Negative Selection Algorithm Based on Antigen Density Clustering

CHAO YANG\textsuperscript{1,2,3}, LIN JIA\textsuperscript{1}, BING-QIU CHEN\textsuperscript{1}, AND HAI-YANG WEN\textsuperscript{1}

\textsuperscript{1}School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China
\textsuperscript{2}Hubei Education Informatization Engineering Technology Research Center, Wuhan 430062, China
\textsuperscript{3}Hubei Key Laboratory of Applied Mathematics, Faculty of Mathematics and Statistics, Hubei University, Wuhan 430062, China

Corresponding author: Chao Yang (stevenyc@hubu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61977021, and in part by the Open Project of Hubei Key Laboratory of Applied Mathematics under Grant HBAM201902.

ABSTRACT

The negative selection algorithm (NSA) is one of the basic algorithms of the artificial immune system. In the traditional negative selection algorithm, candidate detectors are randomly generated without considering the uneven distributions of self-antigens and nonself-antigens, thereby resulting in many redundant detectors, and it is difficult for these detectors to fully cover the area of nonself-antigens. To overcome the problem of low detector generation efficiency, a negative selection algorithm that is based on antigen density clustering (ADC-NSA) is proposed in this paper. The algorithm divides the process of detector generation into three steps: the first step is to calculate the density of the antigens by using the method of antigen density clustering to select nonself-clusters. The second step is to prioritize the abnormal points (nonself-antigens that are not clustered) as the centers of candidate detectors and to generate the detectors via calculation. The third step is to generate the detectors via the traditional algorithm. Detector generation via these three steps can reduce the randomness of the detector generation in the traditional algorithm, thereby improving the efficiency of detector generation. The experimental results demonstrate that on the BCW and KDD-Cup datasets, the negative selection algorithm that is based on antigen density clustering can effectively increase the detection rate while reducing the false-positive rate compared with the traditional negative selection algorithm (RNSA) and two improved algorithms at the same expected coverage.

INDEX TERMS

Artificial immunity, negative selection algorithm, antigen density clustering, detector.

I. INTRODUCTION

The artificial immune system (AIS) is a computational paradigm that is inspired by the biological immune system [1]. The artificial immune system has been widely used in computer security, anomaly detection and prediction [2], [3], [23], [24]. The negative selection algorithm is one of the basic algorithms of AIS. This algorithm was proposed by Forrest [4] in 1994 and has been applied to intrusion detection and data classification.

The negative selection algorithm has been widely used in the fields of network intrusion detection, spam detection, medical diagnosis, and fault detection [5]–[8]. However, the negative selection algorithm has the disadvantages of high detector repeat coverage and loopholes. [9], [10], [11]. In response to these disadvantages, many scholars have proposed improved algorithms. For example, Gonzalez et al. [12] proposed a real-valued negative selection algorithm (RNSA) with an immutable radius. Antigens and antibodies belong to the $[0, 1]^n$ value space to maximize the coverage of nonself-regions to improve the detection efficiency. Ji and Dasgupta [13] proposed a variable-radius negative selection algorithm (V-Detector). The main strategy is to randomly generate the detector center $x$, find the nearest self-antigen to $x$, calculate the distance $r$ between them, and dynamically generate detectors with $x$ as the center and $r$ as the radius. Chen et al. [14] proposed a negative selection algorithm that was based on hierarchical clustering of self-sets (CB-RNSA). After the hierarchical clustering of self-sets, cluster centers are used to replace self-points, which effectively reduces the computational cost of distance calculations. Liu et al. [15] proposed SDS-RNSA, which uses a subspace density search algorithm to calculate the sample subspace region and directly generates detectors in its subspace to increase the detection rate.
Zhengjun et al. [16] proposed a negative selection algorithm that was based on soft subspace clustering of antigens (ASSC-NSA), which uses clustering to calculate the key features and weights of various types of antigens, thereby reducing the influence of redundant features on the detector, which effectively guides the generation of mature detectors. In 2018, Abid et al. [10] designed a layered real-valued NSA (LRNSA). Different layers are formed according to the distance of candidate detectors from the self-antigens, and the detectors belonging to far-self layer are generated by clustering optimization method. In addition, the algorithm makes the detectors stay apart from each other in order to maximize the coverage and decrease the number of mature detectors. In 2019, Fan et al. [11] proposed ASTC-RNSA, the algorithm first uses the Delaunay triangulation method from the perspective of computational geometry to divide its space into simple units for determining the position of the detector. Then the overlap between the simple unit and the self-antigen is removed to form a set of triangulation coverage areas, and finally detectors are generated within this area. Avoid the time-consuming self-tolerance process of traditional NSAs.

According to the discussion above, the focus of improving the NSA algorithm has been on the efficient generation of detectors. Many scholars have improved the traditional algorithm in response to the problems of high redundancy and loopholes that are caused by the random generation of candidate detectors via the traditional algorithm.

This paper proposes a negative selection algorithm that is based on antigen density clustering (ADC-NSA). Partial detectors are generated via density clustering to reduce the repeated coverage and loopholes that are caused by randomly generated detectors in the traditional algorithm. In this paper, the generation of a detector via antigen density clustering is divided into three steps: the first step is to cluster the antigen via the antigen density clustering algorithm and to select the clustered non-self-clusters as the mature detectors; the second step is to use nonself-antigens that are not clustered as abnormal points to generate mature detectors via training; and the third step is to use traditional algorithms to randomly generate candidate detectors and to train them to generate mature detectors. This process reduces the generation of redundant detectors and enables the algorithm to cover the nonself-area with as few detectors as possible, thereby effectively overcoming the problems of high detector repeat coverage and loopholes.

II. PROBLEM DESCRIPTION

Traditional negative selection algorithms generate mature detectors (antibodies) by judging whether the candidate detector matches the self-antigens. Then, the data to be detected are matched with the mature detectors. If the matching is successful, the data are abnormal. The basic definitions and process of the algorithm are as follows:

**Definition 1:** Antigen set. The antigen set is \( A_g = \{x_1, x_2, x_3, \ldots, x_n\} \), where \( x_i \in [0, 1] \), \( n \) represents the total number of sample points, \( x_i \) represents the normalized value of sample point \( i \), and \( A_g \) represents the set of normalized values of all sample points.

**Definition 2:** Self-antigen and nonself-antigen. Self-antigen \( \in A_g \) represents the positive sample of the sample, and nonself-antigen \( \notin A_g \) represents the negative sample of the sample. The area that is covered by the self-antigens in the range of the value space is called the self-region, and the area not covered is called the nonself-region.

**Definition 3:** Affinity. The Euclidean distance \( \text{dist}(x_i, x_j) = \sqrt{\sum_{d=1}^{D} (x_i^d - x_j^d)^2} \) between two points represents the affinity between the two points, where \( x_i^d \) and \( x_j^d \) represent the i-th and j-th sample points, \( d \) represents a feature dimension of the sample points, \( D \) represents the total number of feature dimensions of the sample points, and \( x_i^d \) represents the d-th dimension feature of the i-th sample point.

**Definition 4:** Detectors. A detector is denoted by \( d_i(z_i, r_i) \), where \( z_i \) represents the randomly generated candidate detector center, \( r_i \) represents the distance from the center to its nearest self-cell, and the circle that is formed by \( z_i \) and \( r_i \) corresponds to the mature detector.

As illustrated in Figure 1, the traditional negative selection algorithm simulates the negative selection process in which the immune system recognizes self-cells and nonself-cells. The algorithm randomly generates candidate detectors, and by removing the detectors that have detected the self-antigens, the detectors that can detect any non-self-cell are retained. Finally, a mature detector set is generated for data detection. The main advantages are that no prior knowledge is required and an unlimited number of nonself-antigens can be detected with a limited number of self-antigens [17], [18]. The main disadvantage is that the traditional negative selection algorithm generates random detectors in the detector generation step [11], [19], which leads to problems such as high repeat coverage and loopholes.

**FIGURE 1.** (a) Detector generation (b) Data detection.

In Figure 2, the pentagram represents the self-set, and the open circles represent the mature detectors. All regions except the self-set are nonself-regions. The traditional negative selection algorithm expects that the detectors can cover as many nonself-regions as possible. However, when the antigens are unevenly distributed in the sample space, where the antigens are densely distributed, the gaps between the sample points are narrow, which hinders the efficient generation of detectors. Where antigens are sparse, the randomly generated candidate detectors will inevitably be highly redundant,
which is also called antigen affinity calculation.

The antigen set is calculated using the Euclidean distance, and the distance between any two sample points represents the d-th dimension feature of the i-th sample point, where attributes and is expressed as $x_i$.

The Euclidean distance $d_{ij}$ between sample points $x_i$ and $x_j$ is:

$$d_{ij} = \sqrt{\sum_{d=1}^{D} (x_{id} - x_{jd})^2} \tag{1}$$

The antigen set $A_g = \{x_1, x_2, x_3, \ldots, x_n\}$ contains $n$ sample points, and each sample point has D-dimensional feature attributes and is expressed as $x_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$. $x_{id}$ represents the d-th dimension feature of the i-th sample point, and the distance between any two sample points $x_i$ and $x_j$ in the antigen set is calculated using the Euclidean distance, which is also called antigen affinity calculation.

The local density $\rho_i$ of the sample point $x_i$ is:

$$\rho_i = \sum_{x_j \in A_g} M(d_{ij} - d_c) \tag{2}$$

where $d_c$ represents the cutoff distance (clustering radius). In [25], [27], a $d_c$ was selected so that the average number of neighbors in each sample point was about 2% of the total number of sample points, and $M(m) = \begin{cases} 1, & m \leq 0 \\ 0, & m > 1 \end{cases}$. If $d_{ij}$ is greater than $d_c$, $\rho_i$ is not changed, whereas $\rho_i$ is incremented by one if $d_{ij}$ is less than the $d_c$. This function is used to calculate the true density of $x_i$.

**Definition 3:** The sample point distance $\delta_i$ is defined as follows:

If $\rho_i$ is smaller than $\rho_l$:

$$\delta_i = \min (d_{ij}) \quad \text{(} x_i, x_j \in A_g, j \neq i \text{ and } \rho_j > \rho_i \text{)} \tag{3}$$

If $\rho_i$ is maximal:

$$\delta_i = \max (d_{ij}) \quad \text{(} x_i, x_j \in A_g, j \neq i \text{)} \tag{4}$$

**Definition 4:** The cluster center $c_i$ is determined by the size of the cluster center weight $\gamma_i$. Sort $\gamma_i$ in descending order and set the sample points that correspond to the first K values of $\gamma_i$ as $c_i$ [26].

**Definition 5:** The cluster center weight $\gamma_i$ is defined as:

$$\gamma_i = \delta_i \times \rho_i \tag{5}$$

If $F_i = 1$, the cluster is a nonself-cluster; otherwise, it is a self-cluster. The $\varepsilon$ is the category judgment threshold, and it takes 0.99 in this experiment.

**Definition 6:** The abnormal point $a_i$ is determined according to $\delta_i$ and $\rho_i$. A point with small $\rho_i$ and relatively large $\delta_i$ is called an abnormal point. In this paper, the nonself-antigens that satisfy these conditions and are not clustered are called abnormal points, which will be preferred as the candidate detector centers.

**Definition 7:** The cluster discriminant $F_i$ is defined as:

$$F_i = \begin{cases} 1 & \sum_{i=1}^{n} g (x_i) (\text{non-} \text{self}) > \varepsilon \\ 0 & \text{else} \end{cases} \tag{6}$$

If $F_i = 1$, the cluster is a nonself-cluster; otherwise, it is a self-cluster. The $\varepsilon$ is the category judgment threshold, and it takes 0.99 in this experiment.

**Definition 8:** The expected coverage $c_p$ is defined as:

$$c_p (p, t, m) = \begin{cases} -1 & \text{if } t \geq G \\ 0 & \text{else if } \frac{m}{\sqrt{tp} \times (1 - p)} - \frac{\sqrt{tp}}{1 - p} < Z_n \\ 1 & \text{else} \end{cases} \tag{7}$$

The calculation termination condition is $\text{Flag} = c_p$. If $\text{Flag} = -1$, set $t = m = 0$ and start counting again. $Z_n$ is a very small constant. In this experiment, the value of $Z_n$ is set as 0.001, and $G > \max \left(\frac{5}{p}, 5 / (1 - p)\right)$.

**B. ADC-NSA ALGORITHM**

The basic process of the negative selection algorithm that is based on antigen density clustering is presented as Algorithm 1 and Figure 3:
Algorithm 1 ADC-NSA Algorithm

Input: set \( A_g = \{x_1, x_2, x_3, \ldots, x_n\} \), cutoff distance \( d_c \), expected coverage \( c_p \)
Output: Detectors

1\>: Detectors = \( \emptyset \), self \( \cup \) nonself = \( A_g \), self \( \cap \) nonself = \( \emptyset \).
2\>: Use the antigen density clustering algorithm to calculate \( d_{ij}, \delta_i, \rho_i \) of each sample point, abnormal point \( a_i \) and nonself-clustering center \( c_i \).
3\>: Add the nonself-cluster that is composed of \( c_i \) and \( d_c \) as the first type of mature detector into set Detectors.
4\>: Select the abnormal point \( a_i \) as the center of the candidate detector preferentially, calculate the distance between \( a_i \) and the nearest self-antigen, record it as \( R \), and add the circle with \( a_i \) as the center and \( R \) as the radius as the second type of mature detector to Detectors.
5\>: Randomly generate candidate detectors within the collection, and add the third type of mature detectors, which are generated using the traditional detector generation algorithm, to Detectors.
6\>: Reach \( c_p \) and terminate of the algorithm. Thus, the generation of Detectors ends.

Algorithm 2 Antigen Density Clustering Algorithm

Input: set \( A_g = \{x_1, x_2, x_3, \ldots, x_n\} \) and cutoff distance \( d_c \)
Output: nonself-clustering center \( c_i \) and abnormal point \( a_i \)

1\>: Calculate Euclidean distance \( d_{ij} \) according to formula (1).
2\>: Calculate local density \( \rho_i \) according to formula (2).
3\>: Calculate \( \delta_i \) according to formulas (3) and (4).
4\>: Calculate \( \gamma_i \) according to formula (5) and select \( c_i \) according to definition 4.
5\>: Select the abnormal point \( a_i \) that satisfies the condition in definition 6.
6\>: Determine the cluster category according to formula (6), and select the nonself-clusters.

FIGURE 3. ADC-NSA algorithm flowchart.

1) ANTIGEN DENSITY CLUSTERING ALGORITHM
The antigen density clustering algorithm clusters antigens based on the density of the antigen distribution, in preparation for the generation of the detectors. The algorithm is based on the following assumptions: (1) The density of the clustering center point is higher than that of the surrounding sample points. (2) The distance between the clustering center point and the higher density point is relatively large.

The process of the antigen density clustering algorithm is presented as Algorithm 2:

Algorithm 3 Detector Generation Algorithm
The generation of the detector is divided into three main steps: (1) Use the nonself-clusters that are calculated via antigen density clustering as detectors. (2) Use the abnormal point \( a_i \) preferentially as a candidate detector center to generate a detector via calculation. (3) Use the traditional algorithm to generate the detectors. The generation process of the detectors is illustrated in Figure 4, and the process is presented as Algorithm 3:

FIGURE 4. Detector generation process.
In Figure 4, hollow dots represent self-antigens with a specified radius, solid dots represent nonself-antigens, large circles represent clustering results, and the regions with self-antigens are nonself-regions. The generated mature detectors should cover as many nonself-regions as possible. Among them are nonself-clusters 1, 2, and 3 and self-cluster 1. These four groups of clusters are calculated via the antigen density clustering algorithm according to the distribution of the antigens. Nonself-clusters 1, 2 and 3 (each of which is defined by a cluster center $c_i$ and a cutoff distance $d_i$) are selected as the first category of mature detectors according to the cluster discrimination formula (6). $a_i$ is preferentially used as the center of the candidate detector. Then, calculate the distance between $a_i$ and the nearest self-antigen; this distance is regarded as $R_i$. If the detector that is defined by the abnormal point $a_i$ and the radius $R_i$ contains other abnormal points $a_j$, then remove $a_j$ and select the detector as the second category of mature detectors. If a randomly generated candidate detector point is in the self-antigen or as the second category of mature detectors, then remove it. As illustrated in the figure, the candidate detector center $z_i$ is randomly generated, and the closest self-antigen distance $r_i$ is selected as the radius. At this time, the generated detector is used as the third category of mature detectors. Three categories of mature detectors have been generated.

IV. EXPERIMENT AND ANALYSIS OF THE RESULTS

A. DATASETS AND EVALUATION INDICATORS

The datasets that are used in this paper are from the UCI database [21]. Classic datasets BCW and KDD-Cup99, which are often used for anomaly detection and machine learning, are selected. The BCW dataset originates from breast cancer data that were provided by foreign medical institutions. This dataset has 2 categories, 9 attributes, 241 abnormal data, and 458 normal data. This experiment standardizes and normalizes the data, and divides the testing set and training set of the BCW dataset by using a partition function, As shown in Table 1. The KDD-Cup99 dataset originates from 9-week network data connection information that was collected by a foreign LAN. This dataset includes a training dataset and a test dataset. In this experiment, the experimental data are extracted at a ratio of 1:1. Each connection record in the training dataset contains 41 fixed feature attributes and a class identifier. The data features include basic features, network features and content features.

As shown in Table 2.

TABLE 1. Details of the BCW dataset in the experiment.

| Data type | Label | Total number | Experimental training data | Experimental test data |
|-----------|-------|--------------|-----------------------------|------------------------|
| Normal    | 2     | 458          | Random 70%                  | Random 30%             |
| Abnormal  | 4     | 241          | Random 70%                  | Random 30%             |

TABLE 2. Details of the KDD dataset in the experiment.

| Data type | Description | Attack in training set | Attack in testing set | Experimential training data | Experimental test data |
|-----------|-------------|------------------------|-----------------------|-----------------------------|------------------------|
| Normal    | Normal access | None                   | None                  | 10%                          | 10%                     |
| DOS       | Denial of service attack | Neptune,back, snort, land etc. | apache2, udpstorm, mailbomb etc. | 5%                          | 5%                     |
| Probe     | Detection attack | Portsweep, ipsweep, satan etc. | Mean, smtp etc. | 50%                          | 50%                     |
| R2L       | Remote unauthorized access | Ftpwrite, guess_passwd etc. | Sendmail, nmap, snmpguess etc. | 50%                          | 50%                     |
| U2R       | Illegal elevation of user rights | buffer overflow, loadmodule, etc. | Xterm, httpsncsm etc. | 50%                          | 50%                     |

The evaluation indices are the detection rate (DR) and the false-positive rate (FPR), which are commonly used in the classic binary classification problem. DR = \( \frac{TP}{TP + FN} \) (8) FPR = \( \frac{FP}{FP + TN} \) (9)

In these expressions, TN denotes the number of true-negative types, which are correctly recognized as self-antigens; TP denotes the number of true-positive types, which are correctly recognized as nonself-antigens; FN denotes the number of false-negative types, which are incorrectly recognized as self-antigens; and FP denotes the number of false-positive types, which are incorrectly recognized as nonself-antigens.
B. EXPERIMENTAL DATA PREPROCESSING

This experiment uses the most common z-score normalization (zero-mean normalization) method, also known as standard deviation standardization. This method gives the mean and standard deviation of the original data to standardize the data.

The processed data conforms to the standard normal distribution, that is, the mean is 0, the standard deviation is 1, and its conversion function is: $x^* = \frac{x - \mu}{\sigma}$. Where $\mu$ is the mean of all sample data and $\sigma$ is the standard deviation of all sample data. There are two benefits after data normalization: ① Improve the convergence speed of the model, ② Improve the accuracy of the model.

Due to the large KDD-CUP data set, this paper uses Principal Component Analysis (PCA) to reduce the dimension of this data set, while maintaining the characteristics of the largest variance contribution in the data set. The main steps are: The features are recombined into uncorrelated principal components, which are used to represent the original information. This method can effectively reduce the dimensionality of the sample and improve the calculation accuracy [22].

The key statement in the code is as follows: sklearn.decomposition.PCA (n_components = 99.9, copy = True, whiten = False).

① n_components: If the value is assigned to string, such as n_components = 'mle', the number of features will be automatically selected to meet the required percentage of variance; if no value is assigned, the default is None and the number of features will not change (the feature data will be changed). In this experiment, the percentage of variance is set to 99.9%, that is, the similarity with the feature data before dimensionality reduction is 99.9%.

② copy: True or False, the default is True, that is, whether the original training data needs to be copied.

③ whiten: True or False, the default is False, that is, whether to whiten, so that each feature has the same variance.

C. EXPERIMENTAL PARAMETER SETTINGS

The main parameters of this experiment are the expected coverage $c_p$ and the cutoff distance $d_c$. The experimental results are presented in Figures 5, 6, 7 and 8. Figures 5 and 6 plot the detection rate and the false-positive rate when $c_p$ is 90%-99%. Figures 7 and 8 plot the detection rate and the false-positive rate when $c_p$ is 99% and $d_c$ is 0.1-0.55.

According to the figure, in the BCW dataset, when $c_p$ is 99% and $d_c$ is 0.35, the detection rate of this algorithm reaches its maximal value and the false-positive rate is relatively low. At this time, the detection rate is 99.41%, and the false-positive rate is 3.54%.

In the KDD-Cup dataset, the algorithm performs best when $c_p$ is 99% and $d_c$ is 0.25. The detection rate reaches 99.24%, and the false-positive rate is 3.25%.

The final parameter settings are listed in Table 3:

D. COMPARATIVE ANALYSIS EXPERIMENT

To further evaluate the performance of the algorithm, this paper compares it with three algorithms, namely, RNSA [12], V-Detector [13], and ASSC-NSA [16], on the BCW and KDD-Cup datasets. The statistics are presented in Table 4, and the results of the comparative experiments are presented in Figures 9, 10, 11, and 12.
TABLE 3. ADC-NSA parameter settings on two datasets.

| dataset  | c_p  | self-radius | d_e | DR    | FPR   |
|----------|------|-------------|-----|-------|-------|
| BCW      | 99%  | 0.002       | 0.35| 0.9941| 0.0355|
| KDD-Cup  | 99%  | 0.02        | 0.25| 0.9924| 0.0325|

TABLE 4. Comparison of four algorithms on the bcw and KDD-Cup datasets.

| algorithm   | BCW DR | BCW FPR | KDD-Cup DR | KDD-Cup FPR |
|-------------|--------|---------|------------|-------------|
| ADC-NSA     | 0.9941 | 0.0354  | 0.9924     | 0.0325      |
| ASSC-NSA    | 0.9797 | 0.0367  | 0.9388     | 0.0426      |
| V-D         | 0.9367 | 0.0419  | 0.8256     | 0.0446      |
| RNSA        | 0.8481 | 0.0620  | 0.7469     | 0.0519      |

E. TIME COMPLEXITY ANALYSIS

Based on this experiment, the following assumptions are made: \( N_s \) is the number of self-antigens, \( N_n \) is the number of nonself-antigens, \( d \) is the dimension, \( M \) is the number of nonself-clusters, and \( P \) is the number of abnormal points. By analyzing definitions 1 through 7, we can get that the time complexity of antigen density clustering algorithm is \( O((N_s + N_n)^2) \). Combined with the detector generation algorithm, the time complexity of this experiment is \( O((N_s + N_n)^2 \cdot d) \). Compared with other experiments [11], the time complexity is shown in Table 5.

As shown in Table 5, compared with RNSA and V-Detector, the time complexity of ADC-NSA is much lower than the traditional exponential level [14]. Under the same conditions, the time complexity of this algorithm is better than the improved ASSC-NSA. As the dimension \( d \) increases, the time complexity of this algorithm will also be better than the improved ASTC-RNSA. Overall, the time complexity of the algorithm is slightly lower.
V. CONCLUSION
The traditional negative selection algorithm ignores the influence of the antigen distribution on the generation of detectors during the detector generation stage, thereby resulting in low efficiency of detector generation. Therefore, this paper proposes a negative selection algorithm that is based on antigen density clustering (ADC-NSA): First, the clustering algorithm is used to identify high-density regions, and the clustered nonself-clusters are directly used as mature detectors. Second, the abnormal points are selected as the centers of candidate detectors preferentially in low-density regions, and detectors are generated via training. Finally, detectors are generated via the traditional algorithm. The algorithm can effectively generate detectors in regions with various densities; hence, the algorithm has a higher detection rate and a lower false-positive rate.

At present, the algorithm still has two problems: 1. In the clustering step, the selection of the cutoff distance $d_c$ during clustering still depends on artificial experience; 2. When the detection is performed, the points that fall into the loopholes are not clear. The next research work is how to achieve adaptiveness in the selection of $d_c$ and conduct further research on the determination of the data to be detected that falls into the loopholes.

REFERENCES
[1] X. K. Gang and X. Xinying, “Overview of artificial immune system and algorithm,” Comput. Eng. Appl., vol. 41, no. 20, pp. 77–80, 2005.
[2] G. Dozier, D. Brown, H. Hou, and J. Hurley, “Vulnerability analysis of immunity-based intrusion detection systems using genetic and evolutionary algorithms,” Appl. Soft Comput., vol. 7, no. 2, pp. 547–553, Mar. 2007.
[3] S. T. Powers and J. He, “A hybrid artificial immune system and self-organising map for network intrusion detection,” Inf. Sci., vol. 178, no. 15, pp. 3024–3042, Aug. 2008.
[4] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, “Self-nonself discrimination in a computer,” in Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy, May 1994, pp. 202–212.
[5] M. E. Fumukov, V. K. Polukov, and V. A. Shcherb, “Negative selection and neural network based algorithm for intrusion detection in IoT,” in Proc. 41st Int. Conf. Telecommun. Signal Process. (TSP), Jul. 2018, pp. 1–5.
[6] H. Xiao-juan, L. Lei, and Q. Ning-jia, “A novel spam categorization algorithm based on active learning method and negative selection algorithm,” Acta Electronica Sinica, vol. 1, no. 1, pp. 203–209, 2018.
[7] S. Le and B. Wenyang, “New negative selection algorithm and its application in disease diagnosis,” J. Frontiers Comput. Sci. Technol., vol. 1, pp. 36–42, 2015.
[8] X. Song, F. Gao, Z. Chen, and W. Liu, “A negative selection algorithm based identification framework for distribution network faults with high resistance,” IEEE Access, vol. 7, pp. 109363–109374, 2019.
[9] J. Kim, P. J. Bentley, U. Ackelin, J. Greensmith, G. Tedesco, and J. Twycross, “Immune system approaches to intrusion detection—a review,” Natural Comput., vol. 6, pp. 413–446, Jun. 2007.
[10] A. Abid, M. T. Khan, and C. W. de Silva, “Layered and real-valued negative selection algorithm for fault detection,” IEEE Syst. J., vol. 12, no. 3, pp. 2960–2969, Sep. 2018.
[11] Z. Fan, C. Wen, L. Tao, C. Xiaochun, and P. Haipeng, “An antigen space triangulation coverage based real-valued negative selection algorithm,” IEEE Access, vol. 7, pp. 51886–51898, 2019.
[12] F. A. Gonzalez, D. Dasgupta, and L. F. Nino, “A randomized real-valued negative selection algorithm,” in Proc 2nd Int. Conf. Artif. Immune Syst., pp. 261–272, 2003.
[13] Z. Ji and D. Dasgupta, “V-detector: An efficient negative selection algorithm with probably adequate detector coverage,” Inf. Sci., vol. 179, no. 10, pp. 1390–1406, Apr. 2009.
[14] W. Chen, T. Li, X. Liu, and B. Zhang, “A negative selection algorithm based on hierarchical clustering of self set,” Sci. China Inf. Sci., vol. 56, no. 8, pp. 1–13, Oct. 2011.
[15] Z. Liu, T. Li, J. Yang, and Y. Yang, “An improved negative selection algorithm based on subspace density seeking,” IEEE Access, vol. 5, pp. 12189–12198, 2017.
[16] L. Zhengqun, G. Jiangjin, and Y. Tao, “Improved negative selection algorithm based on antigen soft subspace clustering,” Appl. Res. Comput., vol. 35, no. 3, pp. 680–684, Mar. 2018.
[17] Z. Z. Jin, M. H. Liao, and G. Xiao, “Survey of negative selection algorithms,” J. Commun., vol. 34, no. 1, pp. 159–170, 2013.
[18] J. Zhou and D. Dasgupta, “Real-valued negative selection algorithm with variable-sized detectors,” in Proc. GECCO, 2004, pp. 287–298.
[19] G. Jiangjin and Y. Tao, “Immune evolution negative selection algorithm,” Appl. Res. Comput., vol. 34, no. 5, pp. 1293–1297, May 2017.
[20] Z. Fan, Y. Jin, L. Tao, and Z. Fangdong, “DnyNSA: A novel real-value based negative selection algorithm,” in Proc. IEEE Symp. Serv. Comput. Intell. (SSCI), Nov. 2018, pp. 1104–1109.
[21] UCI Data Set [Online]. Available: http://archive.ics.uci.edu/ml/index.php
[22] N. Lei and S. Zhong-lin, “PCA-AKM algorithm and its application in intrusion detection system,” Comput. Sci., vol. 45, no. 2, pp. 226–230, 2018.
[23] A. da Silva, I. S. Guaranys, B. Arruda, E. C. Gurjao, and R. S. Freire, “A method for anomaly prediction in power consumption using long short-term memory and negative selection,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2019, pp. 1–5.
[24] A. Lasisi, N. Tairan, R. Ghazali, W. K. Mashwani, S. N. Qasem, H. Kumar G.R, and A. Arora, “Predicting crude oil price using fuzzy rough set and bio-inspired negative selection algorithm,” Int. J. Swarm Intell. Res., vol. 10, no. 4, pp. 25–37, Oct. 2019.
[25] Z. Yuan and L. Zhong, “Fuzzy density peaks clustering algorithm based on $k$-nearest neighbors,” Comput. Eng. Softw., vol. 38, no. 4, Aug. 2017.
[26] M. C. Lai, S. Hong, and M. Tao, “A density peak clustering algorithm based on the automatic selection of cluster center points,” Comput. Sci., vol. 43, no. 7, pp. 255–258, 2016.
[27] A. Rodriguez and A. Liao, “Clustering by fast search and find of density peaks,” Science, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
**BING-QIU CHEN** received the B.S. degree in information and computing science from Huanggang Normal University, in 2017. He is currently pursuing the M.E. degree in computer technology with Hubei University, Hubei, China. His research interests include artificial immune systems and machine learning.

**HAI-YANG WEN** received the B.E. degree in software engineering from the Hubei University of Economics, in 2018. He is currently pursuing the M.E. degree in computer technology with Hubei University, Hubei, China. His research interests include artificial immune systems and machine learning.