table 1, the time complexity of traditional algorithms is exponential to the size of selves N_s. When the number of self elements increases, the time cost will rapidly increase. GB-RNSA eliminates the exponential impact and reduces the influence of growth of selves' scale on the time cost. So, GB-RNSA lowers the time complexity of the original algorithm and improves the efficiency of detector generation.

Table 1: Comparisons of time complexity.

Algorithm	Time complexity
SNSA	$O\left(\frac{-\ln(P_f) \cdot N_s}{P(1-P')^{N_s}}\right)$ [7]
RNSA	$O\left(\frac{\lvert D\rvert \cdot N_s}{(1-P')^{N_s}}\right)$ [11]
V-Detector	$O\left(\frac{\lvert D\rvert \cdot N_s}{(1-P')^{N_s}}\right)$ [13]
GB-RNSA	$O\left(\frac{\lvert D\rvert}{1-P'}\left(N_s + \lvert D\rvert^2\right)\right)$

4. Experimental Results and Analysis

This section validates the effectiveness of GB-RNSA through experiments. Two types of data sets are selected for the experiments which are commonly used in the study of real-valued negative selection algorithms, including 2D synthetic datasets [22] and UCI datasets [20]. 2D synthetic datasets

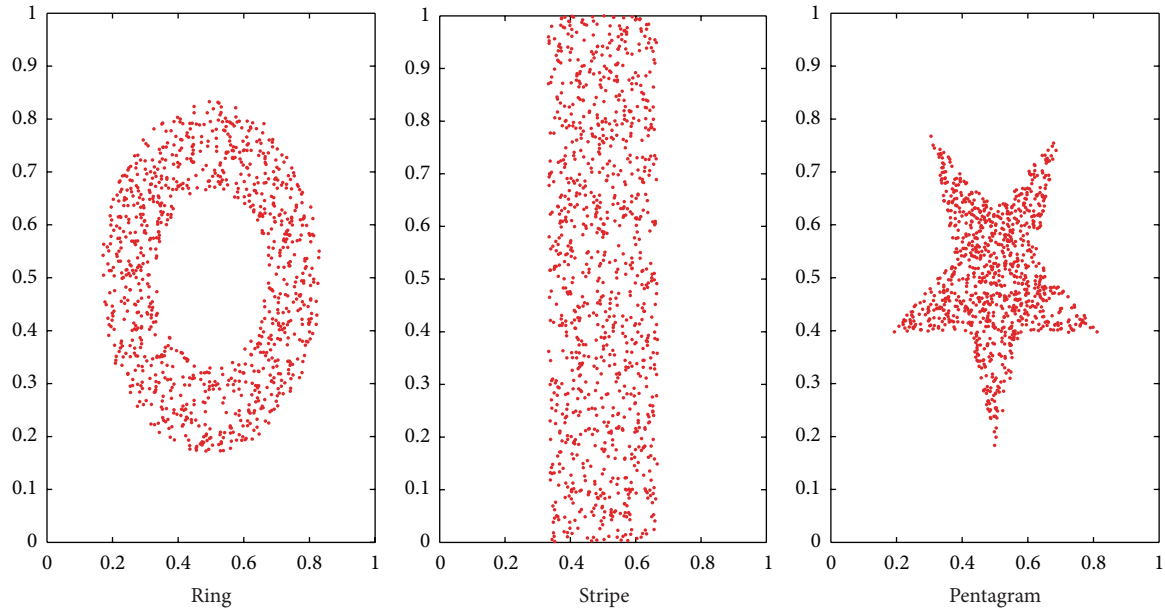

Figure 5: Distributions of Ring, Stripe, and Pentagram datasets.

Table 2: Effects of different self radiuses.

Datasets	Self radius $r_s = 0.02$		Self radius $r_s = 0.1$		Self radius $r_s = 0.2$	
	DR%	*FAR*%	*DR*%	*FAR*%	*DR*%	*FAR*%
Ring	81.55 (1.02)	62.11 (2.14)	61.77 (1.39)	12.04 (1.24)	32.39 (1.42)	0.00 (0.00)
Stripe	80.21 (1.24)	63.34 (1.90)	58.52 (1.18)	11.20 (2.47)	25.93 (1.88)	0.00 (0.00)
Pentagram	77.09 (1.38)	67.02 (2.32)	57.65 (2.31)	13.19 (1.63)	22.78 (1.59)	0.00 (0.00)

Table 3: Effects of different sizes of the training set.

Datasets	Size of the training set $N_s = 100$		Size of the training set $N_s = 500$		Size of the training set $N_s = 800$	
	DR%	*FAR*%	*DR*%	*FAR*%	*DR*%	*FAR*%
Ring	22.54 (1.22)	76.26 (2.05)	86.09 (1.16)	8.21 (1.21)	95.92 (1.37)	0.00 (0.00)
Stripe	18.25 (1.98)	78.92 (2.32)	80.13 (1.87)	9.05 (1.44)	87.63 (1.78)	0.00 (0.00)
Pentagram	12.20 (1.55)	88.29 (2.87)	72.33 (1.91)	11.42 (1.41)	82.18 (1.49)	0.00 (0.00)

Table 4: Experimental parameters of UCI datasets.

Datasets	Record numbers	Properties	Types	Self sets	Nonself sets	Training set and its size	Test set and its size
Iris	150	4	Real	Setosa: 50	Versicolour: 50 Virginica: 50	Setosa: 25	Setosa: 25 Versicolour: 25 Virginica: 25
Haberman's Survival	306	3	Integer	Survived: 225	Died: 81	Survived: 150	Survived: 50 Died: 50
Abalone	4177	8	Real, integer	M: 1528	F: 1307 I: 1342	M: 1000	M: 500 F: 500 I: 500

are authoritative in the performance test of real-valued negative selection algorithms [13, 19, 22], which is provided by Professor Dasgupta's research team of Memphis University. UCI datasets are classic machine learning data sets, which are widely used in the tests of detectors' performance and generation efficiencies [11, 18, 19, 23]. In the experiments, two traditional real-valued negative selection algorithms RNSA and V-Detector are chosen to compare with.

The number of mature detectors *DN*, the detection rate *DR*, the false alarm rate *FAR*, and the time cost of detectors

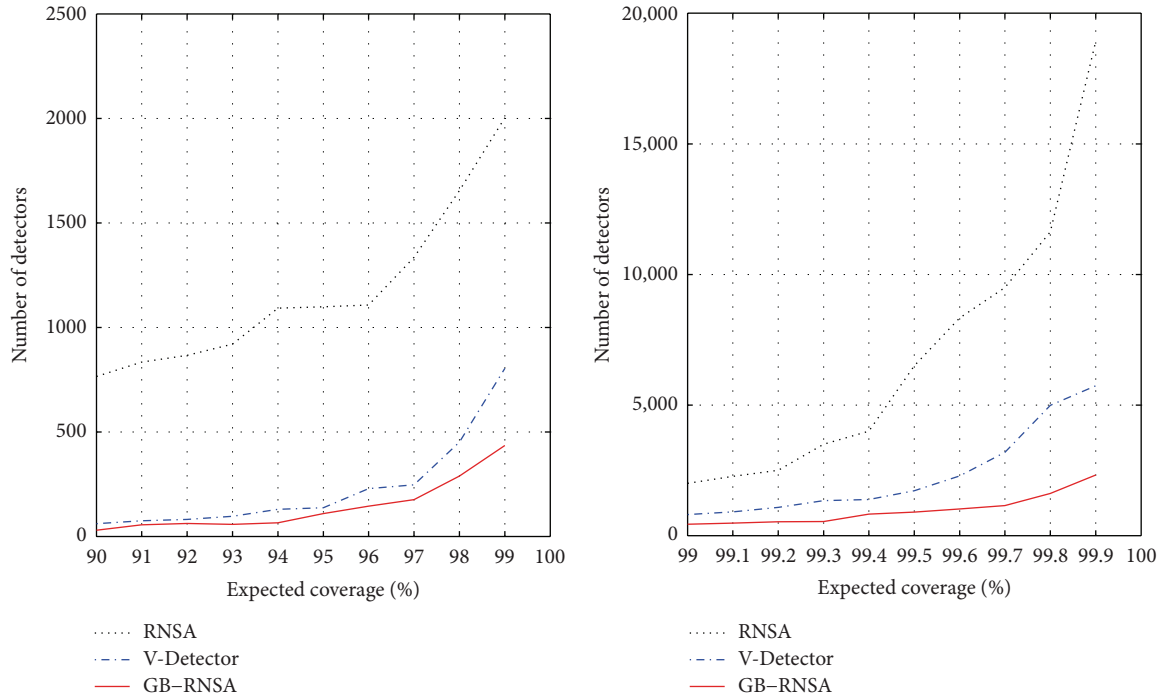

Figure 6: Comparisons of the numbers of detectors for RNSA, V-Detector, and GB-RNSA (dataset of Haberman's Survival is adopted; the radius of self antigen is 0.1).

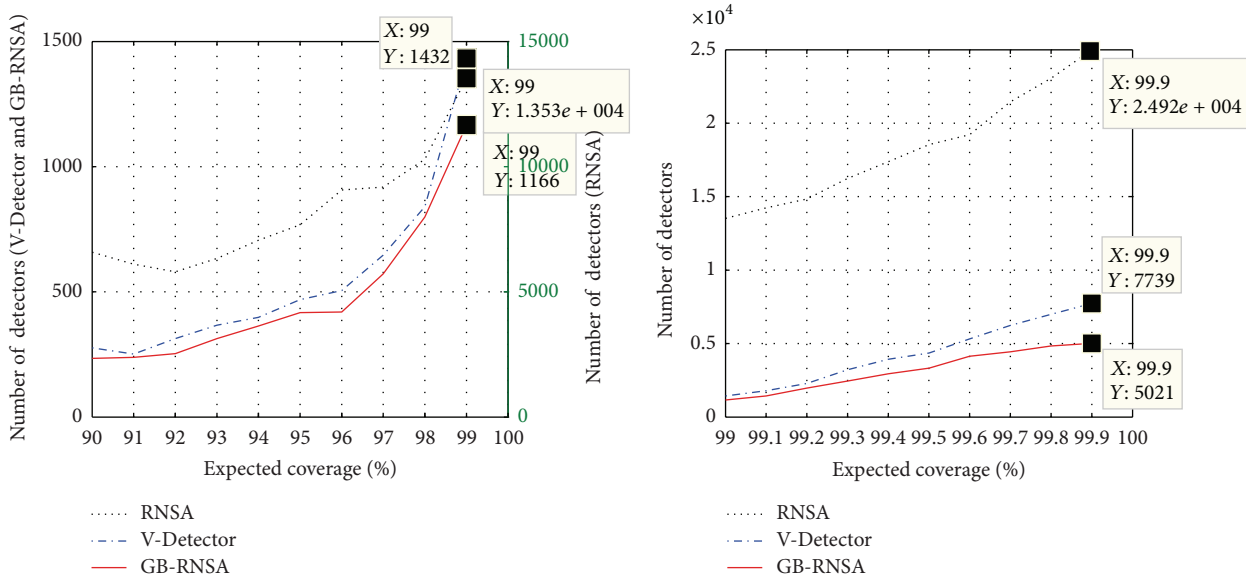

Figure 7: Comparisons of the numbers of detectors for RNSA, V-Detector, and GB-RNSA (dataset of Iris is adopted; the radius of self antigen is 0.1).

generations *DT* are adopted to measure the effectiveness of the algorithms in the experiments. Because the traditional algorithm RNSA uses the preset number of detectors as the termination condition, this paper modified RNSA and uses the expected coverage of non-self space as the termination condition, in order to ensure that the three algorithms are under the same experimental conditions to make valid comparisons.

4.1. 2D Synthetic Datasets. These datasets consist of several different subdatasets. We choose Ring, Stripe, and Pentagram subdatasets to test the performance of detectors generation of GB-RNSA. Figure 5 shows the distributions of these three datasets in two-dimensional real space.

The size of self sets of the three datasets is $N_{Self} = 1000$. The training set is composed of data points randomly selected from the self set, and the test data is randomly

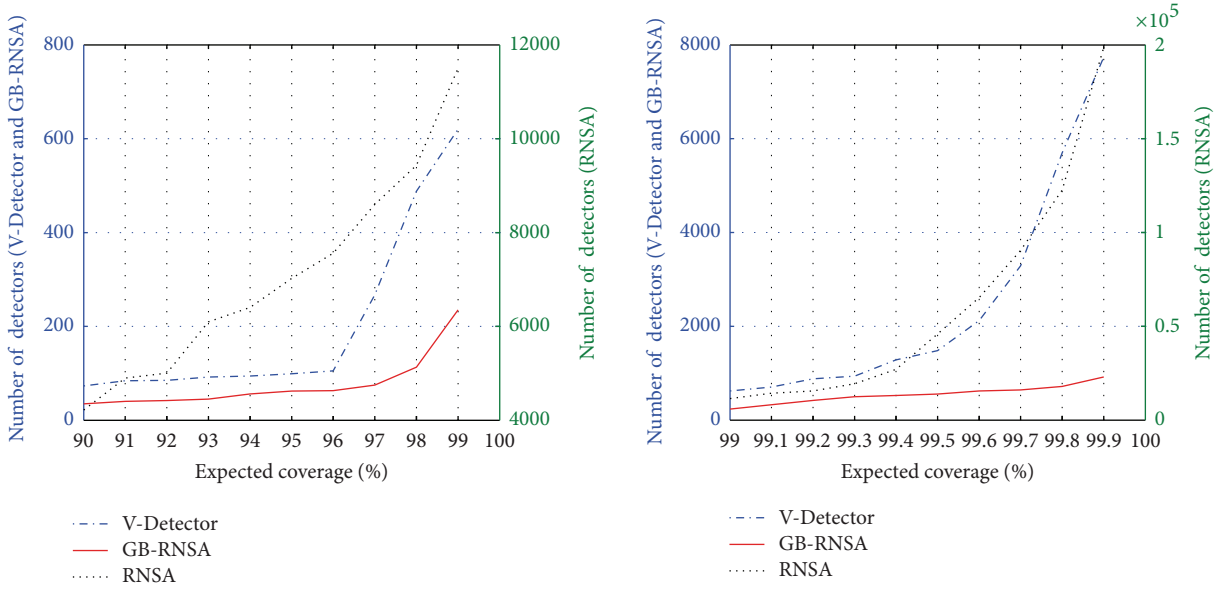

Figure 8: Comparisons of the numbers of detectors for RNSA, V-Detector, and GB-RNSA (dataset of Abalone is adopted; the radius of self antigen is 0.1).

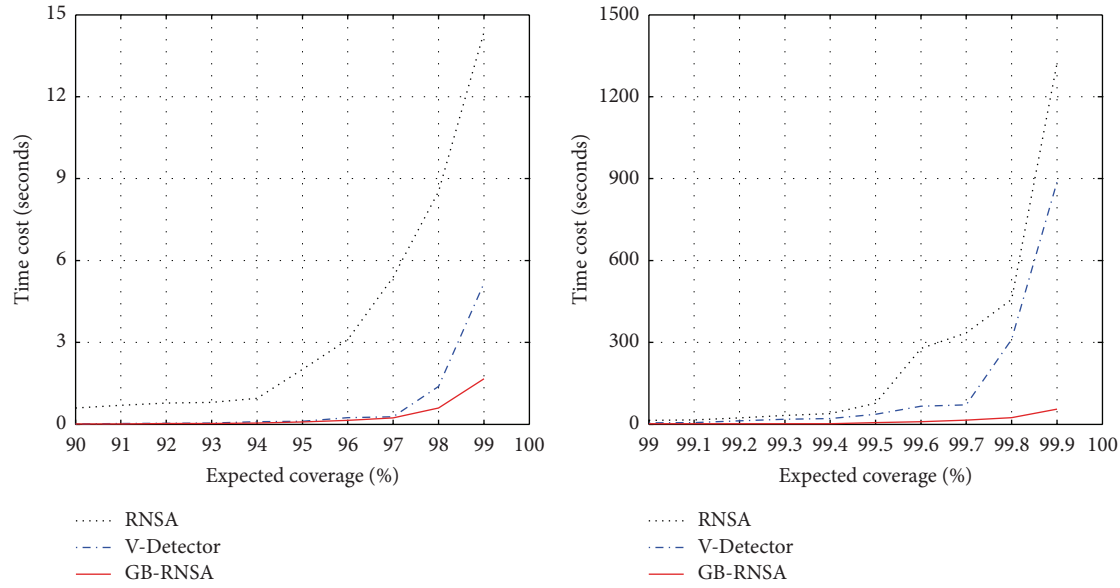

Figure 9: Comparisons of time costs of RNSA, V-Detector, and GB-RNSA (dataset of Haberman's Survival is adopted; the radius of self antigen is 0.1).

selected from the two-dimensional $[0, 1]$ space. The experiments were repeated 20 times and the average values were adopted. Experimental results are shown in Tables 2 and 3, where values within parenthesis are variances. Table 2 lists comparisons of detection rates and false alarm rates of GB-RNSA in the three datasets under the same expected coverage of 90%, the same training set $N_s = 300$, and different self radiuses. As can be seen, the algorithm has higher detection rate and false alarm rate under smaller self radius, while the algorithm has lower detection rate and false alarm rate under greater self radius. Table 3 lists comparisons of detection rates and false alarm rates of GB-RNSA in the three datasets under the same expected coverage of 90%, the same self radius $r_s = 0.05$ and different sizes of training set. The detection rate increases gradually and the false alarm rate decreases gradually while the size of the training set grows.

4.2. UCI Datasets. Three standard UCI data sets including Iris, Haberman's Survival and Abalone, are chosen to do the experiments, and experimental parameters are shown in Table 4. For the three data sets, self set and non-self set are chosen randomly, and records of training set and test set are

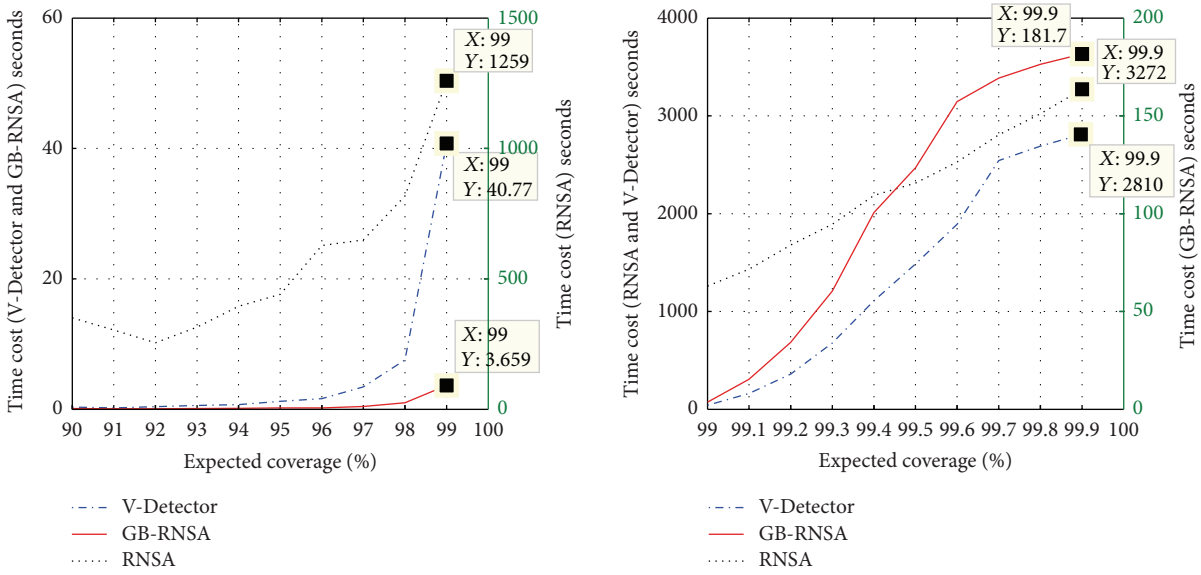

Figure 10: Comparisons of time costs of RNSA, V-Detector, and GB-RNSA (dataset of Iris is adopted; the radius of self antigen is 0.1).

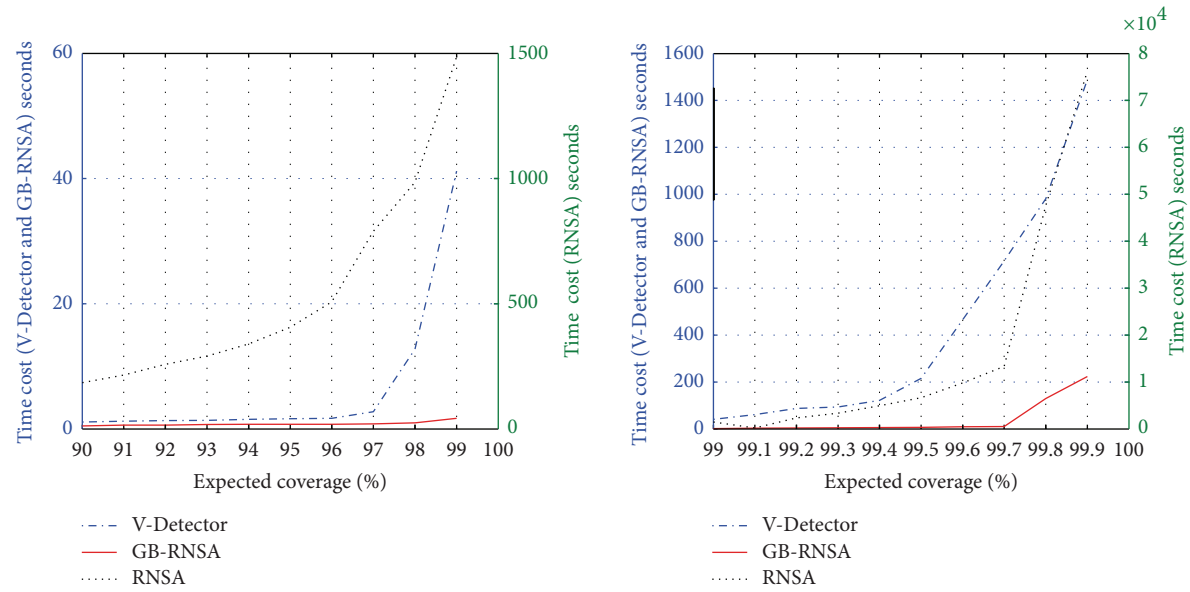

Figure 11: Comparisons of time costs of RNSA, V-Detector, and GB-RNSA (dataset of Abalone is adopted; the radius of self antigen is 0.1).

chosen randomly as well. The experiments were repeated 20 times and the average values were adopted.

4.2.1. Comparisons of the Number of Detectors. Figures 6, 7, and 8 show the number of mature detectors of RNSA, V-Detector, and GB-RNSA on the three data sets. Seen from the figures, with the increase of the expected coverage, the number of detectors which are needed to meet the coverage requirements for the three algorithms correspondingly increases. But the efficiency of GB-RNSA is significantly better than those of RNSA and V-Detector. For the data set of Iris, to achieve the expected coverage 99%, RNSA needs 13527 mature detectors, V-Detector needs 1432, and GB-RNSA needs 1166 which decreases about 91.4% and 18.6%, respectively. For the larger data set of Abalone, to achieve the expected coverage 99%, RNSA needs 11500 mature detectors, V-Detector needs 620, and GB-RNSA needs 235 which decreases about 98% and 62.1%, respectively. Thus, under the same expected coverage, different data dimensions, and different training sets, the number of mature detectors generated by GB-RNSA is significantly reduced compared with RNSA and V-Detector.

4.2.2. Comparisons of the Cost of Detectors' Generations. Figures 9, 10, and 11 show the time costs of detectors' generation of RNSA, V-Detector, and GB-RNSA on the

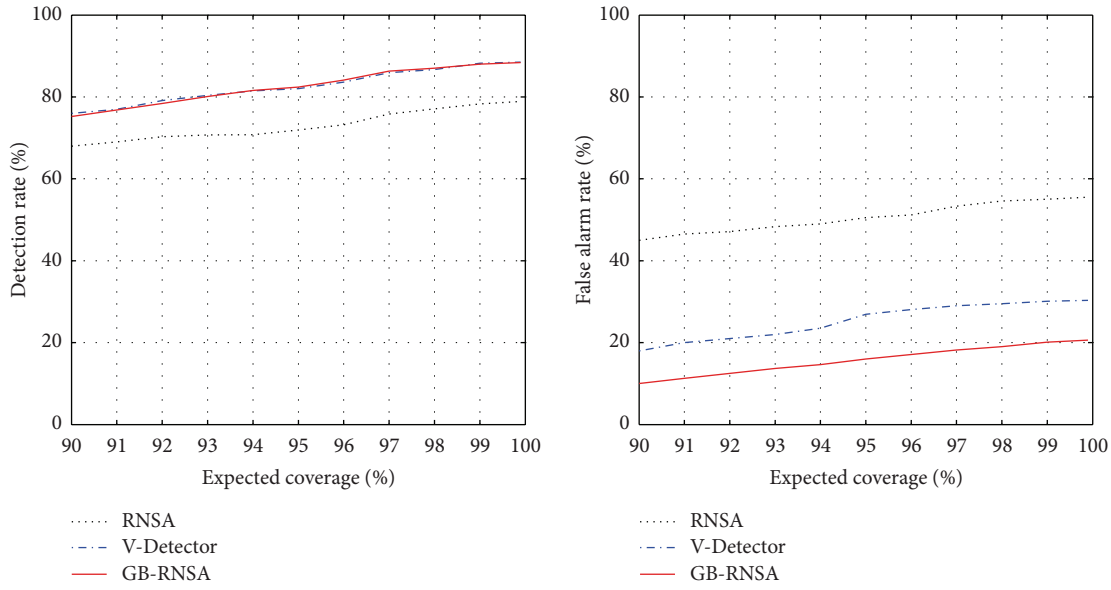

Figure 12: Comparisons of DR and FAR of RNSA, V-Detector, and GB-RNSA (dataset of Haberman's Survivalis is adopted; the radius of self antigen is 0.1).

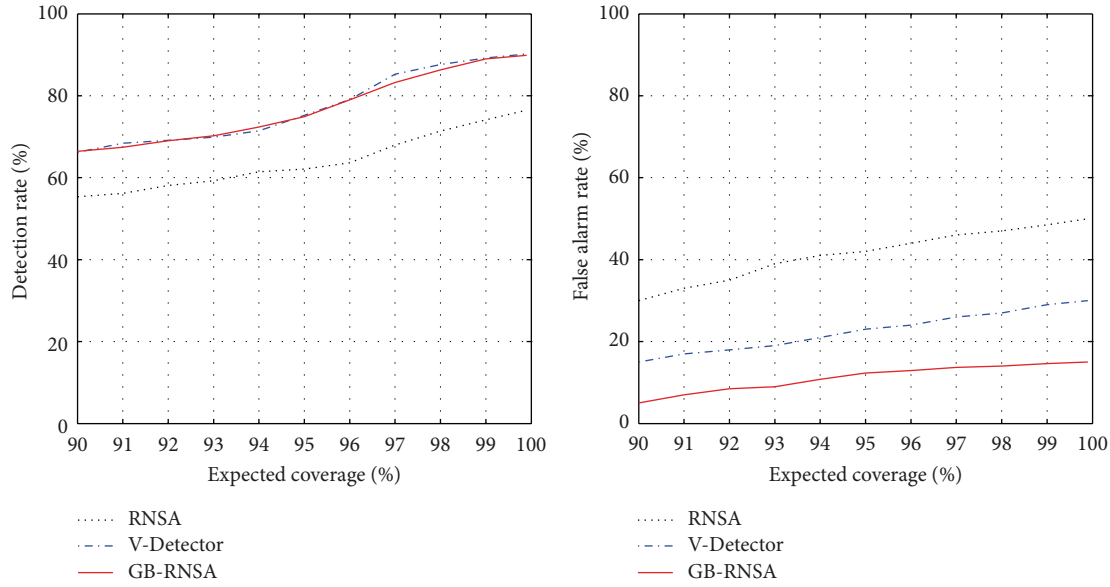

Figure 13: Comparisons of DR and FAR of RNSA, V-Detector, and GB-RNSA (dataset of Iris is adopted; the radius of self antigen is 0.1).

three data sets. As seen from the figures, with the increase of the expected coverage, the time cost of RNSA and V-Detector is in a sharp increase, while that of GB-RNSA is in a slow growth. For the data set of Iris, to achieve the expected coverage of 90%, the time cost of RNSA is 350.187 seconds, that of V-Detector is 0.347 seconds, and that of GB-RNSA is 0.1 seconds which decreases about 99.97% and 71.2%, respectively; when the expected coverage is 99%, the time cost of RNSA is 1259.047 seconds, that of V-Detector is 40.775 seconds, and that of GB-RNSA is 3.659 seconds which decreases about 99.7% and 91.0%, respectively. For the other two datasets, experimental results are similar. Thus, compared with RNSA and V-Detector, the effectiveness of detectors' generation of GB-RNSA is promoted.

4.2.3. Comparisons of Detection Rates and False Alarm Rates. Figures 12, 13, and 14 show the detection rates and false alarm rates of RNSA, V-Detector, and GB-RNSA on the three data sets. As seen from the figures, when the expected coverage is large than 90%, the detection rates of the three algorithms are similar, and that of RNSA is slightly lower, while the false alarm rate of GB-RNSA is obviously lower than those of RNSA and V-Detector. For the data set of Haberman's

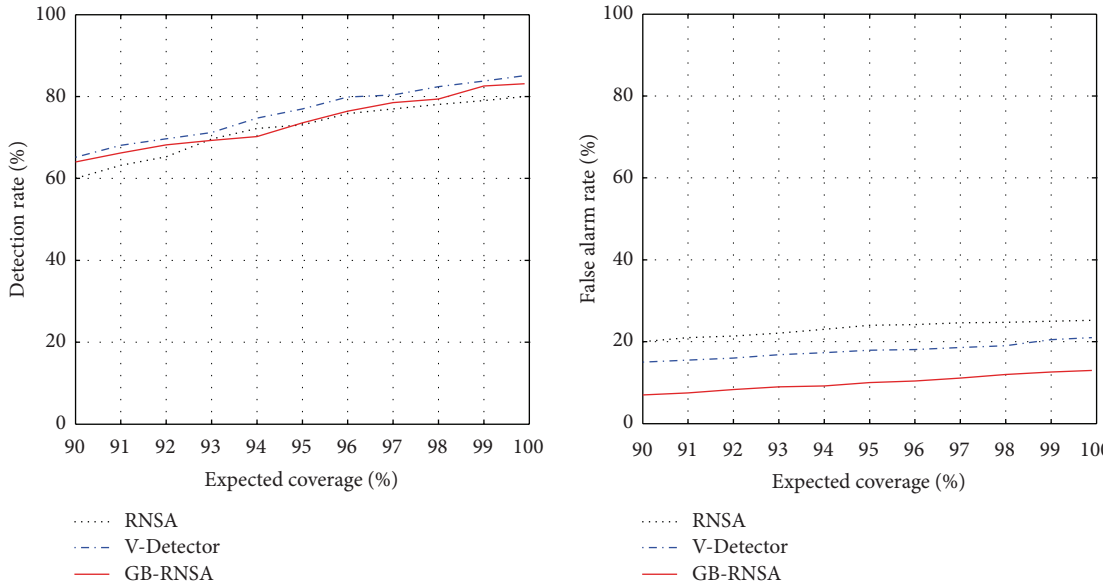

Figure 14: Comparisons of DR and FAR of RNSA, V-Detector, and GB-RNSA (dataset of Abalone is adopted; the radius of self antigen is 0.1).

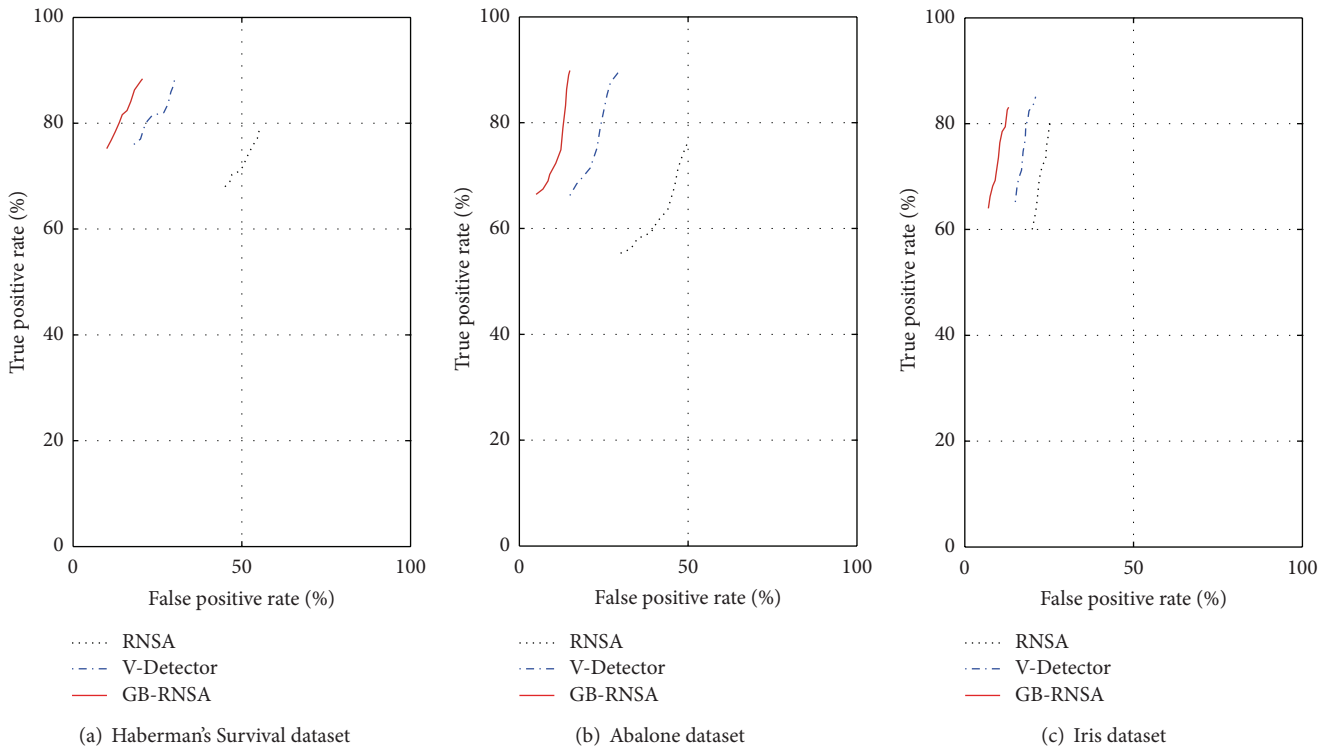

Figure 15: ROC curves of RNSA, V-Detector, and GB-RNSA.

Survival, when the expected coverage is 99%, the false alarm rate of RNSA is 55.2%, that of V-Detector is 30.1%, and that of GB-RNSA is 20.1% which decreases about 63.6% and 33.2%, respectively. For the data set of Abalone, when the expected coverage is 99%, the false alarm rate of RNSA is 25.1%, that of V-Detector is 20.5%, and that of GB-RNSA is 12.6% which decreases about 49.8% and 38.5%, respectively. Thus, under the same expected coverage, the false alarm rate of GB-RNSA is significantly lower compared with RNSA and V-Detector.

The ROC curve is a graphical method for the classification model using true positive rate and false positive rate. In NSAs, true positive rate is the detection rate and false positive rate is the false alarm rate. Figure 15 shows the ROC curves of RNSA, V-Detector, and GB-RNSA on the three data sets.

A good classification model should be as close as possible to the upper-left corner of the graphic. As seen from Figure 15, GB-RNSA is better than RNSA and V-Detector.

5. Conclusion

Too many detectors and high time complexity are the major problems of existing negative selection algorithms, which limit the practical applications of NSAs. There is also a problem of redundant coverage of non-self space for detectors in NSAs. A real-valued negative selection algorithm based on grid for anomaly detection GB-RNSA is proposed in this paper. The algorithm analyzes distributions of the self set in the real space and divides the space into grids by certain methods. The randomly generated candidate detector only needs to match selves who are in the grid where the detector is and in its neighbor grids. And before the candidate detector is added into the mature detector set, certain methods are adopted to reduce the duplication coverage. Theory analysis and experimental results demonstrate that GB-RNSA has better time efficiency and detector quality compared with classical negative selection algorithms and is an effective artificial immune algorithm to generate detectors for anomaly detection.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China under Grant no. 61173159, the National Natural Science Foundation of China under Grant no. 60873246, and the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China, under Grant no. 708075.

References

[1] D. Dasgupta, S. Yu, and F. Nino, "Recent advances in artificial immune systems: models and applications," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1574–1587, 2011.

[2] P. Bretscher and M. Cohn, "A theory of self-nonself discrimination," *Science*, vol. 169, no. 3950, pp. 1042–1049, 1970.

[3] F. Burnet, *The Clonal Selection Theory of Acquired Immunity*, Vanderbilt University Press, Nashville, Tenn, USA, 1959.

[4] N. K. Jerne, "Towards a network theory of the immune system," *Annals of Immunology*, vol. 125, no. 1-2, pp. 373–389, 1974.

[5] P. Matzinger, "The danger model: a renewed sense of self," *Science*, vol. 296, no. 5566, pp. 301–305, 2002.

[6] M. L. Kapsenberg, "Dendritic-cell control of pathogen-driven T-cell polarization," *Nature Reviews Immunology*, vol. 3, no. 12, pp. 984–993, 2003.

[7] S. Forrest, L. Allen, A. S. Perelson, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp. 202–212, May 1994.

[8] T. Li, *Computer Immunology*, House of Electronics Industry, Beijing, China, 2004.

[9] T. Li, "Dynamic detection for computer virus based on immune system," *Science in China F*, vol. 51, no. 10, pp. 1475–1486, 2008.

[10] T. Li, "An immunity based network security risk estimation," *Science in China F*, vol. 48, no. 5, pp. 557–578, 2005.

[11] F. A. González and D. Dasgupta, "Anomaly detection using real-valued negative selection," *Genetic Programming and Evolvable Machines*, vol. 4, no. 4, pp. 383–403, 2003.

[12] Z. Ji, *Negative selection algorithms: from the thymus to V-detector [Ph.D. dissertation]*, University of Memphis, Memphis, Tenn, USA, 2006.

[13] Z. Ji and D. Dasgupta, "V-detector: an efficient negative selection algorithm with "probably adequate" detector coverage," *Information Science*, vol. 19, no. 9, pp. 1390–1406, 2009.

[14] X. Z. Gao, S. J. Ovaska, and X. Wang, "Genetic algorithms-based detector generation in negative selection algorithm," in *Proceedings of the IEEE Mountain Workshop on Adaptive and Learning Systems (SMCals '06)*, pp. 133–137, July 2006.

[15] X. Z. Gao, S. J. Ovaska, X. Wang, and M.-Y. Chow, "Clonal optimization of negative selection algorithm with applications in motor fault detection," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '06)*, pp. 5118–5123, Taipei, Taiwan, October 2006.

[16] J. M. Shapiro, G. B. Lament, and G. L. Peterson, "An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 337–344, Washington, DC, USA, June 2005.

[17] M. Ostaszewski, F. Seredynski, and P. Bouvry, "Immune anomaly detection enhanced with evolutionary paradigms," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference (GECCO '06)*, pp. 119–126, Seattle, Wash, USA, July 2006.

[18] T. Stibor, P. Mohr, and J. Timmis, "Is negative selection appropriate for anomaly detection?" in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '05)*, pp. 569–576, IEEE Computer Society Press, June 2005.

[19] W. Chen, X. Liu, T. Li, Y. Shi, X. Zheng, and H. Zhao, "A negative selection algorithm based on hierarchical clustering of self set and its application in anomaly detection," *International Journal of Computational Intelligence Systems*, vol. 4, no. 4, pp. 410–419, 2011.

[20] "UCI Dataset," http://archive.ics.uci.edu/ml/datasets.

[21] G. Chang and J. Shi, *Mathematical Analysis Tutorial*, Higher Education Press, Beijing, China, 2003.

[22] F. Gonzalez, D. Dasgupta, and J. Gomez, "The effect of binary matching rules in negative selection," in *Proceedings of the Genetic and Evolutionary Computation (GECCO '03)*, pp. 196–206, Springer, Berlin, Germany, 2003.

[23] T. Stibor, J. Timmis, and C. Eckert, "On the appropriateness of negative selection defined over hamming shape-space as a network intrusion detection system," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, pp. 995–1002, September 2005.