Comparative Performance Analysis of Negative Selection Algorithm with Immune and Classification Algorithms

Ayodele Lasisi[1], Rozaida Ghazali[1], and Tutut Herawan[2]

[1] Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia
86400, Parit Raja, Batu Pahat, Johor, Malaysia
lasisiayodele@yahoo.com, rozaida@uthm.edu.my

[2] Faculty of Computer Science and Information Technology
University of Malaya, 50603, Kuala Lumpur, Malaysia
tutut@um.edu.my

Abstract. The ability of Negative Selection Algorithm (NSA) to solve a number of anomaly detection problems has proved to be effective. This paper thus presents an experimental study of negative selection algorithm with some classification algorithms. The purpose is to ascertain their efficiency rates in accurately detecting abnormalities in a system when tested with well-known datasets. Negative selection algorithm with some selected immune and classifier algorithms are used for experimentation and analysis. Three different datasets have been acquired for this task and a comparison performance executed. The empirical results illustrates that the artificial immune system of negative selection algorithm can achieve highest detection and lowest false alarm. Thus, it signifies the suitability and potentiality of NSA for discovering unusual changes in normal behavioral flow.

Keywords: anomaly detection, classification algorithm, data representation, negative selection algorithm.

1 Introduction

The emergence of Artificial Immune System (AIS) which began with the works of Forrest and her group [1] in 1994 by proposing and developing the Negative Selection Algorithm (NSA) opened the door as an important addition within the confines of anomaly detection. The biological process of *negative selection*, which laid the foundation for the abstraction of NSA algorithm, is attributed to specialized white blood cells, called T-cells developed in the bone marrow inhibiting receptors that undergoe a pseudo-random generation procedure. These T-cells are further exposed to $self$ cells in the thymus, and there is an elimination action taken when a reaction occurs between the the T-cells and the $self$ cells with those not reacting being retained and granted permission to leave the

T. Herawan et al. (eds.), *Recent Advances on Soft Computing and Data Mining SCDM 2014*, Advances in Intelligent Systems and Computing 287,
DOI: 10.1007/978-3-319-07692-8_42, © Springer International Publishing Switzerland 2014

thymus into maturation stage. At this stage, they are now fully integrated in the immune system surveillance structure of intuders into the body.

The task of anomaly detection, a two-class classification problem that classifies an element as *normal* or *abnormal* within a given feature space, can be considered as analogous to the immunity of biological immune system [2]. Their targeted aim is to detect abnormal behaviours of system that contradicts to the normal functioning of the system [3, 4]. Varieties of classification-anomaly based techniques exist in literature [5] ranging from neural-network based to rule-based. These classification-anomaly based methods establishes a balanced platform for comparison with the immune algorithm of negative selection algorithm, clonal selection algorithm, artificial immune recognition system (airs), and Immunos-81 variants. The purpose of venturing into algorithmic comparison as will be highlighted in this paper, is to weigh the performances in terms of detection rate and false alarm rate, which will in turn give us a clearer picture of their capacities when anomaly detection is concerned.

The structure of this paper is as follows. In Section 2, an overview of what anomaly detection is all about is presented. This is followed by a brief insight into classification algorithms in Section 3. Negative Selection Algorithm and further exploration of the two data representation types constitutes Section 4. Our experimental results and discussions of these results sums up Section 5. The paper concludes with Section 6.

2 Anomaly Detection

In comprehending the concept behind anomaly detection, the term *anomaly* must be clearly defined. An anomaly, also referred to or used interchangeably depending on the application area as outliers, aberrations, exceptions, or peculiarities, is defined as patterns behaving differently to the normal behavioral flow [5]. Three basic types of anomalies exist namely, point anomalies (single data instance deviation), collective anomalies (deviation of group data instances), and contextual anomalies (context of deviation occurrence). Thus, anomaly detection is the process of identifying or recognizing abnormal behavioral changes in data [5, 6]. It will be of interest to know that anomaly detection has been in existence since the nineteenth century [7]. Improvement and advances in technological approach to effectively detect anomalies has been the distinguishing factor, and this is echoed by Vasarhelyi and Issa [8]. Computer security, medical, computer vision, general purpose data analysis and mining, and sensor network are some of the application areas of anomaly detection [9].

An anomaly detection method is considered good irrespective of the domain, based on the following three criteria [10]: (1) accurately locating and differentiating anomalies from normal behavior, (2) robustness in terms of sensitivity to parameter settings and changes of patterns in datasets, and (3) limited resources required. The training data to be used for modeling the system is of great importance as it aids in selecting the appropriate anomaly detection techniques which are supervised, semi-supervised, and unsupervised learning techniques.

Two phases, training phase and testing phase, make up the supervised technique where both malicious and benign data are applied. The semi-supervised technique on the other hand requires only the normal data with the usage of a training phase. However, unsupervised technique is devoid of a training phase with all the data present prior to initializing the algorithm [11].

3 Classification Algorithm

Classification is the process of generalizing data according to different instances [12] and predicting a certain outcome based on a well-known given input [13]. Various classification techniques are available for solving classification problems ranging from statistical methods, decision trees, neural networks, rule-based methods [14, 15]. The training data serves as input for the classification and rule based algorithm to begin their tasks for which the target values are known. With each of the algorithms having different strengths and weaknesses, they possess the ability to find relations between the predictor attributes' values and target attributes' values in the training data [16]. The selected algorithms for use in this paper are the popularly known and a few artificial immune system classifiers which will be brought to light in the later section of this paper.

4 Negative Selection Algorithm

The human immune system process called *negative selection* gave rise to one of the earliest algorithms in the artificial imunne system domain. The theoretical concept of negative selection is well rooted in central tolerance, an immune mechanism for self-tolerance averting autoimmunity (reacting to self antigens) [17]. T-cells, a special kind of white blood cell called lymphocytes, are generated from the bone marrow and equipped with receptors which takes upon themselves the task of identifying and recognizing specific molecular patterns. The receptors of T-cells are generated in a pseudo-random manner, and are exposed to normal proteins which resides in the thymus of the host body. The reaction of the T-cells with the proteins causes an elimination of the T-cells, and only those which do not react are allowed to migrate from the thymus, causing them to mature and be fully integrated in the immune system [18]. Based on the negative selection principle, Forrest et al. [1] proposed and developed the Negative Selection Algorithm (NSA) for detection applications. Two data representations of NSA are the strings (or binary) negative selection algorithm, and real-valued negative selection algorithm [19].

4.1 Strings Representation of Negative Selection Algorithm

The processes of negative selection made it well suited for computer and network security, and in 1994, [1] proposed a Negative Selection Algorithm (NSA) for discriminating between self and non-self in a computer. It can be applied to virus detection, image inspection and segmentation, and also hardware tolerance [20]. Steps in NSA execution is summarized as follows [21]:

Given a universe U which contains all unique bit-strings of length l, self set $S \subset U$ and non-self set $N \subset U$, where

$$U = S \subset U \qquad \text{and} \qquad S \cap N = \emptyset$$

1. Define self as a set S of bit-strings of length l in U.
2. Generate a set D of detectors, such that each fails to match any bit-string in S.
3. Monitor S for changes by continually matching the detectors in D against S.

Since the first implementation of NSA which uses the Exhaustive Detector Generating Algorithm (EDGA) incorporated into the works of Forrest et al. [1], different variations of the algorithm using strings representation have been reported in literature [22]. This stems from limitations of the original NSA, as the time in generating valid detectors increases exponentially with the size of the self strings (time and space complexity) [23]. Thus, larger number of detectors are been generated. As a measure against the time and space consuming factors, D'Haeseleer et al. [24] developed the linear and greedy detector generation algorithms, with both operating in linear time with respect to the size of the self and detector sets, and greedy algorithm was reported to dissolve the problem with a new collection of generated detectors. Also, Wierzchon [25] introduced binary template with no intention of decreasing the time but rather generating efficient non-redundant detectors. NSMutation proposed in [26] is a modified version of EDGA using somatic hypermutation, and Ayara et al. [27] made a performance comparison of the different strings detector generation algorithms and results showed that NSMutation is more extensible. Still, the strings representation suffers greatly in dealing with real world applications which basically inherits the use of real-valued data.

4.2 Real-Valued Representation of Negative Selection Algorithm

The proposition by Gonzalez et al. [15] employs the use of real-valued data as against strings representation, in an effort to deal with the issues posed by strings (or binary) negative selection algorithm, and termed it Real-Valued Negative Selection Algorithm (RNSA). The algorithm distributes the detectors in the $nonself$ space based on heuristic to optimally maximize the coverage area. Among the advantages of real-valued representation which is a high level representation stated in [28, 29] are increased expressiveness, possibility of extracting high-level knowledge from the generated detectors, and improved scalability in certain cases. The algorithm adopts n-dimensional vectors in real space $[0, 1]^n$ to encode antigens and antibodies, and Euclidean distance to calculate the affinities between them.

Although the RNSA is characterized by the three basic steps in [21], the algorithmic description with additional components in evolving the detectors to cover the $nonself$ space is found in [15]. This is performed through an iterative process with the aim of:

- Moving the detectors away from the $self$ set, and
- Maximizing the coverage space of $nonself$ by keeping the detectors apart

The detectors of the RNSA is fixed and chosen beforehand, and in an effort to dynamically choose the detectors, [30] proposed an improved version of RNSA called Variable-Sized Detectors (V-Detectors). The detectors of V-Detectors terminates training stage when enough coverage has been achieved. For the purpose of study in this paper, the RNSA [15] will be used to have a balanced performance comparison with the chosen classification algorithms.

5 Experimental Results and Analysis

Experiments are performed to compare the performance of Negative Selection Algorithm with some classification algorithms. These algorithms are implemented using both MATrix LABoratory (MATLAB) and Waikato Environment for Knowledge Analysis (WEKA). The selected classifiers from WEKA toolbox are the immune algorithms namely AIRS1 [31, 32], AIRS2 [31, 33], AIRS2Parallel [31, 34], CLONALG [35], Immunos1 [36], Immunos2 [36], and Immunos99 [36]. Selecting from different categories, the standard classification algorithms adopted for use in the experimental study includes Naive Bayes (NB) from bayesian category, Multilayerperceptron (MLP) from neural network, Sequential Minimal Optimization (SMO) from support vector machines category, IBk from instance-based category, J48 from decision tree, and NNge from nearest neighbour category. Dataset have been retrieved from UCI Machine Learning Repository and are all real-valued data which suites well for implementation with negative selection algorithm, and they are Fisher's IRIS data, Balance-Scale (BS), and Lenses data.

The Fisher's IRIS data has largely been employed for use in discriminant analysis and cluster analysis. It is composed of three species of 50 samples each, $Iris\ Setosa$, $Iris\ Versicolor$, and $Iris\ Virginica$ with four numeric features, *sepal length*, *sepal width*, *petal length*, and *petal width*. These features are measured in millimeters within an entire searching space of 4-dimensional hypercube $[0, 1]^4$. The Balance-Scale on the otherhand has 625 data instances with three classes of balance-scale $tip\ to\ the\ left$, $tip\ to\ the\ right$, and *balanced*, while Lenses comprises of 24 data instances and three classes as well. The searching space of both Balance-Scale and Lenses is a 4-dimensional hypercube $[0, 1]^4$.

In other to pass the datasets as input for NSA execution in MATLAB, each class is employed as the training data while the other classes becomes the testing data used to measure the performance of the detectors. For example, one of the species of the Fisher's IRIS data serves as training set, and the other two species are for testing. This process is performed for each class, and the Euclidean distance in (1) is integrated to measure the affinities between the detectors and real-valued coordinates.

$$D = \sqrt{\sum_{i=1}^{n} (d_i - x_i)^2} \tag{1}$$

where $d = d_1, d_2, \ldots, d_n$ are the detectors, $x = x_1, x_2, \ldots, x_n$ are the real-valued coordinates, and D is the distance. The parameters used by real-valued negative selection algorithm are: $r = 0.1$, $\eta_o = 0.005$, $t = 15$, and $\tau = 15$. The number of randomly generated detectors is 1000, and experiments were repeated 10 times with the average values recorded. The above parameters denotes:

- r: radius of detection
- η_o: initial value of the adaptation rate
- t: age of the detector
- τ: the decay rate

To assess the performance of the immune and classification algorithms in WEKA, a 10-fold cross validation is used for testing and evaluating. This process entails dividing the dataset into ten subsets of equal size, with nine subsets making up the training data, leaving the only remaining subset as the test data. Performance statistics are calculated across all 10 trials, and this provide a good platform to ascertain how well the classifiers perform on the various dataset.

5.1 Performance Metric Terms

As a measure for balanced performance comparison between real-valued negative selection algorithm, immune algorithms, and classification algorithms, the detection rate and false alarm rate described in (2) and (3) are used for evaluation.

$$DR = \frac{TP}{TP + FN} \tag{2}$$

$$FAR = \frac{FP}{FP + TN} \tag{3}$$

where TP is the number of $nonself$ elements identified as $nonself$; TN is the number of $self$ elements identified as $self$; FP is the number of $self$ elements identified as $nonself$; FN is the number of $nonself$ elements identified as $self$; DR is the detection rate; FAR is the false alarm rate.

5.2 Simulation Results

The simulation experiments were performed on 2.10 GHz Intel Pentium (R) Processor with 4GB of RAM. As earlier mentioned that MATLAB and WEKA

Table 1. Fisher's IRIS dataset performance result

Algorithm	Detection Rate %	False Alarm Rate %
AIRS1	95.33	2.3
AIRS2	95.33	2.3
AIRS2Parallel	96.0	2.0
CLONALG	95.33	2.3
Immunos1	97.33	1.3
Immunos2	97.33	1.3
Immunos99	96.67	1.7
Naive Bayes	96.0	2.0
Multilayerperceptron	97.33	1.3
SMO	96.0	2.0
IBk	95.33	2.3
J48	96.0	2.0
NNge	96.0	2.0
NSA	**96.73**	**0.53**

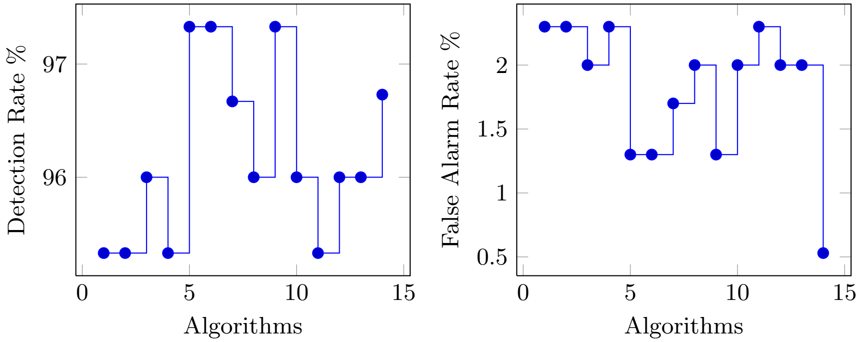

Fig. 1. Graph Plots for Detection Rates and False Alarm Rates (Fisher's IRIS)

have been adopted for use in the performance comparison of the algorithms. The results after series of experiments for each dataset is tabulated and graphed.

In the graph illustrations, the algorithms have been labelled from 1 to 14 with AIRS1 representing the least, followed by AIRS2, while NSA connotes the highest in the order shown from the tables. In Table 1, it can be revealed that Negative Selection Algorithm performed considerably well when compared with the other selected algorithms on the Fisher's IRIS dataset. With a detection rate of 96.73%, only Immunos1, Immunos2, and MLP with 97.33% each were better in performance. With respect to false alarm rate, NSA shows to be more effective with 0.53%. AIRS1, AIRS2, CLONALG, and IBk each produced the highest false alarm rate of 2.3%. The graph representation of the above explanation is reflected in Figure 1 for both the detection rate and false alarm rate.

The overall performance of the Balance-Scale dataset is presented in Table 2 below. The values of the respective algorithms have been adequately plotted in a graph as depicted in Figure 2. The superiority of NSA with detection rate of 98.02% can be confirmed as against the immune and classification algorithms. Majority of the algorithms has detection rate which falls within the 80% range, with only Naive Bayes and MLP that could boast of reaching the 90% range having values of 90.4% and 90.72% respectively. The lower false alarm rate reported by NSA which equals 0.99% superceeds the higher false alarm rate generated by others, with CLONALG having the highest rate of 19.6%.

Table 2. Balance-Scale dataset performance result

Algorithm	**Detection Rate %**	**False Alarm Rate %**
AIRS1	80.48	11.9
AIRS2	80.96	12.2
AIRS2Parallel	81.76	11.5
CLONALG	75.2	19.6
Immunos1	71.84	3.1
Immunos2	86.72	11.3
Immunos99	69.12	5.2
Naive Bayes	90.4	8.2
Multilayerperceptron	90.72	4.2
SMO	87.68	10.5
IBk	86.56	9.5
J48	76.64	17.3
NNge	81.92	10.8
NSA	**98.02**	**0.99**

For Lenses dataset presented in Table 3 and diagrammatically shown in Figure 3, when tested with NSA, yielded a 100% detection rate, and this proves to outclass the other algorithms in which SMO could only attain 87.5% rate. In the same vein, other algorithms produced higher false alarm rate as against low rate of NSA with 1.22% rate. In addition, when compared to the other datasets, lenses data gave the lowest detection rate of 45.83% generated by Immunos99. Following the step laid down by the detection rate, lenses data also produced the highest false alarm rate of 58.3% to the experimentation with other datasets.

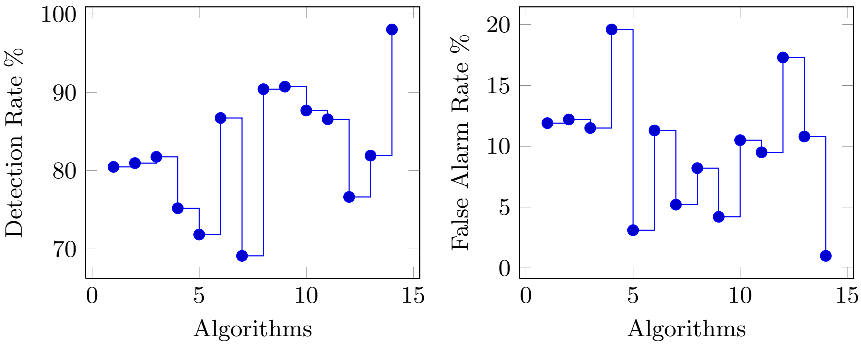

Fig. 2. Graph Plots for Detection Rates and False Alarm Rates (Balance-Scale)

Table 3. Lenses' dataset performance result

Algorithm	**Detection Rate %**	**False Alarm Rate %**
AIRS1	70.83	21.9
AIRS2	79.17	19.7
AIRS2Parallel	75.0	25.5
CLONALG	54.17	35.7
Immunos1	70.83	12.4
Immunos2	58.33	58.3
Immunos99	45.83	42.7
Naive Bayes	79.17	24.4
Multilayerperceptron	70.83	21.9
SMO	87.5	12.8
IBk	79.17	15.0
J48	66.67	23.0
NNge	70.83	21.9
NSA	**100.0**	**1.22**

Therefore, it can be said that NSA is well suited for anomaly detection which rest solely the idea as proposed by Forrest et al. [1], with severals experimental procedures reported in literatures over the years since its inception.

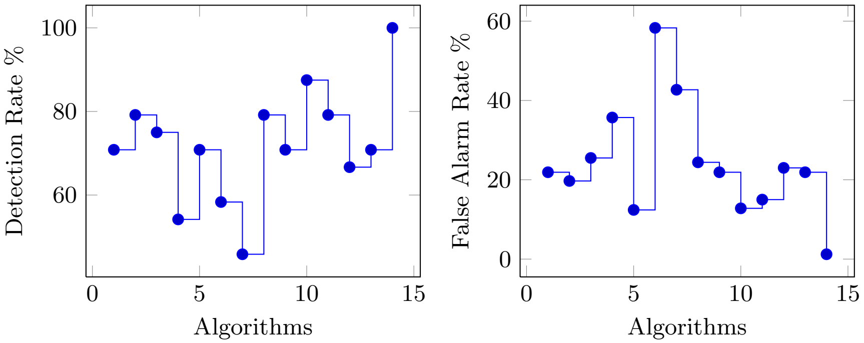

Fig. 3. Graph Plots for Detection Rates and False Alarm Rates (Lenses)

6 Conclusion

A comparative experimental study constitutes the backbone of this paper. Careful selection of classification-based anomaly techniques channel our objective for a proper comparison performance with immune algorithms. The focus is to determine how potent the algorithms could achieve their task when fed with data. The UCI repository provides with standard and benchmarked dataset, and three(3) of those dataset were used for this study. Experiments were carried out on a total of 14 classification and immune algorithms namely AIRS1, AIRS2, AIRS2Parallel, CLONALG, Immunos1, Immunos2, Immunos99, Naive Bayes, Multilayerperceptron, SMO, IBk, J48, NNge, and NSA for each dataset. The negative selection algorithm performed better than all the algorithms for two of the datasets with respect to rate of detection and false alarm, generating values of 98.02% and 0.99% for balance-scale data, also 100% and 1.22% for lenses data. On the remaining dataset (Fisher' IRIS), NSA proved its mettle with a detection rate of 96.73% (except for Immunos1,Immunos2, and MLP with 97.33%). With the verification of the anomaly detection potentials of NSA as reported in this study, and also with numerous improvements at enhancing its recognition qualities, further research will be directed at methods for boosting NSA algorithm.

Acknowledgements. As an expression of gratitude, the authors with deepest appreciation, would like to thank the Office for Research, Innovation, Commercialization and Consultancy Management (ORICC), Universiti Tun Hussein Onn Malaysia (UTHM) and Ministry of Higher Education (MOHE) for financially supporting this Research under the Exploratory Research Grant Scheme (ERGS), Vote No. 0882.

References

1. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy, pp. 202–212. IEEE (1994)
2. Patcha, A., Park, J.M.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks 51(12), 3448–3470 (2007)
3. Boukerche, A., Machado, R.B., Jucá, K.R., Sobral, J.B.M., Notare, M.S.: An agent based and biological inspired real-time intrusion detection and security model for computer network operations. Computer Communications 30(13), 2649–2660 (2007)
4. Dasgupta, D., González, F.: An immunity-based technique to characterize intrusions in computer networks. IEEE Transactions on Evolutionary Computation 6(3), 281–291 (2002)
5. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys (CSUR) 41(3), 15 (2009)
6. Tamberi, F.: Anomaly detection (2007)
7. Edgeworth, F.: On discordant observations. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 23(143), 364–375 (1887)
8. Vasarhelyi, M.A., Issa, H.: Application of anomaly detection techniques to identify fraudulent refunds (2011)
9. Song, X., Wu, M., Jermaine, C., Ranka, S.: Conditional anomaly detection. IEEE Transactions on Knowledge and Data Engineering 19(5), 631–645 (2007)
10. Yao, Y., Sharma, A., Golubchik, L., Govindan, R.: Online anomaly detection for sensor systems: A simple and efficient approach. Performance Evaluation 67(11), 1059–1075 (2010)
11. Amer, M., Abdennadher, S.: Comparison of unsupervised anomaly detection techniques. PhD thesis, Bachelor's Thesis 2011 (2011), `http://www.madm.eu/_media/theses/thesis-amer.pdf`
12. Kumar, R., Verma, R.: Classification algorithms for data mining: A survey. International Journal of Innovations in Engineering and Technology (IJIET) (2012)
13. Kilany, R.M.: Efficient classification and prediction algorithms for biomedical information (2013)
14. Weiss, S.M., Kulikowski, C.A.: Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning and expert systems (1991)
15. Gonzalez, F., Dasgupta, D., Kozma, R.: Combining negative selection and classification techniques for anomaly detection. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002, vol. 1, pp. 705–710. IEEE (2002)
16. Rao, K.H., Srinivas, G., Damodhar, A., Krishna, M.V.: Implementation of anomaly detection technique using machine learning algorithms. International Journal of Computer Science and Telecommunications, ISSN 2047–3338
17. Lederberg, J.: Genes and antibodies do antigens bear instructions for antibody specificity or do they select cell lines that arise by mutation? Science 129(3364), 1649–1653 (1959)
18. Textor, J.: A comparative study of negative selection based anomaly detection in sequence data. In: Coello Coello, C.A., Greensmith, J., Krasnogor, N., Liò, P., Nicosia, G., Pavone, M. (eds.) ICARIS 2012. LNCS, vol. 7597, pp. 28–41. Springer, Heidelberg (2012)

19. Lasisi, A., Ghazali, R., Herawan, T.: Negative selection algorithm: A survey on the epistemology of generating detectors. In: Herawan, T., Deris, M.M., Abawajy, J. (eds.) Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng 2013). LNEE, vol. 285, pp. 167–176. Springer, Heidelberg (2014)
20. Hofmeyr, S.A., Forrest, S.: Architecture for an artificial immune system. Evolutionary Computation 8(4), 443–473 (2000)
21. Stibor, T., Timmis, J., Eckert, C.: The link between r-contiguous detectors and k-cnf satisfiability. In: IEEE Congress on Evolutionary Computation, CEC 2006, pp. 491–498. IEEE (2006)
22. D'Haeseleer, P., Forrest, S., et al.: An immunological approach to change detection. In: Proc. of IEEE Symposium on Research in Security and Privacy, Oakland, CA (1996)
23. Majd, Mahshid, A.H., Hashemi, S.: A polymorphic convex hull scheme for negative selection algorithms. International Journal of Innovative Computing, Information and Control 8(5A), 2953–2964 (2012)
24. D'Haeseleer, P., Forrest, S., Helman, P.: An immunological approach to change detection: Algorithms, analysis and implications. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy, pp. 110–119. IEEE (1996)
25. Wierzchon, S.T.: Discriminative power of the receptors activated by k-contiguous bits rule. Journal of Computer Science & Technology 1(3), 1–13 (2000)
26. de Castro, L.N., Timmis, J.: Artificial immune systems: a new computational intelligence approach. Springer (2002)
27. Ayara, M., Timmis, J., de Lemos, R., de Castro, L.N., Duncan, R.: Negative selection: How to generate detectors. In: Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS), Canterbury, UK:[sn], vol. 1, pp. 89–98 (2002)
28. González, F.A., Dasgupta, D.: Anomaly detection using real-valued negative selection. Genetic Programming and Evolvable Machines 4(4), 383–403 (2003)
29. Gonzalez, F., Dasgupta, D.: Neuro-immune and self-organizing map approaches to anomaly detection: A comparison. In: First International Conference on Artificial Immune Systems, pp. 203–211 (2002)
30. Ji, Z., Dasgupta, D.: Real-valued negative selection algorithm with variable-sized detectors. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 287–298. Springer, Heidelberg (2004)
31. Brownlee, J.: Artificial immune recognition system (airs)-a review and analysis. Swinburne University of Technology, Melbourne, Australia. Tech. Rep. (1-02) (2005)
32. Watkins, A., Timmis, J., Boggess, L.: Artificial immune recognition system (airs): An immune-inspired supervised learning algorithm. Genetic Programming and Evolvable Machines 5(3), 291–317 (2004)
33. Watkins, A., Timmis, J.: Artificial immune recognition system (airs): Revisions and refinements. In: AISB 2004 Convention, p. 18 (2002)
34. Watkins, A., Timmis, J.: Exploiting parallelism inherent in AIRS, an artificial immune classifier. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 427–438. Springer, Heidelberg (2004)
35. De Castro, L.N., Von Zuben, F.J.: Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation 6(3), 239–251 (2002)
36. Brownlee, J.: Immunos-81 the misunderstood artificial immune system, ciscp, faculty of ict, swinburne university of technology. Technical report, Australia, Technical Report 1-02 (2005)