

Research Article

A Quick Negative Selection Algorithm for One-Class Classification in Big Data Era

Fangdong Zhu,¹ Wen Chen,^{1,2} Hanli Yang,³ Tao Li,^{1,2} Tao Yang,¹ and Fan Zhang^{1,4}

¹College of Computer Science, Sichuan University, Chengdu 610065, China

²College of Cybersecurity, Sichuan University, Chengdu 610065, China

³Chongqing University of Technology, Chongqing 400054, China

⁴Chengdu University of Information Technology, Chengdu 610225, China

Correspondence should be addressed to Wen Chen; wenchen@scu.edu.cn and Hanli Yang; yhl@cqut.edu.cn

Received 2 February 2017; Accepted 3 May 2017; Published 12 June 2017

Academic Editor: Zonghua Zhang

Copyright © 2017 Fangdong Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Negative selection algorithm (NSA) is an important kind of the one-class classification model, but it is limited in the big data era due to its low efficiency. In this paper, we propose a new NSA based on Voronoi diagrams: VorNSA. The scheme of the detector generation process is changed from the traditional “Random-Discard” model to the “Computing-Designated” model by VorNSA. Furthermore, we present an immune detection process of VorNSA under Map/Reduce framework (VorNSA/MR) to further reduce the time consumption on massive data in the testing stage. Theoretical analyses show that the time complexity of VorNSA decreases from the exponential level to the logarithmic level. Experiments are performed to compare the proposed technique with other NSAs and one-class classifiers. The results show that the time cost of the VorNSA is averagely decreased by 87.5% compared with traditional NSAs in UCI skin dataset.

1. Introduction

NSA was proposed by Forrest et al. in 1994 [1], which generates immune detectors based on the “Random-Discard” model. Initially, massive immature detectors are randomly generated, and then the ones covering the self-areas are discarded. González et al. presented the real-valued negative selection algorithm (RNSA) in 2003 [2], in which the detectors and antigens are studied in the real-value space. Ji and Dasgupta proposed V-Detector algorithm [3, 4]. It turns the fixed-length detectors in RNSA into the variable-sized detectors to enlarge the detection areas. In 2015, Cui et al. developed BIORV-NSA [5]. In their work, the self-radius can be variable and the detectors, which are recognized by other mature detectors, are replaced by new ones to eliminate the “detection holds.”

In big data era, the low efficiency of NSA becomes an important challenge, which largely limits its applications. In this paper, we design a new NSA based on Voronoi diagrams, named VorNSA. In the VorNSA, a restrained Voronoi diagram is constructed based on the whole training set in the first

step. Then, two types of detectors are generated in the specific location of the Voronoi diagram separately. In order to accelerate the test stage of NSA, in particular for large scale dataset, a new testing strategy VorNSA/MR (VorNSA with Map-Reduce) is proposed. Unlike the testing stage of classic NSAs, data are divided into small groups and calculated to generate the labels separately in Map stage. Then the final labels can be obtained after merging and sorting in the Reduce stage.

The contributions of this work can be summarized as follows. (1) Based on Voronoi diagrams, the optimal position of detectors is calculated directly rather than in a stochastic way. Therefore, the time consumption wasted on excessive invalid detectors is avoided. (2) In the Map/Reduce framework, data are partitioned into several small parts by VorNSA/MR and can be processed in parallel to enhance the self/non-self-discrimination efficiency.

The rest of the paper is organized as follows. In Section 2, we describe the definitions of VorNSA. The original contribution of the paper is presented in Section 3. Experimental results on synthetic datasets and real-world datasets are

shown and discussed in Section 4. Conclusions appear in Section 5.

2. Basic Definition of VorNSA

VorNSA is designed based on Voronoi, which is derived from computation geometry to search the nearest neighbors, and it has been widely utilized in the fields of life sciences [6], material sciences [7], and mobile navigation [8]. The basic definitions are listed as follows.

Definition 1 (site). Site is a set of n distinct points in the feature space. In VorNSA, all the training samples are defined as site points: $S = \{S_1, S_2, \dots, S_n\}$.

Definition 2 (Voronoi diagram). $\text{Vor}(S)$ divides the feature space into n unoverlapped cells based on the given site set S , and each cell $\nu(S_i)$ only contains one site S_i in S , such that any point q in $\nu(S_i)$ satisfies $\text{dist}(q, S_i) < \text{dist}(q, S_j) \forall S_j \in S, j \neq i$, and $\text{dist}()$ can be any distance metrics.

Definition 3 (cell). All the cells construct a mathematic partition of the feature space, and the cell corresponding to site S_i is denoted by $\nu(S_i)$.

Definition 4 (largest empty circle). The largest circle with center p , which does not contain any site in S , is denoted by $C_S(p)$.

Theorem 5. A point p is a vertex of $\text{Vor}(S)$ iff $C_S(p)$ contains at least three sites on its boundary [9].

Definition 6 (I-detector). $\langle c, r \rangle$, where c is the detector position in the feature space, and r is the detector radius, satisfies that c corresponds to one vertex of the Voronoi diagram.

Theorem 7. Given p is the center of an I-detector, there are at least three sites located on the boundary of $C_S(p)$, and these sites are the nearest neighbors of each other.

Proof. According to Definitions 2 and 6, it can be inferred that the center of the I-detector p is an intersection of three or more cells. Suppose that p is intersected by three cells $\nu(S_i), \nu(S_j), \nu(S_k)$, while the sites of these cells are S_i, S_j, S_k . According to Definition 4 and Theorem 5, there is a largest empty circle $C_S(p)$ that does not contain any site of S , and S_i, S_j, S_k are located on its boundary. So S_i, S_j , and S_k are the nearest sites of p among the site sets S . \square

Theorem 8. The bisector between sites S_i and S_j defines an edge of $\text{Vor}(S)$ iff there is a point q on the bisector such that $C_S(q)$ contains both S_i and S_j on its boundary with no other site [9].

Definition 9 (II-detector). $\langle c, r \rangle$, where c is the detector position in the feature space, and r is the detector radius, satisfies that c corresponds to the junction of the edges of $\text{Vor}(S)$ and the unit hypercube $[0, 1]^d$.

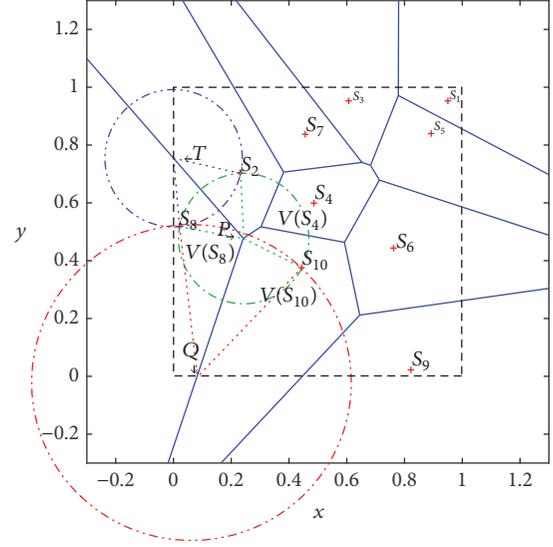


FIGURE 1: The red cross is the site. The blue line is the Voronoi diagram. The circle is the largest empty circle.

Theorem 10. Given q is the center of II-detector, there are two sites located on the boundary of $C_S(q)$, and these sites are the nearest neighbors of each other.

Proof. According to Definitions 2 and 9, it can be inferred that the center of II-detector q is an intersection of two cells. Suppose that q is intersected by two cells $\nu(S_i), \nu(S_j)$, while the sites of these cells are S_i, S_j . According to Definition 4 and Theorem 8, there is a largest empty circle $C_S(q)$ that does not contain any site of S , and S_i, S_j are located on its boundary. So S_i and S_j are the nearest sites of q among the site sets S . \square

As an example in Figure 1, there are 10 sites S_1-S_{10} in set S , and the space is divided into 10 cells $\nu(S_1)-\nu(S_{10})$ by the Voronoi diagram $\text{Vor}(S)$. The green circle is $C_S(P)$, and three sites (S_2, S_8, S_{10}) are located on its boundary. The red circle is $C_S(Q)$, and two sites (S_8, S_{10}) are located on the boundary. The purple circle is $C_S(T)$, and two sites (S_2, S_8) are located on the boundary. P is the center of I-detector, while Q and T are the centers of II-detector.

3. The Details of VorNSA

3.1. The Detector Generation Process of VorNSA

3.1.1. Space Partition Stage. First of all, all the training data are normalized to $[0, 1]^d$ feature space, where d is the data dimension. The normalized training set is denoted by S . Secondly, a bounded Voronoi diagram $\text{Vor}(S)$ is constructed based on S , to divide the unit feature space $[0, 1]^d$ into n cells, where $n = |S|$. Finally, the set $VS = \{\langle \text{Vet}(S_i), S_i \rangle \mid i = 1, \dots, n\}$, where $\text{Vet}(S_i), S_i$ are the vertex and site in a cell $\nu(S_i)$, can be constructed.

3.1.2. I-Detector Generation Stage. According to Definition 6 and Theorem 7, the center of I-detector p is designated by the

```

Input: Training set S, Self radius  $R_S$ , Minimum detector radius  $\delta$ 
Output: Detector set D
(1) normalize S into  $[0, 1]^d$ 
(2) construct voronoi diagram Vor(S) by sites S
(3) get all cells  $\nu(S_i)$  in Vor(S)
(4) construct  $VS = \{\langle \text{Vet}(S_i), S_i \rangle \mid i = 1, \dots, n\}$  by  $\nu(S_i)$ 
(5) foreach  $\langle \text{Vet}(S_i), S_i \rangle$  in VS
(6)   if  $\text{Vet}(S_i)$  has three or more same values in VS
(7)   then  $VS1 = VS1 \cup \langle \text{Vet}(S_i), S_i \rangle$ 
(8) foreach  $\langle \text{Vet}(S_j), S_j \rangle$  in VS1
(9)   compute the detector radius  $R_p$  using Eq. (1)
(10)  if  $R_p > \delta$  then  $D_I = D_I \cup \langle \text{Vet}(S_j), R_p \rangle$ 
(11) foreach  $\langle \text{Vet}(S_i), S_i \rangle$  in VS
(12)  if  $\text{Vet}(S_i)$  has two same values in VS
(13)  then  $VS2 = VS2 \cup \langle \text{Vet}(S_i), S_i \rangle$ 
(14) foreach  $\langle \text{Vet}(S_k), S_k \rangle$  in VS2
(15)  compute the detector radius  $R_Q$  using Eq. (2)
(16)  if  $R_Q > \delta$  then  $D_{II} = D_{II} \cup \langle Q, R_{Qi} \rangle$ 
(17) return  $D = D_I \cup D_{II}$ 

```

ALGORITHM 1: VorNSA (S, R_S, δ).

intersection of three or more cells, and the sites located in the cells are the nearest neighbors of each other. So a new set $VS_1 = \{\langle \text{Vet}(S_j), S_j \rangle \mid j = 1 \dots\}$, where $\text{Vet}(S_j)$ is the position of I-detector and S_j is the nearest sites, can be obtained by $\text{Vet}(S_j) = \{x \mid x = \text{Vet}(S_p) \cap \text{Vet}(S_q) \cap \text{Vet}(S_t), p \neq q \neq t\}$, where $\text{Vet}(S_p)$, $\text{Vet}(S_q)$, and $\text{Vet}(S_t)$ are the vertex sets of cell. Then, generating a mature detector is just through self-tolerating with S_j . According to the principle of self-tolerance, the radius of I-detector can be calculated with

$$R_p = \text{dist}(\text{Vet}(S_j), S_j) - R_S, \quad (1)$$

where R_p is the radius of I-detector, $\text{Vet}(S_j)$ is the center of I-detector, S_j is the nearest sites, and R_S is the radius of self-antigens.

Furthermore, a threshold δ of detector radius is introduced in case of overfitting: If the detector radius R_p is less than δ , the detector will be discarded. Otherwise, it will mature.

3.1.3. II-Detector Generation Stage. The main difference between the I-detector and the II-detector is the location of detector centers. According to Definition 9 and Theorem 10, the position of II-detector q is located on the junction of two cells and the unit hypercube. The sites in the two cells are the nearest neighbors of each other. So a new set $VS_2 = \{\langle \text{Vet}(S_k), S_k \rangle \mid k = 1 \dots\}$, where $\text{Vet}(S_k)$ is the position of II-detector and S_k is the nearest sites, can be obtained by $\text{Vet}(S_k) = \{x \mid x = \text{Vet}(S_p) \cap \text{Vet}(S_q), p \neq q\}$, where $\text{Vet}(S_p)$ and $\text{Vet}(S_q)$ are the vertex sets of cell. Similarly, the radius of II-detector can be computed by (2), and a threshold δ of detector radius is introduced in case of overfitting.

$$R_Q = \text{dist}(\text{Vet}(S_k), S_k) - R_S, \quad (2)$$

where R_Q is the radius of II-detector, $\text{Vet}(S_k)$ is the position of II-detector and S_k is the nearest sites, and R_S is the radius of self-antigens.

Details of the VorNSA can be found in Algorithm 1.

3.2. The Immune Detection Process of VorNSA under Map/Reduce Framework. In the testing stage of traditional NSAs, each piece of data has to be compared with all the detectors to label its classification. This strategy is too time-consuming to be applied in big data era due to its low efficiency. In order to enhance the efficiency in testing stage, an immune detection process of VorNSA under Map/Reduce framework (VorNSA/MR) is proposed. Map/Reduce is a parallel computation framework, which splits the sample set into a group of small datasets and handles them on many cluster nodes simultaneously.

Details of VorNSA/MR (Figure 2) are mainly divided into two parts: Map stage and Reduce stage. First of all, the testing datasets are split into n parts by VorNSA/MR. In the Map stage, each cluster node selects a part of split data to compute the distance with matured detectors. If any distance is less than the detection radius, the testing sample is labeled with the non-self-antigens; otherwise it is labeled with the self-antigens. Then cluster nodes put results to the intermediate value. The Reducer receives the intermediate values, sorts them, and merges them into the final results.

The implements of Map and Reduce stage can be found in Algorithms 2 and 3.

3.3. Theoretical Analysis

Theorem 11. *The time complexity of VorNSA is $(N_S \log N_S + N_S^{\lceil d/2 \rceil} + |D|)$, where N_S is the size of training dataset, d is the dimension of training dataset, and $|D|$ is the size of detectors.*

```

Input: Detector set  $D$ , Split data  $T$ 
Output: Intermediate Value IV
(1) foreach  $T_i$  in  $T$ 
(2)   foreach  $D_k$  in  $D$ 
(3)     Compute the Euclidean distance  $\text{dist}(T_i, D_k)$  between  $T_i$  and  $D_k$ 
(4)     if  $D_{k,r} < \text{dist}(T_i, D_k)$ 
(5)        $T_i$  is Noself Antigen,  $T_i.\text{Label} = 0$ 
(6)       go to line (2)
(7)      $T_i$  is Self Antigen,  $T_i.\text{Label} = 1$ 
(8) IV.Value =  $\langle T.\text{no}, T.\text{Label} \rangle$ 
(9) return IV

```

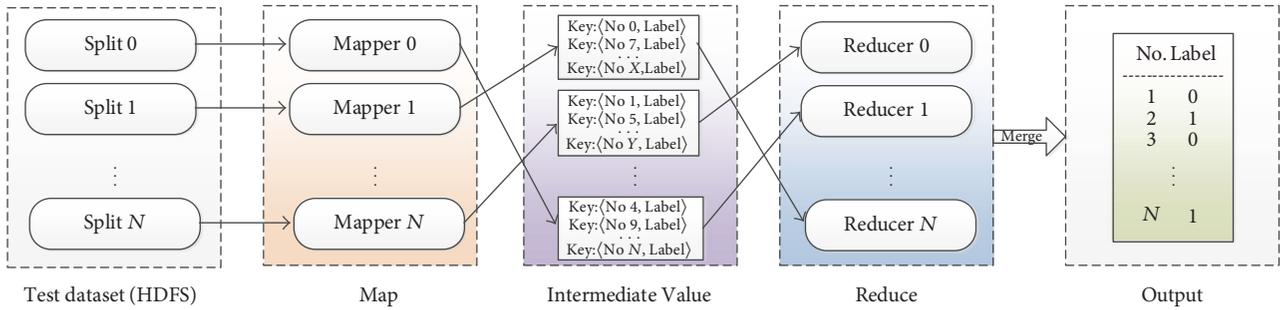
ALGORITHM 2: Mapper (D, T).

FIGURE 2: The details of VorNSA/MR.

```

Input: Intermediate Value IV
Output: Final Value FV
(1) While IV.next  $\neq$  END
(2)   add IV.Value to FV.Value
(3) Sort FV.Value by no
(4) return FV

```

ALGORITHM 3: Reducer (IV).

TABLE 1: The complexity of NSAs.

Algorithm	Time complexity
NNSA [1]	$O\left(\frac{\ln P_f * N_S}{P_m (1 - P_m)^{N_S}}\right)$
RNSA [2]	$O\left(\frac{ D * N_S}{(1 - P_m)^{N_S}}\right)$
V-Detector [4]	$O\left(\frac{ D * N_S}{(1 - P_m)^{N_S}}\right)$
VorNSA	$O(N_S \log N_S + N_S^{\lceil d/2 \rceil} + D)$

Proof. Since VorNSA is divided into three stages, we could analyze the time complexity separately.

The main work in space partition stage is to build a Voronoi diagram, so we borrow the analysis from Voronoi diagrams to estimate the time complexity. The literatures [9–12] prove that a Voronoi diagram with n sites can be computed in $O(n \log n + n^{\lceil d/2 \rceil})$ optimal time under d -dimension space. Therefore, the time complexity can be denoted by $O(N_S \log N_S + N_S^{\lceil d/2 \rceil})$, where N_S is the size of training set, and d is the dimension of training set.

In the second and third stage, the main work is to compute the distance between detectors and sites. Though several detectors are discarded by the threshold δ , the quantity is very small compared with the whole size, so we use the size of detectors $|D|$ instead. According to (1) and (2), we can infer that the time complexity is $O(|D|)$ in the two stages.

Combining the abovementioned, the time complexity of VorNSA is $O(N_S \log N_S + N_S^{\lceil d/2 \rceil} + |D|)$. \square

The time complexity of traditional NSAs is shown in Table 1, where P_m is the match probability between detectors and antigens, P_f is the failure rate, N_S is the size of self-set, $|D|$ is the size of detectors, and d is the data dimension. As shown in Table 1, the time complexity of VorNSA is in logarithmic level with N_S , which is much less than the traditional exponential level compared with NNSA [1], RNSA [2], and V-Detector [4].

4. Experiments and Discussion

In the experiments, we use two evaluation criteria of performance: DR (Detection Rate) and FAR (False Alarm Rate) which is reported in varied literature [2, 3, 13], and they are defined as

$$\begin{aligned}
 \text{DR} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\
 \text{FAR} &= \frac{\text{FP}}{\text{FP} + \text{TN}},
 \end{aligned} \tag{3}$$

TABLE 2: The detail of 4 SDS.

Dataset	Records number	Self-antigens	Non-self-antigens
Cross	10,000	5,531	4,469
Ring	10,000	3,710	6,290
Pentagram	10,000	2,850	7,150
Triangle	10,000	1,476	8,524

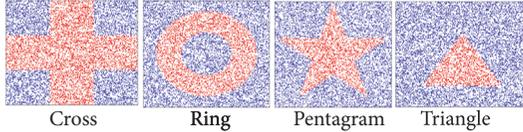


FIGURE 3: The distribution of 4 SDS.

where TP and FN are the counts of true positive and false negative of non-self-antigens, respectively, and TN and FP represent the number of true negative and false positive of self-antigens, respectively.

4.1. Experiments on Synthetic Dataset (SDS). In order to determine the performance of VorNSA among different datasets, 4 SDS proposed by the intelligence security laboratory of Memphis University are introduced in this section. The records of original datasets [3] are 1000, respectively. We expand the number of pieces of data to 10,000 to simulate the environment of big data better. The distributions of datasets are depicted as Figure 3 in which self-antigens are represented by red dots and non-self-antigens are shown by blue points. The details of datasets are listed in Table 2. Additionally, experiment parameters are set as follows: the self-radius is 0.04, self-antigens are randomly obtained from 50 to 1000, and the minimum radius of detectors is 0.005. Each experiment is repeated 25 times independently.

As Figure 4 shows, the trends of experiment results on 4 SDS are approximately the same. It indicates that VorNSA could achieve a high degree of applicability on different datasets. In Figure 4(a), it can be observed that the DR decreases from 95% to 80% with the increment of self-antigens. Besides, in Figure 4(b), the FAR drops from 60% to zeros. The reasons of this phenomenon can be explained as follows: when less self-antigens are trained, some self-antigens cannot be covered by the scope of self. So these self-antigens are identified as non-self-antigens in VorNSA. Due to its strong ability in detecting, the DR and FAR are both high. With the increase of the training numbers, all self-antigens will be covered. Furthermore, the non-self-antigens are covered and identified as self-antigens, in particular those located in the edge of self-set. Therefore, the DR decreases slightly while FAR sharply drops to zeros.

Figure 4(c) shows the quantity of detectors generated by VorNSA is not increasing remarkably with the growth of train set but maintains a relatively stable range. It is implied that VorNSA can effectively control the expansion of detectors. According to Definition 2, with the increment of training samples, the space will be partitioned into smaller cells.

We introduce the minimum detector radius δ . Thus, the inefficient tiny detectors are discarded.

In Figure 4(d), it can be noted that the time consumption of VorNSA on different datasets is similar, and time cost rises slowly even with enormous self-antigens. It suggests that the performance of VorNSA is less affected by the distribution of dataset, because the optimal position of detectors is calculated directly rather than in a stochastic way.

To sum up, we can see that VorNSA can generate fewer but more effective detectors. Besides, the less self-antigens are trained, the higher FAR will be. With the number of self-antigens increasing, the FAR is decreased significantly. Increasing the training set will lead to a rise of the time consumption, and the DR will be slightly decreased. Hence, a smaller self-set will be a smart choice in VorNSA.

4.2. Experiments on Skin Segmentation Dataset. In this section, VorNSA is tested by a group of comparison experiments. The compared algorithms include the classic NSAs (RNSA, V-Detector), a newly proposed NSA (BIORV-NSA) in 2015. To study the different methods, we introduce a classic statistics algorithm for one-class classification: OC-SVM [14], which is implemented by LibSVM [15]. All algorithms run in a computer deployed with Intel Pentium E6600@3.06 G, while the implement of VorNSA refers to an open source toolbox of computational geometry, called MPT 3.0 [16].

The Skin Segmentation dataset is a UCI dataset. It is collected by randomly sampling B, G, and R values of skin texture, which derives from FERET database and PAL database. Total sample size is 245,057 in which 50,859 records are the skin samples and 194,198 records are non-skin ones.

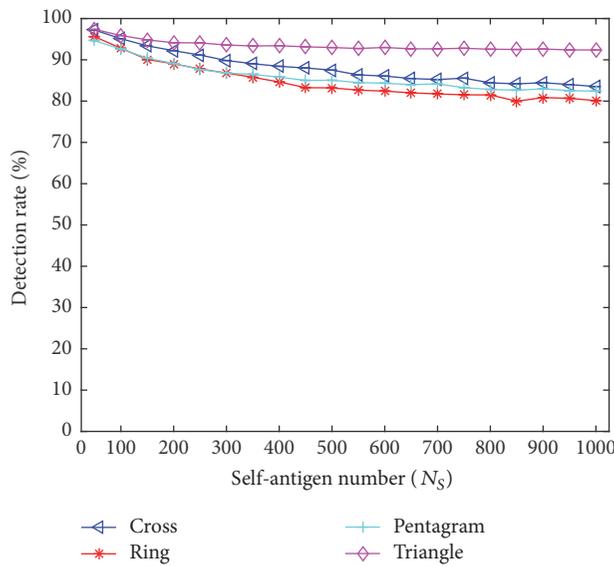
In this experiment, 50 skin samples are randomly obtained as self-antigens. Meanwhile, to verify the performances of VorNSA and VorNSA/MR in large scale dataset, we use all 245,057 records in the datasets. The experiments are preformed 20 times independently, and the evaluation criteria include DR, FAR, detector number (DN), data training time (DT), and data testing time (DTT). The parameters of simulation are set as follows: the OC-SVM uses the RBF kernel functions, and nu is 0.5 and gamma is 0.33. The self-radius of RNSA, V-Detector, and VorNSA are set as the same value (0.1). The maximum number of detectors is 3000 in RNSA, and detector radius is 0.1. The estimated coverage and the maximum self-coverage are 99%. The maximum number of detectors is 1000 in BIORV-NSA, and the self-set edge inhibition parameter is 0.8 and the detector self-inhibition parameter is 1.2. The minimum radius of detectors is 0.005 in VorNSA and VorNSA/MR. The results of experiments are shown in Table 3.

From Table 3, it can be seen that the FAR of OC-SVM is 51.2%, reaching an unacceptable level. As OC-SVM implemented in a different platform, the time consumption is not counted in this paper. The DR of VorNSA (99.2%) is closed to the BIORV-NSA (99.42%), and better than the classic NSAs. Besides, the FAR of VorNSA (1.48%) is lower than BIORV-NSA (3.29%). It indicates that the detectors generated by VorNSA are more applicable than BIORV-NSA and more effective than classic NSAs.

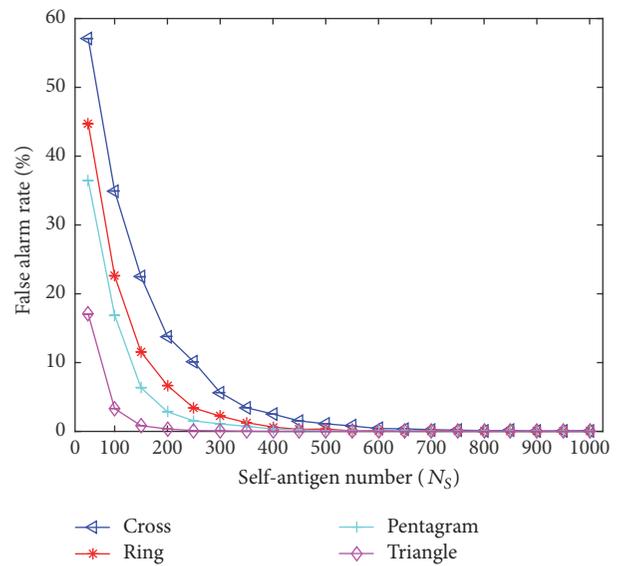
TABLE 3: Results in skin segmentation.

Algorithm	DR (%)		FAR (%)		DN		DT (s)		DTT (s)	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
OC-SVM	99.09	0.7	51.20	6.67	—	—	—	—	—	—
RNSA	98.42	0.63	0.66	1.48	3000.00	0	8.68	0.13	7501.59	400.49
V-Detector	99.05	0.27	1.31	1.22	469.85	174.66	32.12	23.86	948.55	325.50
BIORV-NSA	99.42	0.34	3.29	2.72	1000.00	0	20.00	0.11	1919.83	59.46
VorNSA	99.20	0.16	1.48	1.49	172.25	11.06	1.91	0.77	671.15	89.36
VorNSA/MR*	99.43	0.24	1.56	1.37	176.90	11.96	1.79	0.07	426.70	31.97

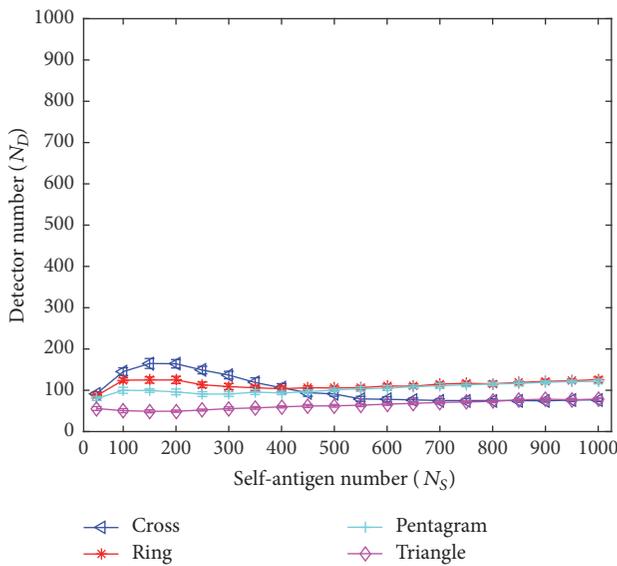
*The VorNSA/MR is deployed at 2 nodes: one is Intel Pentium E6600@3.06 G (2 Core); the other is Inter Core i5-2450M@2.5 G (2 Core).



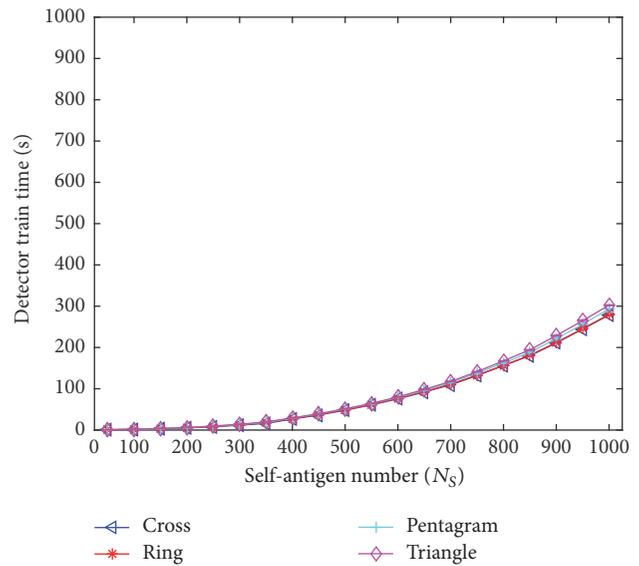
(a) Detection rate



(b) False alarm rate



(c) Detector number



(d) Detector train time

FIGURE 4: Results with different training samples.

Moreover, the DN, DT, and DTT of VorNSA are significantly lower than other NSAs, especially when it integrates the Map-Reduce Testing Framework. For example, the average number of detectors generated by VorNSA is 172.25, lower 63.3% by V-Detector and 82.8% by BIORV-NSA. The average training time of VorNSA is 1.91, lower 78% by RNSA, 94.1% by V-Detector, and 90.5% by BIORV-NSA. So the efficiency of VorNSA is averagely decreased by 87.5% compared with traditional NSAs. The testing time of VorNSA/MR is 426.7, lower 36.4% by VorNSA, 55% by V-Detector, 77.8% by BIORV-NSA, and 94.3% by RNSA.

The main reasons of above results can be explained as follows. In traditional NSAs, a large number of immature detectors are randomly generated without any optimal way and must self-tolerate with all self-antigens to decide whether they are matured or not. As a result, much time has been wasted. The scheme of detector generation of VorNSA is quite different with other NSAs. The optimal position of detectors is directly calculated. Thus, the time consumption on discarding many randomly generated but inappropriate detectors is avoided.

5. Conclusions

In this paper, we propose a new one-class classification algorithm based on Voronoi diagrams (VorNSA) and an immune detection process of VorNSA under Map/Reduce framework (VorNSA/MR) to cope with the challenge of big data. VorNSA alters the generative mechanism of detector from the “Random-Discard” model to the “Computing-Designated” model. VorNSA/MR can divide the sample set into several small parts and can be processed in parallel. Theoretical analyses show that the time complexity of VorNSA decreases from the exponential level to the logarithmic level. Experiments results show that the time consumption of VorNSA is significantly declined.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant nos. 2016YFB0800605 and 2016YFB0800604) and Natural Science Foundation of China (Grant nos. 61402308 and 61572334).

References

- [1] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, “Self-nonsel self discrimination in a computer,” in *Proceedings of the IEEE Symposium on Research in Security and Privacy, (SP '94)*, pp. 202–212, IEEE Computer Society, Oakland, May 1994.
- [2] F. González, D. Dasgupta, and L. F. Niño, “A randomized real-valued negative selection algorithm,” in *In Proceedings of the 2nd International Conference on Artificial Immune Systems*, vol. 2787, pp. 261–272, 2003.
- [3] Z. Ji and D. Dasgupta, “Real-valued negative selection algorithm with variable-sized detectors,” in *Genetic and Evolutionary Computation Conference*, vol. 3102 of *Lecture Notes in Computer Science*, pp. 287–298, Springer, Berlin, Heidelberg, 2004.
- [4] Z. Ji and D. Dasgupta, “V-detector: an efficient negative selection algorithm with ‘probably adequate’ detector coverage,” *Information Sciences*, vol. 179, no. 10, pp. 1390–1406, 2009.
- [5] L. Cui, D. Pi, and C. Chen, “BIORV-NSA: Bidirectional inhibition optimization r-variable negative selection algorithm and its application,” *Applied Soft Computing Journal*, vol. 32, pp. 544–552, 2015.
- [6] D. Sanchez-Gutierrez, M. Tozluoglu, J. D. Barry, A. Pascual, Y. Mao, and L. M. Escudero, “Fundamental physical cellular constraints drive self-organization of tissues,” *EMBO Journal*, vol. 35, no. 1, pp. 77–88, 2016.
- [7] H. W. Sheng, W. Luo, F. Alamgir, J. Bai, and E. Ma, “Atomic packing and short-to-medium-range order in metallic glasses,” *Nature*, vol. 439, pp. 419–425, 2006.
- [8] G. Zhao, K. Xuan, W. Rahayu et al., “Voronoi-based continuous nearest neighbor search in mobile navigation,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2247–2257, 2011.
- [9] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, Springer, 2008, <https://www.amazon.com/Computational-Geometry-Applications-Mark-Berg/dp/3540779736>.
- [10] B. Chazelle, “An optimal convex hull algorithm and new results on cuttings,” in *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*, pp. 29–38, October 1991.
- [11] K. L. Clarkson and P. W. Shor, “Applications of random sampling in computational geometry, II,” *Discrete & Computational Geometry*, vol. 4, no. 1, pp. 387–421, 1989.
- [12] R. Seidel, “Small-dimensional linear programming and convex hulls made easy,” *Discrete & Computational Geometry*, vol. 6, no. 1, pp. 423–434, 1991.
- [13] W. Chen, T. Li, X. Liu, and B. Zhang, “A negative selection algorithm based on hierarchical clustering of self set,” *Science China Information Sciences*, vol. 56, no. 8, pp. 1–13, 2013.
- [14] Y. Chen, X. S. Zhou, and T. S. Huang, “One-class SVM for learning in image retrieval,” in *Proceedings of IEEE International Conference on Image Processing (ICIP) 2001*, pp. 34–37, grc, October 2001.
- [15] C.-C. Chang and C.-J. Lin, “LIBSVM: a Library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, article 27, 2011.
- [16] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, “Multi-parametric toolbox 3.0,” in *Proceedings of the 12th European Control Conference, (ECC '13)*, pp. 502–510, Zurich, Switzerland, July 2013.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

