Negative Selection Algorithm: A Survey on the Epistemology of Generating Detectors

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. Within the Artificial Immune System community, the most widely implemented algorithm is the Negative Selection Algorithm. Its performance rest solely on the interaction between the detector generation algorithm and matching technique adopted for use. Relying on the type of data representation, either for strings or real-valued, the proper detection algorithm must be assigned. Thus, the detectors are allowed to efficaciously cover the non-self space with small number of detectors. In this paper, the different categories of detection generation algorithm and matching rule have been presented. Briefly, the biologial and artificial immune system, as well as the theory of negative selection algorithm were introduced. The exhaustive detector generation algorithm used in the original Negative Selection Algorithm laid the foundation at profering other algorithmic methods based on set of rules in generating valid detectors for revealing anomalies.

Keywords: negative selection algorithm, data representation, detector generation algorithm, matching rule

1 Introduction

Negative Selection Algorithm (NSA), a significant set of rules within Artificial Immune System (AIS), announces its presence in the computer security domain. It is especially directed for use in anomaly detection and change detection. The process imitates T-cells detection execution of foreign invaders in the human body. In other to attain its target of recognition and elimination accordingly, the detector generation algorithm and matching technique plays a crucial role. Depending on the data representation, strings (binary) or real-valued represenation, the appropriate detector generation algorithm and matching rule will be utilized. Forrest et al. [1] used the exhaustive detector generation algorithm based on r-contiguous matching rule. The drawback is that it is a costly computation in terms of time and space. Some other improved detector generation

T. Herawan et al. (eds.), *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, Lecture Notes in Electrical Engineering 285, DOI: 10.1007/978-981-4585-18-7_20,

algorithms were proposed, these are linear-time, binary, greedy, and Negative Selection using Mutation (NSMutation) detector generation algorithms. Also different matching algorithms have been introduced as well. These are related to binary string representation. The detector generation schemes are different for real-valued representation. This goes for the matching techniques too. More information shall be presented in the latter section of this paper.

This paper reports and focus on the detector generation methods of negative selection algorithm. The different matching algorithms were discussed. It examines in detail the process of generating detectors from the original method to the various improvements carried out by researchers. These are geared at reducing the number of detectors to its minimum moving a step higher than the previous methods. It is believed that this review will further enhance our understanding and knowledge behind the detector generation process of NSA. The arrangement of the paper is organized as follows: Section 2 introduces biological immune system. Section 3 briefly describes artificial immune system. The concept of negative selection algorithm is mentioned in section 4. Afterwards, data representation and matching techniques occupies section 5, while detector generation algorithms used in NSA are elaborated in section 6. Conclusion sums up the paper in section 7.

2 Biological Immune System

The Biological Immune System (BIS), an integral part of the vertebrate immunity over centuries, is a dynamic, powerful, intelligent, and interconnection of different components of the body, working in totality to fight, defend, and prevent pathogenic organisms' entrance into the body. The postulation of immunological concept of the body mechanisms defending against pathogens through immunoglobulin called antibodies gave birth to immunity [2]. Two major functions attributed to BIS are: protection from foriegn invaders, and maintaining homeostasis [3, 4]. However, there are still inquiries by immunologist about the precise function of the immune system because of its sophisticated nature. As such, it goes to show that the immune system has inherent capabilities which surpasses what is being obtained now (i.e innate and adaptive system; and humoral and cellular processes) in guarding the body from pathogen invasion [5]. Boukerche et al. [6] gave the properties of immune systems as detection, diversity, learning, and tolerance. The immune system utilizes two lines of defenses known as the innate immune system and adaptive immune system [7, 8]. Innate immunity is the first line defense and its non-specific. Non-specific in that, it does not concentrate on a particular kind of pathogen. Adaptive immunity on the other hand handles such invasions that bypass the innate immunity line of defense. It is specific as it targets, matches a particular pathogen, and stores in memory the structure of that pathogen for faster detection and elimination if encountered again.

3 Artificial Immune System

Artificial Immune System (AIS) is a computational paradigm that has evolved over the past two decades with algorithms developments mimicking the immune system processes. It connects and fosters the immunology, computer science, and engineering disciplines [9, 10]. Theoretical immunology models the immune system for in-depth knowledge of its behaviour [11]. Coupling the theoretical immunology with observed immune functions, principles and models introduces AIS as being able to cope effectively with changing situations and also suitable for problem solving [12]. Such problem solving tasks include but not limited to pattern recognition, learning, memory acquisition, distributed detection, and optimization [13]. Among AIS researchers, three (3) definitions have gained popularity, and only one of these is widely accepted as defined in [12].

Undoubtedly, the works of Bersini and Varela [14], pioneer in the use of metaphor for immune network theory, and Forrest et al. [1] announces the pathway from immunology to computing. The immune network theory was the focus of Hugues Bersini and Francisco Varela, abstracted from the way the natural immune network memorizes and functions leading to models and algorithms. Collaboration with researchers of the same field of interest popularized the concept [14, 15]. Self-non-self discrimination as it applies to computer security was the intention of Stephanie Forrest and her colleagues [1]. They were inspired at how the immune system recognizes self (normal) from non-self (abnormal) and a Negative Selection Algorithm (NSA) was proposed, thus becoming the pioneers of AIS algorithm development. It sets in motion the modeling and development of immune functions and properties of a number of AIS algorithms. A detailed review on theories and algorithms of artificial immune system can be found in [10].

4 Negative Selection Algorithm

In the biological immune system, there exist cells responsible for battling and annihilating intoxicated foreign molecules which are harmful to the body. T-cells, a special kind of white blood cell called lymphocytes, falls under the umbrella of the protecting cells. The receptors of T-cells are generated in a pseudo-random manner which undergoes a censoring process in the thymus called negative selection. In the thymus, the T-cells reacting to self cells are terminated while those not reacting leave the thymus into maturation stage. At this stage, they are equipped with the full functionality of protecting the body. Based on the negative selection principle, Forrest et al. [1] proposed and developed the Negative Selection Algorithm (NSA) for detection applications. Two principal stages of the NSA are the generation stage and the detection stage. The production of detector set is carried out at the generation stage, and these detector set are now ultimately used for change detection. Steps in NSA execution is summarized as follows [16]:

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

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

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.

Clearly, it can be deduced from the algorithmic steps that a kind of matching rule is required, reflected in both stages of the algorithm. This matching rule is hinged to a data representation method, invariably working in togetherness for performing the change detection task. Thus, in the next section, we shall elaborate on the data representation and matching techniques used by NSA for generating detectors.

5 Data Representation and Matching Techniques

The success of the detector generation algorithms depends solely on how the data is being represented and the matching technique adopted. For negative selection algorithm, strings (or binary) representation and real-valued representation has been widely used. Also, there is hybrid of both data representations which consist of different data types such as integer, real value, categorical information, boolean value, text information, etc. String representation has proved advantageous owing to the fact that: 1) it can be eventually represented in binary form; 2) anaylzed easily; and 3) beneficial for either textual or categorical information [17]. However, it suffers from space complexity and scability issues [18]. As a result, real-valued representation emerged in dealing with real value data types and also being suitable for real world applications. While data represented in strings can be used with a variety of matching techniques, euclidean distance is the primary matching technique used for real-valued representation [19]. Other utilized techniques for representing real-valued data and those of string representation are discussed below.

5.1 Matching Rule for Strings Representation

***R*-Contiguous Matching Rule.** The interaction between antigens and antibodies needs a proper representation and there must be an affinity funtion. The r-contiguous matching rule was proposed by Percus et al. [20] in mapping antibodies to antigens, and matching process is defined as follow:

Suppose we denote antigens as set of binary strings $x = x_1, x_2, \ldots, x_n$ and antibodies denoted by a detector $d = d_1, d_2, \ldots, d_n$. This notation shall be used for the rest of this paper. The antibody matches the antigen if (1) holds:

$$\exists i \leq n - r + 1 \;\Big|\; x_j = d_j, \forall\ j = i, \ldots, i + r - 1 \tag{1}$$

where | denotes such that, and $\forall$ is for-all or for any.

The original NSA in [1] made use of the r-contiguous matching rule. With a pre-defined window size r, two binary strings are set to match if identical.

***R*-Chunk Matching Rule.** The r-contiguous matching rule described above, and the matching rules for classifier systems in [21] inspired r-chunk matching rule by Balthrop et al. [22]. The r-chunk rule encapsulate r-contiguous rule in that all the r-bits in the window must be matched with the binary strings. Therefore, any r-contiguous detectors can equally be represented as a set of r-chunk detectors. It is defined as follow:

Given detector $d = d_1, d_2, \ldots, d_m$ and binary strings $x = x_1, x_2, \ldots, x_n$ with $m \leq n$ and $i \leq n - m + 1$. The detector matches the binary strings if and only if (2) is satisfied:

$$x_j = d_j \;\forall\; j = i, \ldots, i + m - 1 \tag{2}$$

However, the distinguishing factor between r-contiguous and r-chunk matching rule is that full length r-contiguous bits develops crossover holes as well as length-limit holes, while r-chunk matching rule is devoid of this by eradicating the problem posed by length-limit holes.

Hamming Distance. Jerne [23] proposed a computational model based on idiotypic network theory which uses binary representation for the antibodies and antigens. Hamming distance is the matching rule implemented for this model. It is defined as follow:

Given detector $d = d_1, d_2, \ldots, d_n$ and binary strings $x = x_1, x_2, \ldots, x_n$, the detector matches binary strings if (3) is satisfied:

$$\sum_i \overline{x_1 \oplus d_i} \geq r \tag{3}$$

where $\oplus$ is the exclusive-or (XOR) operator, the line over $\overline{x_1 \oplus d_i}$ is the NOT operator, and $0 \leq r \leq n$ is a threshold value.

Additionally, variation of the Hamming distance known as Rogers and Tanimoto (R&T) matching rule was compared with several binary matching techniques and results shows it stands out to be the best [24]. This hamming distance has computational issues because it requires a huge number of steps in its execution. Thus, limits its application area.

5.2 Matching Rule for Real-Valued Representation

Euclidean Distance. This method of matching rule is widely incorporated for real-valued representation [19]. Inspite of its suitability for real valued cordinates, it yield undesirable results under large real-valued cases. Therefore, the best performance is achieved with limited real-valued cases [25]. Given the cordinates of detector $d = d_1, d_2, \ldots, d_n$ and binary strings cordinates as $x = x_1, x_2, \ldots, x_n$, the distance D existing between the detectors and binaries is shown in (4):

$$D = \sqrt{\sum_{i=1}^{n} (d_i - x_i)^2} \tag{4}$$

Manhattan Distance. This is an alternative distance measure to euclidean distance, also used for real-valued representation. It executes based on sum of the absolute value of the detectors and binary strings as against the square of the sum in euclidean distance. Given the cordinates of detector $d = d_1, d_2, \ldots, d_n$ and binary strings cordinates as $x = x_1, x_2, \ldots, x_n$, the distance D existing between them is shown in (5):

$$D = \sqrt{\sum_{i=1}^{n} |d_i - x_i|} \tag{5}$$

Minkwoski Distance. Minkwoski distance is an abstraction of the Euclidean distance and Manhattan distance [26], used by Dasgupta et al. [27] for aircraft fault detection. The distance D of the Minkwoski distance is defined as in (6):

$$D = \sqrt[h]{\sum_{i=1}^{n} |d_i - x_i|^h} \tag{6}$$

where h is real number such that $h \geq 1$. When $h = 1$, it represents the Manhattan distance; while for $h = 2$, Euclidean distance is being represented. Hamaker and Boggess [28] presented several other matching techniques which are useful for real-valued representation.

6 NSA Detector Generation Algorithms

Insight into the various detector generation algorithms with respect to the above matching mechanism for strings representation and real-valued representation shall be discussed. For string (or binary) representation, the Exhaustive Detector Generating Algorithm (EDGA) using the r-contiguous bits [20] was incorporated in the original work by Forrest et al. [1]. It imitates the T-cells generation processes of the BIS by random generation of detectors, and matching with self strings for creating a database of legitimate detectors to be used in detection purpose. Time complexity and space complexity need to be considered greatly in determining the degree at which detectors exert their authority. In other to ascertain the computational complexities of the original NSA, a mathematical expression was derived by D'Haeseleer et al. [29]. This, coupled with the experiments carried out by Ayara et al. [30] proves that it is computationally expensive as most randomly generated detectors are discarded.

Furthermore, a modified version to EDGA using somatic hypermutation was proposed in [12] called NSMutation. The proposition of other improved detector generation algorithm, the linear-time detector generating algorithm and greedy detector generating algorithm were reported in [31]. They are more deterministic as against the randomized method of exhaustive detector generating algorithm. The former has higher space complexity than EDGA, whereas for the latter, a higher time complexity is observed but demonstrate to have increased coverage area with limited number of detectors leading to higher detection rate [17]. Still on the deterministic approach, Wierzchon [32] introduced a binary template detector generating algorithm with increased efficiency which produces less number of detector to cover the search space. [30] went on to compare all the above detector generating algorithm and results shows that NSMutation is more extensible. Table 1 below shows the time complexity and space complexity of the above detector generation algorithms [29]. The terms used in the table denotes the following: l is the length of strings; r is the matching threshold; m, matching size; N_S is the population of self data; N_R, population of competent detectors.

Table 1. Complexities for detector generating algorithms based on strings (or binary) representation

Algorithm	**Time**	**Space**
Exhaustive	$O(m^l \cdot N_S)$	$O(l \cdot N_S)$
Linear	$O((l-r+1) \cdot N_S m^r) + O((l-r+1) \cdot m^r) + O(l \cdot N_R)$	$O((l-r+1)^2 \cdot m^r)$
Greedy	$O((l-r+1) \cdot N_S m^r) + O((l-r+1) \cdot m^r \cdot N_R)$	$O((l-r+1)^2 \cdot m^r)$
Binary Template	$O(m^r \cdot N_S) + O((l-r+1) \cdot m^r \cdot N_R)$	$O((l-r+1) \cdot m^r) + O(N_R)$
NSMutation	$O(m^l \cdot N_S) + O(N_R \cdot m^r) + O(N_R)$	$O(l(N_S + N_R))$

Moreover, for real-valued representation, euclidean distance has been the predominant matching rule used by detector generation algorithm [19, 33]. The detector generation scheme can be hyper-rectangle [34], hyper-sphere [35], multi-shape [36], and convex hull [18] based on research work at rightly representing the detectors. They gained attention due to problems that surfaced from binary representation and are deemed fit for solving real world problems. Its first representation was in the characterization of self and nonself space using genetic algorithm in evolving the detectors. These resulted into hyper-rectangles and called detector rules [34]. Thereafter, Real Valued Negative Selection (RNS) that uses a detector generation algorithm resting on the idea of heuristic was proposed [35]. As with the desired goal of other detectors, they tend to maximize the coverage of the non-self space. The matching technique used was fuzzy membership function and thus, the detectors were hyper-spheres.

Also, Gonzalez et al. [37] put forward a RandomizedRNS (RRNS) by replacing the heuristic method with Monte Carlo simulation method. To further proliferate the distribution of detectors in the non-self space, simulated annealing was employed. In multi-shaped detector generation scheme, a structured genetic algorithm was merged with various shapes of detectors. Monte Carlo estimation method evolves the detectors which adequately cover non-self space [36]. A variable detector method called V-detector used euclidean distance in matching and generating detectors. It shows to be efficient in terms of the number of detectors generated [19]. Taking advantage of pseudo-random detector generation algorithmic method, [18] proposed a Convex-Hull NSA (CH-NSA) using a matching algorithm specifying if a point is within the convex hull. It supports dissimilar anatomy of shapes, thus no special preference for any. Number of detectors generated are significantly less, yielding good performance with regards to coverage area.

Generally, the aim of researchers are directed towards the creation of small number of detectors that can competently cover the non-self space. Ma et al. [38] gave a guided rule in effectively generating detectors stated as: (1) generating detector covering the area of shape space, and (2) generating detectors that will be in the surrounding of the inhabitant within the shape space. Strictly adhering to this rule will increase accuracy and performance.

7 Conclusion

The overview of the detector generation algorithm as applied within NSA has been outlined and given light in this paper. While the algorithms have provided researchers with varying options based on data represenation, continual investigation at improving the existing ones marches on. The matching mechanism, integrated with the detector generation algorithm, signifies both as the major components for negative selection algorithm. The r-contiguous bit matching rule and euclidean distance have established themselves as the dominant forces for both string and real-valued representation respectively. Instructions leading to generating less number of detectors was provided, and altogether producing an increased performance. This will spur computer scientist at trying to have the least minimum detectors as possible. While proper recognition has been duly accorded due to the success rate of the detector generation mechanisms of NSA, further experimental investigations are needed at collapsing the detectors with minimum overlap so as to optimize its overall process.

References

1. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Research in Security and Privacy, 1994. Proceedings., 1994 IEEE Computer Society Symposium on, IEEE (1994) 202–212
2. Silverstein, A.M.: Paul ehrlich, archives and the history of immunology. Nature immunology **6**(7) (2005) 639–639
3. Immune, A.: Artificial immune systems. (2006) 107–118
4. Greensmith, J., Whitbrook, A., Aickelin, U.: Artificial immune systems. In: Handbook of Metaheuristics. Springer (2010) 421–448
5. Aickelin, U., Dasgupta, D.: Artificial immune systems. In: Search Methodologies. Springer (2005) 375–399

6. Boukerche, A., Jucá, K.R.L., Sobral, J.B., Notare, M.S.M.A.: An artificial immune based intrusion detection model for computer and telecommunication systems. Parallel Computing **30**(5) (2004) 629–646
7. Janeway Jr, C.A.: How the immune system recognizes invaders. life, death and the immune system. Scientific American **269**(3) (1993) 72
8. Ou, C.M.: Multiagent-based computer virus detection systems: abstraction from dendritic cell algorithm with danger theory. Telecommunication Systems (2011) 1–11
9. Dasgupta, D.: An overview of artificial immune systems. In Dasgupta, D (Ed.), Artificial Immune Systems and Their Applications (1998) 3–19
10. Dasgupta, D., Yu, S., Nino, F.: Recent advances in artificial immune systems: models and applications. Applied Soft Computing **11**(2) (2011) 1574–1587
11. De Castro, L.N., Timmis, J.: Artificial immune systems: a novel approach to pattern recognition. (2002) 67–84
12. de Castro, L.N., Timmis, J.: Artificial immune systems: a new computational intelligence approach. Springer Verlag (2002)
13. De Castro, L.N., Von Zuben, F.J.: Artificial immune systems: Part i– basic theory and applications. Technical Report - RT DCA 01/99, School of Computing and Electrical Enginnering. State University of Campinas, Brazil (1999)
14. Bersini, H., Varela, F.J.: Hints for adaptive problem solving gleaned from immune networks. In: Parallel problem solving from nature. Springer (1991) 343–354
15. Bersini, H., Varela, F.: The immune learning mechanisms: reinforcement, recruitment and their applications. Computing with Biological Metaphors **1**(2) (1994) 166–192
16. Stibor, T., Timmis, J., Eckert, C.: The link between r-contiguous detectors and k-cnf satisfiability. In: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, IEEE (2006) 491–498
17. Ji, Z., Dasgupta, D.: Revisiting negative selection algorithms. Evolutionary Computation **15**(2) (2007) 223–251
18. 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) (2012) 2953–2964
19. Ji, Z., Dasgupta, D.: Real-valued negative selection algorithm with variable-sized detectors. In: Genetic and Evolutionary Computation–GECCO 2004, Springer (2004) 287–298
20. Percus, J.K., Percus, O.E., Perelson, A.S.: Predicting the size of the t-cell receptor and antibody combining region from consideration of efficient self-nonself discrimination. Proceedings of the National Academy of Sciences **90**(5) (1993) 1691–1695
21. Holland, J.H., Holyoak, K.J., Nisbett, R.E., Thagard, P.R.: Induction: Processes of inference, learning, and discovery. computational models of cognition and perception (1986)
22. Balthrop, J., Esponda, F., Forrest, S., Glickman, M.: Coverage and generalization in an artificial immune system. In: Proceedings of the Genetic and Evolutionary Computation Conference, Citeseer (2002) 3–10
23. Jerne, N.K.: Towards the network theory of the immune system. Ann. Immunol.(Inst. Pasteur) **125C** (1974) 373–389
24. Harmer, P.K., Williams, P.D., Gunsch, G.H., Lamont, G.B.: An artificial immune system architecture for computer security applications. Evolutionary computation, IEEE transactions on **6**(3) (2002) 252–280

25. Chen, J., Yang, D., Naofumi, M.: A study of detector generation algorithms based on artificial immune in intrusion detection system. WSEAS TRANSACTIONS on BIOLOGY and BIOMEDICINE **4**(3) (2007) 29–35
26. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2011)
27. Dasgupta, D., KrishnaKumar, K., Wong, D., Berry, M.: Negative selection algorithm for aircraft fault detection. In: Artificial Immune Systems. Springer (2004) 1–13
28. Hamaker, J.S., Boggess, L.: Non-euclidean distance measures in airs, an artificial immune classification system. In: Evolutionary Computation, 2004. CEC2004. Congress on. Volume 1., IEEE (2004) 1067–1073
29. 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)
30. 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). Volume 1., Canterbury, UK:[sn] (2002) 89–98
31. D'Haeseleer, P., Forrest, S., Helman, P.: An immunological approach to change detection: Algorithms, analysis and implications. In: Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on, IEEE (1996) 110–119
32. Wierzchon, S.T.: Discriminative power of the receptors activated by k-contiguous bits rule. Journal of Computer Science & Technology **1**(3) (2000) 1–13
33. Yu, S., Adviser-Dasgupta, D.: Exploration of sense of self and humoral immunity for artificial immune systems: algorithms and applications. 361
34. Dasgupta, D., González, F.: An immunity-based technique to characterize intrusions in computer networks. Evolutionary Computation, IEEE Transactions on **6**(3) (2002) 281–291
35. González, F.A., Dasgupta, D.: Anomaly detection using real-valued negative selection. Genetic Programming and Evolvable Machines **4**(4) (2003) 383–403
36. Balachandran, S., Dasgupta, D., Nino, F., Garrett, D.: A framework for evolving multi-shaped detectors in negative selection. In: Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium on, IEEE (2007) 401–408
37. Gonzalez, F., Dasgupta, D., Niño, L.F.: A randomized real-valued negative selection algorithm. In: Artificial Immune Systems. Springer (2003) 261–272
38. Ma, W., Tran, D., Sharma, D.: A practical study on shape space and its occupancy in negative selection. In: Evolutionary Computation (CEC), 2010 IEEE Congress on, IEEE (2010) 1–7