Immune Multiagent System for Network Intrusion Detection using Non-linear Classification Algorithm

Muna Elsadig Mohamed
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

Brahim Belhaouari Samir
Department of Fundamental and applied Science
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

Azween Abdullah
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

ABSTRACT
The growth of intelligent intrusion and diverse attack techniques in network systems stimulate computer scientists and mathematical researchers to challenge the dangers of intelligent attacks. In this work, we integrate artificial immune algorithm with non-linear classification of pattern recognition and machine learning methods to solve the problem of intrusion detection in network systems. A new non classification algorithm was developed based on the danger theory model of human immune system (HIS).The abstract model of system algorithm is inspired from HIS cell mechanism mainly, the Dendritic cell behavior and T-cell mechanisms. Classification techniques using k-nearest neighbor (k-NN) or Gaussian Mixture (GMM) almost have the common sense that they believe the neighboring data. The algorithm tested use KDD Cup dataset and the result shows a significant improvement in detection accuracy and reducing the false alerts.

General Terms
Network Security

Keywords
Artificial immune system, classification, Intrusion detection system

1. INTRODUCTION
With the explosive advance of the Network Systems and Internet, and the rise of information technology in everyday life the need for networks security has become critical. Meanwhile, the complexity of attacks is on the rise regardless of the beefed-up security measures. Intrusion detection is a system for computer networks security used to identify possible threats and produce alerts to get respond to them rapidly. In network systems, the main security danger comes from insider abuse and from external intrusions.

Immune System presents a valuable metaphor for computer security systems and it is an appealing mechanism because firstly, the Human Immune System (HIS) defends the body with high level of protection features from pathogens, in a Self–Organized, Robust, Distributed and Diverse manner [1]. Secondly, current security systems are not able to handle the dynamic and increasingly complex nature of the computer systems and their security needs. Based on this deficiency, Artificial Immune Systems (AISs) have been successfully applied to a number of network security problem domains that include Intrusion Detection Systems (IDS). However, most of the developed intrusion detection system technologies work fairly well in static systems but have certain deficiencies in dynamic systems, such as the lack of self-adaptation, lack of robustness and they are mostly centralized in design. Hence, it is necessary to construct an effective defense system, which has features of autonomous, self-adaptability, self-detecting, self-monitoring, and self-healing. Currently the trend is towards large and much more dynamically configured systems. The present algorithms integrates an artificial immune intrusion detection system for network security inspired by the immunology theory known as the danger theory, adaptive immune system and pattern recognition classifications. The proposed system is inspired by the Human Immune System (HIS), which is applied to the autonomous defense system. The inspired IDS mechanisms look at the danger model and its application to activate malicious behavior defense in order to create a fully decentralized model.

The second section describes and overview of human system and approaches inspired intrusion detection problems and the IDS design feature. Section three discuss the classification algorithms analysis and section four the simulation results are discussed. Finally the conclusion of the work and the future work are demonstrated.

2. HUMAN IMMUN SYSTEM
The main features of biological immune system adopted by this novel IDS are the features of two interacting subsystems: the innate and adaptive immune systems. While the reality of an innate immune system has long been obvious, it has little impact on the design of AISs [2, 3]. AISs to date have largely been inspired by adaptive immune system. Scientists described the adaptive immune system as a system capable of specific recognition and remembrance of pathogen, while the innate immune system is characterized mainly as the first line of defense and rapid-response system against pathogens. Many immune system approaches to IDS have been introduced. There are three major extractions, and accordingly three different views: conventional algorithm, the negative selection paradigm, and the danger theory to get more information refer to [4, 5, 6].

Danger theory is a new different immunology theory for IDS. The particular characteristic that makes this model different from other immune theories is that according to the danger theory immune response is triggered by unusual deaths of self-cells. Danger theory (DT) recommends that foreign intruders, which are dangerous, will encourage the generation of cellular molecules (danger signals) initiating cellular stress or cell distress (dies by necrotic or abnormal death). Pathogens, which have damaged the body cell, sends danger signal to the dendritic cells or Antigen Presenting Cells (APCs). Antigen is swallowed from the extracellular environment by DCs in their immature state and then processed internally. During processing, antigen is fragmented

and attached to main histo-compatibility complex (MHC) molecules. This MHC antigen complex is then presented under definite conditions on the surface of the DC to T-helper cell in lymphocyte. As well as extracting antigen from their surroundings, DCs also have receptors, which respond to a range of other signaling molecules in their environment. Certain molecules, such as lipopolysaccharide, collectively termed pathogen-associated molecular proteins (PAMPs) are common to the entire classes of pathogens and bind with toll-like receptors (TLRs) on the surface of DCs. While other groups of molecules, termed as danger signals, such as heat shock proteins (HSPs), are associated with damage to host tissue or unregulated necrotic cell death and bind with receptors on DCs. In addition, there are other classes of molecules related to inflammation is called endotoxin, or lipopolysaccharide (LPS). This substance is present in the outer covering of some types of bacteria; also interacts with receptor families present on the surface of DCs. The current maturation state of the DC is determined through the combination of these complex signaling. The number and strength of DC cytokine output depends on its current maturation state. The proposal presented by Aickelin and Cayzer [7] to DT model has encouraged many AIS, mainly computer security developers, to discover the potentials of danger theory [8, 9,10]. Our system uses the danger theory as forcefulness concept for inspiration to intrusion detection system integrated with adaptive immune system mechanisms, mainly DC as classifier, T-cell and B-cell mechanisms. The abstraction of the algorithms design and analysis is in previous work in [11].

Based on the understanding of how both of these systems function, this research mapped a number of features to IDS design [2, 12, 13]. These features provide a general structural framework in designing a new biological IDS generation in general. The combination of these design features form a robust IDS specification and design with optimal performance.

3. IMMUNE AGENT CLASIFICATION ANALYSIS

Classification techniques that use k-nearest neighbor (k-NN) or Gaussian Mixture (GMM) almost have the common sense that they believe the neighboring data. These data has been represented as pixel vectors where any new pixel vector for example x, which will be classified to neighboring k cluster class and the classification will be more accurate compared to NN technique [17]. This is because they are more competent in overlapping area as these methods take more consideration of training data samples that are less numerous. In order to reduce the classification time of k-NN technique (k-NN), we need to cluster our space i.e. the training data into subclasses, where each subclass will be represented by one datum. According to the number of the subclasses we can select two or more representatives. This is followed by applying the classification algorithm NN or k-NN using representative data. The data in the subclasses are random, where they are relatively close to each other. This procedure of classification is known as cluster-k-NN(C-k-NN) which is comparable to 'variable k'-NN.

To estimate the classification time of order N for NN and k-NN respectively;

$$X = O(N), \qquad \frac{X}{N} \to K \; as \; N \to \infty \qquad (1)$$

$$x = o(N), \qquad \frac{x}{N} \to 0 \; as \; N \to \infty \qquad (2)$$

where N is the training data size. The time of classification is subclass number mi dependent, where mi is the number of subclasses in class Ci,. Therefore, the classification time has been reduced by clustering the space. Generally, mi is a small number that does not depend on the training data size. This has been considered in Gaussian functions for estimation of probability density for GMM method, in general, is bounded to the variable N.

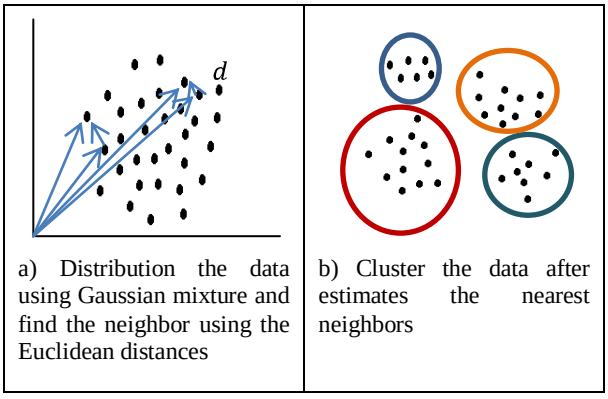
Non-parametric density is commonly estimates by k-NN. The rule used by k-NN technique is influential and generate highly non-linear classification with limited data [18].To classify a pattern x, first we have to find the closed k examples in the dataset and select predominant class Ci, among those k neighbor. Here the problem is if two or more classes are predominant classes.

The original C-k-NN classifier is based on the Euclidean distance between a test sample x and specified training samples, but in the new cluster-k-NN we add the following metric in order to get a better estimation of density probability:

$$d(x, x_i) = \frac{d^s_{euclidean}(x, \hat{x}_{i,j})}{n_{i,j}}, \forall_x \in R^d \qquad (3)$$

where;
s is a positive number,
$x_{i,j}$ belongs to the subclass j of class *i*,
We note $C_{i,,j}$, for all *s* and $\hat{x}_{i,j}$ is the represents $C_{i,,j}$
$n_{i,j} = card(C_{i,,j})$ or the variance of the set $C_{i,,j}$.

Figure 1(a) explains how we can distribute the data and find the nearest neighbors by calculate the distance as in equation (6). In addition, figure 1 (b) how the data has been clustered by finding the nearest neighbors.



a) Distribution the data using Gaussian mixture and find the neighbor using the Euclidean distances

b) Cluster the data after estimates the nearest neighbors

Figure 1: Gaussian Mixture Data Clustering

The Gaussian mixture model has been used as parametric method that can classify as semi-parametric density estimation. It defines a very universal class of functional forms for the density model. In a linear mixture model, the probability density function is expressed as a linear combination of basic functions. A model with M components is explained as distribution mixture according to equation (4).

$$P\left(x/_j\right) = \exp\left\{-\tfrac{1}{2}\left(x-\mu_j\right)^T\left(x-\mu_j\right)\right\}/$$
$$(2\pi)^{\frac{d}{2}}|\sum_j|^{\frac{1}{2}} \quad (4)$$

The parameters to be estimated are the mixing coefficient$P(j)$, the covariance matrix and mean vector μ_j.

In C-k-NN, each Gaussian function $P(j)\ P\left(x/_j\right)$ can be approximated by:

$$\frac{1}{cst+d(x+\mu_i)} \quad (5)$$

Where *cst* is any small number that is added to avoid the division by *0*.

The estimation of the number of subclasses and their representatives for C-k-NN (or the number of Gaussian function M and their means μ_j for GMM) can be derived by K-means cluster or another modified stable algorithm. The number of subclasses is needed as input to fix the number of the clusters. The iteration of clustering is started by one and under the following conditions it is stopped:

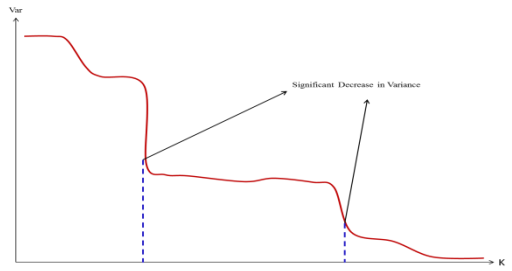


Figure2: Variance in Terms of Number of Cluster "k"

All the representative centroids ($\mu_{i,j}$) have to be closer to their class $C_{i,j}$ than to other classes. This is to reduce misclassification.

a. The variance of each class, var, does not reduce significantly in comparison to the previous iteration. We define the variance of each class as:

$$var = \sum_{i=1}^{n} var_i \quad (6)$$

Where,var_i is the variance of subclass *i*
Considering that:

$$\frac{\Delta var}{var} \leq \alpha \quad 0.0 \leq \alpha \leq 1.0 \quad (7)$$

As criterium of smooth function of var with the number of subclasses are variable. The best value of $\propto$ will be considered in the simulation to obtain the best accuracy. By this judgment of $\propto$, we have completed our dictionary for training phase.
For each class, C_i is represented by:

$$\{\mu_{i,1}, \mu_{i,2}, \ldots\ldots, \mu_{im_i}\}, \quad 1 \leq i \leq cn \quad (8)$$

Where, m_i is the number of subclasses for class C_i and cn is the number of classes.

To classify new pattern *x* we use *k*-NN algorithm on the data set:

$$\{\mu_{i,j}: 1 \leq i \leq cn, 1 \leq j \leq m_i\} \quad (9)$$

By using this technique we reflect on the minimum rule, and assign *x* to class C_i which is verified by:

$$C_i = arg(\min_{\substack{1\leq i\leq cn \\ 1\leq j\leq m_i}}\{d(x,m_{i,j})\}), \quad (10)$$

Where,$\arg(d(x,m_{i,j}) = C_i\ \forall\ 1 \leq cn$.

Due to the dependency of the result on random choice of k initial vectors k-means cluster is unstable. To achieve the stability of k-means cluster, we introduce a different initialization for the algorithm, which gives a better result than the classic *k*-means cluster where:

$$Var_{modified\ algorithm} \leq Var_{classic\ algorithm} \quad (11)$$

By using the modified classification algorithm better results are obtained in terms of accuracy and minimum false errors rates.

3.1Near-to-near algorithm

The algorithm first calculates the distance $d(x_{i,n},x_{i,m})$ between test samples *x* for $x_i \epsilon C_i$ then
starts to cluster or class the samples into N_i *-1* subclasses;

where,
$$card(C_i)=N_i \quad (14)$$

Then we put the two closest data into the same subclass

$$C_{i,1}=\{x_{i,n0},x_{i,m0}\} \quad (15)$$

where;
$$min_{n\neq m}\, d(x_{i,n},x_{i,m}) = d(x_{i,n_0},x_{i,m_0}).,$$

The next step is to put other data into separate subclasses,

$$C_{i,j} = \{x_{i,j}\}, \forall j \in \{1,\ldots N_i\} - \{n_0,m_0\}, \quad (16)$$

The following index n_i and m_i are considered for which

$$min_{\substack{n\neq m \\ (n,m)\neq(n_0,m_0)}} d(x_{i,n},x_{i,m}) = d(x_{i,n1},x_{i,m1}),$$

If $x_{i,n1}$ and $x_{i,m1}$ belong to the same subclass $C_{i,r}$, then this subclass is split into two other subclasses,

$$C_{i,r+1} = C_{i,r} - \{x_{i,n1},x_{i,m1}\} \quad (17)$$

$$C_{i,r} = \{x_{i,n1},x_{i,m1}\} \quad (18)$$

If $x_{i,n1}$ and $x_{i,m1}$ belong to two different subclasses $C_{i,r1}$and $C_{i,r2}$ respectively, Then we put $x_{i,n1}$ in subclass $C_{i,r2}$.*If* $card(C_{i,r2}) > card(C_{i,r1})$.

And *If* $card(C_{i,r2}) \leq card(C_{i,r1})$ Then $x_{i,m1}$is put in $C_{i,r1}$
The distance between the vector to the set is used as *d(vector ,mean of set).*
When k-subclass is obtained, the iteration stops and the initial *K*-vector will be the mean value of each class.

3.2. Near-to-mean Algorithm

This algorithm is to near-to-near algorithm but it deals with the mean of subclass $C_{i,r}$.
At the start, the class is split into two subclasses
$C_{i,1}=\{x_{i,n0},x_{i,m0}\}$,
$$C_{i,2} = \{x_{i,j} | j \notin \{n_0,m_0\}\}, \quad (20)$$

where;
$d(x_{i,n0},x_{i,m0}) = min_{n\neq m} d(x_{i,n},x_{i,m})$,
$C_i$is updated by replacing $x_{i,n0}$ and $x_{i,n0}$ by their average, i.e.

$$C_i^1 = \{..,x_{i,n0-1},s_0,x_{i,n0+1},\ldots,x_{i,m0-1},s_0,x_{i,m0+1},\ldots\}, (21)$$

where; $S_0=(x_{i,n0} + x_{i,m0})/2$.
Next $x_{i,n1}$ and $x_{i,m1}$ are considered such as,

$$d(x_{i,n1},x_{i,m1}) = min\{d(x_{i,n},x_{i,m}) | d(x_{i,n},x_{i,m}) \neq 0\},$$

We replace all data in C_i^1 that are equal to $x_{i,1n}$ or $x_{i,m1}$ by s_1 which is the mean of the union of the two subclasses where $x_{i,1n}$ and $x_{i,m1}$ belong to

$$s_1 = \frac{c_{n1}x_{i,n1}+c_{m1}x_{i,m1}}{c_{n1}+c_{m1}},$$

Where; $C_{n1}$is number of repletion of $x_{i,n1}$inside$C_i^1$and $C_{m1}$is number of repletion of $x_{i,m1}$ inside C_i^1.

The algorithm stops once the number of distinct vector inside C_i^r is equal to *k*.
No need to keep all data, this is one of the features and powerful point of these classification algorithms. Instead of this it needs only the average of each subclass.

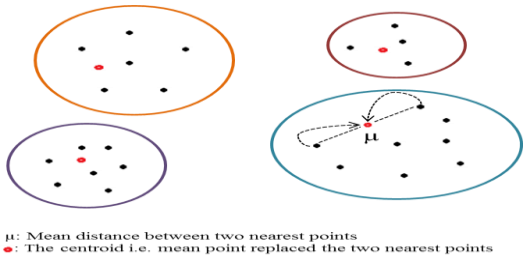


Figure 3: Each cluster has a centroid and the mean between each to nearest is estimated and each two nearest data are replaced by their means

To classify a new data or vector *x*, we use *k*-NN algorithm i.e. we assign *x* to class $C_{\hat{\imath}}$ for which,

$$\hat{\imath} = arg_i min_{ij} d(x,\mu_{i,j}),$$

Where; $arg_i d(x,\mu_{i0,j0}) = i_0$.
More examination of the *k*-NN algorithm is required to find the closest *j*-examples in the dataset and to select the predominant class. The smallest and closest examples in the dataset can be captured and the predominant class which have exactly *k* examples can be selected.

4. IMMUNE AGENT ALOGRITHMS FOR INTUSION DETECTION VALIDATION

The DARPA KDD Cup Intrusion Detection Evaluation Program was formulated and managed by MIT Lincoln Labs. The aim of DARPA was to study and assess research in intrusion detection and test the performance of intrusion detection. A standard set of data to be audited, which comprises a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD Cup intrusion detection contest uses a version of this dataset and summarized network connections with 41-features per connection [19]. Lincoln Labs set up a situation to obtain nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. The original training data was about four gigabytes of compressed binary TCP dump format from simulated period seven weeks of network traffic. This was processed into approximately five million network connection records. Abnormal behaviors fall into four main classes and their categories [20]: DOS: denial-of-service,R2L: unauthorized access from a remote machine, U2R: unauthorized access to local superuser (root) privileges, and Probing: surveillance and other probing. The process of collecting the dataset, resulting in 41 features for each connection between the dataflow from source IP address to destination IP address. To validate IDS abstraction and design algorithm, the KDD cup data set is used. For the 4 different classes of abnormal behavior the algorithms are tested. The results are shown in table 1. The accuracy for the detection for the 4 test is improved.

The false positive error is decreased significantly compared to IMAAD [21] in table 2 while the true positive rate is increased. The accuracy for the U2R test reached 100 % detection of the behavior. And the minimum accuracy attained in R2L test which is 99.6 and still higher than the previous AIS algorithm.

The change of the accuracy values with the values of **α** in the denial of service, probe, U2L and U2R simulation test are illustrated in figure 1,2,3, and 4.
The fast and max value of DoS dataset simulation is estimated at feature number 4.

Table 1 Accuracy of Classification for The 4 Simulations Tests

Abnormal behavior class	**Denial of Services**	**Probe**	**R2L**	**U2R**
Accuracy	99.79 %	99.92%	99.60%	100.0%

Table 2 Comparison between IMAAD and Current System

	Number of Identified Categories		**TP %**		**FP %**	
Simulation test	IMAAD	Current	IMAAD	Current	IMAAD	Current
Denial of Services	5	7	97.2	99.86	2.8	0.07
Probe	4	5	96.5	1.0	3.5	0.08
R2L	4	5	94.5	1.0	4.6	0.0
U2R	3	5	95.2	99.6	0.0	0.0

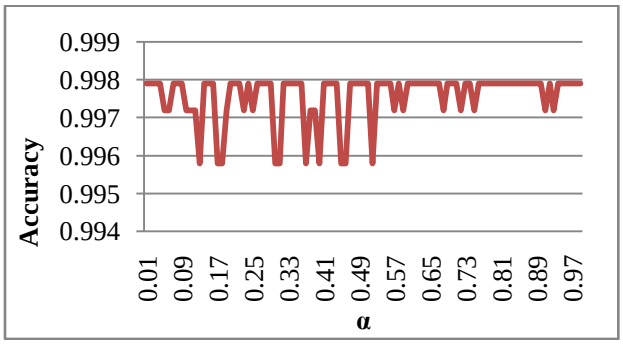


Figure 1: The Denial of services dataset Test. Change of accuracy at best feature with α.

The maximum value is 99.92% for probe dataset simulation test. The fast and max accuracy is estimated at feature 8 for the first and the value is achieved by other features. Figure 6.9 shows the change of the accuracy at the point of the best feature with values of α. The fast and maximum value for the accuracy is estimated at feature number 4 in R2L dataset simulation test.

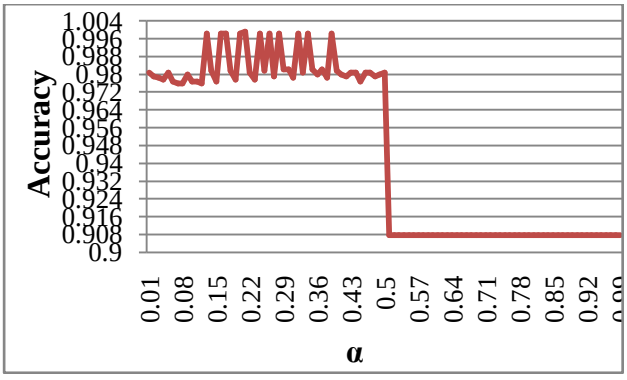


Figure 2: The Probe simulation Test, Change of accuracy at best feature with α.

In the R2L the value of the false negative error is high compared to other test while the false positive error is 0.

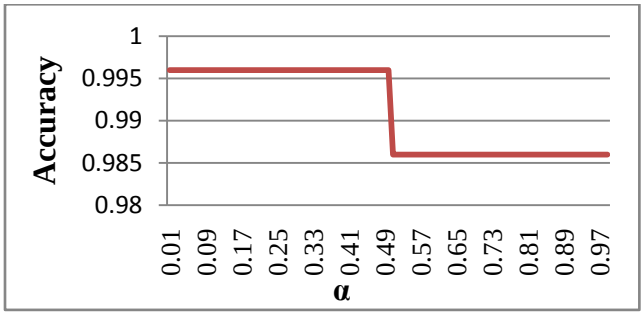


Figure 3.The R2L Simulation test, Change of Accuracy at Best Feature with α.

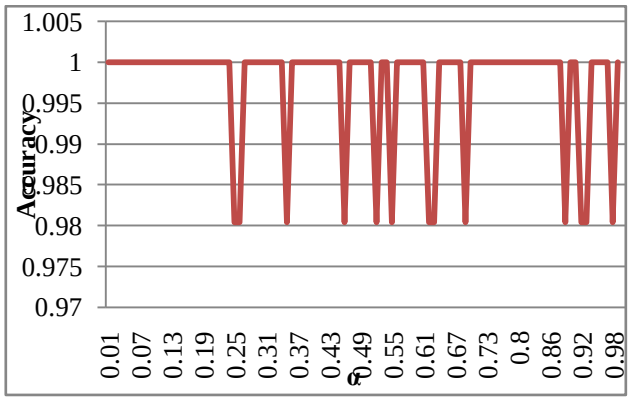


Figure 4.U2Rsimulation test, Change of accuracy at best feature with α.

5. CONCLUSION

In this paper, integration between artificial immune system and pattern recognition algorithms is presented for the problem of network intrusion detection. The abstract model is inspired from human immune system mechanisms in particularly DC, T-cell and B-cell. The algorithm introduced in this work is simulated for immune agents using nonlinear classification methods. The classification algorithm is based on the k-NN mean cluster and Gaussian mixtures techniques. The system algorithms are tested using KDD CUP dataset. Four tests are run performed for different 4 classes of abnormal behaviors. The obtained results show improvements in accuracy of detection and significant decrease in the false errors. The integration between AIS and pattern recognition algorithms for classification presents noteworthy results in intrusion detection system capabilities. Using nonlinear classification algorithm is highly efficient in solving the matter classifying normal and abnormal behaviors. In the future work, the system must be tested for the predictability of prevention response and implemented in real time work. Also we can validate using different standard datasets, for example process behavior datasets.

6. REFERENCES

[1] A. Somayaji, S. Hofmeyr, and S. Forrest, "Principles of a computer immune system," proc of the 1997 workshop on New security paradigms - NSPW '97, 1997, pp. 75-82.

[2] S.M. Garrett, "How do we evaluate artificial immune systems?," Evolutionary computation, vol. 13, Jan. 2005, pp. 145-77.

[3] J. Twycross and U. Aickelin, "Biological Inspiration for Artificial Immune Systems," vol. 4628, 2010, p. 12.

[4] J. Kim, P.J. Bentley, U. Aickelin, J. Greensmith, G. Tedesco, and J. Twycross, "Immune system approaches to intrusion detection – a review," Natural Computing, vol. 6, Jan. 2007, pp. 413-466.

[5] S. Forrest, S.A. Hofmeyr, and A. Somayaji, "Computer immunology," Communications of the ACM, vol. 40, 1997, pp. 88-96.]

[6] J. Kim and P. Bentley, "The Human Immune System and Network Intrusion Detection," proc of the 7th European Conf

on Intelligent Techniques and Soft Computing EUFIT99, 1999.

[7] U. Aickelin and S. Cayzer, "The Danger Theory and Its Application to AIS," proc of the First International Conf on Artificial Immune Systems ICARIS2002, 2002, pp. 141-148.

[8] U. Aickelin and J. Greensmith, "Sensing danger: Innate I immunology for intrusion detection," Information Security Technical Report, vol. 12, 2007, pp. 218-227.

[9] A. Krizhanovsky and A. Marasanov, "An Approach for Adaptive Intrusion Prevention Based on The Danger," The Second International Conf on Availability, Reliability and Security (ARES'07), Apr. 2007, pp. 1135-1142.

[10] J. Greensmith, U. Aickelin, and S. Cayzer, "Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection," Artificial Immune Systems, 2005, p. 153–167.

[11] M. Elsadig, A. Abdullah, and B.B. Samir, "Immune Multi Agent System for Intrusion Prevention and Self-Healing System Implement a Non-Linear Classification," (ITSim), IntSymp in , vol.3, no., pp.1-6, 15-17 June 2010.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, 2009, pp. 1-58.

[13] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS) Recommendations of the National Institute of Standards and Technology," Nist Special Publication.

[14] Duda, R. O., Hart, P. E., and Stork, D. G. Pattern Classification 2nd Edition. Wiley- Interscience, 2000.

[15] E. Eskin, "Anomaly Detection over Noisy Data using Learned Probability Distributions," Proc of the 25th Int Conf on Machine learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 255-262.

[16] K. Chan, M.V. Mahoney, and M.H. Arshad, "A Machine Learning Approach to Anomaly Detection," Tech. Rep. CS-003 06, Department of Computer Science, Florida Institute of Technology Melbourne FL 32901, 2003, pp. 1-13.

[17] B.VDasarstly, Ed.,"Nearest Neighbor (NN) Norms: NN Pattern classification techniques", osAlamitos,AC:IEEE computer Socity press 1990.

[18] D.M.Titterington, A.F.M. Smith, and U.E.Mako,"statistical analysis of finite mixture distriburions.", John Wiley,NewYork,1985.

[19] KDD CUP 99 Data Sethttp://www.sigkdd.org/kddcup/index.php?section =1999&method=info

[20] H.G. Kayacik, A.N. Zincir-Heywood, and M.I. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets," Dalhousie University, Faculty of Computer Science, 2005, pp. 3-8.

[21] S. Liu, T. Li, D. Wang, X. Hu, and C. Xu, "Multi-agent network intrusion active defense model based on immune theory," Wuhan University Journal of Natural Sciences, vol. 12, Jan. 2007, pp. 167-171.