Anomaly Based IDS Using Variable Size Detector Generation in AIS: A Hybrid Approach

T. Pourhabibi and R. Azmi

***Abstract*—Artificial Immune System (AIS) is inspired by biological immune system and provides novel ways to solve complex problems including fault detection, optimization and anomaly detection. Artificial Negative Selection is one the most important branches in AIS that discriminates normal and anomalous samples based on natural immune system self/non-self discrimination mechanism. In this paper a new schema of detector generation approach for negative selection is introduced. Negative selection is typically applied to anomaly detection problems, which can be considered as a type of pattern classification problem and is typically employed as an intrusion detection technique. This new approach, hybrids ideas from Artificial Immune Negative Selection algorithms and Restricted Coulomb Energy neural networks that are specific design of hyper sphere classifiers. While generated detectors have variable radius in real-valued space. The algorithm is tested using real-world datasets, including NSL-KDD99. The experiments in this paper showed the algorithm can tightly control the number of generated detectors.**

***Index Terms*—Detector generation, RCE network, negative selection.**

I. Introduction

Artificial Immune System (AIS) is inspired by biological immune system and provides novel ways to solve complex problems including fault detection, optimization, anomaly detection and so on. Artificial Negative Selection algorithm (NS) is one the most important branches in AIS community that simulates natural immune system self/non-self discrimination mechanism. In Negative Selection algorithm, a set of detectors are used to check incoming data to be abnormal/anomaly (non-self) or normal (self). The portion of the non-self space that is covered by detector set is one of the main concerns in negative selection algorithms. So, generating the detectors covering most portion of non-self space is important.

Among the latest works reviewing negative selection algorithms, as one of the new variations, V-detector has some unique features made it more reliable and efficient than the other negative selection algorithms.

This paper tries to introduce a new variable detector generation for negative selection based on ideas from Restricted Coulomb Energy (RCE) neural networks that are specific design of hyper sphere classifiers.

Manuscript received May 15, 2012, revised May30, 2012.

This work was supported in part by the Iran Telecommunication Research Center (ITRC) under Grant 8971/500.

Authors are with the Computer Department of Alzahra university, Tehran, Iran (e-mail: Tahereh.Pourhabibi@student.alzahra.ac.ir; r.azmi@alzahra.ac.ir).

II. Negative Selection Algorithm

A. General Negative Selection

Negative Selection is known as discriminating self/non-self and detectors are artificial non-self samples with a match threshold. In negative selection algorithms, detectors are generated randomly based on a match rule against a set of normal samples. Detectors that do not match any self are stored and go through negative selection process. So the NSA consists of three phase: defining self, generating detectors and monitoring the occurrence of anomalies (Fig. 1) [1]. Negative selection is typically applied to anomaly detection problems, which can be considered as a type of pattern classification problem, and is typically employed as a (network) intrusion detection technique.

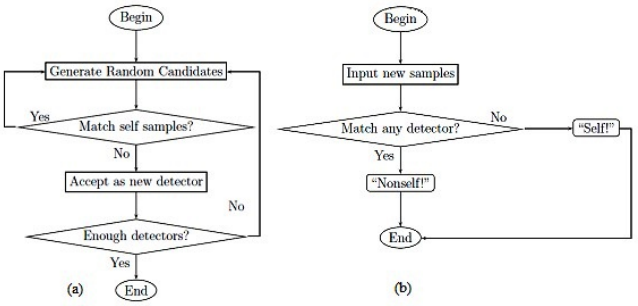


Fig. 1. (a) Detector generation for negative selection (b) Negative selection algorithm [1]

B. V-Vector Negative Selection Algorithm

Among the latest works reviewing negative selection algorithms, V-Detector (variable size detector) has some unique features to be more reliable and efficient than the others negative selection algorithms. This algorithm operates on normalized vector of real-values attributes being points in the d-dimensional unit hypercube, $U=[0\ 1]^d$. Each self sample, $s_i \in S$ is represented as a hypersphere with center at $c_i \in U$ and constant radius r_s, i.e. $s_i = (c_i, r_s)$, i=1,…,l. where l is the number of self samples. Every point $u \in U$ belonging to any hypersphere is considered as a self element. Also, detectors d_j are represented as hyperspheres: $d_j = (c_j, r_j)$, j = 1, . . . ,m. where m is the number of detectors. In contrast to self elements, the radius r_j is not fixed but is computed as the Euclidean distance from a randomly chosen center c_j to the nearest self element (this distance must be greater than r_s, otherwise detector is not created). Formally we define r_j as:

$$r_j = \min_{1 \le i \le l} dist(c_i, c_j) - r_s \quad (1)$$

The algorithm terminates if predefined number T_{max} of detectors is generated or the hypercube is sufficiently well covered by these detectors [2] (Fig. 2).

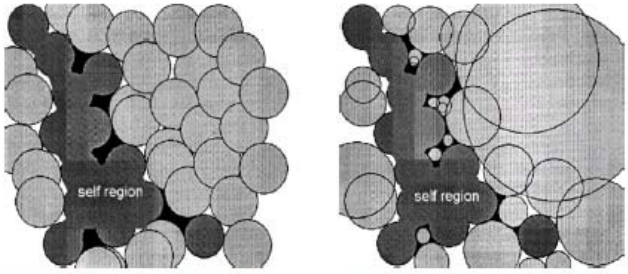


Fig. 2. (a) Constant size detectors (b) Variable size detectors [3]

III. Restricted Coulumb Energy Network

The Restricted Coulomb Energy(RCE), proposed by Reilly, Cooper and Elbaum, is one the first incremental models of neural networks[4] consists of three layers of "neuron cells" with a full set of connections between the first and second layers, and a partial set of connections between the second and third layers, as shown in Fig. 3. Each input layer cell represents a feature (a measurable characteristic) of an incoming pattern (an input signal) that the network assigns to some pattern class. The middle layer cells are called prototype cells, each of which contains information about an example of learned pattern class that occurred in the training data. Each cell on the output layer corresponds to a different pattern class represented in the training data set [5] [6]. In this model, decision units are characterized by their influence region, defined by hyper sphere around the unit, whose radius is equal to the threshold of unit. The state space is then divided into zones, each dominated by different decision units. New units are created with an initial chosen radius if presented template does not fit into one of the influence regions of the units associated with the correct class.

An RCE network cell is characterized by five elements: its class *c*, its weight vector, ω, its cell threshold, ξ, its pattern count, and its smoothing factor, λ. During training, all but the smoothing factor play a role in prototype cell development. The prototype cell weight vector ω represents the set of weighted connections between the prototype cell and each of the input layer cells. In response to a signal on the input layer, each prototype cell computes a distance (Euclidean), d_i. Between the input signal and the prototype vector stored in its weights via

$$d_i = [\sum_{j=1}^{N_D} (\omega_{ij} - x_j)^2]^{1/2} \quad (2)$$

where,

ω_{ij}=weight connecting i^{th} prototype cell and j^{th} input cell

x_j=activity of j^{th} input cell (i.e., the j^{th} feature value of vector *x*)

N_D=number of input cells (i.e., dimension of feature space)

During training, a prototype cell becomes active, if the prototype-to-pattern distance d is less than the cell threshold, λ. This is called the activation rule and the prototype is said to fire. The network is trained through a sequence of input signals, each presented with its correct classification (a labeled training set) [5] [6].

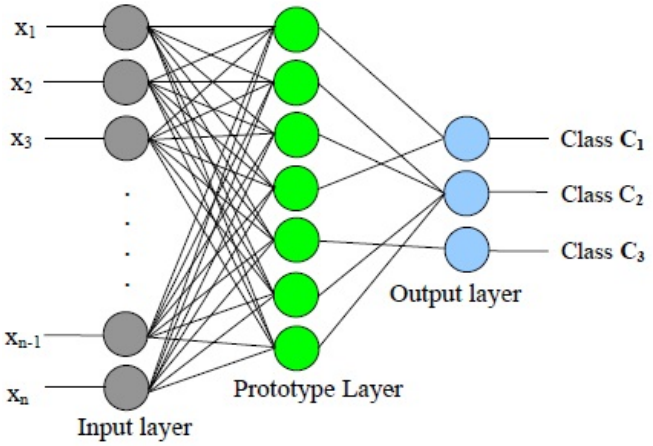


Fig. 3. Architecture of a RCE network [7]

A. Incremental Training Process of RCE Network

The incremental training process of a RCE network can be summarized as follows:

- **Case 1**: Creation of a new prototype cell: For an input feature vector *x* belonging to a class *k*, if it does not trigger any response from the existing prototype cell (if any), a new prototype cell *i* will be created and, this new cell is then connected to the output cell *k* representing the class c_k.
- **Case 2**: Increment of an existing prototype cell's counter *t*: For an input feature vector *x* belonging to a class c_k, if it does trigger a response from an existing prototype cell *i* belonging to the same class c_k, the *t* counter of this existing prototype cell *i* is incremented by 1.
- **Case 3**: Modification of the influence field of an existing prototype cell: For an input feature vector *x* belonging to a class, if it triggers a response from an existing prototype cell *i* that does not belong to same class c_k, this prototype cell's radius of the hyper-spherical influence filed is reduced according to the following calculation:

$$d_i = [\sum_{j=1}^{N_D} (\omega_{ij} - x_j)^2]^{1/2} \quad (3)$$

It is clear that the incremental training process of RCE neural network is very simple and has not issue of convergence. Most importantly, both the number of cells in the prototype layer and the number of cells in the output layer can be dynamically increased during the real-time activation of the RCE neural network [7] (for more information we refer to [8]).

IV. Proposed Detector Generation Algorithm

As we can find from above discussion, in negative selection algorithms, detectors are generated randomly over the state space and one the main concerns in these algorithms,

especially in high dimensional spaces, is the detectors convergence, means the portion of non-self space that is covered by detectors. So it is important to generate detectors that (a) are distributed all over the space, (b) have as big as possible radius to cover all the space. As a result, in our approach, we introduce a detector set generation method that not only uses randomly generated detectors, but also the training non-self samples. In this way we can generate big radius detectors which are distributed all over the space, especially in parts that the existence probability of non-self samples is more.

A. Detector Set Generation

The learning algorithm for proposed detector generation method can be described as shown Fig.4:

1) Setting Initial values :Set Self as normal and non-Self as abnormal training samples; Set rs as self radius; Set Rmax as maximum radius; Set MaxNum as maxiumum number of randomly generated detectors.

2) Generate randomly detector x, and set its radius to Rmax
   a) compute distance d, from x to its nearest self
   b) if d < x.radius +rs and x.radius>0
      reduce the x.radius as in Equation 3;
      go to 2_a
   c) if x.radius>0
      add x to Mature Detector list

3) If number of detectors in Mature Detector list exeed Maxnum go to 4 else go to 2.

4) For all non-self training samples ab,check if ab match any detector in Matture Detector list
   a) If it does not happen any match,we generate a new detector c, centered in ab with radius Rmax
   b) Compute distance d, from detector c to its nearest self
   c) if d<c.radius+rs and c.radius>0
      reduce c.radius as in Equation 3
      Go to 4_b
   d) if c.radius>0
      add c to Mature Detector list

5) Out put the Mature Detector list

Fig. 4. Detector Set Generation Algorithm for Proposed Approach

V. Dataset Description

To test our proposed algorithm, we used NSL-KDD dataset which is a new version of KDD Cup99 dataset [9]. Each record of this data set describes a network connection as a vector of 41 attributes (38 numerical and 3 symbolic) with additional end-label (normal value means self connection). Examples of self and non-self connections are given below [2]:

0, icmp, eco i, SF, 18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00,0.00, 0.00, 1, 153, 1.00, 0.00, 1.00, 1.00, 0.00, 0.00, 0.00, 0.00, ipsweep.

91, udp, domain u, SF, 87, 45, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 147, 140, 0.95, 0.01, 0.18, 0.00, 0.00, 0.00, 0.00, 0.00, normal.

In our experiment we used only 19 of 41 features based on feature selection proposed in [10] and using a min-max approach we normalized the data set as in [11].

VI. Experimental Results

To test the proposed algorithm, we used 125970 Records of normalized KDD Cup 99 train dataset. More precisely, we randomly created 10 subsets from dataset. In each subset we used 70% of data records for train and the remained 30% for test.

Table I shows the average results of three times run of algorithm on ten random selected subsets of NSL_KDD99 dataset. Detector Rate (DR) and False Alarm Rate (FA) and the accuracy are defined as:

$$DR = \frac{TP}{TP + FN} \tag{4}$$

$$FA = \frac{FP}{FP + TN} \tag{5}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Fig. 5.(a) shows the balance between number of detectors and false alarm rate and Fig. 5.(b) shows the balance between accuracy and false alarm rate of the algorithm based on results of Table I. To test our algorithm, we initialize our variables as: r_s =0.1, R_{max}=19, *Maxnum*=500.

TABLE I: Results of Experiments on 10-Random Selected Subsets of NSL-KDD99 Learning Dataset

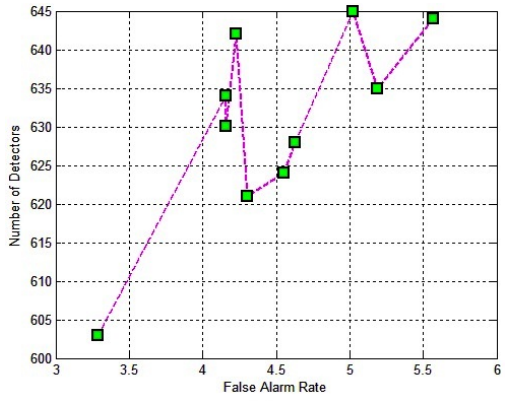
Number of Detectors	*DR* (%)	*FA* (%)	*Accuracy* (%)
645	99.94	5.02	90.15
621	99.78	4.30	90.74
624	99.94	4.55	90.39
603	99.61	3.29	90.55
642	99.94	4.23	90.39
644	99.94	5.57	89.53
634	99.77	4.16	90.45
635	100	5.19	89.82
628	99.89	4.63	90.39
630	99.94	4.16	89.55

VII. Conclusion

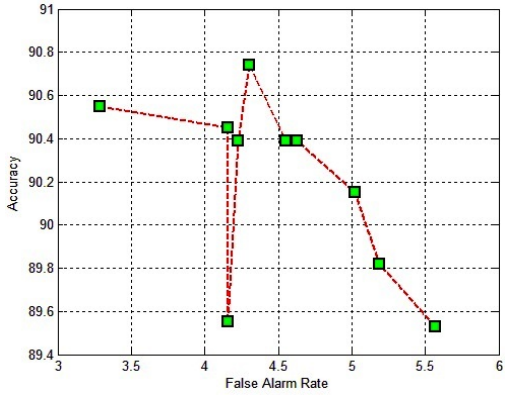
A new approach for detector generation in AIS was proposed. This new technique generates detectors based on the inherent ability of discriminating self/non-self in NSA. Initially, randomly generated detectors have as big as possible radius in the state space and the algorithm leaves them away from self data by decreasing the radius based on RCE network factors. So each generated detector owns its radius. All training abnormal patterns that could not match any randomly generated detector will be added to detectors list by enough getting away from self data. As a result, we can distribute the detectors in parts of space that the

probability of existence abnormal patterns is more.

As we expect, the results show high detection rate and low false alarm rate with least possible number of detectors.



(a)



(b)

Fig. 5. (a) Balance between number of detectors and false alarm rate (b) Balance between accuracy and false alarm rate

ACKNOWLEDGMENT

Thanks to Iran Telecommunication Research Center (ITRC) because of its financial support.

REFERENCES

[1] Z. Ji, "Negative Selection Algorithms: from the Thymus to V-detector,"Ph.D. dissertation, Dept. Computer Science, University of Memphis, Memphis, TN, 2006.

[2] A. Chmielewski and S. T. Wierzcho, "V-detector algorithm with tree-based structures," in *Proc. International Multiconference on Computer Science and Information Technology*. Wisła, Poland, 2006, pp. 9–14.

[3] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review,"Applied Soft Computing, v.10, n.1, January, 2010, p.1-35.

[4] J. Didier Legat, P. Thissen, M.Verleysen and J. Luc Voz, "parallel Implementation of the RCE algorithm, " Mocroelectronics Laboratory, Universite Catholique de louvian.

[5] V. Rajan, K. R. Pattipati and J. Luo, "Fault Diagnosis in Mixed-Signal Circuits via Neural-Network based classification Algorithms," Proc. International Mixed-signal Testing Workshop (IMSTW 2000), Montpelier, France, June 21-23, 2000.

[6] Rajan V, Yang J, Chakrabarty S and Pattipati K, "Machine learning algorithms for fault diagnosis in analog circuits, " in *Proc Systems, Man, and Cybernetics. 1998 IEEE International Conference,* San Diego, 1998, vol 2, pp 1874-1879.

[7] M. L. Yuan and M. Xie, "An incremental representation of conceptual symbols using RCE neural network," presented at the 2nd *International Conference on Development and Learning*, United States, IEEE Computer Society, pp. 102-107

[8] F. Zboril, "Sparse Distributed Memory and Restricted Coulomb Energy Classifier," in *Proc MOSIS'98, MARQ*, Ostrava, Sv. Hostn - Bystrice pod Hostnem, pp171 - 176, 1998.

[9] NSL-KDD Intrusion Detection Data Set, [online]. Available: http://iscx.ca/NSL-KDD/.

[10] D. M. Farid, J. Darmont, N. Harbi, H.H. Nguyen and M.Z. Rahman, "Adaptive Network Intrusion Detection Learning: Attribute Selection and Classification," in *Proc International Conference on Computer Systems Engineering (ICCSE 09)*, Bangkok, Thailand, WASET December, 2009.

[11] M. Sabhnani, G. Serpen, "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context," in *Proc International Conference on Machine Learning: Models, Technology and Application*, Las Vegas, Nevada, USA, June 2003.

T. Pourhabibi was born in Tehran, Iran on July 1983. She received her Bachelor of Science degree in Software Engineering from Shahid Beheshti University, Tehran, Iran on February 2007 and her Master of Science degree in Artificial Intelligence Engineering from Alzahra University, Tehran, Iran on March 2011. Her interest fields of study are: artificial immune system, intrusion detection, machine learning, parallel processing, cloud computing, artificial intelligence and data mining.

Dr. R. Azmi was born in Iran on May 1968. He received his BS degree in Electrical Engineering from Amirkabir university of technology, Tehran, Iran in 1990 and his MS and PhD degrees in Electrical Engineering from Tarbiat Modares university, Tehran, Iran in 1993 and 1999 respectively. Since 2001, he has joined Alzahra university, Tehran, Iran. He was an expert member of Image Processing and Multi-Media working groups in ITRC (From 2003 to 2004), Optical Character Recognition working group in supreme council of information and communication technology (From 2006 to 2007) and Security Information Technology and Systems working groups in ITRC (From 2006 to 2008). He was Project Manager and technical member of many industrial projects. Dr Azmi is founder of Operating System Security Lab (OSSL), Medical Image Processing Lab (MIPL), Face and Facial Expression Recognition Lab (FFERL), Web-based Anomaly Detection Lab (WADL) and Optical Character Recognition Lab (OCRL) in Alzahra University. He is currently an Assistant Professor of Computer Engineering at Alzahra University.