Real-Valued Negative Selection Algorithm with Variable-Sized Detectors

Zhou Ji[1], Dipankar Dasgupta[2]

[1] St. Jude Children's Research Hospital
Memphis, TN 38105
zhou.ji@stjude.org
[2] The University of Memphis
Memphis, TN 38152
ddasgupt@memphis.edu

Abstract. A new scheme of detector generation and matching mechanism for negative selection algorithm is introduced featuring detectors with variable properties. While detectors can be variable in different ways using this concept, the paper describes an algorithm when the variable parameter is the size of the detectors in real-valued space. The algorithm is tested using synthetic and real-world datasets, including time series data that are transformed into multiple-dimensional data during the preprocessing phase. Preliminary results demonstrate that the new approach enhances the negative selection algorithm in efficiency and reliability without significant increase in complexity.

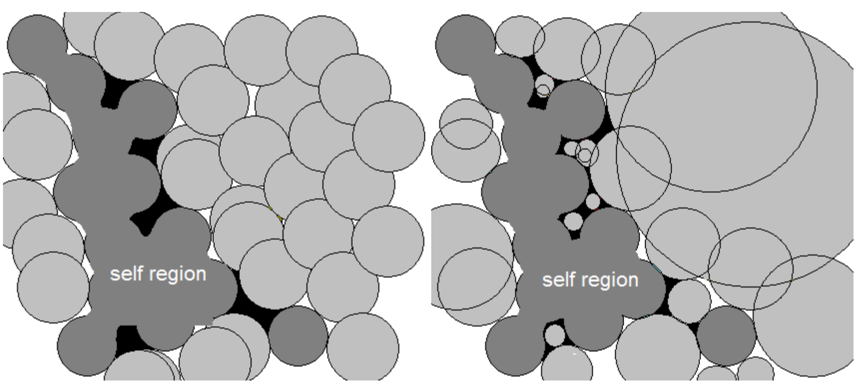
1 Introduction

Soft computing is an increasingly active research area in computational intelligence. Artificial Immune Systems are soft computing techniques that are based on metaphor of the biological immune system. [1][2][3][4]. The immune system shows computational strength from different aspects in problem solving. Most existing AIS algorithms imitate one of the following mechanisms of the immune system: negative selection, immune network, or clonal selection. Negative selection-based algorithm [1][2] has potential applications in various areas, in particular anomaly detection. The inspiration of negative selection comes from the T cell maturation process in the immune system: if a T cell in thymus recognizes any self cell, it is eliminated before deploying for immune functionality. In a similar manner, the negative selection algorithm generates detector set by eliminating any detector candidate that match elements from a collection of self samples. These detectors subsequently recognize non-self data by using the same matching rule. In this way, it is used as an anomaly detection algorithm that only requires normal data to train [5].

Most works in negative selection used the problem in binary representation [6][7]. There are at least two obvious reasons of this choice: first, binary representation provides a finite problem space that is easier to analyze; second, binary presentation is straightforward to use for categorized data. However, many applications are natural to be described in real-valued space. Furthermore, these problems can hardly be processed properly using negative selection algorithm in binary representation [8]. On the other hand, this work and some other works [9][10] demonstrated that despite the intrinsic difficulty of real-valued representation, it can also provide unique opportunity in dealing with higher dimensionality.

Matching rule is one of the most important components in a negative or positive pattern detection algorithm [6][7][8][11][12]. For binary representation, there exist several matching rules like rcb (*r*-contiguous bit), *r*-chunks, and Hamming distance [6][8]. For real-valued representation, however, the Euclidean distance is primarily used [8][9][10][13]. Matching is determined when the distance between a data point and some detector is within a certain threshold. In some cases, variations of Euclidean distance are used, for example, a Euclidean distance defined in a lower dimensional space projected from the original higher dimensional problem space [13].

Independent of the type of matching rule, the detectors usually have some basic characteristics, e.g., the number of bits, *r*, in binary representation, or the distance threshold, Δ, to decide a matching in real-valued representation, that are constant through out the entire detector set. However, the detector features can reasonably be extended to overcome this limitation. The algorithm introduced in this paper demonstrates that allowing the detectors to have some variable properties will enhance the performance of negative detectors. We call this idea and the algorithm based on it as *V-detector*. In the case of real-valued negative selection algorithm, the detectors are in fact hyper-sphere-shaped. The threshold used by Euclidean distance matching rule defines the radius of the detectors. The radius is an obvious choice to make variable considering that the non-self regions to be covered by detectors are very likely to be in different scales. The flexibility provided by the variable radius is easy to realize. However, variable radius is not the only possibility provided by *V-detector*. Detector variability can also be achieved by other ways, such as different detector shapes, variable matching rules, etc.



(a) Constant-sized detectors (b) Variable-sized detectors

Fig. 1. Main concept of Negative Selection and *V-detector*

Figure 1 illustrates the core idea of variable-sized detectors in 2-dimensional space. The dark grey area represents the actual self region, which is usually given through the training data (self samples). The light grey circles are the possible detectors covering the non-self region. Figure 1(a) shows the case where the detectors are of constant size. In this case, a large number of detectors are needed to cover the large area of non-self space. The well-known issues of “holes” are illustrated in black. In figure 1 (b), using variable-sized detectors, the larger area of non-self space can be covered by fewer detectors, and at the same time, smaller detectors can cover the holes. Since the total number of detectors is controlled by using the large detectors, it becomes more feasible to use smaller detectors when necessary.

Another advantage of this new method is that estimated coverage, instead of the number of detectors, can be used as a control parameter. The algorithm can evaluate the estimated coverage automatically when the detector set is generated. On the other hand, we need to set the number of detectors in (advance) when constant sized detectors are used. This will be discussed in more details in the following sections.

2 Algorithm and Analysis

Detector - Set(S, m, r_S)

S : set of self samples

m : number of detectors

r_S : self radius

1 : $D \leftarrow \varnothing$

2 : Repeat

3 : $\quad x \leftarrow$ random sample from $[1, 0]^n$

4 : $\quad$ Repeat for every s_i in $S = \{s_i, i = 1,2,...\}$

5 : $\qquad d \leftarrow$ Euclidean distance between s_i and x

6 : $\qquad$ if $d \le r_S$, go to 2

7 : $\quad D \leftarrow D \cup \{x\}$

8 : Until $|D| = m$

9 : return D

Fig. 2. Detector generation algorithm in negative detection using constant-sized detectors

A negative selection algorithm basically consists of two phases. First, the detector set is generated in the training or generation phase. Then, the new sample is examined using the detector set during the detection phase. To highlight the feature of *V-detector*, let us first describe the real-valued negative detection algorithm using constant-sized detectors, where candidate detectors are generated randomly. Those that match any self samples (training data) using Euclidean distance matching rule are eliminated. The generation phase finishes when a preset number of detectors are obtained. The generation phase of this algorithm is shown in the figure 2. The time complexity of this algorithm is $O(m|S|)$, where m is the preset number of detectors and $|S|$ is the size of training set (self samples). Self radius r_s in this case is the same as the detector radius, which represents the allowed variability of the self points [10].

V - Detector - Set(S, T_{max}, r_s, c_o)

S : set of self samples

T_{max} : maximum number of detector

r_s : self radius

c_0 : estimated coverage

$D \leftarrow \varnothing$

Repeat

    $t \leftarrow 0$

    $T \leftarrow 0$

    $r \leftarrow$ inifinite

    $x \leftarrow$ random sample from $[1,\ 0]^n$

    Repeat for every d_i in $D = \{d_i . i = 1, 2, ...\}$

        $d_d \leftarrow$ Euclidean distance between d_i and x

        if $d_d \leq r(d_i)$ then, where $r(d_i)$ is the radius of d_i

            $t \leftarrow t + 1$

            if $t \geq 1/(1 - c_0)$ then return D

            go to 4 :

    Repeat for every s_i in S

        $d \leftarrow$ Euclidean distance between s_i and x

        if $d - r_s \leq r$ then $r \leftarrow d - r_s$:

    if $r > r_s$ then $D \leftarrow D \cup \{< x, r >\}$, where $< x, r >$ is a detector

    else $T \leftarrow T + 1$

    if $T > 1/(1 -$ maximum self coverage$)$ exit

Until $|D| = T_{max}$

return D

Fig. 3. Detector generation algorithm of *V-detector*

V-detector algorithm also generates candidate detectors randomly. However, when we check the matching rule of Euclidean distance, we keep the distance in record and assign a variable radius based on the minimum distance to each detector that is going to be retained. The detector's detector generation phase is described in figure 3. Comparing with the version of constant-sized detectors, the most important differences lie in steps 13 through 15. Now that we let each detector has its own radius in addition to the location, the radius is basically decided by the closest self sample. Self radius still specifies the variability represented by the training data, but it is not used as detector radius anymore.

The algorithms of detection phase are similar for constant and variable detectors except that matching threshold for each variable-sized detector is difference. In the experiments of this paper, matching is decided by the closest detector.

The control parameters of *V-detector* are mainly self radius r_s and estimated coverage c_0. Maximum number of detectors, shown as T_{max} in figure 3, is preset to be the maximum allowable in practice, which does not need much further discussion. Self radius is an important mechanism to balance between detection rate and false alarm rate, in the other words, the sensitivity and accuracy of the system.

Estimated coverage is a by-product of variable detectors. If we sample m points in the considered space and only one point is not covered, the estimated coverage would be $1-1/m$. Therefore, when we randomly try m times without finding an uncovered point, we can conclude that the estimated coverage is at least $\alpha = 1-1/m$. Thus, the necessary number of tries to ensure estimated coverage α is

$$m = 1/(1-\alpha) \quad \mathbf{(1)}$$

Despite the enhancement, complexity of *V-detector* is not increased comparing to basic negative selection. The computation of radius has linear complexity to the number of the training set size. Steps 13 through 15 has complexity O ($|S|$), where $|S|$ is the set size of training data, just the same as steps 4 through 6 in figure 2's basic negative selection. Furthermore, not only are the complexities of the same order of n, but the times to actually compute the distance, which is potentially a costly step, is the same as well. If the final number of detectors is m, the total complexity if $O(m|S|)$. If m has the same order of magnitude as the preset number in the algorithm using constant-sized detectors, the complexity doesn't change; if m is reduced significantly, the complexity is further improved.

Similarly, the complexity of detection algorithms is $O(m)$ although m has different interpretation in the two methods. The difference in space complexity also only lies in the possible different m, which can always be limited in *V-detector*.

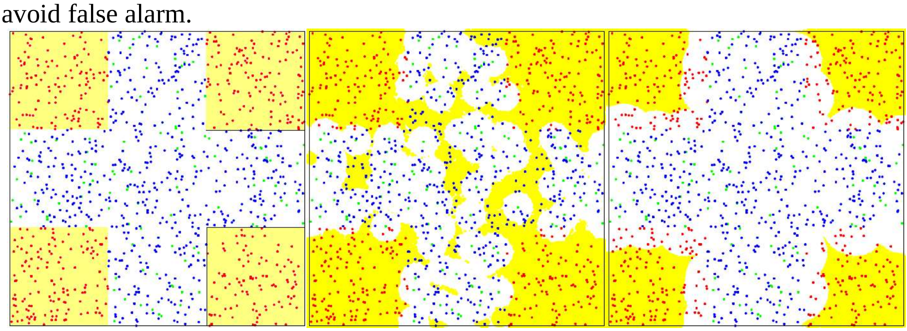
The *V-detector* algorithm normally converges in one of the two ways. Type 1 convergence is when the estimated coverage is reached in step 11 of figure 3. This is the scenario that *V-detector* shows more of its strength in controlling detector number. Type 2 convergence is when the limit of detector number is reached in step 19. Even in this case, the algorithm still has the potential to cover holes better then basic algorithm. There is another possibility that the algorithm will halt, which we are not going to discuss further. If the training data cover almost all space, say 99.99%, the algorithm terminates as a special case at step 18. It may happen when self-radius is set to be too big so that the whole space is all "normal".

The small "holes" are easier to be covered not by just using smaller detectors, rather by using the automatic decision of how small the detectors need to be. The total numbers of detectors, on the other hand, are dealt with by using larger detectors whenever possible.

3 Experiments and Results

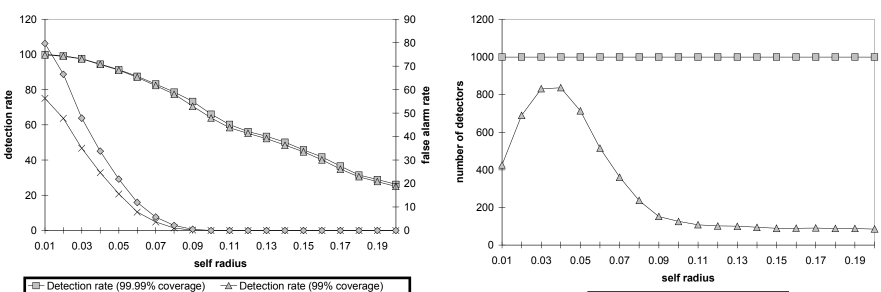
3.1 *V-detector*'s Basic Property on Synthetic Data

A synthetic 2-dimensional datasets are used to demonstrate the properties of *V-detector* algorithm. Figures 4(a) shows a cross-shaped self region over the entire space (unit square) $[0, 1]^2$. The training set is 100 random picked points in the self region and the test set is 1000 random distributed points over the entire space. The shaded area in figures 4(b) and 4(c) shows the coverage achieved by detector set generated using different self radius. Comparing (b) and (c), it is easy to see the effect of self radius on the results. The smaller self radius would result in high detection rate but high false alarm rate too, so it is suitable for the scenario when detecting all or most anomalies is very important. On the other hand, larger self radius would result in low detection rate and low false alarm rate, thus suitable when we need to try the best to avoid false alarm.



(a) Actual self space (b) self radius = 0.05 (c) self radius = 0.1

Fig. 4. Cross-shaped self space



(a) Detection rate and false alarm (b) Number of detectors

Fig. 5. Results on cross-shaped self region

Figure 5 shows the complete trend of self radius' affect on the results for self radius from 0.01 up to 0.2. The results using two different values of estimated coverage, 99% and 99.99%, are presented together to show that parameter's influence. All the results shown in this figure are average of 100 repeated experiments. Detection rate and false alarm rate are defined as

$$DR = TP/(TP+FN), \quad (2)$$

$$FA = FP/(FP+TN), \quad (3)$$

respectively, where TP, FN, FP, TN are the counts of true positive, false negative, false positive, and true negative. As shown in these results, high detection rate and low false alarm rate are the two goals between which we need to balance according to specific application. While *V-detector* algorithm uses much fewer detectors for both cases, more detectors are needed to obtain the estimated coverage when the self radius is small. The shape of self region also have direct effect on the detector number.

3.2 Comparison with Similar Methods on Real-World Data

To study the property and possible advantages of *V-detector*, experiments were also carried out to compare with the results obtained using other anomaly detection methods that only use normal data to train as *V-detector*. Two such AIS methods were reported in [13], namely MILA (*M*ultilevel *I*mmune *L*earning *A*lgorithm) and NSA (*N*egative *S*election *A*lgorithm, single level to compare with MILA). MILA is a multilevel model of combined negative detection and positive detection [13][14]. It provides a very flexible yet complex mechanism for anomaly detection. Single Level NSA to be compared is to some extent similar to the negative selection using constant-sized detectors described earlier in this paper. However, both MILA and Single Level NSA use a subset of all the dimensions of the problem space to calculate the Euclidean distance for the matching rule. While MILA's model involves multiple ways to choose the subset, Single Level NSA can be seen a extended version of rcb (*r*-contiguous bits) – *r* contiguous dimensions out of all the dimensions [13][14]. Nevertheless, the detectors are constant sized both in MILA and in Single Level NSA.

Table 1 shows the comparison using the famous benchmark Fisher's Iris Data (self raduis 0.1, estimated coverage 99%). The results shown are the summary of 100 repeated tests for each method and parameter setting. One of the three types of iris is considered as normal data, while the other two are considered abnormal. The normal data are either completely or partially used to train the system. Although the partial training set may seem small in this case, it is necessary to demonstrate the system's capability to recognize unknown normal data. As we have seen, self radius is an important control parameter of *V-detector* to balance its performance. The results in this table are obtained using self radius $r_s = 0.1$ considering that the Single Level NSA and MILA results were from threshold 0.1. However, we have to note that the threshold used in Single Level NSA or MILA is not strictly comparable to the self radius in *V-detector*. In the results cited here, MILA or Single Level NSA uses a sliding window of size 2, so the distance is defined in 2-dimensional space, not the original 4-dimensional space. The maximum detector set size is set to be 1000 for the reason of comparison too. *V-detector* has comparable detection rate but lower false alarm rate in most cases, especially when fewer training data were used. *V-detector*'s another obvious advantage is the potentially smaller number of detectors. Table 1 also shows that *V-detector* can obtain similar or better results using much smaller detector number in all cases.

The main control parameters, self radius and estimated coverage, can be used to balance between high detection rate and low false alarm rate. In the experiments we just described, false alarm did not really become a problem when all available training data are used, so the issue is more readily illustrated when only partial data are used to train.

Table 1. Comparison between *V-detector* and other methods using Fisher's Iris Data

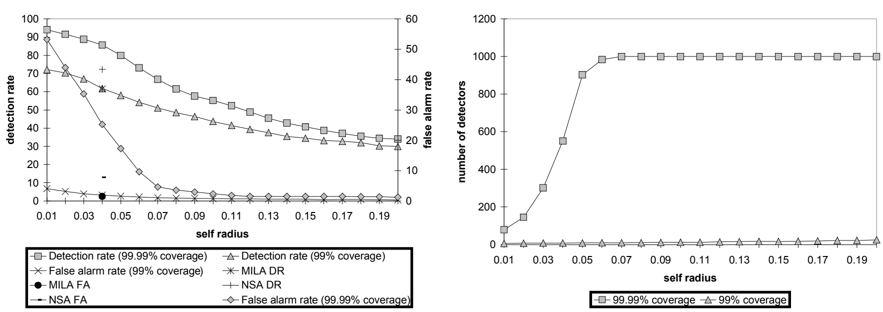
Training Data	Algorithm	Detection Rate		False Alarm rate		Number of Detectors	
		Mean	SD	Mean	SD	Mean	SD
Setosa 100%	MILA	95.16	1.79	0	0	1000*	0
	NSA	100	0	0	0	1000	0
	V-detector	99.98	0.14	0	0	20	7.87
Setosa 50%	MILA	94.02	2.44	8.42	1.56	1000*	0
	NSA	100	0	11.18	2.17	1000	0
	V-detector	99.97	0.17	1.32	0.95	16.44	5.63
Versicolor 100%	MILA	84.37	2.79	0	0	1000*	0
	NSA	95.67	0.69	0	0	1000	0
	V-detector	85.95	2.44	0	0	153.24	38.8
Versicolor 50%	MILA	84.46	2.70	19.60	2.00	1000*	0
	NSA	96	0.45	22.2	1.25	1000	0
	V-detector	88.3	2.77	8.42	2.12	110.08	22.61
Virginica 100%	MILA	75.75	2.01	0	0	1000*	0
	NSA	92.51	0.74	0	0	1000	0
	V-detector	81.87	2.78	0	0	218.36	66.11
Virginica 50%	MILA	88.96	2.04	24.98	2.56	1000*	0
	NSA	97.18	0.71	33.26	0.96	1000	0
	V-detector	93.58	2.33	13.18	3.24	108.12	30.74

* MILA has actually 1000 T-cell detectors and 1000 groups of B-cell detector.

Similar comparison was done for a biomedical dataset, which is blood measurement of a group of 209 patients [15]. Each patient has four different types of blood measurements. These blood measures were used to screen a rare genetic disorder. 134 of the patients are normal; 75 patients are carrier of the disease, the "anomalies" to be detected. Table 2 compares the results from MILA and Single Level NSA, and the results from *V-detector* using self radius 0.1 and self radius 0.05. When all the available normal data were used to train the system, the false alarm didn't occur. However, the detection rate is lower than the cases trained by only part of the normal data. Considering the balance between detection rate and false alarm, and the much less number of detectors used, *V-detector*'s results are comparable. Figure 6 shows the balance over a whole range of self radius. The results by Single Level NSA and MILA are plotted as individual points at a comparable self radius on the graph. *V-detector*'s results appears better if we consider both detection performance and false alarm issue. It further confirms *V-detector*'s advantage in balancing the goals.

Table 2. Comparison between *V-detector* and other methods using biomedical data

Training Data	Algorithm	Detection Rate		False Alarm rate		Number of Detectors	
		Mean	SD	Mean	SD	Mean	SD
100% training	MILA	59.07	3.85	0	0	1000*	0
	NSA	69.36	2.67	0	0	1000	0
	r=0.1	30.61	3.04	0	0	21.52	7.29
	r=0.05	40.51	3.92	0	0	14.84	5.14
50% training	MILA	61.61	3.82	2.43	0.43	1000*	0
	NSA	72.29	2.63	2.94	0.21	1000	0
	r = 0.1	32.92	2.35	0.61	0.31	15.51	4.85
	r=0.05	42.89	3.83	1.07	0.49	12.28	4
25% training	MILA	80.47	2.80	14.93	2.08	1000*	0
	NSA	86.96	2.72	19.50	2.05	1000	0
	r=0.1	43.68	4.25	1.24	0.5	12.24	3.97
	r=0.05	57.97	5.86	2.63	0.77	8.94	2.57



(a) Detection rate and false alarm (b) Number of detectors

Fig. 6. Balance between detection rate and false alarm rate (biomedical data)

3.3 Application on Time Series Data

V-detector algorithm is used to detect ball bearing fault. The raw data are the time series of measured acceleration of ball bearings [16]. As preprocessing, the time series is first transformed into multiple-dimensional data using two common methods of signal analysis. The first method is basically DFF (Discrete Fourier Transform). It takes overlapped segments of 64 points from the raw time series. The step between the segments is chosen to be 8. Fast Fourier Transform (FFT) is performed with Hanning windowing to each segment [17][18]. Half of the Fourier transform coefficients are taken as data points to be detected. The data is thus 32-dimensional. The second method uses statistical moments to represent the property of each segment of 128 points [19]. The moments of first (mean), second (variance), third, four, and fifth order are used, so the resulted data points become 5-dimensional.

Table 3. Detection Results on Fast Fourier Transform of Different Ball Bearings

Ball bearing conditions	Total number of data points	Number of detected anomalies	Percentage detected
New bearing (normal)	2739	0	0%
Outer race completely broken	2241	2182	97.37%
Broken cage with one loose element	2988	577	19.31%
Damage cage, four loose element	2988	337	11.28%
No evident damage; badly worn	2988	209	6.99%

Table 4. Detection Results on Statistical Moments of Different Ball Bearings

Ball bearing conditions	Total number of data points	Number of detected anomalies	Percentage detected
New bearing (normal)	2651	0	0%
Outer race completely broken	2169	1674	77.18%
Broken cage with one loose element	2892	14	0.48%
Damage cage, four loose element	2892	0	0%
No evident damage; badly worn	2892	0	0%

Table 3 shows the results using Fourier transform. Table 4 is the corresponding results using statistical moments. Throughout all the different conditions of ball bearing, Fourier transform seems to be more sensitive to detect any anomaly than statistical moments. Both methods detect better when the damage is more severe.

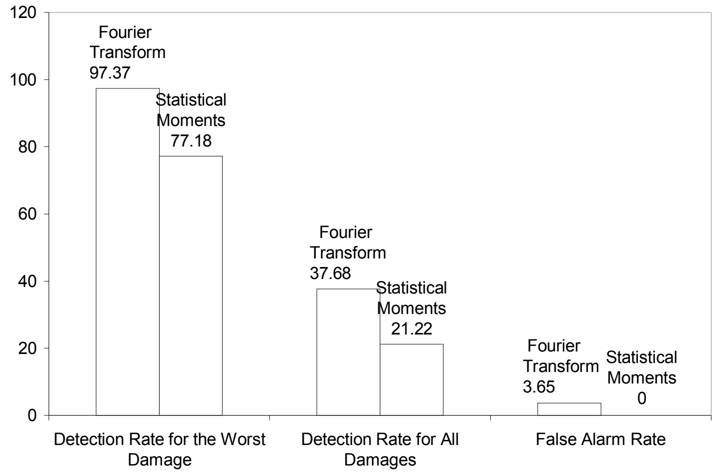


Fig. 7. Summary of Detection Results on Ball Bearing Data

Figure 7 summarizes the performance in terms of detection rate and false alarm. Detection rates are evaluated on two different assumptions: first, only the worst damage (broken race) is considered as fault to be detected; second, all the three types of damages or breakings are regarded as real fault. For complete new ball bearing, there is no false alarm. Assuming the last type of condition (no evident damage) does not count as fault to be detected, false alarm rate is also plotted in figure 7.

4 Conclusion

The paper proposed an extension of real-valued negative selection algorithm with a variable coverage detector generation scheme. Experimental results demonstrated that *V-detector* scheme is more effective in using smaller number of detectors because of their variable sizes. Moreover, it provides a more concise representation of the anomaly detectors derived from the normal data. The detector set generated by *V-detector* is more reliable because the estimated coverage instead of the arbitrary number of detectors is obtained by the algorithm at the end of the run.

The following are some advantages of *V-detector* algorithm:

- Time to generate detectors and to examine new samples is saved by using smaller number of detectors. It also requires less space to store them.
- Holes can be better covered. The smaller detectors are more acceptable because fewer detectors are used to cover the large non-self region.
- The coverage estimate is very useful to provide prediction of the algorithmic performance even if the detection rate (for some specific cases) is not very high due to incomplete or noisy data.

The influence of estimated coverage as a control parameter needs further study, including more experiments and formal analysis. The implication of self radius, or how to interpret each self sample (training data), is also an important topic to be explored. Future works along the line of variable detectors will be variable shape of detectors, variable number of dimensions, etc. It also has potential use for the problems that have very high dimensions but where only a few dimensions affect the detection process. Limited number of detector dimensions has additional benefit of extracting knowledge or rules in a more comprehensible form.

Acknowledgement: This work was supported in part by NIH Cancer Center Support Core Grant CA-21765 and the American Lebanese Syrian Associated Charities (ALSAC).

References

1. de Castro, L. N., et al, Artificial Immune System: A New Computational Intelligence Approach, Springer-Verlag. 2002
2. Dasgupta, D., et al, Artificial Immune System (AIS) Research in the Last Five Years, IEEE Congress of Evolutionary Computation (CEC), Canberra, Australia, 2003
3. Hofmeyr, S., and S. Forrest, Architecture for an artificial immune system, Evolutional Computation Journal, vol. 8, no.4, 2000
4. de Castro, L., N., J. I. Timmis, Artificial Immune Systems as a Novel Soft Computing Paradigm, Soft Computing Journal, vol. 7, issue 7, 2003
5. Dasgupta, D., et al, An Anomaly Detection Algorithm Inspired by the Immune System, in *Artificial Immune System and Their Application*, ed. by D. Dasgupta et al, 1999
6. Esponda, F., S. Forrest, P. Helman, A Formal Framework for Positive and Negative Detection Scheme, IEEE Transaction on Systems, Man, and Cybernetics, 2003
7. Ayara, M., J. Timmis, R. de Lemos, L. de Castro, and R. Duncan, Negative Selection: How to Generate Detectors, 1st International Conference on Artificial Immune System (ICARIS), UK, 2002
8. Gonzalez, F., D. Dasgupta, J. Gomez, The Effect of Binary Matching Rules in Negative Selection, Genetic and Evolutionary Computation Conference (GECCO), Chicago, 2003
9. Gonzalez, F., D. Dasgupta, L. F. Nino, A Randomized Rea-Valued Negative Selection Algorithm, 2nd International Conference on Artificial Immune System (ICARIS), UK. 2003
10. Gonzalez, F., D. Dasgupta, Anomaly Detection Using Real-Valued Negative Selection, Genetic Programming and Evolvable Machine, vol. 4. pp. 383-403, 2003
11. Ceong, H. T., et al, Complementary Dual Detectors for Effective Classification, 2nd International Conference on Artificial Immune System (ICARIS), UK, 2003
12. Kim, J., et al, An evaluation of negative selection in an artificial immune system for network intrusion detection, in Proceedings Genetic and Evolutionary Computation Conference (GECCO), San Francisco, 2001
13. Dasgupta, D., et al, MILA – Multilevel Immune Learning Algorithm, Genetic and Evolutionary Computation Conference (GECCO), Chicago, 2003
14. Ji, Z., Multilevel Negative/Positive Selection in Real-Valued Space, Research Report, The University of Memphis, December 21st, 2003
15. StatLib – Datasets Archive, {http://lib.stat.cmu.edu//dataset/}
16. Structural Integrity and Damage Assessment Network, Public Datasets, {www.brunel.ac.uk/research/cnca/sida/html/data.html}
17. Paul Bourke, Analysis {http://astronomy.swin.edu.au/~pbourke/analysis/}
18. Interstellar Research, FFT Windowing {http://www.daqarta.com/ww00wndo.htm}
19. Institute for Communications Engineering, Higher-order Statistical Moments, {http://speedy.et.unibw-muenchen.de/forsch/ut/moment/}