International Journal of Innovative Computing, Information and Control
Volume **8**, Number **5(A)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 2953–2964

A POLYMORPHIC CONVEX HULL SCHEME FOR NEGATIVE SELECTION ALGORITHMS

Mahshid Majd, Ali Hamzeh and Sattar Hashemi

Department of Computer Science and Engineering and Information Technology
School of Electrical and Computer Engineering
Shiraz University
Mollasadra Ave., Shiraz, Iran
{ majd; ali }@cse.shirazu.ac.ir; s_hashemi@shirazu.ac.ir

Received November 2010; revised March 2011

Abstract. *Artificial Immune System (AIS) is known as the collection of all researches including computational models and knowledge discovery algorithms inspired by the natural immune system. One of the most widely used techniques in the field of artificial immune system is Negative Selection Algorithm (NSA). The principle of this algorithm is finding a set of detectors which can discriminate the self and non-self areas. Each detector in NSA defines a subspace of problem space where no self data is located. As the obtained set of detectors paves the way of discriminating the self and non-self data, NSA finds its main application in anomaly detection problems. What significantly affects the detection performance of NSA is choosing a proper shape/representation for detectors to model the regularities of non-self area precisely in one hand, and on the other hand, number of generated detectors should be tuned in a delicate manner. Using a general and polymorphic representation scheme for covering non-self area can elegantly alleviate the above mentioned issues. This paper presents a novel representation method based on convex hulls, named CH-NSA, which is general enough to support polymorphic shapes with variable properties. Convex hull is a general form representation which supports not only regular symmetric shapes such as rectangles, spheres and ellipse but also irregular asymmetric shapes. The experimental results show that applying this new representation to well-known benchmark problems in the literature enhances the accuracy of NSA by significantly decreasing the number of needed detectors compared with the other common representation methods.*
Keywords: Artificial immune system, Negative selection algorithm, Convex hull representation

1. Introduction.

Artificial Immune Systems (AISs) are biological inspired techniques, based on theories and knowledge of vertebrate immune system. They represent an identification mechanism capable of performing several tasks like pattern recognition, learning, memory acquisition, distributed detection and optimization [1]. The common techniques inspired by the immune system are Clonal Selection Algorithm (CSA), Negative Selection Algorithms (NSA) and Immune Network Algorithms. This paper focuses on NSA which has potential applications in various research areas such as anomaly detection. Anomaly detection refers to the problem of discriminating the incoming data as normal or anomaly data, by giving a collection of normal data as the training samples. Computer security is a well known application of anomaly detection in which case only positive samples are available and we aim at detecting negative ones. NSA has been widely used for such problems because of its ability to model the problem space using just positive samples.

In NSA, a group of detectors are generated by some random processes to cover those parts of problem space where negative samples might appear. These detectors are then used to categorize the new data into positive or negative. A common representation of NSA detectors was binary representation due to its simplicity [2, 3]. Almost all representations can theoretically be translated into binary form, but it imposes some limitations such as binary string size, accessible distance measures and comprehensibility. Hence, real valued representation has received great attention in the recent decades.

This paper intends to introduce a new representation of detector definition and matching mechanism for NSA which is designed for real valued problem spaces. There are some previous efforts to find the proper representation for detectors in real world problems, such as hyper rectangle [4], hyper sphere [5] and multi-shape [6]. Here, the proposed representation is based on convex hull shape which delineates a convex region in the problem space. Since convex hull provides asymmetric shape, it can partition the problem space into complex regions and model complicated boundaries. Indeed, convex hull is general form of other geometric shapes such as hyper rectangle and hyper sphere. To the best of our knowledge, this paper for the first time proposes the use of convex hull for detector representation.

The performance of the proposed method, called CH-NSA, is comparable to those methods which use other representations. Moreover, experimental results show that in term of true positive rate, the proposed convex based representation could lead to a better performance with less number of detectors in well known benchmark [6, 7, 8, 9].

The remaining of this paper is organized as follows. Section 2 briefly reviews some related negative selection algorithms. Section 3 introduces convex hull representation for NSA and also describes the implementation details of CH-NSA. Experiments are given in Section 4. Next, in Section 5, the applicability of the proposed method is discussed and the influence of the main control parameters is illustrated. The last section includes summary and conclusion.

2. Related Work.

NSA was firstly proposed by Forrest [2] and soon later D'haeseler et al. [3] presented an efficient implementation of it, named *greedy algorithm*. The greedy algorithm used a binary representation of data space and r-continues matching rule which is designed especially for string and binary data. Two strings are matched using r-continues matching rule if r-continues bits of them are equal in corresponding positions [11]. In [10], it is mentioned that binary representation has three advantages. First, it is easy to analyze. Second, it is a proper representation for textual and categorical information and third, any data can be changed into binary format. Despite these advantages, this representation made the algorithm hard to be used in real world problems with real valued data. In addition, a large number of detectors were generated to achieve a good level of detection.

Soon later, the real valued representation was used in [4] to characterize the self/non-self space. Each detector in the algorithm was represented by a lower and upper bound for all dimensions of the data space which means any data instance lay within this bend is matched with the detector. The resulted detectors were in fact hyper rectangles and are called "detector rules". A genetic algorithm was used as the detector generator to evolve good detectors which goodness was defined as having less coverage of self space and less overlap with other detectors.

Next, Gonzales et al. suggested a real valued representation for NSA and called the algorithm Real Valued Negative Selection or RNS [5]. In general, a real valued detector covers a region of non self space and any data situated in this region is recognized by the detector and classified as non self data. They used a detector generation algorithm based on a heuristic which tries to distribute detectors in the non-self space to maximize the coverage. They used an iterative process to change the position of detectors in a way that detectors have a minimum overlap with each other and cover no self data. Fuzzy membership function was used as matching rule and therefore their detectors were hyper spheres with a non self recognition fuzzy radius or in general, fuzzy spheres.

As an improvement to RNS, in [12] instead of the heuristic approach mentioned above, Monte Carlo simulation technique was used to estimate the volume of the non self space and decide the number of needed detectors. In addition, simulated annealing was applied to optimizing the distribution of detectors in non-self space. The algorithm was called Randomized Real Valued Negative Selection or RRNS. The new algorithm was claimed to be better than RNS in providing a theoretical basis for analysis but it did not give a better performance.

A new approach to detector representation named V*-detector* was introduced in [8]. It emphasized on having individual specific properties for detectors and 'V' in V*-detector* came from the word 'variable'. Euclidean based matching rule was used and matching threshold was variable in detector set. This means detectors were hyper spheres with individual specific radius. This variable radius is dynamically defined equal to the distance between center of hyper sphere and nearest self data given in the training dataset. The flexibility of the size property provided a notable improvement in detector set coverage, compared with the most of the earlier NSA reported. Besides, significantly less number of detectors was generated.

Already, the idea of having detectors with variable properties has been examined in some other ways. For example in [13], detectors with different sizes were used and in [14], a variable sized hyper sphere was used based on $Minkowsky$ distance. In training phase, the detectors which fell in self space were moved away from training data. They also applied detector cloning and random exploration.

Next, in [6] a frame work for multi-shape detector representation was presented. Tested shapes were hyper sphere, hyper rectangle and hyper ellipses and could be extended to other shapes. An evolutionary approach was applied to generate hyper shapes and Monte Carlo estimation technique was used to ensure enough coverage by detectors. Using variable sized and variable shape detectors resulted in a better coverage of non-self space with small number of detectors.

One of the latest efforts on reducing the number of detectors was an improvement to V*-detector* by [7]. The algorithm combined a statistical estimation of coverage with the detector generation mechanism to control the number of generated detectors. The method would finish detector generation when estimated coverage is close to target value, having predefined statistical confidence in estimating coverage. The used statistical inference was Hypothesis testing and the null hypothesis was defined as "The coverage of the non-self region by all the existing detectors is below percentage $p_{\min}$".

Finally, there are some other works [15, 16, 17] where it is tried to overcome the inefficiency of using random generation mechanism. They used some other optimization techniques like genetic algorithm to help the detector generation component in producing effective detectors.

3. **Proposed Approach.** Why using convex hulls? As it is mentioned in previous section, there had been some valuable efforts to achieve a proper representation for the detectors in some classic benchmark problems. Yet, convex hull whose particular properties make it a fine choice for complicated problem spaces is not examined. Due to the fact that convex hulls offer asymmetric shapes, bring the idea that it can successfully represent complex boundaries. On the other hand, all the other geometric shapes can be approximated through convex hulls with sufficient number of points. This property

shows the generality of convex hull which means that at least it can solve those problems answered by the other geometric shapes.

Convex hull representation is not a complete stranger in knowledge representation field. In fact, it has been used previously in learning classifier systems literature [18] where representation showed to be a challenging issue. The reported experimental results of this representation showed a significant improvement at a 99.99% confidence level. It is also mentioned that "convex conditions find solutions that are on average at least as complex as those produced by hyper rectangles (interval condition)". This result and the studied characteristics of convex hulls motivated us to test this representation in NSA as well.

A convex hull can be defined for a set of points which are in any dimensions including infinite dimensions. A convex polyhedron in n dimension space is composed of some faces which is the intersection of the convex polygon and a supporting hyperplane. A hyperplane is said to support a convex polyhedron CH iff some $x \epsilon CH$ are on the hyperplane and all the other $x \epsilon CH$ lies on the same side of the hyperplane. A face of CH is called a k-$face$ if it is in k dimension. For CH in n dimension, the $(n-1)$-$faces$ are called $facets$, the $(n-2)$-$faces$ are called $ridges$, and the 0-$faces$ are called $vertices$.

As previously mentioned, NSA contains three components, detector representation, matching rules and detector generation method. In what follows, we adapt NSA's components for convex hull representations.

3.1. **Detectors.** A detector with this representation simply can be defined as a set of points by their Cartesian coordinates when they define the convex hull's vertices $P_0, P_1, \ldots, P_n$. It is assumed that the vertices in P are ordered counterclockwise, using P_0 as origin. It does not matter which vertex is chosen as the origin point since all the vertices of a convex hull are external points. In case of a tie, we take the one with lowest y coordinate. The decision to list vertices counterclockwise instead of clockwise is arbitrary. The convex can have a fixed number of vertices or a variable number of them. The number of convex hulls vertices can be defined in two ways: first, a fixed number of points and second, a variable number of points chosen randomly within a lower and upper bound. In the former case, the larger the number of points is fixed on, more complex regions are covered. However, generating such a complex detector would be harder. In the latter case, the lower bound should obviously set to three. The upper bound has a direct effect on the complexity of detector. Since a complex convex hull is less feasible to generate, smaller amount of such detectors would be generated. A detailed description of convex hull behavior can be found in Section 4.

3.2. **Matching rule.** Convex hull detectors match all the points located in their corresponding convex hull. Therefore, matching rule will be the algorithm which defines whether a point is in the convex hull or not. A point is inside a convex polygon iff it lies on the same side of all the facets of the convex polyhedron (polygon). Now we need to determine whether a point is on the right side of a facet or the left side of it. This can be implemented by calculating the area of the hypertriangle (or triangle for 2-D) which connects the vertices of the facet and the corresponding point. The area of a hypertriangle $P_0P_1 \ldots P_{n+1}$ where $P_0P_1 \ldots P_n$ are the vertices of the facet, P_{n+1} is the new point and $P_i = (x_{i,0}, x_{i,1}, \ldots, x_{i,n})$ is

$$\text{area} : \frac{1}{2} * \begin{pmatrix} x_{0,0} & x_{1,0} & \ldots & x_{n,0} \\ x_{0,2} & x_{1,2} & \ldots & x_{n,2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{0,n} & x_{1,n} & \ldots & x_{n,n} \\ 1 & 1 & \ldots & 1 \end{pmatrix} \tag{1}$$

If the area is positive then the point P_{n+1} is to the left of the facet. If it is negative the point P_{n+1} is to the right of the facet and if it is zero, it is on the facet.

3.3. **Detector generation.** Let us first concentrate on generating one detector. Since a convex hull detector is represented by its points, the number of convex hulls vertices is considered as the detector size.

The first approach one may come up with is to generate n random points in Cartesian coordinate and then use a convex hull algorithm such as [19] to find the convex region that covers these points. Next step is to check the detector against self data. If it matches with even one self data it must be eliminated and another detector should be generated. Despite the simple implementation of this approach, it has a disadvantage. It causes elimination of a huge number of detectors in the second step since it has no control mechanism to keep the detector away from the self region space.

A simple control mechanism which has been used for spherical detectors [8] is to choose the center of sphere out of the self space. Also its radius is defined equal to Euclidean distance of the nearest self instance to the center. This control mechanism can be extended to convex hulls. Therefore, instead of generating n random point, a center point is produced randomly. The center is checked to be out of the self space. Also as an optimization mechanism to keep number of generated detectors under control, the center is examined to find the number of already generated detectors it matches with. If it is more than a predefined ($\theta_{overlap}$) percentage of the whole number of generated detectors by now, the center is eliminated and a new one is produced. Otherwise, the center point is accepted. Then, n points are generated in polar coordinate centered in the accepted center point. Again, to reduce possible self coverage by the detector, it is tried to choose n points out of the self space. Thus, the points are generated as the following. The angular space around the center point is divided into n parts. For each part, a random angle is generated, and then the radius corresponding to each angle is defined equal to distance between the center point and nearest self data point located close to the corresponding angle. As usual [6, 7, 8], if one of the radiuses is less than a predefined value ($theta_{self}$), the detector is ignored and a new center would be generated.

3.3.1. *Convex hull construction algorithm.* To solve the convex hull problem in 2-D or higher dimensions, there exist numerous proposed methods among them Gift Wraping is a well-known one for computing the convex hull of a set of points in two or more dimensions [20]. The special case of this method is Jarviss March algorithm in 2-D [19]. The idea of this method is based on the fact that all points of a set lie on the same side of their convex hull edges (or on the edges). Jarviss March algorithm initializes the first vertex of convex hull, v_0, with an external point which is defined as the furthest point from all points in some direction. Next, the algorithm tries to find the adjacent point of v_0 which is the point that builds a convex hull edge with v_0. Due to the mentioned fact, an edge between two points is a convex hull edge iff all other points lie on the same side of the edge. The algorithm continues with finding the adjacent vertex of the lastly found vertex v_i and this process repeats until one reaches v_0. This approach can be extended to higher dimensions called Gift Wrapping algorithm. After generating convex hull of the detector, it is checked if it contains any self data or not. As usual, the detector which includes any self data instance is eliminated.

3.3.2. *Complexity.* Figure 1 shows the flow diagram of the detector generation phase in CH-NSA. The control parameters of the algorithm are self radius, θ_{self}, maximum number of allowed overlapped detectors, $\theta_{overlap}$, and the number of convex hull vertices which is called the detector size, $|d|$. From Figure 1, it can be observed that this detector generation

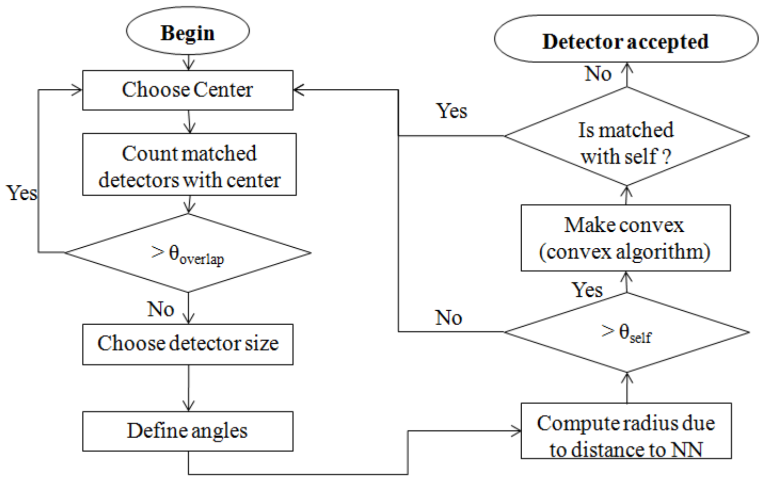


Figure 1. Flow diagram of CH-NSA

mechanism can be divided into three phases. The first phase is to find a proper center for the convex hull which can be assumed to have the complexity of O_1. The complexity of this phase is dependent on the value of parameter $\theta_{overlap}$ which would be discussed in next section. The second phase is to generate the proper radiuses and produce convex hull with complexity O_2 which is equal to size of the training dataset, $|S|$, in addition to the complexity of generating a convex hull with $|d|$ vertices. The complexity of generating a convex hull of n points in D dimensions using Gift Wrapping is $O(n^{([D/2]+1)})$ which can be negligible for low dimension problem space. The last phase is to check the detector against self data with complexity O_3 equal to the size of the training dataset. Therefore, the whole complexity could be estimated from Equation (2).

$$\text{O(CH-NSA)} = \left(O_1 + |S| + |d|^{\left[\frac{D}{2}\right]+1} + |S|\right) * |detectorset| \tag{2}$$

where, $|detectorset|$ indicates the number of generated detectors.

4. **Results and Discussion.** This section presents the experimental results in four subsections. First, the tested datasets are described and second, the experimental setup for control parameters of proposed method is given in detail. The next subsection presents the achieved results of CH-NSA in comparison with the well known *V-detector* algorithm. Finally in the last subsection, the behavior of the proposed algorithm and the effect of control parameters are analyzed.

4.1. **Experiments design.** To verify the basic behavior of CH-NSA, experiments were carried out using six synthetic data which have been used in literature as benchmark problems [6, 7, 8]. These datasets are selected due to their various shapes, small size and few dimensions which would lead to better visual understanding of the algorithm behavior. The different shapes are designed as the self space which are normalized in the unit square $[0, 1]^2$. These shapes are shown in Figure 2. The datasets contain certain known characteristics of common problem spaces such as sharp corners, curvilinear spaces and disjoint parts.

In Figure 2, the white area presents the self space and the remaining is considered as non-self space. The training data is composed of 1000 random points sampled from self space and the test data consists of 1000 random points from both non-self and self space. Each instance of the training data has two real valued attribute according to the position

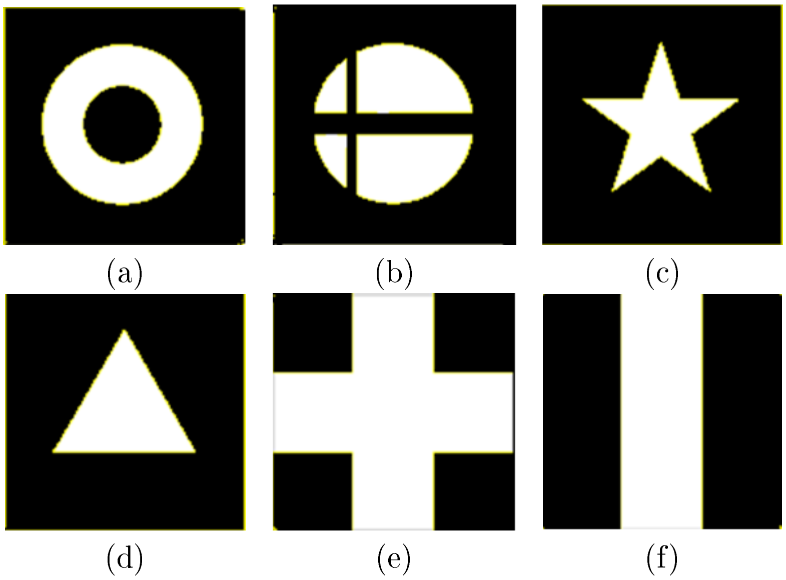


FIGURE 2. Different types of shape used as synthetic data, drawn from [7]: (a) ring; (b) intersection; (c) pentagram; (d) triangle; (e) cross; (f) stripe

of corresponding sample pixel. In addition to these two attributes, test data contains a class label for each instance which determines self/non-self status of the sample.

4.2. **Experimental settings.** To obtain performance of CH-NSA, the average and standard deviation of the accuracy over 50 independent runs of the algorithm is calculated. As it is mentioned earlier, the algorithm has four control parameters; θ_{self}, $\theta_{overlap}$, convex size $|d|$ and number of detectors $|S|$. To compare performance of the *V-detector* algorithm and CH-NSA, the parameters are configured as follows; θ_{self} is set to 0.05 as it is common in the literature, [6, 7, 8]. $\theta_{overlap}$ is set to $\frac{1}{15}$ experimentally, $|d|$ is randomly chosen among 4, 5 and 6 and number of detectors, $|S|$, is set to 50 the same as in V*-detector* to have a fair comparison. Also, the one-tailed pairwise t-test is applied to check whether the difference between results of two methods over the datasets is statistically significant or not. The null hypothesis of this test is that CH-NSA and V*-detector* perform the same on average over 50 independent runs against the alternative that the average performances are not equal.

4.3. **Results.** Table 1 highlights the difference between the CH-NSA and V*-detector* algorithms. V*-detector* algorithm is implemented as in [7] and is justified by results mentioned there.

TABLE 1. The results of applying CH-NSA and V*-detector* algorithm in selected datasets

Data Set	CH-NSA		V-detector		p-value
	TP	FP	TP	FP	
Pentagram	99.60 ± 0.2	1.7 ± 0.7	96.4 ± 1.7	0.5 ± 0.7	$2.7802e-021$
Intersection	98.60 ± 0.4	3.7 ± 1	90.6 ± 2.7	4.3 ± 2.2	$1.4308e-035$
Ring	98.70 ± 0.5	4.8 ± 1	93.2 ± 3.2	6.3 ± 1	$1.1045e-018$
Cross	99.18 ± 0.4	2.8 ± 0.6	95.12 ± 2.2	2.2 ± 0.4	$3.1861e-026$
Stripe	99.3 ± 0.3	1.3 ± 0.7	99.4 ± 0.8	2.2 ± 0.6	$1.5321e-031$
Triangle	99.75 ± 0.2	0.8 ± 0.9	99.4 ± 0.5	1.0 ± 0.7	$2.1584e-024$

The table contains the average of true positive rate (TP), false positive rate (FP) and their standard deviation over 50 independent runs. Also, the obtained p-value of pairwaise t-test with 5% significance level is presented. The results show that with the same number of detectors, CH-NSA can achieve better TP and FP rate in almost all datasets.

It was stated earlier that the results presented in Table 1 is obtained by a fixed number of detectors, 50. Nevertheless, using proposed method, acceptable results can be achieved with a very small number of detectors. This is shown in Figure 3 where, results of CH-NSA and *V-detector* algorithm with different values for $|S|$ on pentagram, intersection, cross and ring benchmarks are illustrated. Comparison of the two methods shows that the proposed method results in slightly few detectors to reach 90 percentage of true positive rate.

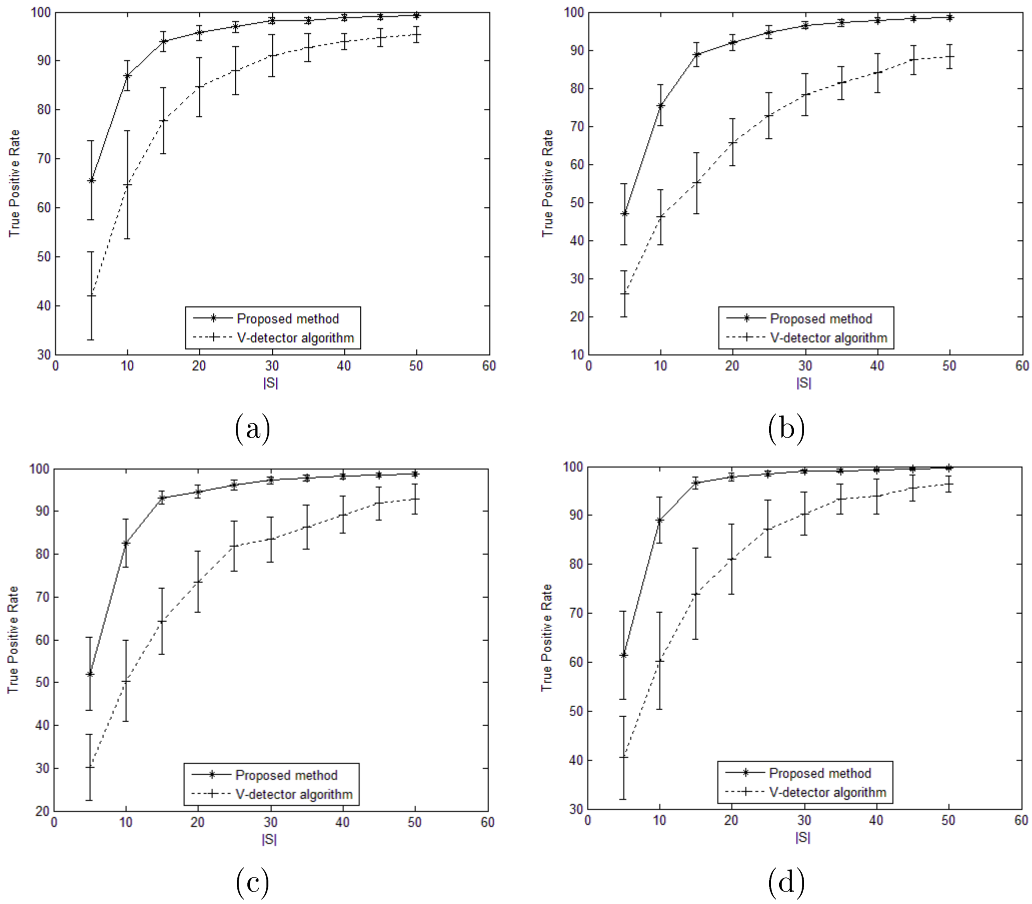


Figure 3. Comparison of proposed method and *V-detector* algorithm in different values of $|S|$ on: (a) cross; (b) intersection; (c) ring; (d) pentagram

4.4. **Discussions.** To determine the reason of this improvement in achieved performance, first take a look at the visual results in Figure 4 where the results of both algorithms on pentagram and intersection benchmarks are compared. As it seems, CH-NSA can better cover the corner areas. This is perfectly obvious in the results, for example in pentagram shape, Figure 4(a). Although *V-detector* tries to fill the corners with small spheres, it does not succeed in a complete coverage of those corners.

As shown in the Figures 4(c) and 4(d), it seems that the spherical shape is not a good choice to be matched with the self data boundaries. The self boundaries can be categorized in four main groups: straight lines, corners, convex and concave curves. The detectors

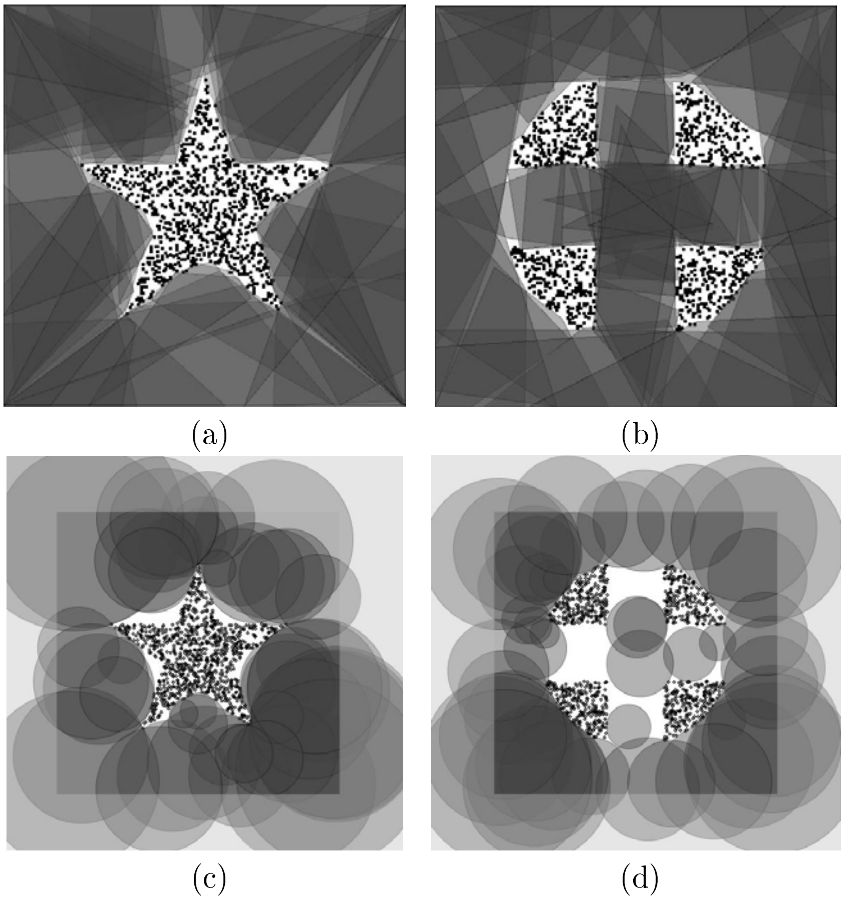


FIGURE 4. Comparison of CH-NSA and *V-detector* in pentagram and intersection shapes. (a) Result of pentagram shape by CH-NSA; (b) results of intersection shape by CH-NSA; (c) result of pentagram shape by *V-detector*; (d) result of intersection shape by *V-detector*.

with spherical representation can effectively cover the convex curves, obviously because sphere is a convex curve. However, it can estimate the other three groups only if a large number of them are used. In contrast, the convex hull representation has better ability to match with straight lines and corners. As it was mentioned in the previous section, a convex hull can be created using a center, a list of angles and their corresponding radiuses. The strategy for determining the radius is the same in both convex hulls and spheres; they are computed equal to the distance to the nearest self data. However, the interesting point is that convex hull has an asymmetric shape which needs several numbers of different radiuses whereas sphere is symmetric shape and may have just one radius. It is possible to interpret this property as degree of freedom. Therefore, convex hull has higher degree of freedom in contrast with sphere. Although sphere shape has intuitive ability in matching with curve boundaries and convex hull shape does not, high degree of freedom gives a hand to convex hull in matching with these curvilinear boundaries. The result of CH-NSA for intersection shape which contains both curvilinear and sharp corners confirms the above claim.

One of the notable features in intersection benchmark is disjoint self spaces. Due to this property, the detector generation mechanism must be in a way that the whole non-self regions around the self space would be covered. *V-detector* has a simple detector generation mechanism; the centers of detectors are generated randomly and the overlap of detectors is not considered. Instead, the covered area by all detectors is estimated with a hypothesis test. The number of generated random points is proportionate to the number of detectors. Therefore, the narrow areas between the self spaces would have less chance to be selected as center. This problem is apparent in the intersection shape results.

A straightforward solution to above drawback is to consider the overlapping centers. This is done in CH-NSA by rejecting those random centers which are covered in more than a predefined number of already generated detectors. To perceive the influence of the remarked solution, we changed the *V-detector* algorithm so that instead of estimating coverage, it eliminates the overlapped detectors as explained above. In this way, the non covered narrow areas would have more chance to be selected as center what in turn can boost the performance of our model. Table 2 presents the results of new *V-detector* in comparison with its original form.

TABLE 2. The results of applying CH-NSA and *V-detector* algorithm in selected datasets

Data Set	V-detector		changed V-detector	
	TP	FP	TP	FP
Pentagram	96.4 ± 1.7	0.5 ± 0.7	98.55 ± 0.5	1.0 ± 0.1
Intersection	90.6 ± 2.7	4.3 ± 2.2	93.92 ± 1.7	5.0 ± 2
Ring	93.2 ± 3.2	6.3 ± 1	98.20 ± 1.6	6.5 ± 1
Cross	95.12 ± 2.2	2.2 ± 0.4	98.74 ± 0.7	2.6 ± 0.4

Figure 5 shows the effect of control parameter $\theta_{overlap}$ on the true positive rate, and also, the proportion of elapsed time for generating center to that of needed to generate detectors. Different values of $\theta_{overlap}$ are tested on pentagram shape. $\theta_{overlap}$ can be defined as a fraction of the number of generated detectors. In this experiment $\theta_{overlap}$ is defined as:

$$\theta_{overlap} = \frac{1}{t} \times |detectorset| \quad (3)$$

For each marker in Figure 5, the parameter value is reported as a function of t. The smaller $\theta_{overlap}$, the higher detection rate, and the more time is taken to get the results. Therefore, this parameter can be used as a balance between high detection rate and low elapsed time. It is worth mentioning that elapsed time is not a problem since the whole time expended for detection is not too long.

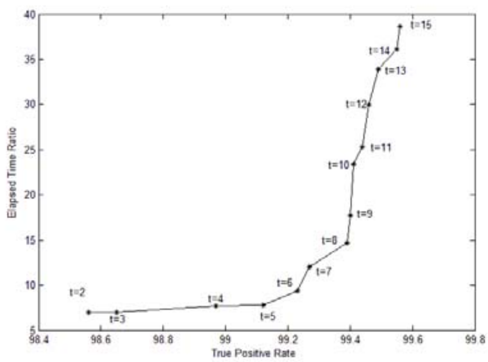
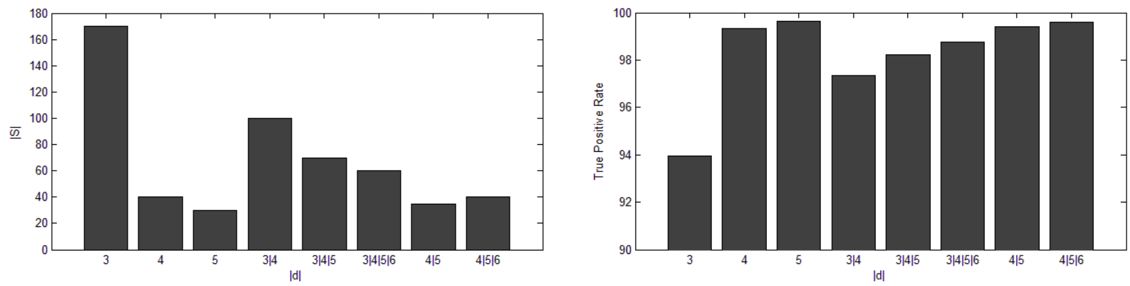


FIGURE 5. Results of different values of $\theta_{overlap}$ on pentagram shape. The parameter value is reported as a function of t.

Figure 6 shows the complete affect of detector size, $|d|$ on detection rate and number of detectors needed to reach 99 percentage of true positive rate. Two series of experiments are conducted to explore the effect of value of $|d|$. In the first round of experiments, the value of $|d|$ is set to a predefined number 3, 4 or 5 where this value remain intact in the meanwhile. In the second round of experiments though, the value of $|d|$ is randomly selected from the list (3, 4, 5). Please note that the notation $x|y$ stands for selecting between x and y in a random fashion. The results are obtained from pentagram benchmark. It is clear that using larger $|d|$ causes increased true positive rate.



(a) (b)

Figure 6. Results of different values of $|d|$ on pentagram benchmark. (a) Affect of $|d|$ on $|S|$ needed to reach 99% of true positive rate; (b) affect of $|d|$ on true positive rate.

Since in this method, vertices of convex hull are generated by a pseudo random process, larger number of vertices can make greater convex hull in terms of covered area. Therefore, with smaller number of such convex hulls, higher coverage can be achieved. So, it can be said that larger $|d|$ causes an increase in complexity of generating centers.

5. **Conclusions and Future Work.** Here, Negative Selection Algorithm is investigated and regarding its structure, it is described that the representation component is one of its main sources of power. It is its representation component which determines available tools to determine the shape of self region and therefore it affects its ability to detect anomaly and its performance. In this work, a new extension of NSA named CH-NSA with convex hull based representation is presented. CH-NSA is tried to cover the self space using convex regions to achieve better maximum coverage with lower number of detectors. To explore the advantages of CH-NSA over the well known NSA, named *V-detector* algorithm, in terms of true positive rate, standard deviations and number of detectors, a number of experiments on well known benchmarks are conducted.

Regarding to the reported results, CH-NSA has showed a good performance regarding both the coverage area and the needed number of detectors. Also, it can outperform the variable size spherical representation due to both mentioned performance criterion in well-known benchmark problems. As it was mentioned in the discussion section, this better performance is somehow due to the asymmetric shape and higher flexibility of convex hulls in covering complex areas. The flexibility of convex hull representation brings the benefit to CH-NSA algorithm in the sense that it provides less number of detectors with more efficient detection of new sample, and also, promising true positive rate.

The detector generation mechanism used in this algorithm is a pseudo random process which considers the uncovered area of problem space. Hence, it seems that a good topic as a future work can be designing a detector generation mechanism which can distinguish the best detectors in terms of its covered area. Also, the influence of higher dimensional data sets needs further study including more experiments and analysis.

REFERENCES

[1] L. N. de Castro and F. J. Von Zuben, Artificial immune systems: Part I – Basic theory and applications, *Technical Report, No.DCA-RT 01/99*, School of Computing and Electrical Engineering, State University of Campinas, Brazil, 1999.

[2] S. Forrest, A. S. Perelson, L. Allen and R. Cherukuri, Self-nonself discrimination in a computer, *Proc. of the 1994 IEEE Symposium on Security and Privacy, IEEE Computer Society*, Washington, USA, pp.202-212, 1994.

[3] P. D'haeseleer, S. Forrest and P. Helman, Immunological approach to change detection: Algorithms, analysis and implications, *Proc. of the 1996 IEEE Symposium on Security and Privacy. SP. IEEE Computer Society*, Washington, USA, 1996.

[4] D. Dasgupta and F. Gonzalez, An immunity-based technique to characterize intrusion in computer networks, *IEEE Transactions on Evolutionary Computation*, vol.6, no.3, pp.1081-1088, 2002.

[5] F. A. Gonzalez and D. Dasgupta, Anomaly detection using real-valued negative selection, *Proc. of Genetic Programming and Evolvable Machines*, vol.4, pp.383-403, 2003.

[6] S. Balachandran, D. Dasgupta, F. Nino and D. Garrett, A framework for evolving multishaped detectors in negative selection, *Proc. of IEEE Symposium Series on Computational Intelligence*, Honolulu, pp.401-408, 2007.

[7] Z. Ji and D. Dasgupta, V-detector: An efficient negative selection algorithm with p̈robably adequatëdetector coverage, *Information Science*, vol.179, no.10, pp.1390-1406, 2009.

[8] Z. Ji and D. Dasgupta, Real-valued negative selection algorithm with variable-sized detectors, *Proc. of GECCO*, pp.287-298, 2004.

[9] G. Li, T. Li, J. Zeng and H. Li, An improved V-detector algorithm of identifying boundary, *Proc. of the 8th International Conference on Machine Learning and Cybernetics*, pp.3209-3214, 2009.

[10] Z. Ji and D. Dasgupta, Revisiting negative selection algorithms, *Evolutionary Computation*, vol.15, no.2, pp.223-251, 2007.

[11] J. K. Percus, O. E. Percus and A. S. Perelson, Predicting the size of the T-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination, *Proc. of National Academy of Sciences USA*, vol.90, no.5, pp.1691-1695, 1993.

[12] F. Gonzalez, D. Dasgupta and L. F. Nino, A randomized real-value negative selection algorithm, *Proc. of the 2nd International Conference on Artificial Immune System*, pp.261-272, 2003.

[13] P. J. C. Branco, J. A. Dente and R. V. Mendes, Using immunology principle for fault detection, *IEEE Transactions on Industrial Electronics*, vol.50, no.2, pp.362-373, 2003.

[14] D. Dasgupta, K. KrishnaKumar, D. Wong and M. Berry, Negative selection algorithm for aircraft fault detection, *Proc. of the 3rd International Conference on Artificial Immune Systems*, pp.1-13, 2004.

[15] X.-Z. Gao, S. J. Ovasaka and X. Wang, A GA-based negative selection algorithm, *International Journal of Innovative Computing, Information and Control*, vol.4, no.4, pp.971-979, 2008.

[16] Q. Wang and X. K. Feng, A detector generation algorithm based on negative selection, *The 4th International Conference on Natural Computation*, pp.605-611, 2008.

[17] H. Xia, S. Ju and T. Cai, A detector generating algorithm based on interval partition, *International Journal of Distributed Sensor Networks*, vol.5, no.1, pp.27, 2009.

[18] P. L. Lanzi and S. W. Wilson, Classifier conditions based on convex hull, *ILLIGAL Report, No.2005024*, 2005.

[19] R. A. Jarvis, On the identification of the convex hull of a finite set of points in the plane, *Information Processing Letters*, vol.2, no.1, pp.18-21, 1973.

[20] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *33.3: Finding the Convex Hull. Introduction to Algorithms*, 2nd Edition, MIT Press and McGraw-Hill, 2001.