Applied Intelligence (2022) 52:482–500
https://doi.org/10.1007/s10489-021-02323-4

Using known nonself samples to improve negative selection algorithm

Zhiyong Li[1,2] · Tao Li[1]

Accepted: 4 March 2021 / Published online: 4 May 2021

Abstract

Negative selection algorithm is the core algorithm of artificial immune system. It only uses the self for training and generates detectors to detect abnormalities. Holes are feature space areas that the detector fails to cover, it is the root cause of the performance degradation of the negative selection algorithm. The conventional method generates a large number of detectors randomly to repair the holes, which is time-consuming and not effective. To alleviate the problem, we propose a V-Detector-KN algorithm in this paper. V-Detector is the abbreviation of the real-valued negative selection algorithm with Variable-sized Detectors, KN represents Known Nonself. The V-Detector-KN algorithm uses the known nonself as the candidate detector to further generate the detector based on the V-Detector randomly generated detector, so as to realize the repair of holes. Compared with the conventional method to randomly generate detectors to repair holes, our proposed V-Detector-KN method uses known nonself to repair holes, reducing the randomness and blindness of hole repair. Theoretical analysis shows that the detection rate of our algorithm is not lower than that of the conventional V-Detector algorithm. The results of experiment comparing with other 6 algorithms on 7 UCI data sets show the superiority of our proposed algorithm.

Keywords Artificial immune system · Negative selection algorithm · V-Detector · Anomaly detection

1 Introduction

Artificial immune system is an adaptive system developed from the mechanism and theory of biological immune system [1]. Once proposed, it quickly became a hot topic of extensive discussion and attention in academia, Nature [2] and Science [3] had carried out special reports on it. Negative Selection Algorithm (NSA) is the core algorithm of artificial immune system [4]. It was proposed by professor Forrest of the university of new mexico in 1994 [5]. In biological immunity, T cells must undergo a self-tolerance process in the thymus to mature, and T cells that recognize any self will die. Therefore, mature T cells do not recognize self and can be used for nonself detection. Inspired by the self-tolerance of T cells, NSA randomly generates candidate detectors and deletes those that recognize self (normal sample) during the training process, and uses detectors that do not recognize self to detect nonself (abnormal sample).

Negative Selection Algorithms has experienced two breakthroughs from binary space to real value space and from fixed radius detector to variable radius detector [5–7]. Early NSA used binary strings to define antibodies (detectors) and antigens (sample), and calculated the similarity between them through r-continuous bit matching rules [5]. In 2003, Gonzalez and Dasgupta extended NSA to real-valued space and proposed a Real-valued Negative Selection Algorithm (RNSA) to normalize antigens and antibodies to $[0, 1]^n$ real value space, and use minkowsky distance to calculate the affinity [6]. In 2004, Zhou Ji proposed real-valued negative selection algorithm with variable-sized detectors (V-Detector) [7]. V-Detector still uses the binary tuple $< cd, rd >$ to represent the detector, but the detector radius rd no longer uses the preset fixed value, but changes dynamically according to the position of the nearest self. Due to the variable size of radius, detectors with larger radius can cover nonself space more efficiently, thus reducing the number of detectors; detectors with smaller radius can better cover holes, thus increasing

✉ Tao Li
taoli_scu@163.com

Zhiyong Li
lizhiyong@uoh.edu.cn

[1] School of Cyber Science and Engineering, Sichuan University, Chengdu, 610065, China

[2] Information Technology Center, Honghe University, Mengzi, 661199, China

the coverage of nonself space. V-Detector became the state-of-the-art NSA at that time [8]. To our knowledge, so far, many scholars are still improving the V-Detector [9–12].

Since no prior knowledge is needed, unlimited abnormal data can be detected only with a limited number of normal samples, making the NSA gradually become one of the most popular tools in the research fields of network intrusion detection [9, 13–15], network anomaly detection [16–18], malicious code detection [19], fault detection [20–22] and civil engineering damage detection [23]. The above application fields can all be regarded as the problem of distinguishing self from nonself. For example, by defining normal network traffic as self, we can train detectors to detect network intrusions (nonself) [9, 13–15]. By defining the normal state of the network, we can train the detector to detect network anomalies (nonself) [16–18]. In the field of fault detection, we usually define the normal state as the self and the abnormal state as the nonself [20–22].

NSA is widely used, but the hole problem has not been solved well, which restricts its development. Holes are feature space areas that the detector fails to cover. In the testing phase, the NSA classifies the samples covered by the detector as nonself, and the samples not covered by the detector as self. The nonself in the hole is incorrectly classified as the self because it is not covered by the detector, resulting in a decrease in the detection rate. Therefore, hole problem is a major factor causing the performance degradation of V-Detector algorithm [24].

In order to reduce holes and increase the coverage of nonself space, so as to improve the detection rate, many scholars have conducted extensive and in-depth research. In 2014, Idris proposed a differential evolution based NSA (referred to as NSA-DE), which optimizes the distribution of detectors through differential evolution to reduce the number of holes [25]. In 2015, Cui Lin proposed a bidirectional inhibition optimization r-variable negative selection algorithm (BIORV-NSA). BIORV-NSA includes self set edge inhibition strategy and detector self inhibition strategy. Self set edge inhibition strategy defines a generalized radius for self individual area, making self individual radius dynamically be variable, so it is possible to cover more nonself space to reduce holes [26]. In 2016, Saurabh put forward the concept of self-tuning detectors and detector power in NSA with the intension to make a detector evolve and facilitate better and correct self and nonself coverage [18]. In 2017, targeted that in the high-dimensional space, antigens (data samples) distribute sparsely and unevenly, and most of them reside in low-dimensional subspaces, which makes it difficult to generate detectors in high-dimensional space and there are a large number of holes. Tao Yang proposed the antigen space density based real-value NSA (ASD-RNSA), guide the candidate detector to evolve and eventually generate in a relatively dense subspace of antigen distribution, so as to achieve the repair of holes in high-dimensional space [11]. In 2018, Ziwen proposed an improved V-Detector algorithm: Detectors with big radius are firstly generated to cover most of the nonself space and detectors with small radius are later generated to cover small holes around self samples. Therefore, two detector generation rules are defined and applied in two different stages. Finally, a detector optimization processing is proposed to delete redundant detectors for reducing the storage consumption and improving the detection efficiency [9]. In 2018, when Tianliang Lu used V-Detector for Shellcode detection, he encoded the immune detector as a hyperellipsoid to reduce holes and improve the coverage of nonself space [19]. In 2019, Fan Zhang employed delaunay triangulation method to divide the self space into simplicial cells for determining the position of the detector, thereby reducing the number of holes and detectors [27]. In 2020, Chao Yang proposed a negative selection algorithm that is based on antigen density clustering (ADC-NSA). 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 [12].

Previous research mainly repaired holes by optimizing the distribution of detectors [11, 21, 25–27], adopting new detector generation method [9, 14, 18, 23], and changing the shape representation of detectors and antigens [19]. The purpose was to increase the coverage of the nonself space by the detector to improve the detection rate. However, previous studies have not considered the use of known nonself. In real scenarios such as network intrusion detection and anomaly detection, some nonself samples are actually available. These known nonself samples can better reflect the distribution of real nonself space than randomly generated detectors. In addition, machine learning algorithms such as SVM [28] and KNN [29] use both self and nonself for training in the training phase. However, conventional NSA only uses the self for training, which makes it often at a disadvantage when compared with machine learning algorithms. In this paper, we propose a V-Detector-KN algorithm. V-Detector is the abbreviation of the real-valued negative selection algorithm with Variable-sized Detectors, KN represents Known Nonself. The V-Detector-KN algorithm first uses V-Detector to generate detectors. On this basis, considering that the known nonself can better reflect the true distribution of the nonself than the randomly generated detector, we take the known nonself as candidate detector center to generate the detector. Not only can the detector be generated in the spatial region where it is difficult to generate the detector in a random manner, but also due to the clustering of the nonself, the known nonself generated detector can be used to detect the unknown nonself.

To summarize, the main contributions of this paper are as follows:

1. For the first time, we make use known nonselfs as candidate detectors to generate detectors to improve the V-Detector algorithm.
2. The use of known nonself can generate detectors in the feature space where the random generation method is difficult to generate detectors, thereby filling the holes formed between randomly generated detectors.
3. Compared with randomly generated detectors, known nonself-generated detectors can better reflect the true distribution of nonself, and can be used to detect unknown nonself around known nonself.

2 V-Detector with known nonself algorithm

In this section, we first introduce the V-Detector algorithm, which is the baseline of our proposed algorithm. On this basis, we introduce our proposed V-Detector-KN algorithm.

2.1 V-Detector algorithm

The negative selection algorithm consists of two phases: training phase and detection phase. In the training phase, a randomly generated candidate detector is compared with the self set. If the candidate detector identifies any self, that is, the affinity of the candidate detector and the self sample meets a certain threshold, the candidate detector will be deleted; otherwise, the candidate detector will become a mature detector for subsequent detection. In the detection phase, the mature detector can be used to detect the antigen to be detected. if the antigen matches any mature detector, it is nonself, otherwise it is self. The main concept of the negative selection algorithm is shown in Fig. 1 [30].

The main concepts of V-Detector algorithm are consistent with NSA, but its detector radius adopts the variable radius idea. The pseudocode in Algorithm 1 describes the detector generation process of the V-Detector algorithm [7]. A candidate detector is randomly generated in the n dimensional feature space (step 6). The candidate detector must successfully go through two tolerance stages before becoming a mature detector. The first stage is tolerated with existing mature detectors (step 7-12), and candidate detectors covered by any mature detector are deleted to avoid overlapping coverage of nonself space. Candidate detectors that successfully go through the first stage are tolerated with the self in the second stage (step 13-18), and candidate detectors that recognize any self are deleted to prevent the detector from recognizing the self [7]. The three convergence conditions of the V-Detector algorithm are: first, the expected coverage rate for nonselfs has been reached (step 11); second, a predetermined number of detectors have been generated (step 19); third, the maximum self coverage (MSC) has been reached (step 18), which means that after repeated multiple times, no detector can be generated, that is, the newly generated detector is not covered by the existing mature detectors, but it is covered by the self samples, so it is difficult to continue generating detectors in the feature space.

Algorithm 1 Detector generation algorithm of V-Detector.

```
V-Detector-Set($S$, $T_{max}$, $r_s$, $c_0$)
$S$: set of Self samples
$T_{max}$: maximum number of detector
$r_s$: self radius
$c_0$: estimated coverage
1 $D \leftarrow \emptyset$
2 Repeat
3    $t \leftarrow 0$
4    $T \leftarrow 0$
5    $r \leftarrow infinite$
6    $x \leftarrow$ random sample form $[1, 0]^{\mathrm{n}}$
7    Repeat for every $\mathrm{d_i}$ in $\mathrm{D} = \{\mathrm{d_i}, \mathrm{i} = 1, 2, \ldots\}$
8       $d_d \leftarrow$ Euclidean distance between $d_i$ and $x$
9       if $\mathrm{d_d} \leq \mathrm{r}(\mathrm{d_i})$ then, where $\mathrm{r}(\mathrm{d_i})$ is the radius of $d$
10          $t \leftarrow t + 1$
11          if $t \geq 1/(1 - c_0)$ then return $D$
12          go to 4:
13     Repeat for every $\mathrm{s_i}$ in S
14        $d \leftarrow$ Euclidean distance between $s_i$ and $x$
15        if $\mathrm{d} - \mathrm{r_s} \leq \mathrm{r}$ then $\mathrm{r} \leftarrow \mathrm{d} - \mathrm{r_s}$ :
16     if $\mathrm{r} > \mathrm{r_s}$ then $\mathrm{D} \leftarrow \mathrm{D} \cup \{< x, r >\}$, where $< x, r >$
is a detector
17     else $T \leftarrow T + 1$
18     if $T > 1/(1 - maximum\ self\ coverage)$ exit
19 until $|D| = T_{\mathrm{max}}$
20 return D
```

2.2 V-Detector-KN algorithm

In real application scenarios, we can often get some negative samples, which we call the known nonselfs. For example, in the field of network anomaly detection, we can regard known abnormal samples as known nonself. These known nonself not only reflect the true distribution of nonself, but there may be other unknown nonself around the known nonself, that is, nonself may be clustered together. Therefore, we can use a known nonself generated detector

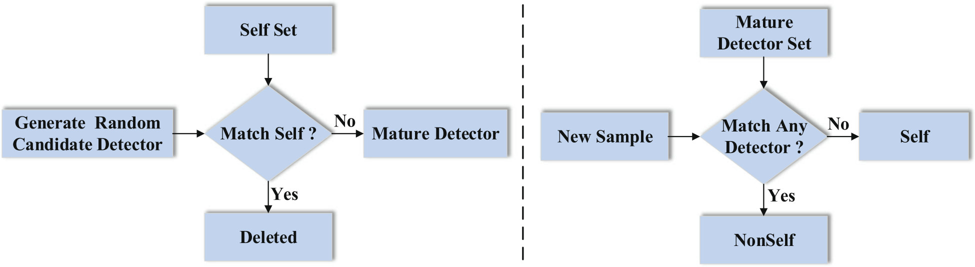


Fig. 1 Main concepts of negative selection algorithms

to detect unknown nonself around it. In addition, the known nonself can generate a detector in a feature space where it is difficult to generate a detector in a random manner. For example, as the number of detectors increases, the coverage of the nonself space by the detectors increases, while the coverage area of the holes decreases. The probability of a randomly generated detector falling into the hole will be reduced, making it difficult to generate a detector in the hole. However, it is easy to generate a detector by a known nonself that is not covered by the detector. The framework of the V-Detector-KN algorithm is shown in Fig. 2.

The V-Detector-KN algorithm uses known nonselfs as candidate detectors to generate mature detectors on the basis of V-Detector generating detectors. Like randomly generated candidate detector, known nonself as candidate detector must undergo two tolerance stages to become a mature detector. First, tolerate with randomly generated mature detectors, and delete candidate detectors covered by any mature detector to avoid overlapping coverage of nonself space. Secondly, the candidate detector that successfully undergoes the first tolerance stage tolerates with self. If the candidate detector is within the coverage of any self, delete it; otherwise, take the candidate detector as the center and the distance to the nearest self as the radius to generate the detector. Algorithm 2 shows the pseudocode for generating the detector by known nonselfs. First, take out known nonself as candidate detector one by one (step1); Second, tolerate with existing mature detectors to avoid overlapping coverage (step2-step5); Finally, tolerate with the self, and delete candidate detectors that recognize the self. Candidate detectors that successfully go through the two tolerance stages will become mature detectors and be added to the detector set (step6-step9).

Algorithm 2 Detector generation by known nonself samples.

```
$S$: set of Self samples
$KN$: set of Known Nonself samples
$num$: number of detectors in $D$
$r_s$: self radius
$r \leftarrow infinite$
1 Repeat for every $\mathrm{kn_j}$ in KN $= \{\mathrm{kn_j}, \mathrm{j} = 1, 2, \ldots\}$
2   Repeat for every $\mathrm{d_i}$ in D $= \{\mathrm{d_i}, \mathrm{i} = 1, 2, \ldots\}$
3     $d_d \leftarrow$ Euclidean distance between $d_i$ and $kn_j$
4     if $\mathrm{d_d} \le \mathrm{r}(\mathrm{d_i})$ then, where $\mathrm{r}(\mathrm{d_i})$ is the radius of $d$
5       go to 1:
6   Repeat for every $\mathrm{s_i}$ in S
7     $d \leftarrow$ Euclidean distance between $s_i$ and $kn_j$
8     if $\mathrm{d} - \mathrm{r_s} \le \mathrm{r}$ then $\mathrm{r} \leftarrow \mathrm{d} - \mathrm{r_s}$
9   if $\mathrm{r}>\mathrm{r_s}$ then D $\leftarrow$ D$\cup\{< kn_j, r >\}$, where$< kn_j, r >$
    is a detector
10 return D
```

3 Algorithm analysis

In this section, we first analyze and compare the performance of the proposed V-Detector-KN algorithm and the V-Detector algorithm, and then analyze the time complexity of the proposed V-Detector-KN algorithm. To increase readability, we summarize the notations used in this section in Table 1.

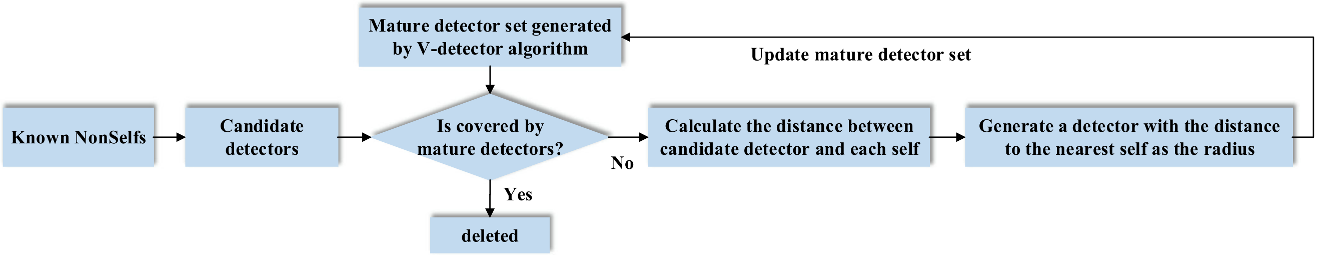

Mature detector set generated by V-detector algorithm
Update mature detector set
Known NonSelfs
Candidate detectors
Is covered by mature detectors?
No
Calculate the distance between candidate detector and each self
Generate a detector with the distance to the nearest self as the radius
Yes
deleted

Fig. 2 V-Detector-KN algorithm framework

Table 1 Notations

Notation	Description
$Data$	Data samples in the data set
$Self$	Normal samples in the data set
$nonself$	Abnormal samples in the data set
N	The number of samples in the $Data$
N_s	The number of $self$
N_n	The number of $nonself$
α	The proportion of the self used for training
β	The proportion of known nonself in nonself
P_m	The probability of matching between any given detector and any data sample
P_n	The probability of the detector identifies the self that is not included in the self training set
P_s	The probability of the detector successfully undergoing self-tolerance phase
N_d	The number of detectors
$V_{detector}$	The volume of detectors
V_{cube}	The volume of the antigen space
r_{d_i}	The detector radius

3.1 Performance comparison and analysis of V-Detector and V-Detector-KN

We first analyzed the probability of the candidate detector successfully experiencing self-tolerance. Secondly, we analyzed the false alarm rate of mature detectors, and finally compared the detection rates of V-Detector and V-Detector-KN.

3.1.1 Probability of candidate detector successfully experiencing self-tolerance

Supposing that $Data$ is the sample set to be classified, it includes self subset $Self$ and nonself subset $NonSelf$, we have $Self \cup NonSelf = Data$, $Self \cap NonSelf = \emptyset$. The number of samples in $Data$ is N, the number of samples in the self subset $Self$ and the nonself subset $NonSelf$ are N_s and N_n respectively, then we have $N = N_s + N_n$. The V-Detector algorithm only uses the self for training. Since the proportion of the self for training is α, $0 \leq \alpha \leq 1$, the number of self used for training is $N_s \cdot \alpha$. Suppose that P_m is the matching probability between any given detector and any data sample, P_s is the probability that the detector successfully undergoing self-tolerance stage, we have,

$$P_s = (1 - P_m)^{N_s \cdot \alpha} \tag{1}$$

Proof Let event A be "the given detector does not match any self in the self training set". Obviously, the detector in the event A is the detector that successfully undergone self-tolerance, i.e., the detector does not match any self. In event A, the number of occurrences of matching between the detector and self meets the binomial distribution, i.e., $Y \sim b(n, p)$ [31], where $n = N_s \cdot \alpha$, $p = P_m$, hence, $P(A) = P(Y = 0) = (P_m)^0 \cdot (1 - P_m)^{N_s \cdot \alpha} = (1 - P_m)^{N_s \cdot \alpha}$, proved. □

In the V-Detector algorithm, P_m can be obtained by calculating the proportion of the detector capacity in the total capacity of the entire antigen space [32].

$$P_m = \frac{N_d \cdot V_{\text{detector}}}{V_{\text{cube}}} = \frac{N_d \cdot \Pi_{i=1}^{n} r_{d_i} \cdot \pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2} + 1\right)} \tag{2}$$

In (2), N_d is the number of detectors, $V_{detector}$ is the volume of detector, V_{cube} is the volume of the antigen space, and r_{d_i} is the detector radius. Γ is the gamma function, defined as [32]:

$$\Gamma\left(\frac{n}{2} + 1\right) = \begin{cases} \left(\frac{n}{2}\right)! & \text{if } n \text{ is even} \\ \frac{\pi^{n/2} \cdot n!}{2^n \left(\frac{n-1}{2}\right)!} & \text{if } n \text{ is odd} \end{cases} \tag{3}$$

3.1.2 Analysis of false alarm rate of mature detector

In the self set $Self$, the number of self not used for training is $N_s \cdot (1 - \alpha)$, the probability that a given detector can detect at least one self that is not used for training is $1 - (1 - P_m)^{N_s \cdot (1-\alpha)}$. Hence, for any mature detector that has successfully undergone self-tolerance, the probability that the detector identifies the self that is not included in the self training set is P_n:

$$\begin{aligned} P_n &= (1 - P_m)^{N_s \cdot \alpha} \cdot \left[1 - (1 - P_m)^{N_s \cdot (1-\alpha)}\right] \\ &= (1 - P_m)^{N_s \cdot (\alpha - 1)} \end{aligned} \tag{4}$$

Proof Let event A be "the given detector does not match any self in the self training set", and event B be "the given detector matches at least one self that is not used for training". Obviously, the detector described by event A are detectors that have successfully experienced self-tolerance, and the probability P_s can be calculated by (1). The detector in event B can detect at least one self that is not used for training. According to the definition of P_n, there is $P_n = P(A) \cdot P(B)$, where $P(A) = (1 - P_m)^{N_s \cdot \alpha}$. In Event B, the number of occurrences of matching between the detector and self also satisfies the binomial distribution, namely $Y \sim b(n, p)$, where $n = N_s \cdot (1 - \alpha)$, $p = P_m$, we have,

$$\begin{aligned} P(B) &= 1 - P(Y = 0) = 1 - (P_m)^0 \cdot (1 - P_m)^{N_s \cdot (1-\alpha)} \\ &= 1 - (1 - P_m)^{N_s \cdot (1-\alpha)} \end{aligned}$$

Therefore,

$$P_n = (1-P_m)^{N_s \cdot \alpha} \cdot \left[1-(1-P_m)^{N_s \cdot (1-\alpha)}\right] = (1-P_m)^{N_s \cdot (\alpha-1)}$$

proved. □

3.1.3 Comparative analysis of detection rate

For any given nonself data sample, the probability of being recognized as nonself by a mature detector, that is, the detection rate P_{tp} is:

$$P_{tp} = 1 - (1 - P_m)^{N_d \cdot (1-P_n)} \tag{5}$$

Proof Let C be the event "nonself data sample matches any detector". According to the definition of detection rate, $P_{tp} = P(C)$. In event C, the number of matches between the nonself data sample and the detector satisfies the binomial distribution $Y \sim b(n, p)$, where $p = P_m$. Since $N_d \cdot P_n$ is the number of mature detectors that identify self, $N_d \cdot (1 - P_n)$ is the number of mature detectors that do not identify self, i.e., $n = N_d \cdot (1-P_n)$, hence, $P_{tp} = P(C) = 1-P(Y = 0) = 1 - (1 - P_m)^{N_d \cdot (1-P_n)}$, proved. □

V-Detector-KN algotithm first generate detectors through the V-Detector algorithm. Supposing that the number of mature detectors generated by the V-Detector is N_{d1}, the detection rate is:

$$P_{tp1} = 1 - (1 - P_m)^{N_{d1} \cdot (1-P_n)} \tag{6}$$

Then, based on the V-Detector generation detectors, V-Detector-KN generate the detector by using known nonself, supposing that the number of mature detectors generated by the V-Detector-KN is N_{d2}, then the algorithm detection rate is:

$$P_{tp2} = 1 - (1 - P_m)^{N_{d2} \cdot (1-P_n)} \tag{7}$$

P_{tp1} is the detection rate of the V-Detector algorithm, and P_{tp2} is the detection rate of our proposed algorithm V-Detector-KN, next we compare the detection rates of the two:

$$\begin{aligned} P_{tp2} - P_{tp1} &= 1 - (1 - P_m)^{N_{d2} \cdot (1-P_n)} \\ &\quad - \left(1 - (1 - P_m)^{N_{d1} \cdot (1-P_n)}\right) \\ &= (1-P_m)^{N_{d1} \cdot (1-P_n)} \\ &\quad - (1-P_m)^{N_{d2} \cdot (1-P_n)} \end{aligned} \tag{8}$$

In (8), $N_{d2} \geq N_{d1}$ is constant. When $N_{d2} = N_{d1}$, it means that the new detector cannot be generated by known nonself, so $P_{tp2} - P_{tp1} = 0$, that is, the detection rate of V-Detector-KN and V-Detector is equal; when $N_{d2} > N_{d1}$, $\frac{(1-P_m)^{N_{d1} \cdot (1-P_n)}}{(1-P_m)^{N_{d2} \cdot (1-P_n)}} = (1 - P_m)^{(N_{d1}-N_{d2}) \cdot (1-P_n)}$. since $0 < (1 - P_m) < 1$, and $(N_{d1} - N_{d2}) \cdot (1 - P_n) < 0$. According to the nature of the exponential function [33], we have $(1 - P_m)^{(N_{d1}-N_{d2}) \cdot (1-P_n)} > 1$, hence, $P_{tp2} - P_{tp1} > 0$. This shows that our V-Detector-KN algorithm is superior to the conventional V-Detector algorithm in detection rate, and the more detectors generated by known nonselfs, the greater the advantage of our algorithm is.

3.2 Algorithm time complexity

In terms of algorithm time complexity, the V-Detector-KN algorithm we proposed is consistent with the V-Detector algorithm, which can be expressed as [24]:

$$O\left(\frac{N_d * N_s}{(1 - P_m)^{N_s}}\right) \tag{9}$$

In (9) , N_d is the number of detectors, N_s is the number of self training set, and P_m is the matching probability between any given detector and any data sample. Compared with the V-Detector algorithm, the V-Detector-KN algorithm use known nonself to generate new detectors. Therefore, the number of self-sets N_d has increased.

4 Parameter analysis of the V-Detector-KN algorithm

The main parameters of the V-Detector-KN algorithm include the self radius r_s, the expected coverage rate c_0, the maximum self coverage MSC, the proportion of self used for training α, $0 \leq \alpha \leq 1$, and the proportion of known nonself β, $0 \leq \beta \leq 1$. The first 4 parameters are the general parameters of the NSA algorithm, and we use the synthetic data set (SDS) for experimental analysis. The last parameter β is a unique parameter of our algorithm, we will analyze it in the demonstration example in Section 5. The SDS was proposed by the famous artificial immune research institution, the intelligent security laboratory of the university of memphis, and has been generally accepted by the international academic community [8, 27, 32]. The four SDS data sets we used in this study are: Ring, Rectangle, Pentagram and Cross. The composition of self antigens and nonself antigens of these four data sets is shown in Table 2. The data distribution is shown in Fig. 3, yellow dots represent self antigens and the blue dots represent nonself antigens.

Detection Rate (DR), False Alarm Rate (FAR), number of detectors, detector training time and detection time are often used to evaluate the performance of NSA [11, 12, 27]. In this paper, we continue to use these five evaluation metrics for evaluation. The calculation methods of DR and FAR are as follows:

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

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

In (10) and (11), TP (True Positive) represents that the nonself is correctly classified as nonself, FN (False Negative) represents that the nonself is incorrectly classified as self, and FP (False Positive) represents that the self is incorrectly classified as nonself, TN (True Negative) represents that the self is correctly classified as self [11].

The experiment uses the control variates method to analyze the impact of each parameter on the algorithm performance. The control variates method only changes one of the parameters at a time, while controlling the rest of the parameters unchanged, so as to study the impact of the changed parameter on the performance of the algorithm, study each parameter separately, and finally obtain the algorithm parameters comprehensively. In the experiment, the self radius r_s ranges from 0.01 to 0.1; the proportion of the self α used for training in the self set varies from 0.1 to 1; both the expected coverage rate c_0 and the maximum self coverage MSC are range from 0.90 to 0.99. When analyzing a certain parameter, we set the remaining parameters as default values, which are shown in Table 3. The default parameters first obtain the corresponding value range according to the literature [8, 11, 32], and then use the controlled variable method to experiment to obtain specific values.

Table 2 The detail of 4 Synthetic data sets

Data set	Records number	Self	Nonself
Ring	10000	3710	6290
Rectangle	10000	3386	6614
Pentagram	10000	2850	7150
Cross	10000	5531	4469

4.1 The impart of self radius on algorithm performance

Figure 4 shows the impact of self radius r_s on the performance of the V-Detector-KN algorithm. Figure 4a and b respectively show that the detection rate and false alarm rate decrease as the proportion of the self training set increases. However, we expect the algorithm to have a high detection rate and a low false alarm rate. Therefore, we need to achieve tradeoff between the detection rate and the false alarm rate. Generally, in the scenarios sensitive to abnormal samples, the value of the self radius can be reduced, and in the scenarios sensitive to false alarms, the self radius can be increased. Figure 4c, d and e respectively show that the number of detectors, detector generation time and detection time converge rapidly as the self radius increases, which is consistent with the V-Detector algorithm.

4.2 The impact of the proportion of self used for training on algorithm performance

Figure 5 shows the impact of the self proportion used for training on the performance of the V-Detector-KN algorithm. Figure 5a shows that the detection rate decreases as the proportion of self increases. This is because the candidate detector must undergo self-tolerance process to become a mature detector. As the proportion of the self increases, the number of self increases, the probability of candidate detectors successfully experiencing self-tolerance decreases, and the number of generated detectors decreases, resulting in a decrease in the detection rate. Figure 5b shows that the false alarm rate converges as the self proportion increases. When the proportion of self is low, the number of self is small, and the self area cannot effectively cover all self samples. Some self samples fall into the non self area and are incorrectly identified as nonself, resulting in a high false alarm rate. As the proportion of self increases and the number of self increases, the self area will completely cover all self samples, so the false alarm rate converges to 0. Figure 5c shows that as the self proportion increases, the number of detectors decreases. The reason is that as the proportion of self increases, the number of self used for training increases. The probability of successful tolerance between candidate detectors and the self is reduced, which leads to a decrease in the number of detectors. Figure 5d and e respectively show that the detector generation time and detection time decrease with the increase of the self proportion. Detector generation time and detection time are both related to the number of detectors.

Table 3 Default values of parameters in V-Detector-KN

Parameter	Default value	Parameter	Default value
r_s^a	0.01	c_0^b	0.99
α^c	0.9	MSC^d	0.99
β^e	0.9		

[a] r_s: self radius; [b] c_0: expected coverage rate; [c] α: proportion of training self in self ; [d] MSC: maximum self coverage; [e] β:proportion of known nonself in nonself

4.3 The impact of expected coverage rate on algorithm performance

The expected coverage rate refers to the proportion of the nonself space covered by the detector in the nonself space. Due to the overlapping coverage of the detectors, it is difficult to calculate the expected coverage rate. Therefore, Zhouji proposed a "point estimation" Monte Carlo method for estimate [8].

$$m \geq 1/(1-\varepsilon) \tag{12}$$

In (12), m is the number of randomly sampled samples, and ε is the expected coverage rate. If $\varepsilon = 0.8$, then $m \geq 5$ can be calculated. That is, if 5 consecutive random sampling points are covered by the mature detector set, it means that the expected coverage rate of the nonself space by the detector set has reached 80%.

Figure 6 shows the impact of expected coverage rate on the performance of the V-Detector-KN algorithm. The detection rate of the V-Detector algorithm increases with the increase of the expected coverage rate, but in our algorithm, the increase in the detection rate is not significant, as shown in Fig. 6a. This is because our algorithm uses known nonself generated detectors based on the detectors generated by V-Detector, thus repairing the holes formed between the detectors when the expected coverage is low. Figure 6b shows that the false alarm rate increases with the expected coverage rate. This is because as the expected coverage rate increases, the number of detectors generated by V-Detector increases, which not only increases the coverage of the nonself space, but also increases the coverage of the self space, leading to an increase in the false alarm rate. Figure 6c and e respectively show that the number of detectors and the detection time increase as the expected coverage rate increases. The number of detectors determines the detection time. Figure 6d shows that the detector generation time increases with the increase of the expected coverage rate, the reason is that the random sampling number m increases.

4.4 The impact of maximum self coverage on algorithm performance

In addition to tolerance with existing mature detectors, candidate detectors must also tolerate with selfs. The maximum self coverage rate reflects the probability of randomly generated candidate detector falling into the self region. It can also be estimated by Monte Carlo method of point sampling [8], refer to (12).

Figure 7 shows the impact of the maximum self coverage rate on the performance of the V-Detector-KN algorithm. Figure 7a shows that the detection rate increases with the increase of the maximum self coverage rate. The reason is that with the increase of the maximum self coverage rate, the number of generated detectors increases, and the coverage rate of the detector to the non self space increases, thereby increasing the detection rate. However, the increase in the maximum self coverage rate also brings an increase in the false alarm rate, because the detector may cover the self area, as shown in Fig. 7b. Figure 7c shows that the number of detectors increases as the maximum self coverage rate increases. The reason is that as the maximum self coverage rate increases, the number of random sampling increases, thereby increasing the probability of generating a detector. Figure 7d and e respectively show that the detector generation time and detection time increase with the increase of the maximum self coverage rate.

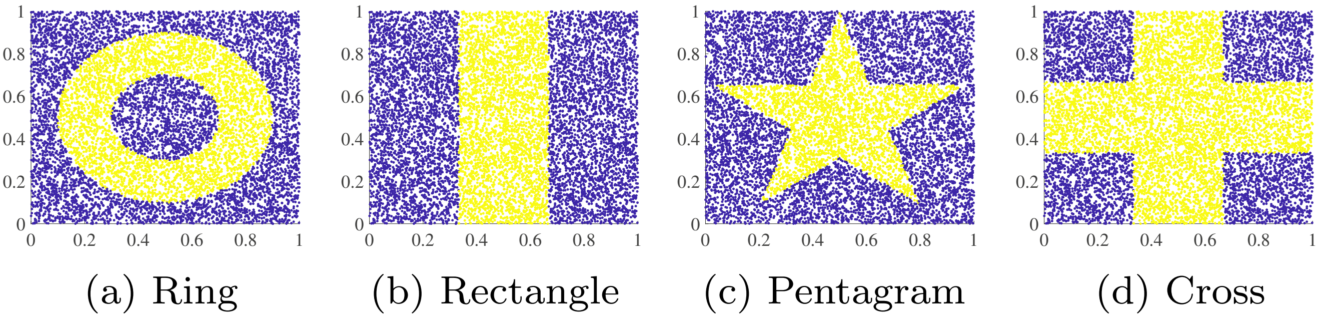
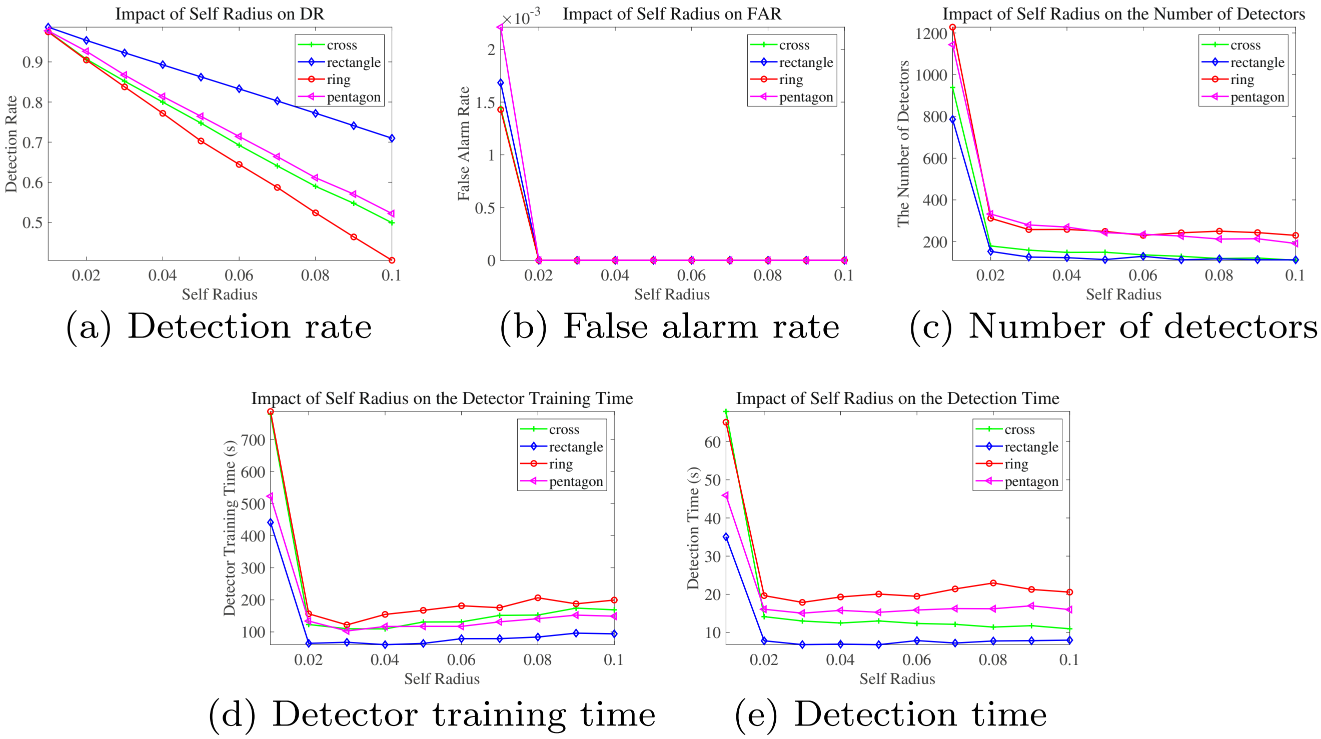


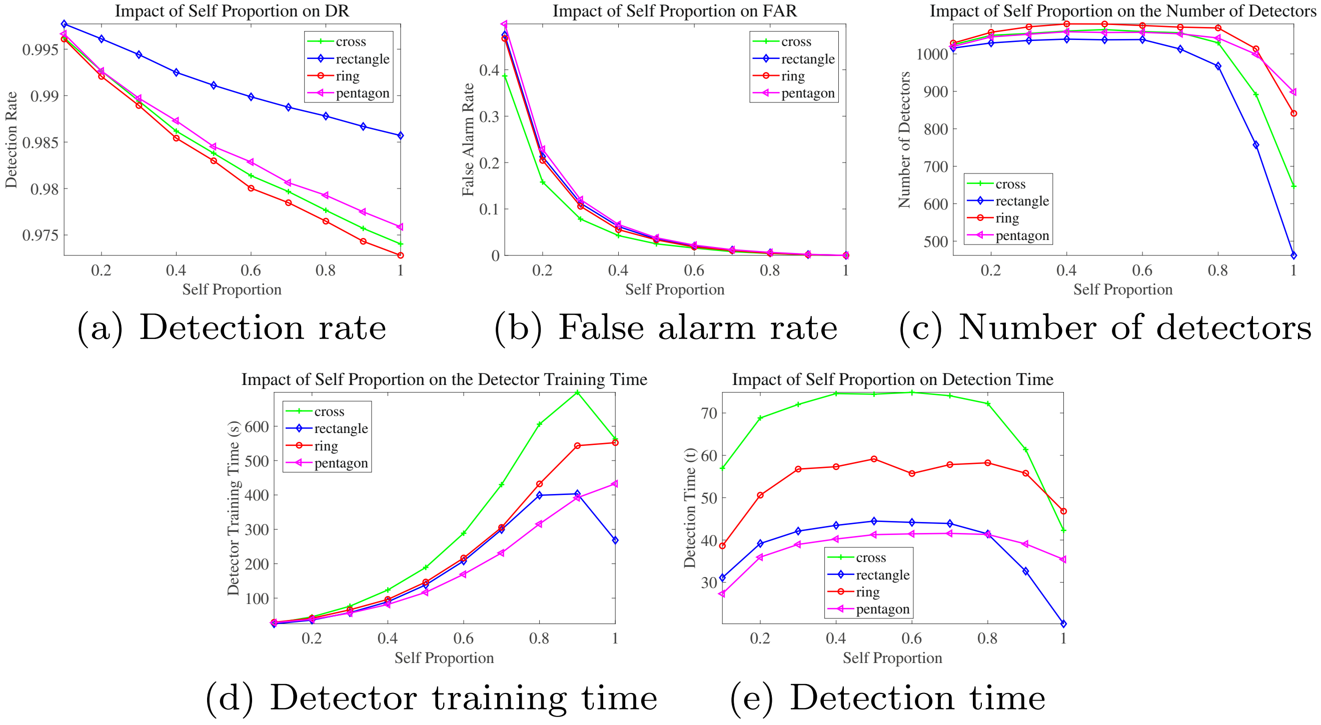
Fig. 3 The distribution of SDS data set



(a) Detection rate (b) False alarm rate (c) Number of detectors

(d) Detector training time (e) Detection time

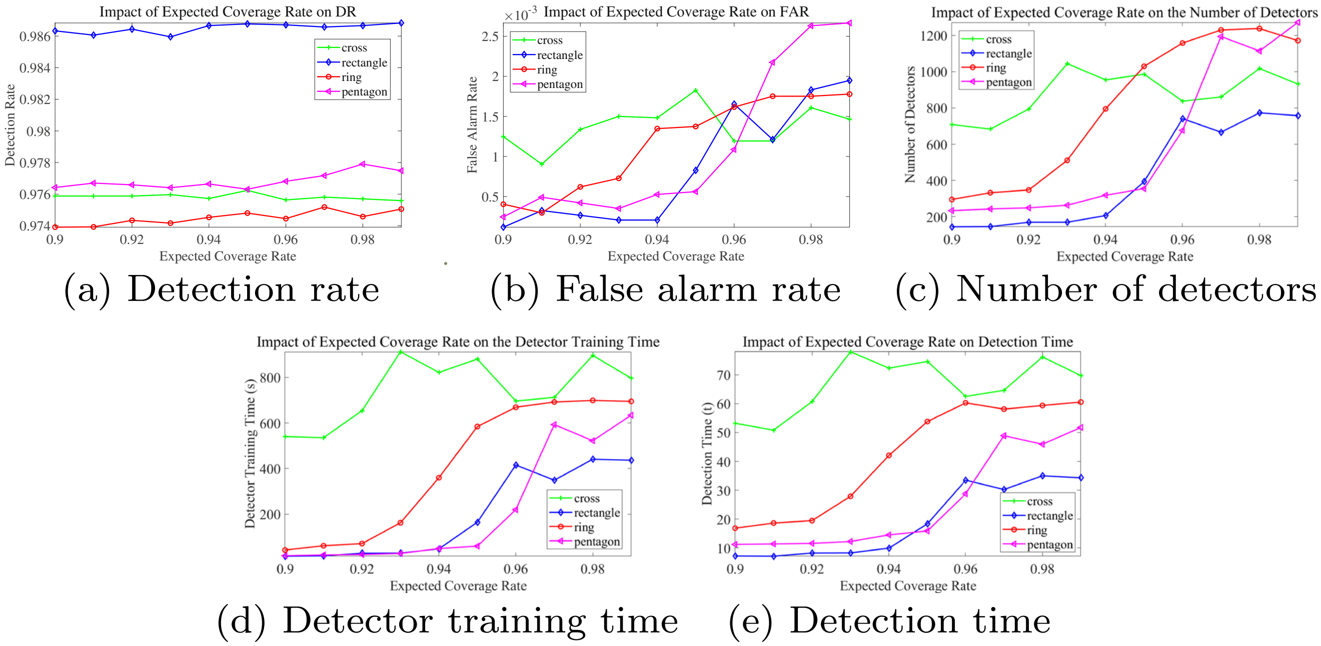
Fig. 4 The impact of self radius on algorithm performance



(a) Detection rate (b) False alarm rate (c) Number of detectors

(d) Detector training time (e) Detection time

Fig. 5 The impact of self proportion on algorithm performance



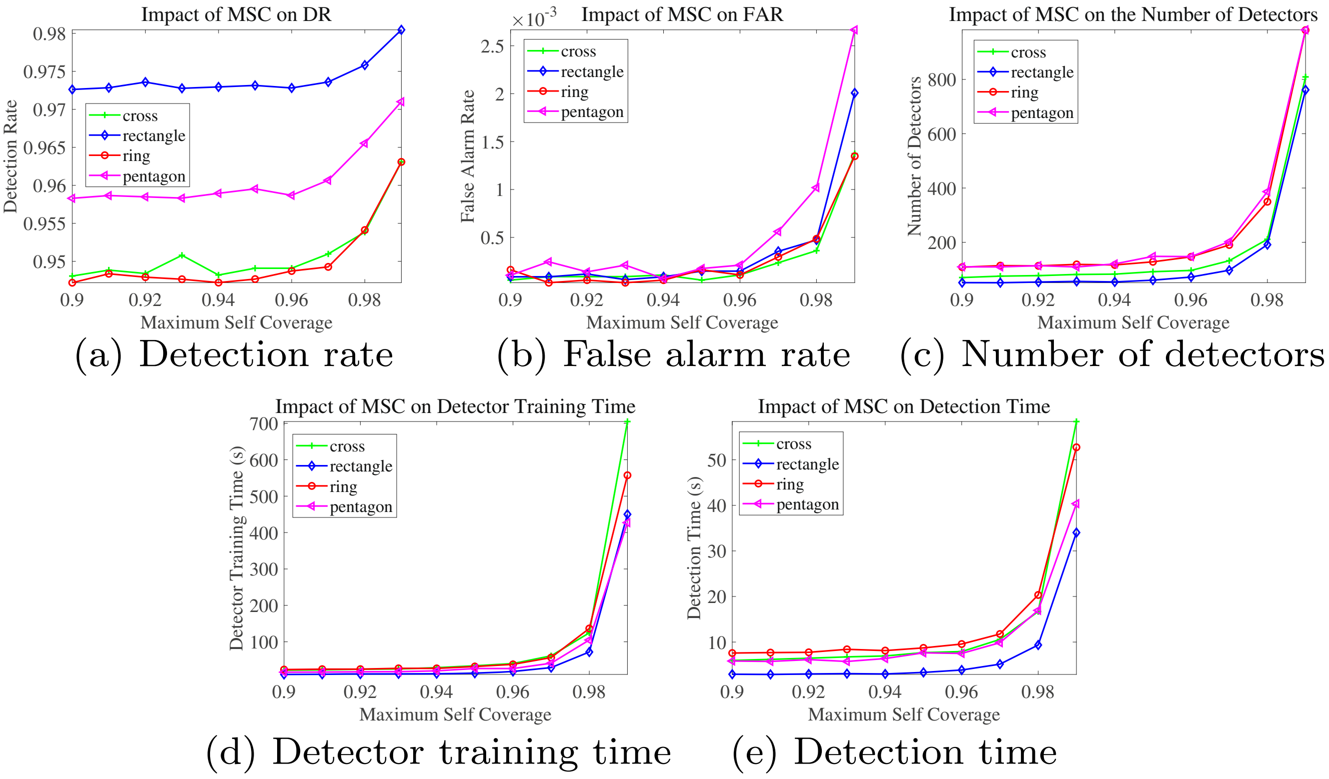
(a) Detection rate (b) False alarm rate (c) Number of detectors

(d) Detector training time (e) Detection time

Fig. 6 The impact of expected coverage rate on algorithm performance

5 Example demonstration

In order to better understand our proposed V-Detector-KN algorithm, haberman's survival data set is utilized for further explanation. Haberman's survival data set contain cases from a study that was conducted between 1958 and 1970 at the university of chicago's billings hospital on the survival of patients who had undergone surgery for breast cancer. It



(a) Detection rate (b) False alarm rate (c) Number of detectors

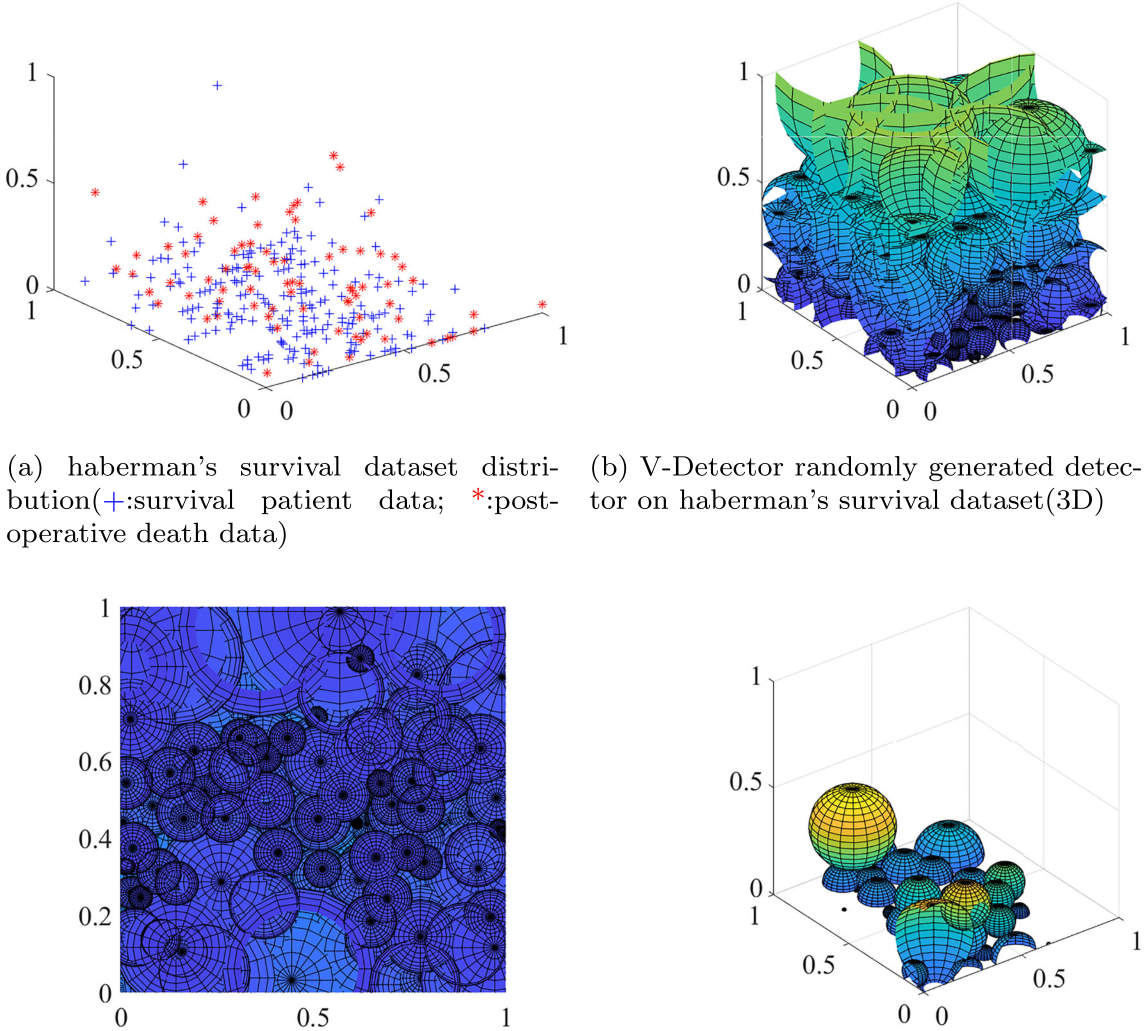
(d) Detector training time (e) Detection time

Fig. 7 The impact of maximum self coverage on algorithm performance

contains 306 post-operative patient data records, including 225 post-operative survival patient data (class label=1) and 81 post-operative death data (class label=2). Each record contains 3 attributes and 1 class label [34].

5.1 Demonstration of known nonself generating detector

We first demonstrate that the known nonself can further generate detectors based on the V-Detector algorithm. On the haberman's survival data set, we regard the dead patients data as self, and the surviving patients data as nonself. Figure 8a shows the distribution of self and nonself. The red "*" represents self and the blue "+" represents nonself. On the haberman's survival data set, the data samples are not uniformly distributed, but densely distributed in the bottom area, and sparsely distributed in the top area. Moreover, the self and nonself are interlaced in the dense area at the bottom, making it difficult to randomly generate detectors in this area. Figure 8b shows the distribution of the detectors randomly generated by the V-Detector. In the top area where the samples are sparsely distributed, large radius detectors can be generated to cover larger nonself space, while in the bottom area where the samples are densely distributed, only small radius detectors can be generated. Many holes are formed between these small radius detectors, as shown



(a) haberman's survival dataset distribution(+:survival patient data; *:post-operative death data)

(b) V-Detector randomly generated detector on haberman's survival dataset(3D)

(c) holes on haberman's survival dataset by V-Detector(bottom view)

(d) known nonself generated detectors

Fig. 8 Demonstration example on haberman's survival data set

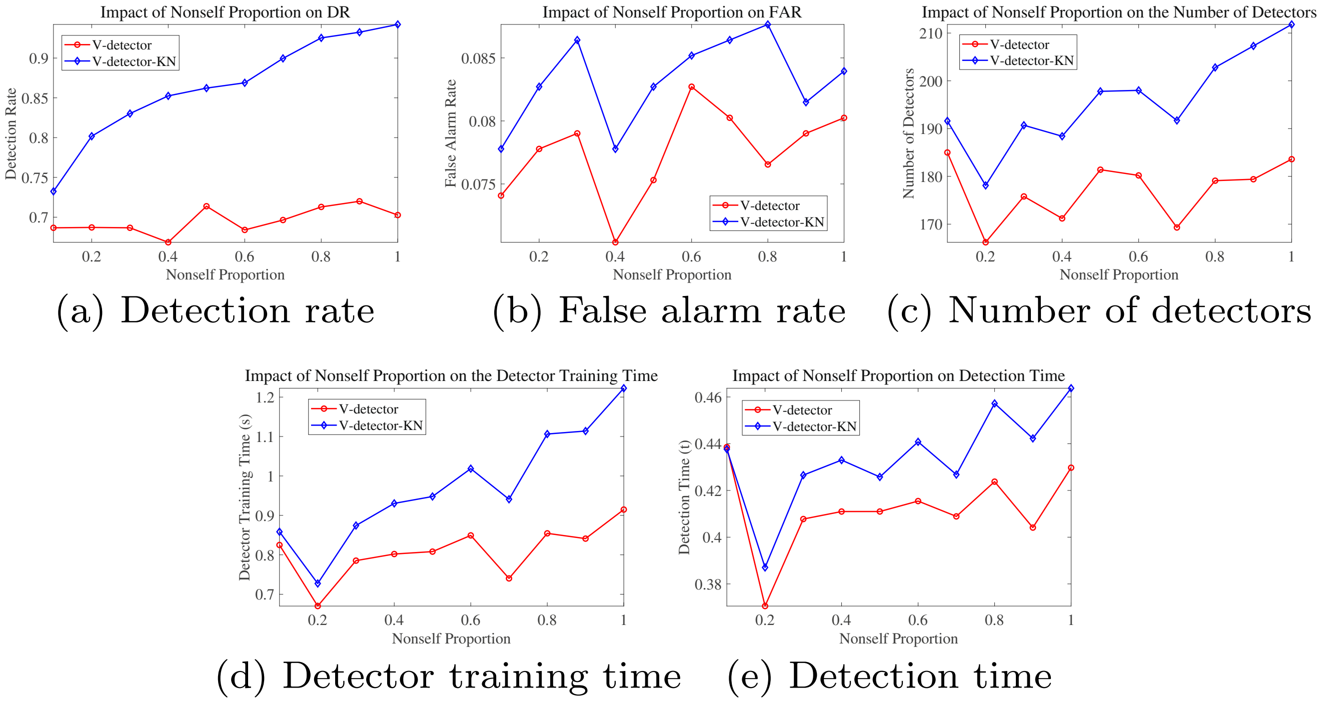
in the bottom view of Fig. 8c. To repair these holes, we introduce known nonself, and generate detectors through known nonself to achieve hole repair. Figure 8d shows the detectors generated by known nonself. This shows that our algorithm can generate detectors in a feature space where it is difficult to generate detectors in a random manner.

5.2 The impact of known nonself proportion on algorithm performance

The proportion of known nonself refers to the proportion of known nonself in nonself. We take the haberman's survival data set as an example to analyze the impact of the nonself proportion on the performance of the V-Detector-KN algorithm. We assume that the proportion of the known nonself in the nonself is β, $0 \leq \beta \leq 1$. When $\beta = 0$, it means that there is no known nonself, and all nonself are unknown. At this time, our algorithm degenerates to the V-Detector algorithm.

When the data set is small, k-fold cross-validation is often used. This method divides the available data set into k disjoint equal-sized data subsets, and then each subset is taken as the test set, and the remaining $k - 1$ subsets are taken together as the training set to learn the classifier. This process is performed a total of k times and produces k test accuracy. The final estimated accuracy on this data set is the average of these k test accuracy. 10-fold and 5-fold cross-validation are often used [35].

In this experiment, we randomly selected the nonself with the proportion of β as the known non-self, and then select the self with the 10-fold cross-validation method to verify the impact of the known nonself proportion β on the performance of the algorithm. Figure 9 shows the performance comparison between V-Detector and V-Detector-KN under different known nonself proportion. Figure 9a shows that the detection rate of the V-Detector-KN algorithm increases with the increase in the proportion of known nonself. Figure 9b shows that the false alarm rate of our algorithm is higher than that of the V-Detector algorithm, but not more than 1%. Figure 9c shows that the number of detectors generated by V-Detector-KN is more than that of the V-Detector algorithm. This proves that our method can generate detectors based on the V-Detector algorithm to improve its performance. Figure 9d shows that compared to the V-Detector algorithm, the detector generation time of the V-Detector-KN algorithm has increased, because the known nonself must successfully tolerate with the self and the existing mature detectors to become a mature detector. Figure 9e shows that the detection time of V-Detector-KN is longer than that of the V-Detector algorithm, because the detection time is positive correlated with the number of detectors. The detailed performance comparison between the V-Detector-KN algorithm and the V-Detector algorithm is shown in Table 4.



(a) Detection rate (b) False alarm rate (c) Number of detectors

(d) Detector training time (e) Detection time

Fig. 9 Performance comparison of V-Detector and V-Detector-KN

Table 4 Performance comparison of V-Detector and V-Detector-KN on the haberman's survival data set

β[a]	V-Detector						V-Detector-KN					
	DR[b]		FAR[c]		ND[d]	T_{train}[e]	DR		FAR		ND	T_{train}
	avg[f]	var[g]	avg	var			avg	var	avg	var		
0.1	68.67%	4.11%	7.41%	1.65%	185	0.82	73.24%	3.68%	7.78%	1.84%	191.6	0.86
0.2	68.71%	6.04%	7.78%	1.02%	166.2	0.67	80.18%	4.05%	8.27%	1.31%	178.1	0.73
0.3	68.67%	6.87%	7.90%	2.19%	175.8	0.79	83.02%	3.90%	8.64%	1.43%	190.7	0.87
0.4	66.84%	7.38%	7.04%	1.65%	171.2	0.80	85.24%	1.99%	7.78%	1.93%	188.4	0.93
0.5	71.38%	3.07%	7.53%	1.97%	181.4	0.81	86.22%	1.92%	8.27%	1.43%	197.8	0.95
0.6	68.40%	9.98%	8.27%	1.17%	180.2	0.85	86.89%	2.17%	8.52%	0.91%	198	1.02
0.7	69.64%	6.66%	8.02%	1.20%	169.3	0.74	89.96%	1.58%	8.64%	1.01%	191.7	0.94
0.8	71.29%	6.94%	7.65%	1.13%	179.1	0.85	92.53%	1.73%	8.77%	1.36%	202.8	1.11
0.9	72.00%	7.59%	7.90%	1.19%	179.4	0.84	93.24%	0.81%	8.15%	1.04%	207.3	1.11
1	70.27%	7.82%	8.02%	1.33%	183.6	0.91	94.22%	1.13%	8.40%	1.13%	211.8	1.22

[a] β: proportion of known nonself in nonself; [b] DR: Detection Rate; [c] FAR: False Alarm Rate; [d] ND: Number of Detector; [e] T_{train}: Detector training time; [f] avg: avg is the average of 20 repeated experiments; [g] var: var is the variance of 20 repeated experiments

As shown in Table 4, when the proportion of the known nonself is 90%, the detection rate of V-Detector on the haberman's survival data set is 72.00%, while that our algorithm is 93.24%, an increase of about 30%. As for the number of detectors, compared with the 179.4 of the V-Detector algorithm, the V-Detector-KN algorithm is 203.5, an increase of 15.5%. The percentage increase in detection rate (29.51%) is higher than the percentage increase in the number of detectors (15.5%). This shows that the known nonself generated detector can cover the unknown nonself. In addition, the variance of detection rate becomes smaller, indicating that the stability of the V-Detector-KN algorithm is better than that of the V-Detector. However, compared to the false alarm rate of V-Detector algorithm of 7.9%, the false alarm rate of V-Detector-KN is 8.15%, an increase of about 3%.

6 Comparative experiment on UCI data set

In this section, we compare our algorithm with other 6 algorithms on 7 UCI real data sets. The algorithms for comparison include V-Detector [7], improved V-Detector [9], ADC-NSA [12], BIORV-NSA [26], KNN [29] and SVM [36]. The data set for comparison uses 7 real UCI data sets, which are often used to evaluate and compare artificial intelligence algorithms [37, 38]. Five performance evaluation metrics are used to evaluate the performance of the algorithm (see Section 4).

In the comparative experiment, we still use the 10-fold cross-validation method described in Section 5.2. In this study, in order to ensure that the number of self and non-self acquired each time is equal, we first divide the data set into self and nonself sets, and then use 10-fold cross-validation for self and non-self sets respectively. Therefore, each test set is the union of 1 self subset and 1 nonself subset, and the training set is the union of the remaining 9 self subsets and 9 nonself subsets. The experiment is repeated 10 times until all samples have been verified at least once. The parameters used in the V-Detector-KN algorithm are shown in Table 3 in Section 4.

6.1 Iris data set

The Iris data set is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant [39]. The class labels of the iris data set are 1-3. We treat instances with a class label equal to 1 as self, and the rest as non-self. The experimental results are shown in Table 5.

On the iris data set, all 7 algorithms can reach a detection rate of 100%. In terms of false alarm rate, both SVM [36] and KNN [29] algorithms are 0, which has advantages over negative selection algorithms. In terms of the number of detectors, the improved V-Detector [9] algorithm has the least number of detectors, because the algorithm adopt detector optimization mechanism to remove redundant detectors. The detector generation time and detection time of the BIORV-NSA [26] algorithm are longer than other algorithms, because the algorithm uses the maximum number of detectors as the termination condition, that is, the algorithm terminates when the number of detectors reaches the preset value of 1000. The ADC-NSA

Table 5 Experimental results of iris data set

Algorithm	DR(%)		FAR(%)		Generated Detectors		Detector Training Time(s)		Detection Time(s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
V-Detector	100	0	1	1.70	11	3.33	0.013	0.005	0.008	0.002
BIORV-NSA	100	0	3.8	0.02	1000	0	6.7	0.15	0.63	0.06
KNN	100	0	0	0						
SVM	100	0	0	0						
ADC-NSA	100	0	1	1.70	11	3.33	0.013	0.005	0.008	0.002
Improved V-Detector	100	0	1	1.40	4.1	1.2	0.06	0.004	0.01	0.001
V-Detector-KN	100	0	0.8	0.01	9.8	2.8	0.013	0.003	0.007	0.002

[12] 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. On the iris data set, the ADC-NSA algorithm generates 2 clusters, but since these two clusters are self clusters, the algorithm degenerates to the V-Detector algorithm. The improved V-Detector [9] algorithm firstly generated detectors with big radius to cover most of the nonself space, and then generated detectors with small radius to cover small holes around self samples, finally, optimize the detector to remove redundant detectors.

6.2 Banknote authentication data set

The banknote authentication data set contains 1372 instances, the class labels are 0 and 1, and the number of attributes is 4 [40]. We regard the class label equal to 1 as the self, and the class label equal to 0 as the nonself. The experimental results are shown in Table 6. On the banknote data set, BIORV-NSA [26], SVM [36] and V-Detector-KN all achieve 100% detection rates, but the BIORV-NSA algorithm requires far more detectors than the V-Detector-KN algorithm, and the false alarm rate is higher than the V-Detector-KN algorithm. In terms of the number of detectors generated, the improved V-Detector [9] algorithm still has the least number of detectors.

6.3 Skin segmentation data set

The skin segmentation data set contains 245057 instances, the class labels are 1 and 2, and the number of attributes is 3 [41]. We regard the class label equal to 1 as the self, and the class label equal to 2 as the nonself. The experimental results are shown in Table 7. On the skin segmentation data set, the V-Detector-KN algorithm we proposed has the highest detection rate, reaching 99.97%. The SVM [36] algorithm fails on this data set, and its detection rate is only 36.80%. The ADC-NSA [12] algorithm degenerates to the V-Detector algorithm on the skin segmentation data set. The reason is that the algorithm needs to calculate the distance between antigens when performing antigen density clustering. However, when the number of instances in the data set is too large, the distance matrix will exceed

Table 6 Experimental results of banknote authentication data set

Algorithm	DR(%)		FAR(%)		Generated Detectors		Detector Training Time(s)		Detection Time(s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
V-Detector	98.50	1.05	0.33	0.23	89.5	20.3	0.70	0.20	0.71	0.14
BIORV-NSA	100	0	1.57	0.007	1000	0	11.1	0.28	6.6	0.13
KNN	99.74	0.55	0	0						
SVM	100	0	0	0						
ADC-NSA	99.28	1.2	0.38	0.26	104.9	19.4	1.73	0.52	0.79	0.13
Improved V-Detector	98.60	4.23	0.2	0.19	11	6.7	0.98	0.11	0.26	0.04
V-Detector-KN	100	0	0.28	0.26	95.1	18.8	0.84	0.22	0.74	0.16

Table 7 Experimental results of skin segmentation data set

Algorithm	DR(%)		FAR(%)		Generated Detectors		Detector Training Time(s)		Detection Time(s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
V-Detector	98.55	1.32	0.0007	0.001	79.6	25.9	46.6	17.4	57.7	13.4
BIORV-NSA	99.90	0.05	0.001	0.001	1000	0	493.9	12.0	580.7	13.3
KNN	99.93	0.02	0.014	0.019						
SVM	36.80	0.32	1	0						
ADC-NSA	98.55	1.32	0.0007	0.001	79.6	25.9	46.6	17.4	57.7	13.4
Improved V-Detector	95.96	3.58	0.0002	0.0006	10.0	4.2	82.57	5.45	27.54	4.42
V-Detector-KN	99.97	0.001	0.001	0.002	190.1	20.4	142.2	15.9	130.0	13.3

Table 8 Experimental results of pima indians diabetes data set

Algorithm	DR(%)		FAR(%)		Generated Detectors		Detector Training Time(s)		Detection Time(s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
V-Detector	27.58	3.56	1.18	0.49	98.8	14.3	0.75	0.15	0.79	0.11
BIORV-NSA	45.26	3.22	2.28	0.33	1000	0	11.47	0.43	8.25	0.38
KNN	56.34	9.24	16.60	6.26					0.008	0.0002
SVM	100	0	0	0						
ADC-NSA	27.58	3.56	1.18	0.49	98.8	14.3	0.75	0.15	0.79	0.11
Improved V-Detector	17.05	7.53	0.54	0.23	14.2	7.1	1.78	1.96	0.40	0.22
V-Detector-KN	95.93	1.00	2.70	0.53	223.9	13.3	1.64	0.16	1.61	0.10

Table 9 Experimental results of balance scale data set

Algorithm	DR(%)		FAR(%)		Generated Detectors		Detector Training Time(s)		Detection Time(s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
V-Detector	95.43	0.97	7.81	0.60	501.0	51.1	6.66	1.71	2.05	0.20
BIORV-NSA	99.11	0.61	7.36	1.01	1000	0	9.24	0.48	3.72	0.17
KNN	80.12	3.15	0	0					0.006	0.001
SVM	100	0	0	0						
ADC-NSA	95.43	0.97	7.81	0.60	501	51.1	6.66	1.71	2.05	0.20
Improved V-Detector	85.46	7.09	4.17	1.53	18.7	12.4	86.74	176.9	0.60	0.56
V-Detector-KN	99.85	0.21	7.95	0.78	507.2	60.1	6.39	2.10	1.89	0.19

Table 10 Experimental results of breast cancer wisconsin (diagnostic) data set

Algorithm	DR(%)		FAR(%)		Generated Detectors		Detector Training Time(s)		Detection Time(s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
V-Detector	81.84	4.84	0.22	0.18	38.9	11.5	0.19	0.06	0.18	0.05
BIORV-NSA[a]	9.76	2.18	4.88	0.01	10000	0	576.3	4.15	60.06	1.06
KNN	93.90	4.98	1.11	1.43						
SVM	100	0	0	0						
ADC-NSA	81.84	4.84	0.22	0.18	38.9	11.5	0.19	0.06	0.18	0.05
Improved V-Detector	79.67	3.97	0.20	0.27	6.0	4.6	0.58	0.05	0.12	0.02
V-Detector-KN	99.43	0.62	0.73	0.55	58.7	8.9	0.29	0.07	0.26	0.04

[a] $BIORV-NSA$: In WDBC data set, the maximum number of BIORV-NSA generative detectors is set to 10000

the maximum range and an error will be reported. The Improved V-Detector [9] algorithm reduces the storage space by deleting redundant detectors, but it also leads to a decrease in the algorithm's detection rate.

6.4 Pima Indians diabetes data set

The pima Indians diabetes data set contains 768 instances, the class labels are 0 and 1, and the number of attributes is 8 [42]. We regard the class label equal to 0 as the self, and the class label equal to 1 as the nonself. The experimental results are shown in Table 8. On the pima Indians diabetes data set, the SVM [36] algorithm has the highest detection rate, reaching 100%. The detection rate of our proposed V-Detector-KN algorithm is 95.93%, ranking second. Note that the remaining 5 algorithms fail on this data set, which also shows the stability of our proposed algorithm.

6.5 Balance scale data set

The balance scale data set contains 625 instances, the class labels are 1 to 3, and the number of attributes is 4 [43]. We regard the class label equal to 3 as the self, and the rest as the nonself. The experimental results are shown in Table 9. The V-Detector-KN algorithm we proposed has the highest detection rate, reaching 99.85%. The BIORV-NSA [26] algorithm comes next with 99.11%. The detection rate of KNN [29] algorithm is only 80.12%. In terms of the number of detectors, the improved V-Detector [9] algorithm has an average of 18.7 detectors, which has a great advantage over other NSAs. The reason is that the algorithm uses a detector optimization strategy to delete redundant detectors, but it also leads to a decrease in detection rate. On the Balance Scale data set, the ADC-NSA [12] algorithm generates 1 cluster, which is self-clustering. Therefore, the algorithm degenerates to the V-Detector algorithm.

6.6 Breast cancer wisconsin (diagnostic) data set

The breast cancer wisconsin (diagnostic) data set contains 569 instances, the class labels are 0 and 1, and the number of attributes is 31 [44]. We regard the class label equal to 0 as the self, and the class label equal to 1 as the nonself. The experimental results are shown in Table 10. On the breast cancer wisconsin (diagnostic) data set, the detection rate of the V-Detector-KN algorithm we proposed is lower than SVM, but higher than the other 5 algorithms. The BIORV-NSA [26] algorithm fails in this data set. The reason is that

Table 11 Experimental results of haberman's survival data set

Algorithm	DR(%)		FAR(%)		Generated Detectors		Detector Training Time(s)		Detection Time(s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
V-Detector	71.82	5.61	8.15	1.04	185.5	18.7	0.93	0.26	0.43	0.03
BIORV-NSA	73.87	3.34	8.15	1.04	1000	0	6.53	0.19	2.06	0.16
KNN	76.05	8.77	62.78	10.61						
SVM	100	0	0	0						
ADC-NSA	71.82	5.61	8.15	1.04	185.5	18.7	0.93	0.26	0.43	0.03
Improved V-Detector	41.24	19.84	5.56	2.12	73.6	61.1	0.35	0.36	0.18	0.12
V-Detector-KN	93.16	1.88	8.52	1.23	204.5	26.2	1.12	0.31	0.45	0.05

BIORV-NSA fails to effectively generate more detectors in the high-dimensional feature space. On the breast cancer wisconsin (diagnostic) data set, the ADC-NSA algorithm generates 1 cluster, which is self clustering. Therefore, the algorithm degenerates to the V-Detector algorithm.

6.7 Haberman's survival data set

The haberman's survival data set contains 306 instances, the class labels are 1 and 2, and the number of attributes is 3 [34]. We regard the class label equal to 2 as the self, and the class label equal to 1 as the nonself. The experimental results are shown in Table 11. On the haberman's survival data set, since self and nonself are distributed in low-dimensional subspace and interleaved with each other, as shown in Fig. 8a, it is difficult for V-Detector to generate detectors in dense spaces. Therefore, the performance of V-Detector and its improved algorithm is not ideal. The V-Detector-KN algorithm we proposed solves this problem by using known nonself generating detectors. Although the detection rate is not as good as SVM [36], it is higher than the other 5 algorithms. The KNN algorithm fails on haberman's survival data set. Although it achieves a high detection rate, it also has a high false alarm rate. The reason is that the positive and negative examples in the haberman's survival data set are intertwined with each other, and it is difficult to distinguish them by distance metric clustering.

7 Conclusion

The known nonselfs reflect the distribution of the real nonself in the feature space, but the conventional NSA and its improved algorithm did not make use of it, resulting in these precious known nonself samples being wasted. In this study, we proposed the V-Detector-KN algorithm. The algorithm first uses the conventional V-Detector algorithm to generate the detector, and then uses the known nonself as the candidate detector center to generate the detector. The V-Detector-KN algorithm has the following three characteristics: First, the detector can be generated in the feature space where the V-Detector algorithm is difficult to generate the detector, such as Haberman's survival data set, 31-D breast cancer wisconsin (diagnostic) data set and pima indians diabetes data set. On the above three data sets, the V-Detector algorithm is difficult to generate detectors due to the overlapping of self and non-self or the high dimensionality of the data set, while the algorithm we proposed can effectively generate detectors. Secondly, the known nonself generated detector can achieve effective coverage of the unknown nonself, thereby increasing the detection rate. This also narrows the gap with machine learning algorithms that use both positive examples (self) and negative examples (nonself) for training. Last but not least is the stability and interpretability of the algorithm. Experimental results on 7 UCI real data sets show that the detection rate of our proposed V-Dtector-KN algorithm ranks first in 3 data sets and second in 4 data sets. Note that, our algorithm is effective on 7 data sets, while the other 6 algorithms are failed on at least one data set. The SVM algorithm failed on the skin segmentation data set, the BIORV-NSA algorithm failed on the breast cancer wisconsin (diagnostic) data set. On the pima indians diabetes data set, V-Detector, improved V-Detector, ADC-NSA, BIORV-NSA and KNN 5 algorithms all failed, and only the V-Detector-KN algorithm we proposed and SVM algorithm are effective. In addition, our algorithm is interpretable, that is, what is covered by the detector is nonself, and what is not covered by the detector is self. We believe that how to make better use of the known nonself to improve the performance of the NSA algorithm is a topic worth studying. In future work, we plan to apply our algorithm to the field of network intrusion detection.

Acknowledgements This work has been supported by the National Key Research and Development Program of China (Grant No.2020YFB1805400), National Natural Science Foundation of China (Grant No. U19A2068, No.U1736212, No.62032002).

References

1. Farmer JD, Packard NH, Perelson AS (1986) The immune system, adaptation, and machine learning[J]. Phys D Nonlinear Phenom 22(1-3):187–204
2. Klarreich E (2002) Inspired by immunity. Nature 415:648–670
3. Balthrop J, Forrest S, Newman MEJ et al (2004) Technological networks and the spread of computer viruses. Science 304:527–529
4. Jin Z-Z, Liao M-H, Xiao G (2013) Survey of negative selection algorithms. J Chin Inst Commun 34.1:159–170
5. Forrest S., Perelson A. S., Allen L et al (1994) Self-nonself discrimination in a computer[C]. In: Proceedings of the 1994 IEEE symposium on security and privacy, IEEE computer society
6. González F. A, Dasgupta D (2003) Anomaly detection using real-valued negative selection [J]. Genetic Programming and Evolvable Machine 4(4):383–403
7. Zhou J, Dasgupta D (2004) Real-valued negative selection algorithm with detectors[C]. In: Proceedings genetic and evolutionary computation conference (GECCO), pp 287–298
8. Ji Z, Dasgupta D (2009) V-detector: An efficient negative selection algorithm with probably adequate detector coverage[J]. Inf Sci 179(10):1390–1406
9. Sun Z, Xu Y, Liang G et al (2018) An intrusion detection model for wireless sensor networks with an improved V-detector algorithm[J]. IEEE Sensors J 18(5):1971–1984
10. Xinping XU, Wang R, Jiang L et al (2018) Research on fault diagnosis of rotor based on improved V-detector algorithm[J]. DEStech Transactions on Engineering and Technology Research
11. Yang T, Chen W, Li T (2017) An antigen space density based real-value negative selection algorithm[J]. Appl Soft Comput 61:860–874

12. Yang C, Jia L, Chen BQ et al (2020) Negative selection algorithm based on antigen density clustering[J]. IEEE Access 8:44967–44975
13. Hofmeyr S, Forrest S. (2000) Architecture for an artificial immune system[J]. Evol Comput 8(4):443–473
14. Bhuvaneswari G, Manikandan G (2019) An intelligent intrusion detection system for secure wireless communication using IPSO and negative selection classifier[J]. Clust Comput 22(5):12429–12441
15. Clotet X, Moyano J, León G (2018) A real-time anomaly-based IDS for cyber-attack detection at the industrial process level of critical infrastructures[J]. Int J Crit Infrastruct Prot 23:11–20
16. Aissa NB, Guerroumi M, Derhab A (2020) NSNAD: negative selection-based network anomaly detection approach with relevant feature subset[J]. Neural Comput Appl 32:3475–3501
17. Li D, Liu S, Zhang H (2016) A boundary-fixed negative selection algorithm with online adaptive learning under small samples for anomaly detection[J]. Eng Appl Artif Intell 50:93–105
18. Saurabh P, Verma B (2016) An efficient proactive artificial immune system based anomaly detection and prevention system[J]. Exp Syst Applic 60:311–320
19. Lu T, Zhang L, Fu Y (2018) A novel Immune-Inspired shellcode detection algorithm based on hyperellipsoid detectors[J]. Secur Commun Netw 2018:1–10
20. Outa R et al (2020) Prognosis and fail detection in a dynamic rotor using artificial immunological system. Engineering Computations
21. Abid A, Khan MT, De Silva CW et al (2018) Layered and real-valued negative selection algorithm for fault detection[J]. IEEE Syst J 12(3):2960–2969
22. Dong LI, Liu S, Zhang H (2017) A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples[J]. Pattern Recogn 64:374–385
23. Barontini A, Perera R, Masciotta MG et al (2019) Deterministically generated negative selection algorithm for damage detection in civil engineering systems[J]. Eng Struct 197:109444
24. Ji Z, Dasgupta D (2007) Revisiting negative selection algorithms[J]. Evol Comput 15(2):223–251
25. Idris I, Selamat A, Omatu S (2014) Hybrid email spam detection model with negative selection algorithm and differential evolution. Appl Artif Intell 28:97–110
26. Cui L, Pi D, Chen C. (2015) BIORV-NSA bidirectional inhibition optimization r-variable negative selection algorithm and its application[J]. Appl Soft Comput 32:544–552
27. Fan Z, Wen C, Tao L et al (2019) An antigen space triangulation coverage based real-value negative selection algorithm[J]. IEEE Access 7:51886–51898
28. Hsu CW, Chang CC, Lin CJ (2016) A practical guide to support vector classification[J]
29. Liu R, Yang B, Zio E et al (2018) Artificial intelligence for fault diagnosis of rotating machinery: A review[J]. Mech Syst Signal Process 108:33–47
30. Dasgupta D. (2006) Advances in artificial immune systems[J]. IEEE Comput Intell Mag 1(4):40–49
31. Evans MJ, Rosenthal JS (2010) Probability and statistics: the science of uncertainty (Second Edition). W. H. Freeman and company. ISBN-10: 1429224622
32. Wen C, Tao L (2017) Parameter analysis of negative selection algorithm[J]. Inf Sci 420:218–234
33. Exponential function, Last accessed: 2020-08-01. [Online]. Available:https://en.wikipedia.org/wiki/Exponential_function
34. Lim T-S (1999) Haberman's survival data set, UCI machine learning repository. Last accessed: 2021-01-10. [Online]. Available:https://archive.ics.uci.edu/ml/datasets/Haberman' s+Survival
35. Christensen R (2015) Thoughts on prediction and cross-validation[J]. Department of Mathematics and Statistics University of New Mexico
36. Chang C-C, Lin C-J (2019) LIBSVM–a library for support vector machines. Version 3.24 released on September 11, 2019. Last accessed:2020-12-01. Available:https://www.csie.ntu.edu.tw/cjlin/libsvm/
37. Li D, Liu S, Gao F et al (2020) Continual learning classification method with new labeled data based on the artificial immune system[J]. Appl Soft Comput 106423:94
38. Tao X, Li Q, Ren C et al (2019) Real-value negative selection over-sampling for imbalanced data set learning[J]. Expert Syst Appl 129:118–134
39. Fisher RA (1988) Iris data set, UCI machine learning repository. Last accessed: 2021-01-10. [Online]. Available:http://archive.ics.uci.edu/ml/datasets/Iris
40. Volker L (2013) Banknote authentication data set, UCI machine learning repository. Last accessed: 2021-01-10. [Online]. Available:https://archive.ics.uci.edu/ml/datasets/banknote+authentication
41. Bhatt R, Dhall A (2012) Skin segmentation data set, UCI machine learning repository. Last accessed: 2021-01-10. [Online]. Available:https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation
42. Turney P (1990) Pima Indians diabetes data set, [Online]. Last Accessed: 2021-01-10. Available: http://networkrepository.com/pima-indians-diabetes.php
43. Siegler RS (1994) Balance scale data set, UCI machine learning repository, [Online]. Last Accessed: 2021-01-10. Available:http://archive.ics.uci.edu/ml/datasets/Balance+Scale
44. Wolberg WH, Street N, Mangasarian OL (1995) Breast cancer wisconsin (Diagnostic) data set, UCI machine learning repository. Last accessed: 2021-01-10. [Online]. Available:https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

Zhiyong Li received the BE degree from the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi, China, in 2003, and MS degree from the School of Information Science and Engineering, Yunnan University, Kunming, Yunnan, China, in 2008. He is currently working toward the PhD degree at the School of Cyber Science and Engineering, Sichuan University, Chengdu, Sichuan, China. His current research interests include artificial immune system, cyberspace security and data mining.

Tao Li received his Ph.D. degree in computer science from the University of Electronic Science and Technology of China, in 1994. He is currently a Professor with the School of Cyber Science and Engineering, Sichuan University, China. He is the Chief Scientist of the National Key Research and Development Program of China. He is also an editorial board member of Immune Computation and several other international academic journals. His main research interests include cyber security, artificial immune system and data mining. He has published nearly 300 papers in IEEE, ACM, Chinese Science, Science Bulletin, Natural Science Progress and other important journals and academic conferences.