Hindawi
Complexity
Volume 2018, Article ID 2520940, 20 pages
https://doi.org/10.1155/2018/2520940

Research Article

A Clone Selection Based Real-Valued Negative Selection Algorithm

Ruirui Zhang[1] **and Xin Xiao**[2]

[1]*School of Business, Sichuan Agricultural University, Chengdu 610000, China*
[2]*School of Computer Science, Southwest Minzu University, Chengdu 610000, China*

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

Received 30 December 2017; Revised 9 October 2018; Accepted 21 November 2018; Published 3 December 2018

Academic Editor: Mohammed Chadli

Excessive detectors, high time complexity, and loopholes are main problems which current negative selection algorithms have face and greatly limit the practical applications of negative selection algorithms. This paper proposes a real-valued negative selection algorithm based on clonal selection. Firstly, the algorithm analyzes the space distribution of the self set and gets the set of outlier selves and several classification clusters. Then, the algorithm considers centers of clusters as antigens, randomly generates initial immune cell population in the qualified range, and executes the clonal selection algorithm. Afterwards, the algorithm changes the limited range to continue the iteration until the non-self space coverage rate meets expectations. After the algorithm terminates, mature detector set and boundary self set are obtained. The main contributions lie in (1) introducing the clonal selection algorithm and randomly generating candidate detectors within the stratified limited ranges based on clustering centers of self set; generating big-radius candidate detectors first and making them cover space far from selves, which reduces the number of detectors; then generating small-radius candidate detectors and making them gradually cover boundary space between selves and non-selves, which reduces the number of holes; (2) distinguishing selves and dividing them into outlier selves, boundary selves, and internal selves, which can adapt to the interference of noise data from selves; (3) for anomaly detection, using mature detector set and boundary self set to test at the same time, which can effectively improve the detection rate and reduce the false alarm rate. Theoretical analysis and experimental results show that the algorithm has better time efficiency and detector generation quality according to classic negative selection algorithms.

1. Introduction

The negative selection algorithm (NSA) first proposed by American scholar Forrest [1] is one of the most important anomaly detection algorithms in artificial immune field. The idea of negative selection algorithm comes from the negative selection behavior of T lymphocytes in immune tolerance of thymus [2]. An immune explanation for this behavior is as follows. In the thymus tolerance issue, T lymphocytes which identify self antigens will be in apoptosis or inactivated, and those cells which do not identify selves will mature after a period of tolerance and exercise their immune function in peripheral lymphoid tissues. The proposition of negative selection algorithm greatly promotes research and application in the anomaly detection field of artificial immune systems. Specifically, the idea of negative selection algorithm is often applied in these areas such as fault detection, virus detection, network intrusion detection, and machine learning [2–4].

The negative selection algorithm put forward by Forrest used binary string to represent antigen and antibody adopted r-continuous matching rule to compute matching degree between antibody and antigen and was successfully applied to anomaly detection system. Then, Balthrop et al. [5] pointed out holes of r-continuous matching rule and put forward the improved r-chunk matching mechanism. Zhang [6] proposed r-variable negative selection algorithm, and He [7] proposed negative selection algorithm with variable-length detectors.

But the binary representation has insufficiency in dealing with numeric data and multidimensional space problems;

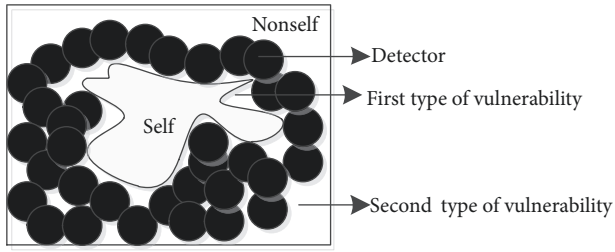

Figure 1: Two kinds of holes.

Gonzalez and Dasgupta [8] proposed a real-valued negative selection algorithm with position-developed detectors (RNSA). The study of real-valued negative selection algorithm is introduced in the following. The work in [9] introduced the super ellipsoid detector into the negative selection algorithm, the work in [10] introduced the super rectangle detector, and they needed less detectors to achieve the same coverage compared to spherical detectors. Ji [11] and Ji and Dasgupta [12] put forward a variable-sized real-valued negative selection algorithm (V-Detector). The algorithm dynamically determined the radius of a mature detector by calculating the smallest distance between the center of candidate detector and the self antigen. The work in [13, 14] proposed a negative selection algorithm based on grid. The algorithms adopted certain method to divide space into several grids, which reduced the tolerance range of randomly generated candidate detectors. The work in [15] proposed a negative selection algorithm based on hierarchical clustering of self sets, which improved the coverage rate of non-self space for detectors through the self set preprocessing. The work in [16] divided detectors into self detectors and non-self detectors which cover self space and non-self space, respectively, and used self detector instead of self elements to reduce the computational cost.

Some studies introduced other artificial intelligence technologies into the negative selection algorithm to improve the efficiency of detector generation. The work in [17–19] proposed a negative selection algorithm based on genetic principles. The work in [20] combined particle swarm optimization strategy and the negative selection algorithm. The work in [21] introduced the wavelet transform to the negative selection algorithm.

Many problems in negative selection algorithms such as the representation of detectors, the affinity calculation method, and detector generation mechanisms were studied. There are many achievements, but some problems have not been solved effectively.

(1) Loopholes. Loopholes generally refer to all the non-self space which is not covered by detectors in the negative selection algorithms and can be specifically divided into two categories: the first kind of vulnerability is the non-self space which cannot be covered by detectors in theory due to restricted detector coding and implementation way; the second is the non-self space which is not covered by detectors but theoretically feasible. Figure 1 illustrates two kinds of holes when the detector size is immutable.

In order to obtain good detection performance, it is necessary to reduce loopholes. For the first kind of holes, regardless of specific differences of detector representations (binary string or real value expression), there are two main solutions: one is the variable detector shape scheme [9, 10] and the second is variable detector size scheme [11, 12]. As shown in Figure 2, two schemes can eliminate the first kind of vulnerability in theory. Although the first solution is possible in theory, its implementation is very difficult. In comparison, the second solution is more feasible, and its concrete implementation effect is better [11–16].

For the second kind of loopholes, there are two solutions at present. One is the exhaustive method to generate all the detectors [22], and the second is to randomly generate detectors which satisfy certain requirements [11, 12]. Main problems of the first solution include the high time complexity and excessive detectors. It can only be applied to the situation that the number of detectors may be limited and cannot be used in many practical applications (for instance, detectors use real-valued coding, and the number is unlimited as a result). The second solution is widely used in the practical applications, but the scheme has random uncertain problems and cannot completely eliminate the existence of loopholes. While the second type of vulnerability can be covered by detectors in theory, the effective covering of holes which locate in the areas between self and non-self space is very difficult. Figure 3 shows a schematic diagram. In the figure, the self space is expressed by a self element, and it can be found that, even in the simplest environment, using a finite number of detectors cannot completely cover all the loopholes in the second category in theory.

(2) Too Many Detectors. The work in [11, 22] pointed out that, in the negative selection algorithm proposed by Forrest et al., the detector generation efficiency is very low. Candidate detectors become mature by negative selection. It is assumed that N_s is the size of self training set, P' is the matching probability between any antigen and antibody, and P_f is the failure rate (the probability of a non-self antigen cannot be matched by any antibody). The number of candidate detectors is $N_c = -\ln(P_f)/(P'(1-P')^{N_s})$, and the algorithm's time complexity is $O(N_c \bullet N_s)$. Therefore, with the increase of the size of the training set, the number of candidate detectors increases exponentially, the time cost in detector generation phase is higher, and the time cost in detection phase is higher as well. In addition, excessive detectors can cause redundant cover between detectors. Some negative selection algorithms made mature detectors to merge or cluster, which formed big-radius detectors instead of original detectors [19].

(3) Contradiction between Detection Rate and False Alarm Rate. For anomaly detection, high detection rate and low false alarm rate are two directions. Existing negative selection algorithms focused on improving the efficiency of detector generation in order to cover the non-self space as much as possible. They did not consider the balance of the detection rate and the false alarm rate, which lacked a good adjustment mechanism of the two rates. The classic algorithm V-Detector [11, 12] proposed two solutions, which were point-aware

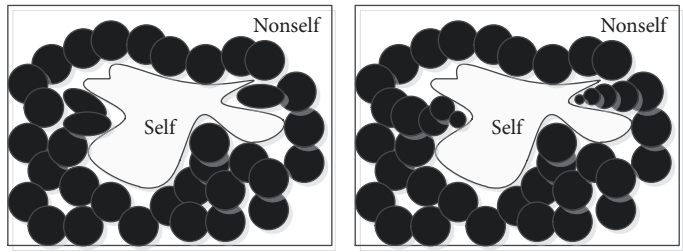

Figure 2: Solutions for the first kind of holes.

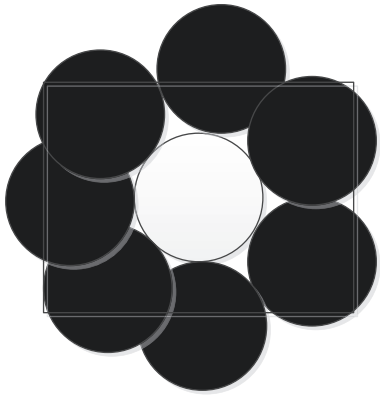

Figure 3: A situation of the second type of holes.

and boundary-aware, respectively. The two solutions have problems. The detection rate of point-aware solution is low, and the false alarm rate of boundary-aware solution is too high.

(4) Not Considering Noise Data in Self Training Set. In many anomaly detection applications, self training set contains noise data. And negative selection algorithms are on the basis of reliable training data; it is widespread that they cannot adapt to the existence of noise data.

How to generate efficient detector set is the key to the negative selection algorithms. This paper proposes a real-valued negative selection algorithm based on clonal selection, named CB-RNSA. The main contributions lie in (1) introducing the clonal selection algorithm and randomly generating candidate detectors within the stratified limited ranges based on clustering centers of self set; generating big-radius candidate detectors first and making them cover space far from selves, which reduces the number of detectors; then generating small-radius candidate detectors and making them gradually cover boundary space between selves and non-selves, which reduces the number of holes; (2) distinguishing selves and dividing them into outlier selves, boundary selves, and internal selves, which can adapt to the interference of noise data from selves; (3) for anomaly detection, using mature detector set and boundary self set to test at the same time, which can effectively improve the detection rate and reduce the false alarm rate.

In this paper, the rest of the sections are arranged as follows. The second section introduces the background of this paper, including the typical real-valued negative selection algorithm RNSA, the variable-sized real-valued negative selection algorithm (V-Detector), and immune theory-clonal selection algorithm. The third section introduces the implementation strategies of the algorithm, including the basic idea, outlier selves discovery mechanism, cluster discovery mechanism, clonal selection mechanism, etc. The fourth section analyzes the algorithm, including time complexity analysis and detector self-reaction rate analysis. The fifth section verifies the effectiveness of the algorithm through experiments which use 2D comprehensive data sets and UCI data sets and compare with classic negative selection algorithms. Conclusions are given in the sixth section.

2. Related Work

2.1. Basic Concepts of NSA. The system state can be expressed by the feature vector $x = (x_1, x_2, \ldots, x_n)$. n is the system dimension, and each feature of the vector is normalized to the real-valued interval $[0, 1]$. The entire state space of the system can be expressed as $\Omega = [0, 1]^n$. The system state space can be further divided into self space *Self* and non-self space *Nonself*. In anomaly detection, self space *Self* is composed of states when the system is normal, and non-self space *Nonself* consists of states when the system is abnormal.

In artificial immune systems, antigens are on behalf of the entire state of the system, the self set represents self space of the system, and the non-self set represents non-self space, which are defined respectively as follows.

Definition 1 (antigens). $Ag = \{ag \mid ag = \langle x, r_s\rangle = \langle x_1, x_2, \ldots, x_n, r_s\rangle,\ x_i \in [0, 1],\ 1 \le i \le n,\ r_s \in [0, 1]\}$ represents all the samples in the space. The parameter ag is an antigen in the set and consists of two parts, x and r_s. The parameter x is the position of sample ag in the real-valued space. The parameter r_s is the radius of ag and represents change threshold. Therefore, ag is a hypersphere in space.

Definition 2 (self set). $Self \subset Ag$ represents all the normal samples in the antigen set. $Self = \{ag \mid ag \subset Ag \cap ag \text{ is normal}\}$.

Definition 3 (non-self set). $Nonself \subset Ag$ represents all the abnormal samples in the antigen set. $Nonself = \{ag \mid ag \subset$

$Ag \cap ag$ is abnormal}. The self/non-self has different meanings in different areas. For network intrusion detection, non-self represents network anomalies, and self represents normal network activities. For virus detection, non-self represents virus signatures, and self represents legal codes. *Self* and *Nonself* meet (1).

$$\begin{aligned} Self \cap Nonself &= \emptyset, \\ Self \cup Nonself &= Ag \end{aligned} \tag{1}$$

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

Definition 5 (detector set). $= \{d \mid d = \langle \mathrm{y}, r_{\mathrm{d}} \rangle = \langle y_1, y_2, \ldots, y_n, r_d \rangle,\ y_j \in [0,1],\ 1 \le j \le n,\ r_d \in [0,1]\}$. The parameter d is one of the elements from the detector set, and its structure is the same as antigen which consists of two parts, y and r_d, respectively. The parameter y represents the position of detector d, and the parameter r_d is the radius of detector d.

Definition 6 (matching rule). $f(ag, d)$ represents the affinity between antigen ag and detector d, that is, the matching degree between data structures. In real-valued space, we can measure the affinity by calculating the distance between two feature vectors, which is usually expressed by Minkowski distance. For two vectors x and y in n-dimensional space, m-order Minkowski distance function is as

$$f_d(ag, d) = \left(\sum_{i=1}^{n} |x_i - y_i|^m \right)^{1/m} \tag{2}$$

Minkowski distance is often called the m-order norm. For real-valued negative selection algorithms, it is thought that, for different values of m, test range of detectors has different geometry. This paper adopts the 2-order norm to express the matching rule, namely, the Euclidean distance. Then, (2) is rewrote as

$$f(ag, d) = \sqrt{\sum_{i=1}^{n} (ag.x_i - d.y_i)^2} \tag{3}$$

Definition 7 (detection system). *DS* consists of three parts, namely, $DS = (D, Self, f)$. In the process of detector generation, if $f(ag, d) \le r_s + r_d$, detector d causes the immune self-reaction and cannot be a mature detector. In the process of detector testing, if $f(ag, d) \le r_d$, detector d recognizes ag as a non-self. It is assumed that A is mapping from self set *Self* and candidate detector d to a classification $\{0, 1\}$, where 0 indicates that d is immature, and 1 means that d is mature. A is such a function as

$$A(Self, d) = \begin{cases} 0, & \exists ag' \in Self, f(d, ag') \le r_s + r_d \\ 1, & otherwise \end{cases} \tag{4}$$

Suppose B is mapping from detector set D and antigen ag to be identified to a classification $\{0, 1\}$, where 0 indicates that ag is self, and 1 means that ag is non-self. B is such a function as

$$B(D, ag) = \begin{cases} 1, & \exists d' \in D, f(d', ag) \le r_d \\ 0, & otherwise \end{cases} \tag{5}$$

When the detection system is working, *TP* is set as correct positive, that is, the number of non-selves correctly recognized by detectors; *TN* is set as correct negative, that is, the number of selves correctly recognized by detectors. Two kinds of errors may occur. False positive *FP* happens when a self sample is identified as a non-self. False negative *FN* occurs when a non-self sample is identified as a self. They can be defined as follows. Given a test set, $\mathrm{U}_{test} \sqsubset Ag$ which consists of two sets, self S_{test} and non-self N_{test}. When *FP* happens, the pattern collection ε^+ can be defined as

$$\varepsilon^+ = \{ag \mid ag \in S_{test} \wedge B(D, ag) = 1\} \tag{6}$$

When *FN* happens, the pattern collection ε^- can be defined as

$$\varepsilon^- = \{ag \mid ag \in N_{test} \wedge B(D, ag) = 0\} \tag{7}$$

When *TP* happens, the pattern collection ζ^+ can be defined as

$$\zeta^+ = \{ag \mid ag \in N_{test} \wedge B(D, ag) = 1\} \tag{8}$$

When *TN* happens, the pattern collection ζ^- can be defined as

$$\zeta^- = \{ag \mid ag \in S_{test} \wedge B(D, ag) = 0\} \tag{9}$$

Definition 8 (detection rate). *DR* is the ratio of the number of non-self samples being correctly identified by detectors to the total number of non-selves and is expressed as

$$DR = \frac{TP}{TP + FN} \tag{10}$$

Definition 9 (false alarm rate). *FAR* is the ratio of the number of self samples being wrongly identified by detectors to the total number of selves and is expressed as

$$FAR = \frac{FP}{FP + TN} \tag{11}$$

2.2. RNSA. The artificial immune system removes those immune cells which respond to selves through the negative selection algorithm, so as to realize the self tolerance. RNSA adopts real-valued vector to describe the configuration space and uses fixed-size detectors. The termination condition of the algorithm is reaching the default number of detectors [8]. Algorithm 1 shows the process of the negative selection algorithm.

Figure 4 shows the relationships between the self set, the non-self set, the detector set, etc. Ag is the antigen space; U_d is the detector space. Although in many cases $Ag = U_d$, we draw them, respectively, in order to describe clearly. *Self* is

```
Procedure. The negative selection algorithm RNSA
Begin
   Generate a large number of candidate detectors at random;
   While a given size detector set has not been generated do
      Calculate affinities between the candidate detector and every self element;
      If the candidate matches any element of self set;
      Then clear the candidate;
      Else put the candidate in the detector set;
   End;
   Use the collection of resistant detectors to test abnormal variations;
End.
```

Algorithm 1: The process of RNSA.

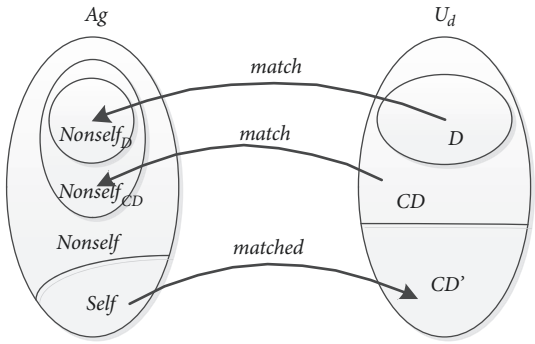

Figure 4: Relationships between sets.

the self set, *Nonself* is the non-self set, *CD* is the candidate detector set, *D* is the detector set which is selected from *CD*, $Nonself_{CD}$ is non-selves which are identified by candidate detectors, and $Nonself_D$ is non-selves which are identified by detectors. Therefore, collection of holes is $Nonself\text{-}Nonself_{CD}$, and collection of non-selves which cannot be tested by detectors is $Nonself\text{-}Nonself_D$.

2.3. V-Detector. V-Detector uses vectors in real-valued space to express detectors and antigens, and the radius of detectors is variable [11, 12]. Firstly, the algorithm randomly generates the center y of the candidate detector $d = \langle y, r_d \rangle$ and then calculates distance *f(ag, d)* between y and every self $ag = \langle x, r_s \rangle$ in the training set. If $\min(f(ag, d)) > r_s$, accept the detector. The radius of the detector is calculated by the following formula:

$$r_d = \begin{cases} \min\left(f\left(ag, d\right)\right) - r_s, & if \text{ point} - \text{aware} \\ \min\left(f\left(ag, d\right)\right), & if \text{ boundary} - \text{aware} \end{cases} \quad (12)$$

Point-aware and boundary-aware are two technologies to determine detector radius. The algorithm of boundary-aware has high detection rate and high false alarm rate. This is because the boundary-aware makes detection range of detectors and coverage of some selves overlap. Non-self space is basically covered, which improves the detection rate, but is contrary to the assumptions of negative selection algorithm "vectors close to selves are selves, and vectors far from selves are non-selves [1, 11]", which can lead to high false alarm rate. Therefore, this article uses the point-aware technology.

Figure 5 shows the contrast of RNSA and V-Detector when the expectation coverage is 50%. The self data are top 25 elements of classification "Iris-versicolor" from IRIS data set [23]. In order to display conveniently in two-dimensional space, we take the element's sepalL and sepalW as self antigen's properties. Blue filled circles are self elements, cyan filled circles are mature detectors, and unfilled area is holes. In RNSA, the detector size is constant and is difficult to determine accurately, which causes many loopholes in the non-self space and low detection rate. In V-Detector, the detector size is variable, and the algorithm makes big detectors cover most of the non-self space and small detectors cover holes, which not only reduces the number of detectors, but also reduces the number of vulnerabilities. But, these two algorithms have faced problems proposed in the introduction section, such as loopholes which influence the detection rate, too many candidate detectors, redundant cover between mature detectors, and high time cost of testing.

2.4. Clonal Selection Algorithm. Clonal selection principle is used to illustrate basic features of the immune response to antigen stimulation in the immune system [24–26]. When external bacteria or virus invades the body, B cells begin to a large number of cloning and destroy invaders. Those cells who can identify antigens will achieve the purpose of hyperplasia by asexual reproduction according to the degree of recognition. The higher the affinity between cells and antigens is, the more cells can produce offspring. In the process of cell division, individual cells also experience a variation process, which results in higher affinity with antigens; the higher the affinity between parent cells and antigens is, the less parent cells experience variation. Algorithm 2 shows the process of clonal selection algorithm.

3. The Algorithm Theory

3.1. The Process of the Algorithm. The main idea of the algorithm is as follows. Firstly, the algorithm analyzes the space distribution of the self set and gets the set of outlier selves and several classification clusters. Then, the algorithm considers centers of clusters as antigens, randomly generates initial immune cell population in the qualified range, and executes the clonal selection algorithm. That is to say, the algorithm carries out the immune selection operation, clonal

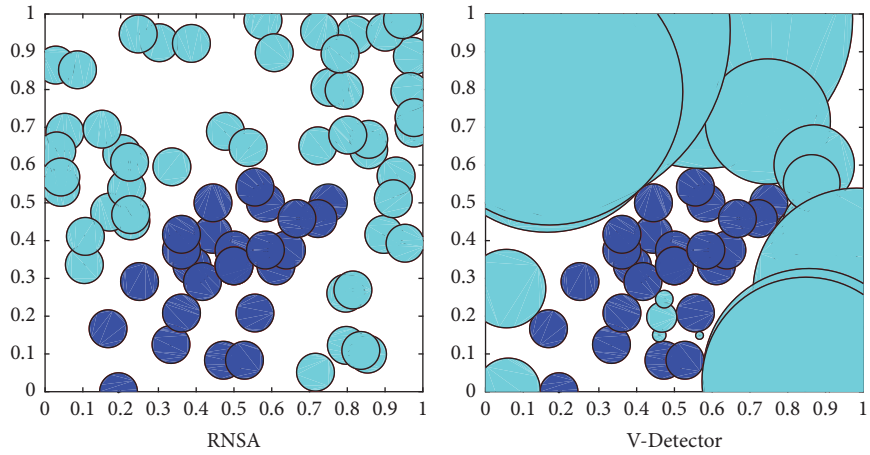

Figure 5: Contrast of RNSA and V-Detector.

```
Procedure. Clonal selection algorithm
Begin
   Randomly generate a population of immune cells;
   While not meet the convergence condition do
      While not search all antigens do
         Choose those cells which have high affinity with antigen;
         Generate copies of immune cells; the higher the affinity is, the more copies are.
         Mutate according to the affinity; the higher the affinity is, the smaller the variation is;
      End;
   End;
End.
```

Algorithm 2: The process of clonal selection algorithm.

amplification operation, and hypermutation operation on the immune cell population. The affinity between antigens and immune cells is inversely proportional to their distance, and the convergence condition of clone selection algorithm is to achieve the expected coverage of non-self space. At this point, the finite range for generating immune cells is far from self space, where the coverage rate is low. At the end of the clonal selection algorithm, immune cell population is viewed as candidate detectors of the first level. The radius of the candidate detector is dynamically determined by computing the distance between its center and the closest self, and then the candidate detector joins the mature detector set through tolerance. In addition, the algorithm adds selves which are closest to detectors into boundary self set. The candidate detectors of this level have biggest radius and cover non-self space away from selves. Afterwards, the algorithm changes the limited scope to make the range of next level more convergent. The algorithm continues to consider centers of clusters as antigens, randomly generates initial immune cell population in the qualified scope, and executes the clonal selection algorithm. When the clonal selection algorithm ends, immune cell population is the candidate detectors of the second level. The candidate detectors of this level have larger radius, are close to detectors of the first level, and cover the non-self space a bit near selves. Then repeat the process until candidate detectors cover non-self space close to selves which are, namely, the boundary areas between self space and non-self space. When the algorithm terminates, mature detector set and boundary self set are obtained. Algorithm 3 shows the process of CB-RNSA.

The main contributions lie in (1) introducing the clonal selection algorithm and randomly generating candidate detectors within the stratified limited ranges based on clustering centers of self set; generating big-radius candidate detectors first and making them cover space far from selves, which reduces the number of detectors; then generating small-radius candidate detectors and making them gradually cover boundary space between selves and non-selves, which reduces the number of holes; (2) distinguishing selves and dividing them into outlier selves, boundary selves, and internal selves, which can adapt to the interference of noise data from selves; (3) for anomaly detection, using mature detector set and boundary self set to test at the same time, which can effectively improve the detection rate and reduce the false alarm rate. Theoretical analysis and experimental results show that the algorithm has better time efficiency and detector generation quality according to classic negative selection algorithms.

Figures 6 and 7 show the contrast between CB-RNSA, RNSA, and V-Detector. The self data are top 25 elements of classification "Iris-versicolor" from IRIS data set [23]. Blue filled circles are self elements, cyan filled circles are mature

Procedure. CB-RNSA
Input: self training set *Train*, expected coverage rate p_0
Output: detector set D, boundary self set $Self_o$, outlier self set $Self_d$
n_0: the sampling frequency of non-self space, $n_0 > \max(5/p_0, 5/(1-p_0))$
i: the number of non-self samples
m: the number of non-self samples which are covered by detectors
CD: candidate detector set $CD = \{d \mid d = \langle y, r_d \rangle = \langle y_1, y_2, \ldots, y_n, r_d \rangle, y_j \in [0,1], 1 \le j \le n, r_d \in [0,1]\}$
Clusters: cluster set *Clusters* = {*cluster*}
l: the number of candidate detector level
Begin
  Initialize self training set *Train*, $i = 0$, $m = 0$, $CD = \emptyset$, $Self_o = \emptyset$, $n_0 = \lceil \max(5/p_0, 5/(1-p_0)) \rceil$;
  Initialize outlier self set $Self_d$ according to Procedure outlier selves discovery algorithm;
  Initialize cluster set *Clusters* according to Procedure clusters discovery algorithm;
  While l does not reach the maximum number of levels for candidate detectors do
    Consider centers of *Clusters* as antigens, randomly generate initial immune cell population in the qualified range;
    While true do
      Select immune cells;
      Generate copies of immune cells;
      Mutate according to affinities;
      Compute distances between mutated individual d_{new} and every self in the training set *Train*;
      If d_{new} is recognized by some self Then discard d_{new};
      Else
        Find the closest self ag to d_{new}, and add it to boundary self set $Self_o$;
        i ++;
        Compute distances between d_{new} and every detector in the detector set D;
        If d_{new} is not identified by any detector Then put it into the candidate detector set CD;
        Else m ++;
      End if;
    If the number of non-self samples i reaches the sample times n_0 Then
        Compute current coverage rate p;
        If p reaches the expected coverage rate p_0, break;
        Else incorporate candidate detector set CD with D, reset i, m, CD;
    End if;
  End;
  l ++;
  Changes the limited range of candidate detectors;
 End;
End.

Algorithm 3: The process of CB-RNSA.

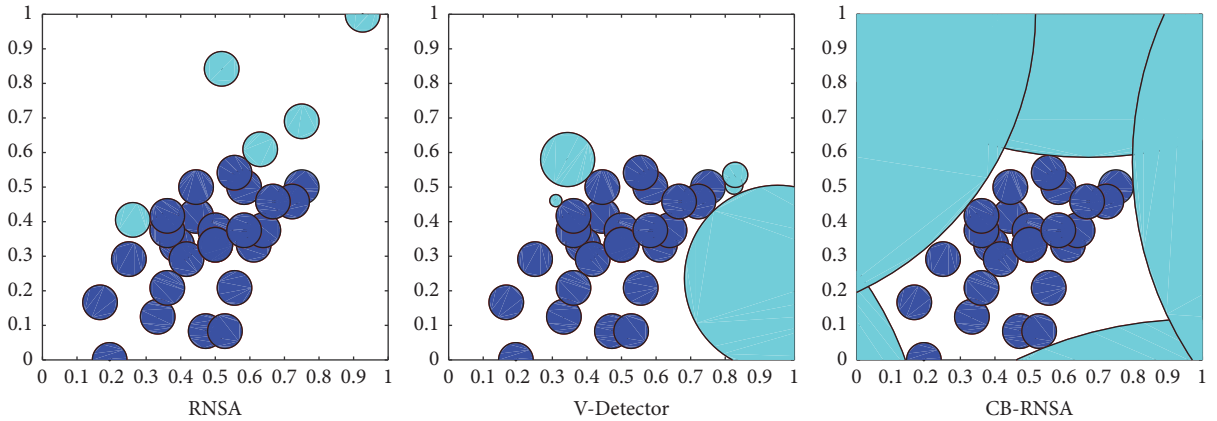

Figure 6: Contrast between CB-RNSA, RNSA, and V-Detector (five mature detectors are produced at the same time).

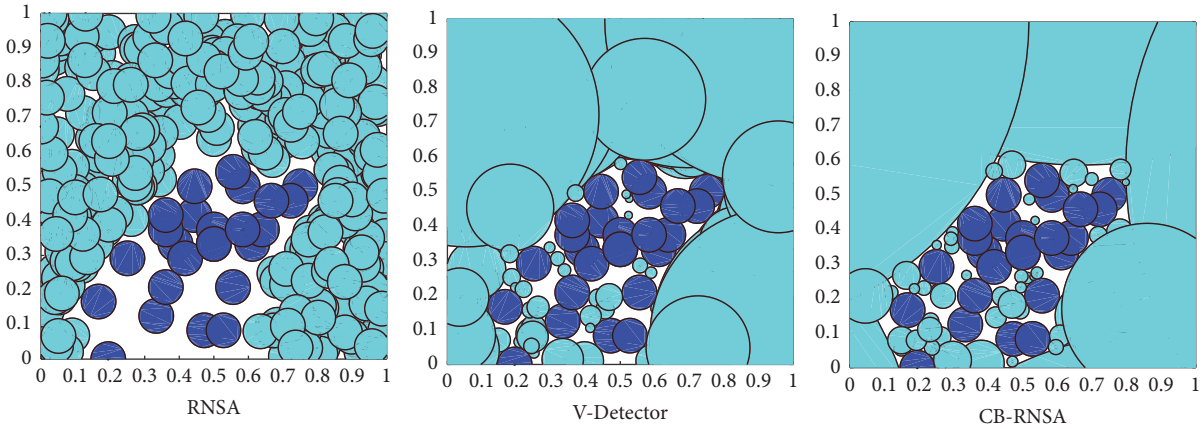

Figure 7: Contrast between CB-RNSA, RNSA, and V-Detector (to achieve the expected coverage rate 90%, RNSA, V-Detector, and CB-RNSA need 357, 81, and 35 mature detectors, respectively, where the self radius is 0.05, and the detector radius of RNSA is 0.05).

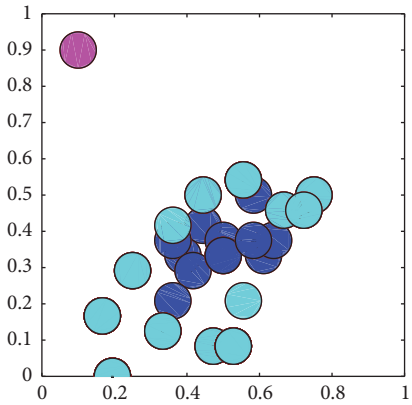

Figure 8: Self classification.

detectors, and unfilled area is holes. In RNSA and V-Detector, with rise of coverage rate, redundant cover between mature detectors in non-self space increases, which causes excessive detector quantity and unnecessary self tolerance. In CB-RNSA, because the clonal selection algorithm is introduced to limit the range of randomly generated candidate detectors, it is preferred that detectors are generated in space of low coverage, which reduces the number of detectors and redundancy.

3.2. Classification of Selves. Most of the negative selection algorithms do not distinguish between selves. But for the continuous space of selves, information within different self is different. We divide selves into three groups, outlier selves, boundary selves, and internal selves, as shown in Figure 8. Magenta filled circles are outlier selves, cyan filled circles are boundary selves, and blue filled circles are internal selves.

Suppose the collection of elements *Nei(ag, r)* whose distance with self *ag* is less than *r* is expressed as (13), and they are called neighbors.

$$Nei(ag, r) = \left\{ag' \mid ag' \in Self \wedge f\left(ag, ag'\right) < r\right\} \quad (13)$$

Definition 10 (outlier self set). $Self_d \subset Self$. The outliers may be caused by noise data. $Self_d = \{ag \mid ag \in Self \wedge |Nei(ag, r)| < \rho \bullet |Self|\}$ means that when the number of neighbors of a self is less than a certain value, the self is classified as an outlier. ρ is the parameter for outlier.

Definition 11 (boundary self set). $Self_o \subset Self$, and they are distributed in edges of self space and non-self space. The boundary self quantity is far less than the number of internal selves, and a lot of false positives and omissions appear in the borders. Because self space and non-self space are complementary, we use detector set to define boundary self set; that is to say, the self which is closest to a detector is boundary. $Self_o = \{ag \mid ag \in Self \wedge (\exists d \in D, f(ag, d) = \min(\forall ag' \in Self, f(ag', d)))\}$.

Definition 12 (internal self set). $Self_i \subset Self$, and they are surrounded by boundary selves. $Self_i = Self - Self_d - Self_o$.

3.3. Outlier Selves Discovery Mechanism. As one of the important research fields of knowledge discovery, there are many effective outlier detection algorithms at present [27–29]. This paper adopts the algorithm based on distance proposed by Knorr [27], and Algorithm 4 shows the process of the algorithm.

3.4. Clusters Discovery Mechanism. In anomaly detection, it is thought that most of these data are normal, and abnormal data are the minority. In this paper, the algorithm first analyzes the space distribution of selves and performs a clustering pretreatment on the self set. Similar selves are classified in the same cluster, and then a number of clusters are generated. Centers of clusters are used as a benchmark to generate candidate detectors. The purpose of self set clustering is to determine the randomly generation scope of candidate detectors, and candidates are generated in the qualified range in order to avoid detector redundancy in high coverage.

Definition 13 (cluster). $cluster = \{c \mid c = \langle x, r_c, C\rangle = \langle x_1, x_2, \ldots, x_n, r_c, \{ag \mid ag \in Self\}\rangle, x_i \in [0, 1], 1 \leq i \leq$

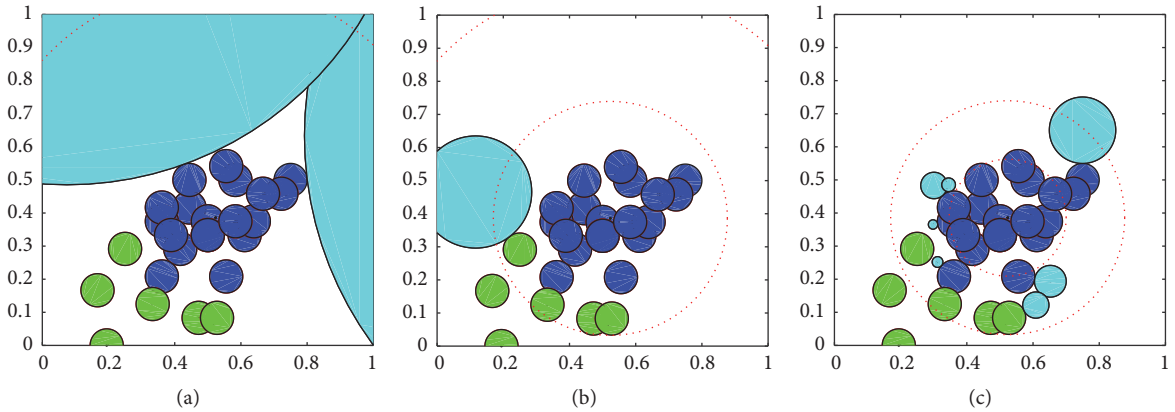

Figure 9: Limited scopes and generations of candidate detectors (based on the cluster of blue filled circles).

```
Procedure. Outlier selves discovery algorithm
Input: the self training set Train
Output: outlier self set Self_d
Begin
  While self ag is not detected do
    m = 0;
    for other self ag' in the self training set Train do
      Compute the distance between ag and ag' f(ag, ag');
      If f(ag, ag') < r then m ++;
    End;
    If m < ρ • |Self| then add ag into the outlier self set Self_d;
  End;
End.
```

Algorithm 4: The process of outlier selves discovery algorithm.

$n\}$. The parameter x is the center vector of the cluster in n-dimensional space, the parameter r_c is the radius of the cluster, and the parameter C is the collection of selves within the cluster. r_c is computed by the following formula:

$$cluster.r_c = \frac{\left(\sum_{ag \in Self} \sum_{ag' \in Self - ag} f\left(ag, ag'\right)\right)}{\left(|Self| \bullet \left(|Self| - 1\right)\right)} \tag{14}$$

The algorithm randomly selects a self as an initial element for a cluster at first and then judges whether any element in the cluster and other self ag are neighbors, that is to say, whether meeting $\exists ag' \in Cluster.C, ag \in Nei(ag', r_c)$, if so, self ag will be within the cluster. After other selves are judged, if there is a self which does not belong to any cluster, that means $Self - \sum_{cluster' \in Clusters}\{ag \mid ag \in cluster'.C\} \neq \emptyset$, the algorithm continues to randomly select a self as an initial element of a new cluster. The above operations are performed until all the selves belong to a cluster. The cluster center x is computed by the following formula (15). Algorithm 5 shows the process of clusters discovery algorithm.

$$cluster.x_i = \frac{\left(\sum_{ag \in cluster.C} ag.x_i\right)}{n} \tag{15}$$

3.5. Clonal Selection Mechanism. Clonal selection algorithm is often used for solving the optimization problems. This paper adopts the clonal selection algorithm to search optimal detectors in the non-self space. The center of each cluster is viewed as an antigen. Initial immune cell population is randomly generated in limited scope, and the clonal selection algorithm is performed on the group. The limits control the position of vector x of immune cell ag in each dimension and are defined as a super sphere loop based on the center of the cluster $cluster.x$. The distance between immune cell and the center of the cluster should be limited between $[r_{low}, r_{high}]$, that means, $r_{low} \leq f(ag, cluster) \leq r_{high}$. r_{low} is the minimum distance between immune cell and cluster center, $r_{low} = 0$, and r_{high} is the maximum distance between these two, $r_{high} = \sqrt{n}$. Candidates will be generated by layer. According to the detector radius, from big to small, a detector of larger coverage will be in priority to be produced, to avoid repeated coverage with existing mature detectors and achieve less detectors covering as much as possible non-self area. Set level is l, and the limited scope of the l level is to meet (16), where the value of l should be satisfied (17); that is, $\max(l) = \lceil \log_2(\sqrt{n}/r_s) + 1 \rceil$.

$$\frac{\sqrt{n}}{2^l} \leq f\left(ag, cluster\right) \leq \frac{\sqrt{n}}{2^{l-1}} \tag{16}$$

$$\frac{\sqrt{n}}{2^{l-1}} \geq r_s \tag{17}$$

Figure 9 shows limited scopes of the first level, second level, and third level of candidate detectors based on one of

```
Procedure. Clusters discovery algorithm
Input: the self training set Train
Output: cluster set Clusters={cluster}
Begin
   While self ag is not belong to any cluster do
      Generate a new cluster for ag;
      For other self ag′ in selves which are not classified do
       If ag′ is neighbor to any element in cluster, then put ag′ into cluster;
      End;
   End;
End.
```

Algorithm 5: The process of clusters discovery algorithm.

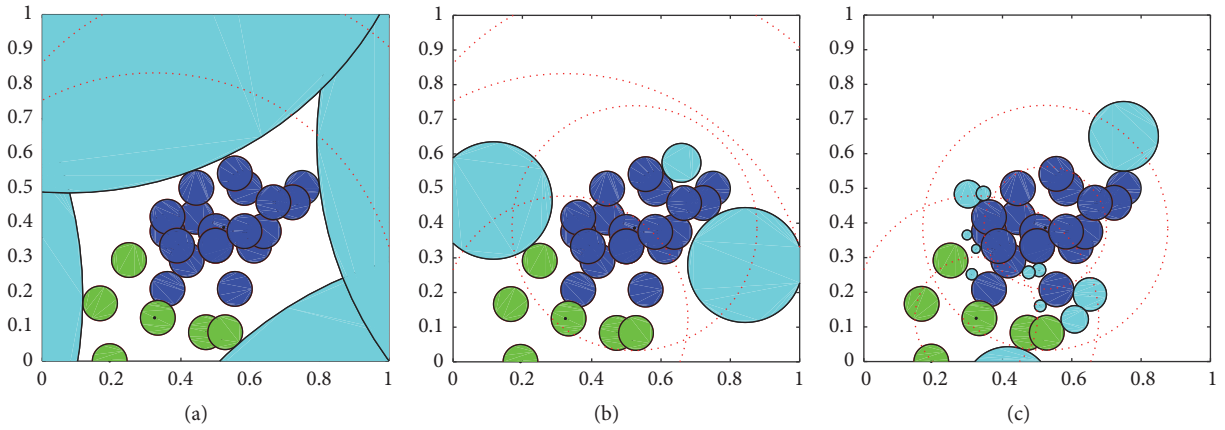

Figure 10: Limited scopes and generations of candidate detectors.

the clusters and generations of candidate detectors. Figure 10 shows limited scopes of the top three levels based on all the clusters and the generation of candidate detectors. Blue filled circles and green filled circles are self elements, cyan filled circles are mature detectors, and regions between two concentric dotted circles are the randomly generation scopes.

In the clonal selection algorithm, the main operations include immune selection Ts, clonal amplification Tc, and hypermutation Tm. Immune selection operation is to choose a certain number of immune cells with high-affinity in order to search in more valuable space. In this algorithm, because immune cells should cover the non-self space as much as possible, it is not necessary to choose by affinity, and the population of all immune cells go into the next operation. Set AB as the population of immune cells and AB' as the population after this operation, and the probability of selection is as

$$P\left(Ts\left(AB\right) = AB'\right) = 1 \tag{18}$$

Clonal amplification operation simulates cloning mechanism of the immune response, and the higher affinity with antigens the cell has, the more offspring the cell can produce. In this algorithm, the number of copies nc is related to the level l where candidate detector is. nc is computed by (19), where nc_{max} is the maximum number of copies and nc_{min} is the minimum.

$$nc = nc_{min} + \left(nc_{max} - nc_{min}\right)\left(1 - \frac{1}{l}\right) \tag{19}$$

The above formula reflects the clone expansion scales for immune cells in different limited scopes. When the hierarchy is small, candidate detectors are in low-coverage areas and the radius is larger. We hope to cover more non-self space with less detectors, so the cloning scale is smaller. When the hierarchy is big, candidate detectors are in high coverage areas and the radius is smaller. We hope to cover holes in the boundary of self space and non-self space, so the cloning scale is larger.

Hypermutation operation is to produce immune cells with higher affinity and enhance the diversity of the population. Mutation operator usually can produce small disturbance and can also produce a wide range of disturbance, which makes the mutation have abilities of local search and global search and makes the algorithm have stronger optimal search performance. In this paper, Gauss mutation is adopted and the formula is as

$$ag'.xi = ag.xi + \alpha \bullet N\left(0, 1\right) \tag{20}$$

where ag' is the mutated immune cell, $N(0, 1)$ is the Gaussian random variable, the mean is 0, and the deviation is

1. α is the control parameter to adjust the variation amplitude. In this paper, a dynamic adaptive α is adopted and only related to candidate detector level l. The mutation mechanism is as (21). In the process of the generation of candidate detectors, when the level is small, individuals search with larger probability, which is good for global search. When the level is large, individuals search with smaller probability, which is more conducive to local search.

$$\alpha = \frac{1}{\left(1 + e^{l-k_0}\right)} \tag{21}$$

k_0 is the cut-off point for α that means the cut-off for the algorithm from global search of big probability to local search of small probability.

After immune cells perform mutation, they also need to meet (16). So, new cells should meet the formula, or they will be discarded. This will produce a large number of useless cells, which needs a large amount of calculations. Set $ag.x$ as the origin, we use n-spherical coordinates to rewrite (16) by (22).

$$\begin{aligned} ag'.x_1 &= ag.x_1 + \rho \bullet \cos\left(\theta_1\right) \\ ag'.x_2 &= ag.x_2 + \rho \bullet \sin\left(\theta_1\right)\cos\left(\theta_2\right) \\ ag'.x_{n-1} &= ag.x_{n-1} + \rho \\ &\quad \bullet \sin\left(\theta_1\right)\sin\left(\theta_2\right)\ldots\cos\left(\theta_{n-1}\right) \\ ag'.x_n &= ag.x_n + \rho \bullet \sin\left(\theta_1\right)\sin\left(\theta_2\right)\ldots\sin\left(\theta_{n-1}\right) \end{aligned} \tag{22}$$

where the parameter ρ is random variable between $[\sqrt{n}/2^l, \sqrt{n}/2^{l-1}]$, the parameters $\theta_1, \theta_2, \ldots, \theta_{n-2}$ are random variables between $[0, \pi]$, and the parameter θ_{n-1} is random variable between $[0, 2\pi]$. They are expressed as (23), (24), and (25).

$$\rho = \frac{\sqrt{n}}{2^l} + \left(\frac{\sqrt{n}}{2^{l-1}} - \frac{\sqrt{n}}{2^l}\right) \bullet \alpha \bullet N(0,1) \tag{23}$$

$$\theta_i = \pi \bullet \alpha \bullet N(0,1) \quad 1 \le i \le n-2 \tag{24}$$

$$\theta_{n-1} = 2\pi \bullet \alpha \bullet N(0,1) \tag{25}$$

3.6. Anomaly Detection. Anomaly detection is used to determine whether data is abnormal. In this paper, the anomaly detection process combines the detector set D and boundary self set $Self_o$ together for testing. So, we modify the definition of B, and B is expressed as (26). B is mapping from detector set D, boundary self set $Self_o$, and antigen ag to be identified to a classification $\{0, 1\}$, where 0 indicates that ag is self, and 1 means that ag is non-self.

$$B\left(D, Self_o, ag\right) = \begin{cases} 1, & \left(\exists ag' \in Self_o, f\left(ag', ag\right) > r_s\right) \bigwedge \left(\exists d \in D, f\left(d, ag\right) \le r_d\right) \\ 0, & otherwise \end{cases} \tag{26}$$

3.7. Coverage Rate of Non-Self Space. The greater the coverage of detectors for the non-self space is, the greater the probability of detecting non-self is. Due to overlaps between detectors, direct calculation of coverage rate p for non-self space is very difficult. Monte Carlo method is used to calculate the approximate value $\widehat{p}$. Suppose a randomly generated non-self set $N \in Nonself$, the calculation formula of non-self space coverage rate of detector set D is as

$$\begin{aligned} \mathrm{p} &= \frac{\int_{\vec{x} \in D} d\vec{x}}{\int_{\vec{x} \in Nonself} d\vec{x}} \approx \widehat{p} \\ &= \frac{\sum_{ag \in N} \left(\exists d \in D, f\left(d, ag\right) \le r_d\right))}{|N|} \end{aligned} \tag{27}$$

Set p_0 as the target coverage rate. When the estimated value $\widehat{p}$ is greater than or equal to p_0, it is thought that detectors meet the requirements of the target coverage. Because $\widehat{p}$ is a random variable, it is inevitable that the actual coverage rate p is less than p_0. To reduce the probability of this situation, the algorithm introduces the hypothesis test for processing.

Set α as the significance of hypothesis test, z_a as the α quantile of standard normal distribution, n_0 as the sample size, and x as the number of samples covered by detectors. The calculating formula of maximum x_{max} is as

$$x_{max} = \left\lfloor z_\alpha \sqrt{n_0 p_0 \left(1 - p_0\right)} + n_0 p_0 \right\rfloor \tag{28}$$

Obviously $n_0 > x_{max}$, then $n_0 > z_\alpha \sqrt{n_0 p_0 (1 - p_0)} + n_0 p_0$, that is, $n_0 > p_0 {z_\alpha}^2 / (1 - p_0)$. In order to meet the requirements and make the estimated coverage rate $\widehat{p} = x/n_0$ approximately obey the normal distribution, the calculation formula of sample size n_0 is as

$$n_0 = \left\lceil \max\left(\frac{5}{p_0}, \frac{5}{1 - p_0}, \frac{p_0 {z_\alpha}^2}{1 - p_0}\right)\right\rceil \tag{29}$$

Because $5 > p_0 {z_\alpha}^2$, the above equation can be rewriting as $n_0 = \lceil \max(5/p_0, 5/(1 - p_0)) \rceil$.

The actual coverage rate of detectors for the non-space is p, the null hypothesis of hypothesis testing H_0 is "$\mathrm{p} = p_0$", and the alternative hypothesis H_1 is "$\mathrm{p} > p_0$". In the process of the algorithm, if x is less than or equal to x_{max}, receive H_0 and update detector set; if x is greater than the x_{max}, accept H_1 and exit.

Table 1: Contrast of time complexity.

Algorithms	Time complexities
NSA	$O\left(-\frac{\ln P^f \bullet \lvert Self\rvert}{P_m \bullet (1-P_m)^{\lvert Self\rvert}}\right)$[1]
RNSA	$O\left(\frac{\lvert D\rvert \bullet \lvert Self\rvert}{(1-P)^{\lvert Self\rvert}}\right)$[3]
V-Detector	$O\left(\frac{\lvert D\rvert \bullet \lvert Self\rvert}{(1-P)^{\lvert Self\rvert}}\right)$[4, 5]
CB-RNSA	$O\left(ng \bullet \left\lfloor \log_2\left(\frac{\sqrt{n}}{r_s}\right)+1\right\rfloor + (nc_{max}+nc_{max}\bullet n+\lvert Self\rvert)\bullet\frac{\lvert D\rvert}{(1-P)}+\lvert D\rvert^2\right)$

4. Analysis of the Algorithm

4.1. Time Complexity Analysis

Theorem 14. *The time complexity of CB-RNSA for detectors generation is* $O(ng \bullet \lfloor \log_2(\sqrt{n}/r_s)+1\rfloor + (nc_{max}+nc_{max}\bullet n+\lvert Self\rvert)\bullet\lvert D\rvert/(1-P)+\lvert D\rvert^2)$, *where* n *is the spatial dimension,* $\lvert Self\rvert$ *is the size of self set,* $\lvert D\rvert$ *is the size of detector set,* P *is the self-reaction rate of detectors,* ng *is the scale of initial immune cell population, and* nc_{max} *is the maximum number of clonal copies.*

Proof. CB-RNSA carries out preprocessing operations on the self set in the first place and then executes clonal selection algorithm in every qualified level to generate candidate detectors.

The number of computation times of the outlier selves discovery algorithm is not exceeded $\lvert Self\rvert\bullet(\lvert Self\rvert-1)/2$, and the time complexity is $O(\lvert Self\rvert\bullet(\lvert Self\rvert-1)/2)$. The number of computation times of the clusters discovery algorithm does not exceed $\lvert Self\rvert\bullet(\lvert Self\rvert-1)/2$, and the time complexity is $O(\lvert Self\rvert\bullet(\lvert Self\rvert-1)/2)$. We mainly consider the time complexity of the detector generation process and do not consider the time complexity of preprocessing operations.

In the i layer, the time complexity of randomly generating initial immune cell group is $O(ng)$. Immune selection operation chooses the entire immune cells, and the time complexity is not considered. For each immune cell, the calculation number of the clonal amplification operation is not more than nc_{max}, the time complexity is $O(nc_{max})$ which is a constant value, and the time complexity of hypermutation operation is $O(nc_{max}\bullet n)$. The time complexity of calculating whether immune cells fall into self space is $O(\lvert Self\rvert)$. Suppose the probability of immune cells falling into self space is P_i, the number of immune cells in level i is N_i, and then the number of immune cells which are not excluded in this step is $N_i\bullet(1-P_i)$. The time complexity of computing whether immune cells are covered by mature detectors is $O(\lvert D\rvert)$. Then viewing this immune cell as a candidate detector, the time complexity of calculating the radius is $O(\lvert Self\rvert)$. Results of previous operations can be used for this operation, so, we do not consider the time complexity.

Therefore, the overall time complexity of generating detectors in the i level is $O(ng+N_i\bullet(nc_{max}+nc_{max}\bullet n+\lvert Self\rvert)+(1-P_i)\bullet N_i\bullet\lvert D\rvert)$. The maximum number of i is $l=\lfloor\log_2(\sqrt{n}/r_s)+1\rfloor$; the overall time complexity of generating detectors for CB-RNSA is $O(\sum_{i=1}^{l}(ng+N_i\bullet(nc_{max}+nc_{max}\bullet n+\lvert Self\rvert)+(1-P_i)\bullet N_i\bullet\lvert D\rvert))$. Suppose P is the average self-reaction rate of detectors, the number of candidate detectors is $N_0=\sum_{i=1}^{l}N_i\approx\lvert D\rvert/(1-P)$, and then the time complexity of CB-RNSA is $O(ng\bullet\lfloor\log_2(\sqrt{n}/r_s)+1\rfloor+(nc_{max}+nc_{max}\bullet n+\lvert Self\rvert)\bullet\lvert D\rvert/(1-P)+\lvert D\rvert^2)$. Proved. □

NSA, RNSA, and V-Detector are the influential negative selection algorithms and are widely used in intrusion detection, abnormal diagnosis, pattern recognition, etc. Table 1 lists the time complexity contrast of the three negative selection algorithms and CB-RNSA, where P_m is the probability of detectors identifying any antigen and P_f is the detection failure rate. As can be seen from the table, the time complexity of the traditional negative selection algorithms is in exponential relationship with self set size $\lvert Self\rvert$. When the size of self elements increases, the time spending increases rapidly, even to the unbearable point. The time complexity of CB-RNSA is related to the spatial dimension n, the size of self set $\lvert Self\rvert$, the detector set size $\lvert D\rvert$, and the self-reaction rate P. There is no exponential relationship with $\lvert Self\rvert$, and the size of the detector set $\lvert D\rvert$ is far less than the other three algorithms, which reduces the time complexity and improves the detectors generation efficiency.

4.2. Self-Reaction Rate Analysis of Detectors.

Under the established matching rules, the matching probability P of any given detector with any antigen is a constant [1]. For the r-continuous matching rule of NSA, it satisfies $P=2^{-r}((l-r)/2+1)$, where l is the length of the string and r is the consecutive matching digits.

In the real-valued algorithm, calculation is different. The work in [3, 4] pointed out that, in RNSA and V-Detector, P can be acquired by calculating the ratio of self capacity to the total antigen capacity, and P is also known as the reaction rate of detectors, which means the probability of detectors covering self space. In addition, we can use the number of all candidate detectors to measure detector generation cost. Suppose achieving desired non-self space coverage rate, the number of detectors is $\lvert D\rvert$, and the candidate detectors can be calculated by $N_0=\lvert D\rvert/(1-P)^{num}$, where the number of selves is num. The larger the self-reaction rate of detectors is, the greater the number of candidate detectors for generating $\lvert D\rvert$ mature detectors is, and the higher the detector generation cost is. To simplify the discussion, it is assumed that there is no overlap between selves.

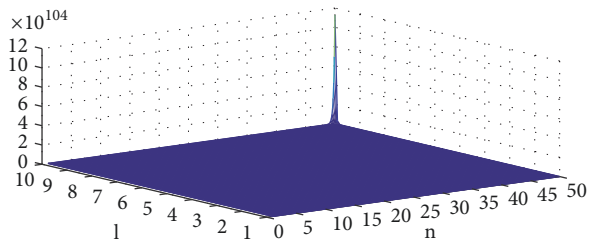

FIGURE 11: Influences of n and l on $\boldsymbol{\theta}$ (set $n_l/|Self| = 0.2$).

For RNSA and V-Detector, detectors are randomly generated in $[0,1]^n$ space. Therefore, the self-reaction rate of detectors P_1 is the ratio of $|Self|$ hypersphere volumes to unit hypercube volume. Suppose V_{self} is the volume of a self. P_1 is expressed as

$$P_1 = \frac{|Self| \bullet V_{self}}{1} = \frac{|Self| \bullet \pi^{n/2} \bullet r_s^n}{\Gamma(n/2+1)} \tag{30}$$

For CB-RNSA, detectors are randomly generated within different limited space. Therefore, the self-reaction rate of detectors P_2 is the ratio of volumes of the limited space which are covered by selves to volumes of the limited space. In the l layer, the limited space is a super sphere loop between two spheres with radiuses $\sqrt{n}/2^l$ and $\sqrt{n}/2^{l-1}$, respectively. Selves may or may not intersect with this loop. Suppose the number of selves within this loop is n_l. To simplify the discussion, suppose there is no element half intersecting with the loop. The self-reaction rate of detectors P_2 is as (31). Suppose V_l is the volume of a sphere for the l layer.

$$P_2 = \frac{n_l \bullet V_{self}}{V_l - V_{l+1}} = \frac{n_l \bullet \left(\left(r_s^n \bullet \pi^{n/2}\right)/\Gamma(n/2+1)\right)}{\left(\left(\sqrt{n}/2^{l-1}\right)^n \bullet \pi^{n/2}\right)/\Gamma(n/2+1) - \left(\left(\sqrt{n}/2^{l}\right)^n \bullet \pi^{n/2}\right)/\Gamma(n/2+1)} = \frac{n_l \bullet r_s^n}{\left(\sqrt{n}/2^{l-1}\right)^n - \left(\sqrt{n}/2^{l}\right)^n} \tag{31}$$

To compare the self-reaction rates of the three algorithms, set $\theta = P_2/P_1$ and is expressed as

$$\begin{aligned}\theta = \frac{P_2}{P_1} &= \frac{\left(n_l \bullet r_s^n\right) / \left(\left(\sqrt{n}/2^{l-1}\right)^n - \left(\sqrt{n}/2^{l}\right)^n\right)}{\left(|Self| \bullet \pi^{n/2} \bullet r_s^n\right)/\Gamma(n/2+1)} \\ &= \frac{n_l}{|Self|} \bullet \frac{\Gamma(n/2+1)}{\pi^{n/2}\left(\left(\sqrt{n}/2^{l-1}\right)^n - \left(\sqrt{n}/2^{l}\right)^n\right)}\end{aligned} \tag{32}$$

When l is small, the super sphere loop is far from the center of the cluster. Selves of this cluster are disjoint with the loop and selves of other clusters may intersect with the loop. With the increase of l, the loop will be more and more near the center of cluster, selves of this cluster are likely to intersect with the loop, and other selves are disjoint with the loop. So, $n_l < |Self|$. When θ is less than 1, the self-reaction rate of CB-RNSA is less than RNSA and V-Detector, that means the detector generation cost of CB-RNSA is less. Figure 11 shows the variation of θ with changes of the data dimension n and the limited level l. As can be seen from the figure, when n and l are small, θ is far less than 1.

5. Experimental Results and Analysis

This section verified the effectiveness of GB-RNSA through experiments. Experiments chose the representative real-valued negative selection algorithms RNSA and V-Detector for comparisons. Experimental data adopt two types of data sets which are commonly used in the study, including 2D comprehensive data sets [30] and UCI data sets [23]. 2D comprehensive data sets are provided by the team of professor Dasgupta from the university of Memphis and are authoritative for the real-valued negative selection algorithm performance test [11, 12, 14]. UCI data sets are classical in machine learning and are widely used in the performance tests and detector generating efficiency tests [8–16].

In specific comparisons, in order to avoid the influence of different exit conditions on the algorithms, all algorithms adopt the same exit criteria "to reach the expected non-self space coverage rate". The number of mature detectors DN, detection rate DR, false alarm rate FAR, and time cost of detector generation DT are adopted to measure the effectiveness of algorithms.

5.1. 2D Comprehensive Data Sets. The data sets contain a number of subdata sets; Figure 12 shows distributions in the two-dimensional space of self data from three subdata sets, Cross, Intersection, and Ring. Without loss of generality, experiments chose the three data sets.

Self set size of these three data sets $|Self| = 1000$. The training set is composed of randomly selected selves, and the test data are composed of random points in the $[0,1]$ space. The experiments repeat 20 times and averaged values were obtained. Tables 2 and 3 show the results of the experiments, and values in parentheses are variances. Table 2 lists contrasts of detection rate and false positive rate of CB-RNSA under the same expected coverage rate 90%, the same training set size N_s =500, and different radiuses of selves. It can be seen that detectors trained from the smaller radius of selves have higher detection rate and false positive rate, and detectors trained from the bigger radius have lower detection rate and false positive rate. Therefore, we should adopt smaller radius of selves to train detectors for applications whose environment is sensitive to abnormal data, and bigger radius for applications whose environment is sensitive to false positives. Table 3 lists contrasts of detection rate and false positive rate of CB-RNSA under the same expected coverage rate 90%, the same radius of selves $r_s = 0.05$, and different training set sizes. It can be seen that, with the rise

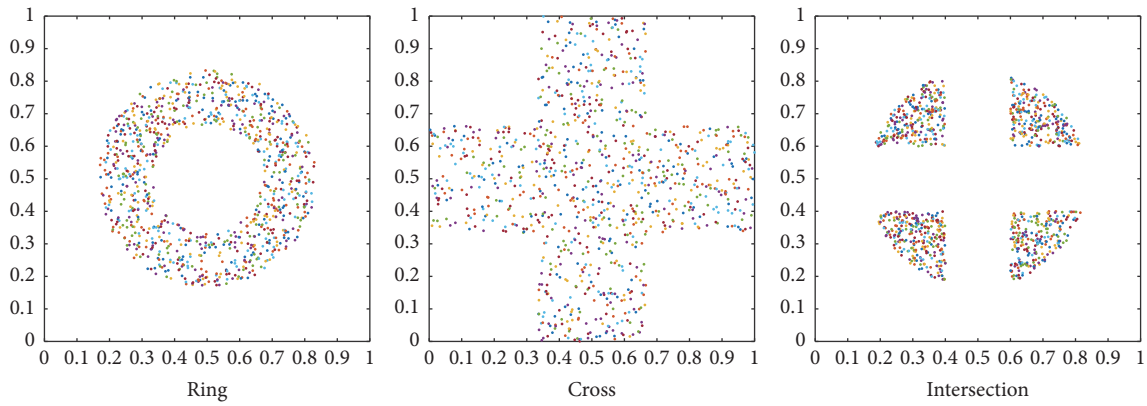

Figure 12: Distributions of Cross, Intersection, and Ring.

Table 2: Influence of different self radiuses on CB-RNSA.

Data sets	$r_s = 0.02$		$r_s = 0.05$		$r_s = 0.1$		$r_s = 0.2$	
	DR%	*FAR*%	*DR*%	*FAR*%	*DR*%	*FAR*%	*DR*%	*FAR*%
Ring	93.31 (1.32)	24.24 (1.44)	86.43 (1.01)	9.22 (1.15)	71.14 (1.55)	0.00 (0.00)	43.21 (2.42)	0.00 (0.00)
Cross	94.19 (1.16)	24.76 (1.71)	88.02 (1.24)	8.28 (1.72)	72.52 (1.41)	0.00 (0.00)	46.29 (2.76)	0.00 (0.00)
Intersection	90.11 (1.68)	26.63 (1.82)	84.91 (1.51)	9.45 (1.14)	63.13 (1.89)	0.00 (0.00)	38.64 (2.53)	0.00 (0.00)

Table 3: Influence of different training set sizes on CB-RNSA.

Data sets	$N_s = 100$		$N_s = 500$		$N_s = 800$		$N_s = 1000$	
	DR%	*FAR*%	*DR*%	*FAR*%	*DR*%	*FAR*%	*DR*%	*FAR*%
Ring	24.87 (1.21)	69.32 (1.74)	86.43 (1.01)	9.22 (1.15)	97.92 (1.01)	0.00 (0.00)	98.87 (1.00)	0.00 (0.00)
Cross	26.46 (1.45)	67.41 (2.01)	88.02 (1.24)	8.28 (1.72)	98.19 (1.05)	0.00 (0.00)	99.42 (0.31)	0.00 (0.00)
Intersection	21.51 (1.87)	71.23 (2.17)	84.91 (1.51)	9.45 (1.14)	96.58 (1.10)	0.00 (0.00)	98.08 (1.13)	0.00 (0.00)

of training set size, the detection rate increases gradually, and the false positive rate reduces gradually. This is because more selves participating in training is good for effective screening of detectors, which can reduce the number of self-reacted detectors, and makes detectors cover non-self space more accurately.

Figures 13 and 14 are comparisons of a run of RNSA, V-Detector and CB-RNSA in Cross data set and Intersection data set, respectively. Blue filled circles are self elements, cyan filled circles are mature detectors, and unfilled area is holes. As can be seen from the figures, there is less redundant coverage between detectors of CB-RNSA, and detector quantity is less, which makes less detectors achieve the same coverage expectation.

2D comprehensive data sets are clean. In order to test conditions of self set containing noise data, we added a small amount of noise data in the Ring data set. Figure 15 shows comparison of a run of RNSA, V-Detector, and CB-RNSA in Ring data set. Blue filled circles are self elements, white filled circles are noise data, cyan filled circles are mature detectors, and unfilled area is holes. CB-RNSA detects the outlier selves and ignores them because they are noise data. In the case of noisy data interference, detectors of CB-RNSA fully covered the non-self space, and RNSA and V-Detector cannot effectively handle it.

5.2. UCI Data Sets. Experiments selected four standard UCI data sets, including Haberman's Survival, Abalone, Breast Cancer Wisconsin Original (BCW1 for short), and Breast Cancer Wisconsin Diagnostic (BCW2 for short), and experimental parameters are shown in Table 4. Of these four data sets, self set and non-self set were randomly selected; training set and testing set were randomly selected as well. Experiments were repeated 20 times and averaged values were gained.

5.2.1. Comparisons of the Number of Detectors. Figure 16 shows comparisons of the number of mature detectors of RNSA, V-Detector, and CB-RNSA. As can be seen from

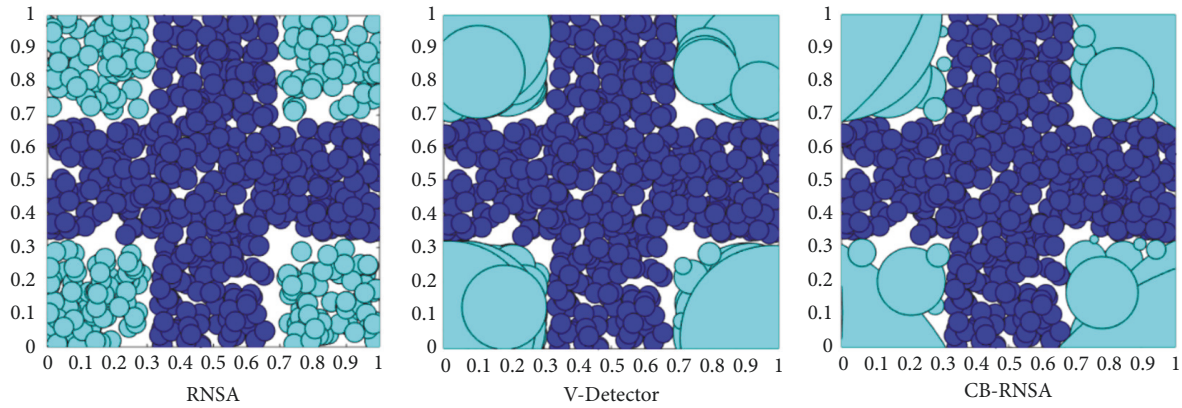

Figure 13: Comparisons of RNSA, V-Detector, and CB-RNSA in Cross (to achieve the expected coverage rate 90%, RNSA, V-Detector, and CB-RNSA need 289, 61, and 37 mature detectors, respectively, where the training set size is 500, the self radius is 0.03, and the detector radius of RNSA is 0.03).

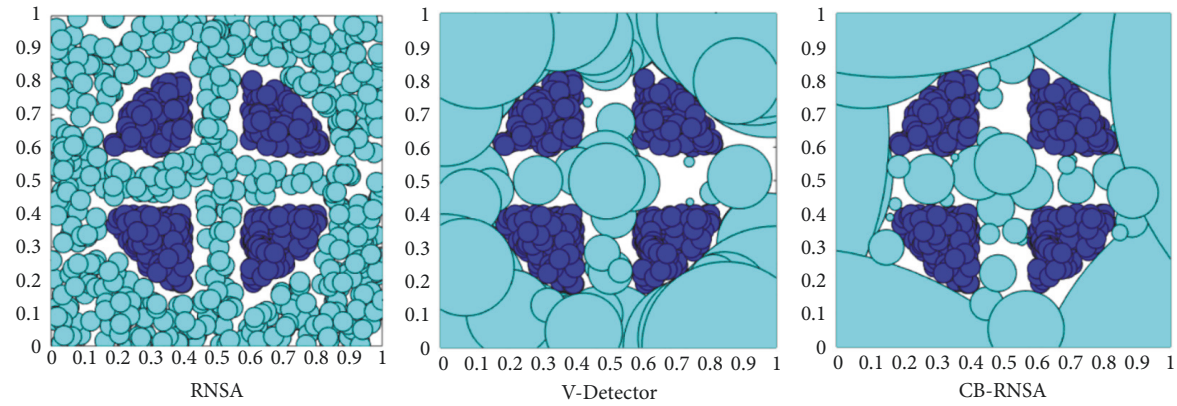

Figure 14: Comparisons of RNSA, V-Detector, and CB-RNSA in Intersection (to achieve the expected coverage rate 90%, RNSA, V-Detector, and CB-RNSA need 590, 85, and 40 mature detectors, respectively, where the training set size is 500, the self radius is 0.03, and the detector radius of RNSA is 0.03).

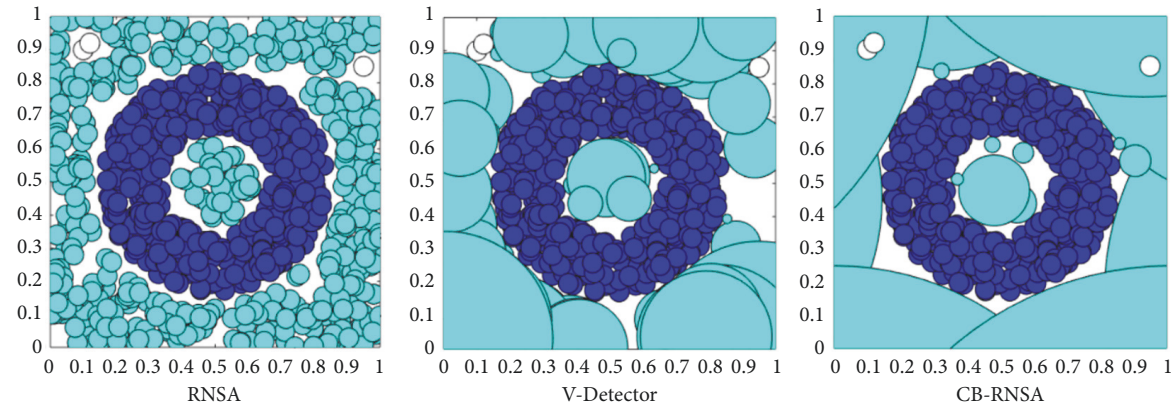

Figure 15: Comparisons of RNSA, V-Detector, and CB-RNSA in Ring (to achieve the expected coverage rate 90%, RNSA, V-Detector, and CB-RNSA need 477, 51, and 17 mature detectors, respectively, where the training set size is 500, the self radius is 0.03, and the detector radius of RNSA is 0.03).

the diagram, while the expected coverage rate increases, the number of mature detectors of three algorithms rises correspondingly. But the efficiency of CB-RNSA is superior to the other algorithms. For Haberman's Survival data set, in order to achieve the expected coverage rate of 99%, RNSA needs 1033.1 mature detectors, V-Detector needs 351.4 detectors, and CB-RNSA needs 165.2 detectors which declines by 84.0% and 53.0%, respectively. For large data set Abalone, in order to achieve the expected coverage rate of 99%, RNSA needs 12893.2 mature detectors, V-Detector needs 1194.0 detectors,

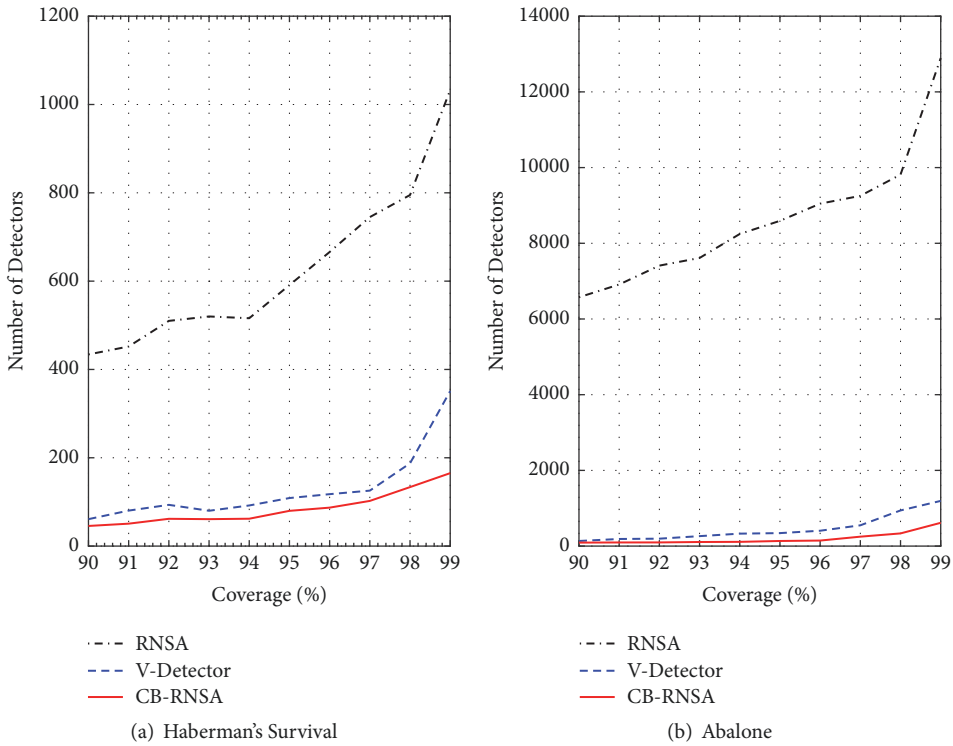

Figure 16: Comparisons of the number of detectors of RNSA, V-Detector, and CB-RNSA.

Table 4: Experimental parameters.

Data sets	Number of records	Dimension	Type	Self set	Non-self set	Training set and its size	Testing set and its size
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
BCW1	683	9	Integer	Benign: 444	Malignant: 239	Benign: 200	Benign: 150 Malignant: 150
BCW2	569	30	Real	Benign: 357	Malignant: 212	Benign: 200	Benign: 150 Malignant: 150

and CB-RNSA needs 615.4 detectors which declines by 95.2% and 48.5%, respectively. So, in the expectation of the same coverage rate, under different data dimensions and different training sets, the number of mature detectors of CB-RNSA has greatly reduced compared to RNSA and V-Detector.

5.2.2. Comparisons of the Cost of Detectors Generation. Figure 17 shows comparisons of the cost of detectors generation of RNSA, V-Detector, and CB-RNSA. For Haberman's Survival data set, when the expected coverage rate rises from 90% to 99%, the time price of RNSA increases from 7.7s to 291.0s, the time price of V-Detector increases from 0.6s to 29.8s, and the time price of CB-RNSA increases from 0.4s to 13.7s. For Abalone data set, when the expected coverage rate rises from 90% to 90%, the time price of RNSA increases from 241.7s to 2412.8s, the time price of V-Detector increases from 2.4s to 228.6s, and the time price of CB-RNSA increases from 1.3s to 82.5s. Therefore, with the rise of expected coverage rate, the time costs of RNSA and V-Detector increase very quickly, and the time cost of GB-RNSA increases more slowly.

5.2.3. Comparisons of Detection Rate and False Alarm Rate. Figures 18 and 19 show comparisons of detection rates and false alarm rates of RNSA, V-Detector, and CB-RNSA. As can be seen from the diagram, while the expected coverage rate is greater than 90%, detection rates of three algorithms have little differences, and that of RNSA is lower; false alarm rate of CB-RNSA is obviously lower than that of RNSA and V-Detector. For BCW1 data set, when the expected coverage rate is 99%, false alarm rate of RNSA is 55.2%, false alarm

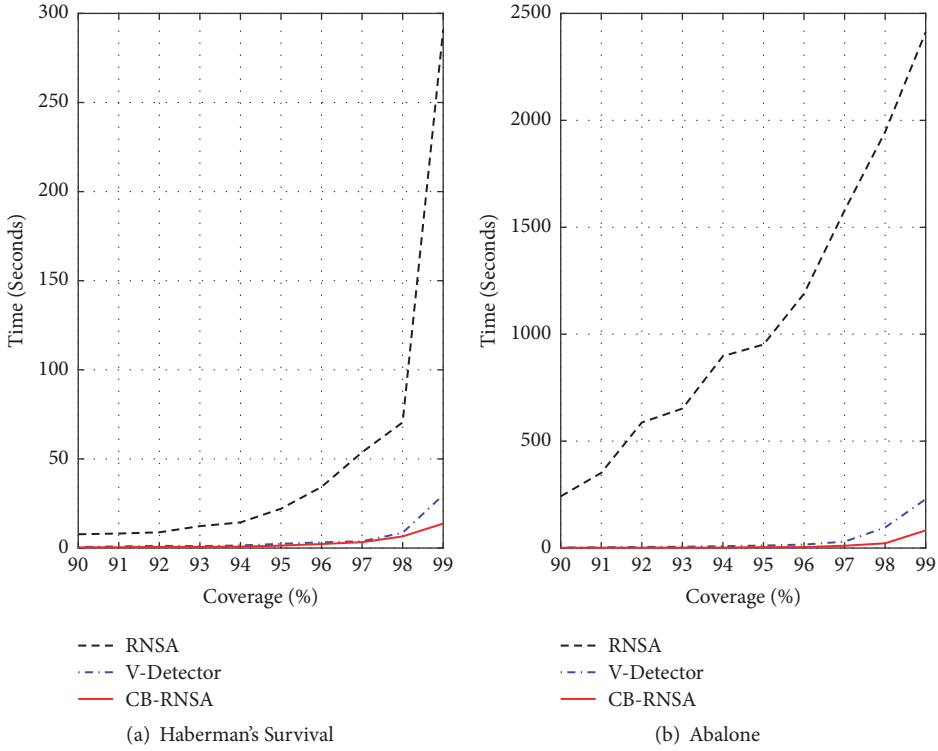

Figure 17: Comparisons of time costs of RNSA, V-Detector, and CB-RNSA.

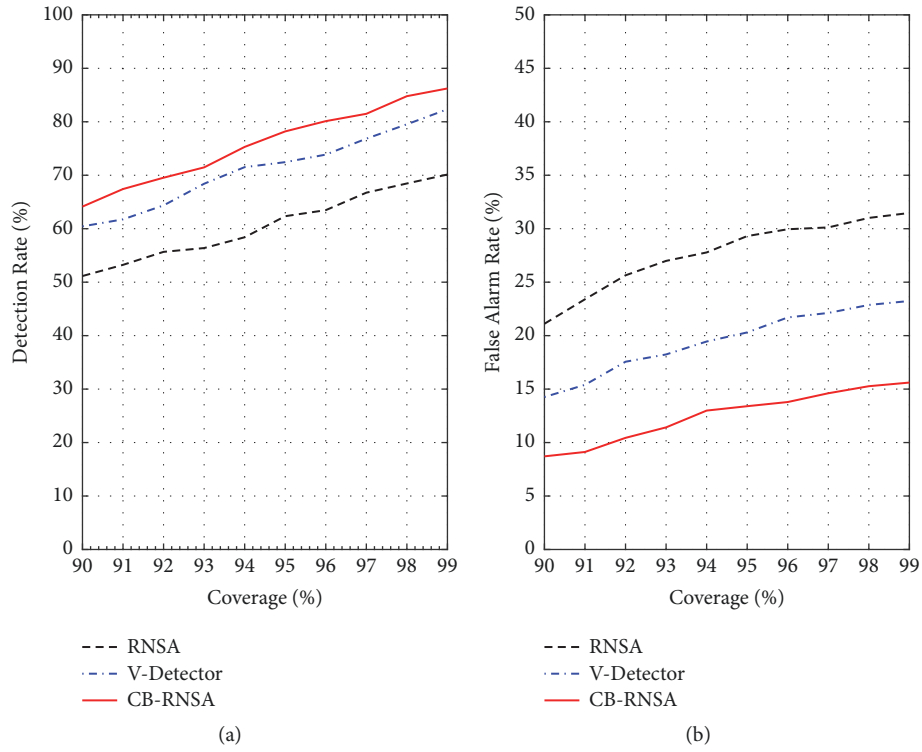

Figure 18: Comparisons of detection rates and false alarm rates of RNSA, V-Detector, and CB-RNSA (BCW1).

rate of V-Detector is 30.1%, and false alarm rate of CB-RNSA is 20.1% which declines by 63.6% and 33.2%, respectively. For high dimensional data set BCW2, when the expected coverage rate is 99%, false alarm rate of RNSA is 25.1%, false alarm rate of V-Detector is 20.5%, and false alarm rate of CB-RNSA is 12.6% which declines by 49.8% and 38.5%, respectively. On the one hand, CB-RNSA introduced clonal selection algorithm and limited the generation range

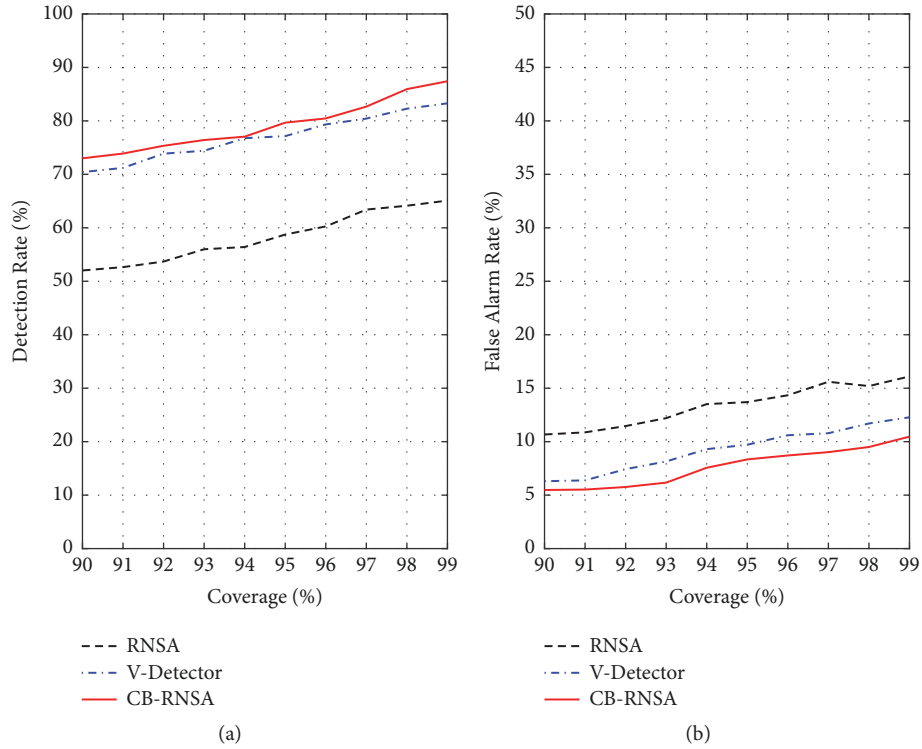

Figure 19: Comparisons of detection rates and false alarm rates of RNSA, V-Detector, and CB-RNSA (BCW2).

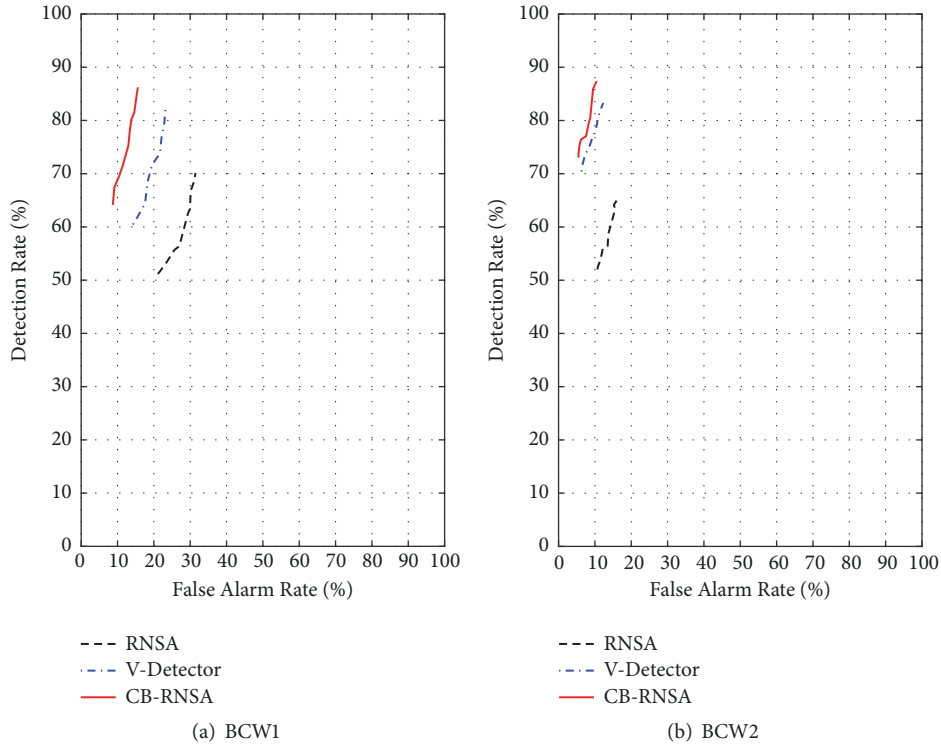

Figure 20: Comparisons of ROC curves of RNSA, V-Detector, and CB-RNSA.

of detectors, which made detectors generated in the low-coverage non-self space and improved the coverage rate. On the other hand, detectors and boundary selves were adopted for testing at the same time, the definition of anomaly was stricter, which reduced the rate of false positive.

The ROC curve is a graphical method for classification model based on detection rates and false alarm rates. Figure 20 shows comparisons of ROC curves of RNSA, V-Detector, and CB-RNSA under two kinds of data sets, BCW1 and BCW2. A good classification mode curve should be

distributed in the left-top of graphic as soon as possible. As can be seen from the diagram, CB-RNSA is superior to RNSA and V-Detector.

6. Conclusions

Excessive detectors, high time complexity, and loopholes are main problems which current negative selection algorithms have face and greatly limit the practical applications of negative selection algorithms. This paper proposes a real-valued negative selection algorithm, named CB-RNSA. The algorithm introduces the clonal selection algorithm and randomly generates candidate detectors within stratified limited ranges based on clustering centers of self set, which reduces the number of detectors and the number of holes. Selves are divided into outlier selves, boundary selves, and internal selves, which adapts to the interference of noise data. When the algorithm runs for anomaly detection, mature detector set and boundary self set are used at the same time, which effectively improves the detection rate and reduces the false alarm rate. Theoretical analysis and experimental results show that the algorithm has better time efficiency and detector generation quality according to classic negative selection algorithms.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Sichuan Provincial Education Department of China Funded Project (035Z2258) for providing financial aid.

References

[1] 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.

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

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

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

[5] J. Balthrop, F. Esponda, S. Forrest, and M. Glickman, *Coverage and generalization in an artificial immune system. GECCO 2002*, Morgan Kaufmann Publishers Inc, NY, USA, 2002.

[6] H. Zhang, L. F. Wu, Y. S. Zhang, and Q. K. Zeng, "An algorithm of r-adjustable negative selection algorithm and its simulation analysis," *Chinese journal of computers*, vol. 28, no. 010, pp. 1614–1619, 2005.

[7] S. He, W.-J. Luo, and X.-F. Wang, "Negative selection algorithm with the variable length detector," *Ruan Jian Xue Bao/Journal of Software*, vol. 18, no. 6, pp. 1361–1368, 2007.

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

[9] 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 GECCO 2005 - Genetic and Evolutionary Computation Conference*, pp. 337–344, USA, June 2005.

[10] 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.

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

[12] Z. Ji and D. Dasgupta, "V-detector: an efficient negative selection algorithm with "probably adequate" detector coverage," *Information Sciences*, vol. 179, no. 10, pp. 1390–1406, 2009.

[13] R. Zhang, T. Li, X. Xiao, and Y. Shi, "A Real-valued Negative Selection Algorithm Based on Grid for Anomaly Detection," in *Abstract and Applied Analysis*, vol. 2013, p. 15, 2013.

[14] C. Wen, D. Xiaoming, L. Tao, and Y. Tao, "Negative selection algorithm based on grid file of the feature space," *Knowledge-Based Systems*, vol. 56, pp. 26–35, 2014.

[15] W. Chen, X.-J. Liu, T. Li, Y.-Q. Shi, X.-F. 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.

[16] M. Gong, J. Zhang, J. Ma, and L. Jiao, "An efficient negative selection algorithm with further training for anomaly detection," *Knowledge-Based Systems*, vol. 30, pp. 185–191, 2012.

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

[18] X. Z. Gao, S. Ovaska, X. Wang, and M. Chow, "Multi-Level Optimization Of Negative Selection Algorithm Detectors With Application In Motor Fault Detection," *Intelligent Automation & Soft Computing*, vol. 16, no. 3, pp. 353–375, 2010.

[19] M. R. Abdolahnezhad and T. Banirostam, "Hybrid Email Spam Detection Method Using Negative Selection and Genetic Algorithms," *IJARCCE*, vol. 5, no. 4, pp. 956–960, 2016.

[20] I. Idris, A. Selamat, N. Thanh Nguyen et al., "A combined negative selection algorithm-particle swarm optimization for an email spam detection system," *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 33–44, 2015.

[21] F. P. A. Lima, A. D. P. Lotufo, and C. R. Minussi, "Wavelet-artificial immune system algorithm applied to voltage disturbance diagnosis in electrical distribution systems," *IET Generation, Transmission & Distribution*, vol. 9, no. 11, pp. 1104–1111, 2015.

[22] P. D'Haeseleer, S. Forrest, and P. Helman, "An immunological approach to change detection: algorithms, analysis and implications," in *Proceedings of the 17th IEEE Symposium on Security and Privacy*, pp. 110–119, IEEE Computer Society Press, Las Alamitos, Calif, USA, May 1996.

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

[24] F. M. Burnet, *The clonal selection theory of acquired immunity*, Cambridge University Press, Cambridge, UK, 1959.

[25] P. A. De Castro and F. J. Von Zuben, "The clonal selection algorithm with engineering applications," in *Proceedings of the GECCO '00, Workshop on Artificial Immune Systems and Their Applications*, pp. 36-37, Morgan Kaufman Publisher, San Fransisco, 2000.

[26] L. N. de Castro and F. J. von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239–251, 2002.

[27] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *The VLDB Journal*, vol. 8, no. 3-4, pp. 237–253, 2000.

[28] F. Angiulli, R. Ben-Eliyahu-Zohary, and L. Palopoli, "Outlier detection using default reasoning," *Artificial Intelligence*, vol. 172, no. 16-17, pp. 1837–1872, 2008.

[29] S. M. Guo, L. C. Chen, and J. S. H. Tsai, "A boundary method for outlier detection based on support vector domain description," *Pattern Recognition*, vol. 42, no. 1, pp. 77–83, 2009.

[30] F. González, D. Dasgupta, and J. Gómez, "The Effect of Binary Matching Rules in Negative Selection," in *Genetic and Evolutionary Computation — GECCO 2003*, vol. 2723 of *Lecture Notes in Computer Science*, pp. 195–206, Springer, Berlin, Germany, 2003.