Applied Soft Computing 11 (2011) 4055–4063

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Artificial immune system based on interval type-2 fuzzy set paradigm[☆]

A. Visconti[a,*,1], H. Tahayori[b]

[a] *Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano, via Comelico 39/41, 20135 Milano, Italia*
[b] *Department of Computer Science, Ryerson University, 350 Victoria Street, Toronto, Ontario M5B 2K3, Canada*

ARTICLE INFO

Article history:
Received 17 April 2008
Received in revised form
22 September 2010
Accepted 12 December 2010
Available online 5 February 2011

Keywords:
Artificial immune system
Interval type-2 fuzzy set
Centroid
Artificial lymphocytes
Intrusion detection system

ABSTRACT

This paper discusses the design and engineering of a biologically-inspired intrusion detection system, based on *interval type-2 fuzzy set paradigm*, for protecting computer networks. To this end, we have proposed a performance-based *Artificial Immune System* (AIS) that mimics the workings of an *adaptive immune system* and consists of a number of running artificial white blood cells, which search, recognize, store and deny anomalous behaviors on individual hosts. The proposed AIS monitors the system through analyzing the set of parameters to provide general information on its state. For the analysis, we have suggested a dynamic technique based on interval type-2 fuzzy set paradigm that enable identifying the system status – i.e. Non-Attack, Suspicious-Non-Attack, Non-Decidable, Suspicious-Attack, Attack. In conclusion, for proving the effectiveness of the suggested model, an exhaustive testing is conducted and results are reported.

1. Introduction

Developed in the 1990s, Artificial Immune Systems (AISs) are inspired by the workings of the biological immune systems, focusing on their capability to recognize elementary *self* components of the body – endogenous or innocuous – and elementary *non-self* components of the body – exogenous or potentially pathogenic. Artificial immune systems proposed in the last decade were mainly studied and implemented for solving real-world problems. It is argued that AIS is aimed to improve the efficiency of systems in different fields such as anti-spam, computer security, clustering, data mining, hardware fault detection, road traffic anomalies, pattern recognition, and so on. In particular, the most representative applications of artificial immune systems are from the area of computer security and fault detection [1].

It is known that biological immune systems draw up several lines of defense to protect living organisms from all kinds of unwanted invaders. The lines of defense are grouped in chemical and physical barriers, innate immune system and adaptive immune system. Skin, mucus secretions, stomach pH, white blood cells could be enumerated as examples of different biological barriers. Similarly, computer networks rely on several lines of defense, e.g. firewall, login policies, antivirus, antispam, etc. to protect servers against unwanted invaders or, more generally, unwanted behaviors originated from within or out of the system. Firewall in a network, equivalent to skin, filters external access requests, and blocks all connection attempts that violate certain criteria. Login policies corresponds to physiological barriers, let regular users get into the network while block the access of outsiders. Antivirus in a network, correspondent to a set of white blood cells, monitors the system, recognizes and kills viruses known a priori, and sometimes updates its database with newly emerged viruses.

Artificial immune systems for the protection of computer networks, based on the recognition of self/non-self behavior, require unambiguous definitions of all permitted or non-permitted actions in a system. A number of ground-breaking solutions to this problem are proposed [2–6], but no single solution has achieved one hundred percent precision. Nevertheless, real-world applications – e.g. IDS – have the necessity of (a) providing a strong, reliable discrimination between normal and abnormal behavior and (b) maintaining a complete database of “good or bad behavior” to be used by the self/non-self recognition algorithms. The choice of self/non-self behaviors is crucial because some bad behavior not stored in the database may not be recognized as network attacks. More-

[☆] Expanded version of a talk presented at the 3rd International Workshop on Hybrid Artificial Intelligence Systems, HAIS 2008.

[*] Corresponding author.
E-mail addresses: andrea.visconti@unimi.it (A. Visconti), htahayor@scs.ryerson.ca (H. Tahayori).

[1] This research was partially funded by the State of Italy via FIRST (Fondo per gli Investimenti nella Ricerca Scientifica e Tecnologica).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.12.011

over, large databases of self/non-self behaviors entail a substantial degree of slowness that is not acceptable in real-time applications.

1.1. Motivation for adaptive immune system

In order to overcome the problems mentioned above, we have designed a biologically-inspired intrusion detection system based on the concepts of the *adaptive immune system*. Being more slow than the innate one, the acquired immune system is the only one that remembers the previously encountered attacks, recognizes new attacks of unwanted intruders entering the system, and provides a specific response to endogenous, or exogenous attack.

Diagnosing an abnormal behavior of a specific type requires knowing which, if any, set of parameters characterizes the anomaly. This set of parameters is called *antigen signature*. Some such signatures are well-known and can be easily recognized automatically, some are just less well-defined and can be more difficult to be recognized, whilst others are completely unknown.

To this end, we analyze and improve existing solutions in computer security through designing, engineering and testing an intrusion detection system that recognizes anomalous values of parameters of a given system. These anomalous values can be interpreted as a sign of an attack to the system comparable with, e.g. fever in the case of presence of infection in body.

1.2. Our contributions

In this paper, we present a performance-based artificial immune system inspired by the workings of adaptive immune systems that implements flexible and efficient type-2 fuzzy set techniques for analyzing anomalous trends of some system parameters, and enables different countermeasures for different kinds of attack. In particular, our AIS is based on several agents that mimic the behavior of white blood cells. These white blood cells – or lymphocytes – cooperate in order to protect the system against exogenous or endogenous attacks. Every white blood cell is a separate Java^{TM} process that monitors and checks the parameters of the system, and learns to identify the most important subset of system parameters. Such subset, that show an abnormal trend, is examined more closely using an algorithm based on interval type-2 fuzzy set.

Our contributions are summarized as follows.

(i) We suggest a dynamic technique based on interval type-2 fuzzy set paradigm for analyzing system parameters when one or more of them show an anomalous trend. Instead of using type-1 fuzzy set that while its membership function is determined then it is certain and thus no uncertainty would be associated with it, type-2 fuzzy set provides more degrees of freedom to manipulate and minimize the effect of uncertainties in our system [7]. There are a number of sources of uncertainties that make using type-2 fuzzy set reasonable. First, a subset of system parameters may be sufficient for describing a specific anomalous behavior, but which and how many parameters belong to such subset is not well defined. Second, once chosen a specific subset of system parameters, the alarm thresholds for some such system parameters may have different values to different experts. Third, measurements of system parameters may be imprecise because a high workload situation introduces an excessive level of background noise. Fourth, also the real data used for the training phase may be noisy. These four reasons are convincing enough to design the recognition algorithm based on type-2 fuzzy set.

(ii) We engineer a flexible and efficient type-2 fuzzy set based-algorithm which requires (a) constructing an *interval type-2 fuzzy map* for each system parameter through surveying the experts in the filed which will depict how far the changes in the system parameter indicate unwanted behavior, (b) denoting the meanings of the granules *Attack, Suspicious-Attack, Non-Decidable, Suspicious-Non-Attack*, and *Non-Attack* as an intervals, for constituting the frame of reference, (c) constructing type-2 fuzzy map of the system, and finally (d) computing the centroid of the map, (e) identifying the region in which the centroid belongs – i.e. Attack, Suspicious-Attack, Non-Decidable, Suspicious-Non-Attack, and Non-Attack.

1.3. Organization of the paper

In Section 2 we briefly present the state of art of artificial immune system in the field of network security. In Section 3, we discuss the principles of immune systems, in particular, we focus our attention on the adaptive immune systems, while in Section 4, we describe the design and engineering of our artificial immune system. In Section 5, interval type-2 fuzzy set is shortly introduced, while the interval type-2 fuzzy set algorithm for granulating system status is discussed in the Section 6. In Section 7, an intensive testing is presented, and finally, conclusions are provided in Section 8.

2. Related works

In the last decade, in order to improve the efficiency of computer security systems like intrusion detection systems, anti-spam systems, and so on, a number of researches were conducted based on the workings of the biological immune system in particular innate and adaptive immune systems. Many analogies between computer security systems and biological immune systems that can be used for solving practical engineering problems would be find in [1,3,5,8,9].

Interesting approaches suggested in [2,10] introduce the possibility of using a sequences of system calls, executed by running UNIX processes, as discriminator between normal and abnormal behavior. Moreover, references [11–13] discuss the design and testing of Lisys, a LAN traffic anomaly detector that monitors TCP SYN packets for detecting unusual requests, alerting administrator when abnormal connections are detected. If an anomaly is confirmed, the signature of the anomaly is stored; in future an alert will be released whenever the same request is made. In [1], Dasgupta described collecting data on normal and abnormal behaviors in host-based and network-based systems. Such data, collected in a realistic context, contain information that may be used for automatically detecting, analyzing and controlling future anomaly behaviors due to new and unpredictable network attacks. Tarakanov et al. in [4] described a formal model for an intrusion detection system inspired by the workings of the acquired immune system. The introduced artificial immune system in [4] is based on a rigorous mathematical approach, applied the singular value decomposition [14] to the matrix of connection logs, mapping the users' requests into a real two-dimensional vector space. Authors argued that similar self/non-self requests lump together. Aickelin and Cayzer [15] and Aickelin and Bentley [16] have suggested interesting approaches based on the Danger Theory [17]. This new theory has shifted control of immunity to the tissues that need protection, hypothesizing that such tissues send signals to the immune system, which in turn will choose the appropriate immune response. Inspired by the behavior of innate immune system, Pagnoni and Visconti in [18] illustrated an artificial immune system based on the working of the innate immune system. The authors argued that the main idea of an intrusion detection system is not to recognize and kill a specific intruder in the most effective way, but rather to find and kill any intruder as quickly as possible. In addition to the artificial immune systems previously mentioned, authors in [6,19,20] suggested applying the negative selection algorithm for detecting network intrusions.

3. Biological immune system

Biological Immune System is an automatic, but complex system to protect organisms against invading enemies. Its tasks are to recognize, neutralize or degrade unwanted invaders. To succeed in this challenge, several mechanisms are evolved over millions of years for recognizing enemies that infect the living organisms and cause diseases. Biological immune systems [21] draw up several lines of defense to protect living organisms from all kinds of foreign, and potentially dangerous, intruders. This defense system includes chemical and physical barriers, innate immune system and adaptive immune system. *Chemical* and *physical barriers* provide the first line of defense in the fight against invaders. These barriers try to protect the body against pathogens that enter an organism, and consequently reduce the probability that such pathogens will lead to an illness. When they fail to stop unwanted intruders, invaders are attacked by the cells of the second line of defense: the *innate immune system* [21]. The innate immune system attacks invaders in a generic way with no necessity of previous exposure to them, and activate the adaptive immune system. The *adaptive* or *acquired immune system* [21] provides the third line of defense in the fight against intruders. Its ability to kill invaders is based on the capacity of recognizing several kinds of pathogens and remembering specific antigen signatures after the resolution of the infection. The cells involved in the acquired reaction are *lymphocytes* such as memory B cell, killer T cell, helper T cells, and so on. When activated, these lymphocytes are able to recognize unwanted invaders, mount a specific immune response against them, and remember specific signatures for unwanted invaders. Unfortunately, these mechanisms are not perfect and sometimes fail.

4. Design and engineering of artificial immune system

We designed and engineered our AIS based on the discrimination between self/non-self system behaviors, focusing on the observation that getting into a system without leaving any track is virtually impossible. We search for these tracks by monitoring the system parameters. In order to identify such tracks, that are the fingerprint of an attack, the values of different system parameters were surveyed, stored and analyzed. Some of them did not show any significant change during an attack, while others did.

4.1. Recognition phase

The proposed artificial immune system consists of a set of processes which runs on a server, and acts as *artificial lymphocytes*. Such processes are autonomous agents that act and cooperate as biological lymphocytes in order to discover suspicious values of system parameters and face external attacks. *Artificial helper T-cells* spent their life time collecting the sample data of system parameters, computing the mean and the standard deviation of such data, comparing the actual values to those previously stored into memory, alerting the AIS if system parameters are considerably changed, and updating the previous values with new values. If some monitored parameter show an erratic trend, or unrealistic peaks, system parameters are inspected more closely – see details in Section 6 – and the artificial immune system is set up in alert state. When the system is in alert state, *artificial memory B-Cells* are able to identify if the actual abnormal behavior was, or not, previously identified. If it was, *Artificial killer T-cells* will take proper action – e.g. in the presence of a denial of service attacks, killer T-cells deny unwanted requests, banning the IP addresses of the senders, etc. – otherwise, if the attack type is unknown, a notification is sent to the system administrator, alerting him of the security threat and increasing the sensibility of the system. All digital cells are separate Java^{TM} processes, so if one process crashes it will not affect the working of the artificial immune system. Designing the system as a set of different processes offered greater security and stability, at the price of a more difficult communication between processes. In order to solve this problem, we implemented a communication protocol. When authenticated, processes can communicate with each other by calling specific functions. Using these functions, artificial lymphocytes are able to stimulate groups of cells, or the entire artificial immune system, when the system is under attack.

4.2. Training phase

Before activation, our artificial immune system passes through a *training phase* that defines the number of running lymphocytes, sets up the parameters of all processes, and selects the most sensible helper T-cells. The number of running lymphocytes is not constant, but is optimized experimentally because it depends on the hardware features and the average workload of the server. Such number can vary between a lower and an upper bound, improving the performance of the AIS when the system is under attack. Furthermore, if the number of artificial lymphocytes falls under the optimized threshold, new processes are automatically created. The training phase is also responsible for setting up the parameters of all processes. Indeed, the digital lymphocytes learn to identify self/non-self behaviors, analyzing the values of system parameters under the supervision of an expert. Finally, the AIS randomly generates Helper T-cells and selects only those that are safe and most sensible, using a negative selection algorithm and a fitness evaluation [22]. After the training phase, the artificial immune system is ready to be activated.

5. Interval type-2 fuzzy set

As defined by Zadeh in [23], a type-2 fuzzy set is a fuzzy set with fuzzy membership function. In more formal form, a fuzzy set of type-2 $\tilde{A}$ in a set X is the fuzzy set which is characterized by a fuzzy membership function $\mu_{\tilde{A}}: X \rightarrow [0, 1]^J$ with $\mu_{\tilde{A}}(x)$ being a fuzzy set in [0, 1] (or in the subset J of [0, 1]) denoting the fuzzy membership grade of x in $\tilde{A}$ [24]. Adopting the notions used in [7,25–27],

$$\tilde{A} = \sum_{x \in X} \sum_{u \in J_x} \mu_{\tilde{A}} \frac{(x, u)}{(x, u)} = \sum_{x \in X} \left[\sum_{u \in J_x} f_x(u)/u \right] /x \quad (1)$$

denotes a type-2 fuzzy set $\tilde{A}$ over X, where $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$, $u \in J_x \subseteq [0, 1]$, and equivalently $f_x(u) \in [0, 1]$ and, $u \in J_x \subseteq [0, 1]$. Here, x is *primary variable*, J_x represents the *primary membership of x, and $f_x(u)$ is referred to as secondary grade. Footprint of Uncertainty of a type-1 fuzzy set* $_{\tilde{A}}$ *is defined as*

$$FOU(\tilde{A}) = \bigcup_{x \in X} J_x = \bigcup_{x \in X} [\underline{u}(x), \bar{u}(x)] \quad (2)$$

A discrete interval type-2 fuzzy set $\tilde{A}$ is characterized by

$$\tilde{A} = \sum_{x \in X} \left[\sum_{u \in J_x} 1/u \right] /x \quad (3)$$

Be noticed that an interval type-2 fuzzy set can be fully described by its *Footprint Of Uncertainty* (FOU) [28], regarding to the uniform distribution sitting on top of the FOU. FOU of an interval type-2 fuzzy set $\tilde{A}$, on the other hand would be fully characterized by two type-1 fuzzy sets, named *Upper Membership Function* (UMF) and *Lower Membership Function* (LMF), respectively defined

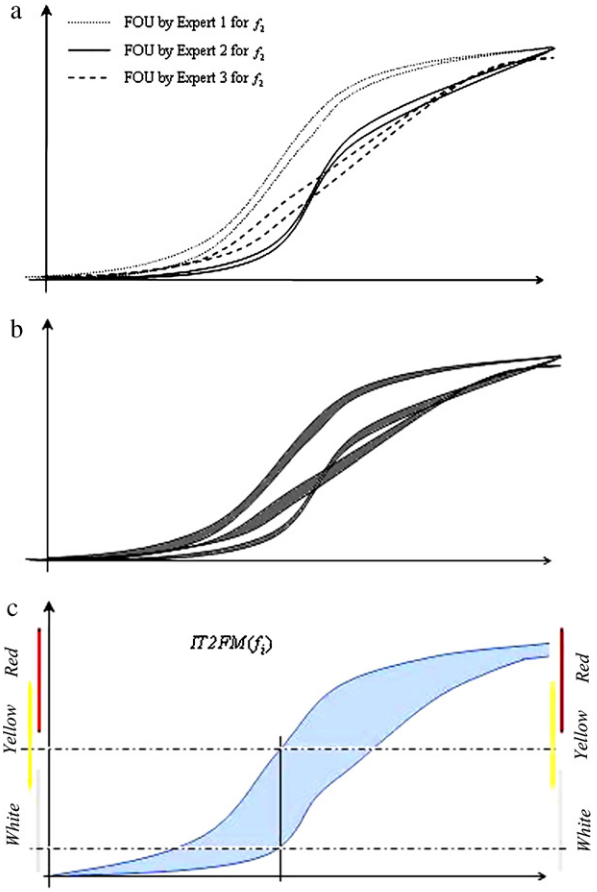

Fig. 1. The figures (a)–(c) show how to reach to an IT2FM of system parameter f_i: (a) three FOUs given by three experts; (b) aggregating and filling the FOUs; (c) interval type-2 fuzzy map of the system parameters f_i.

as $\text{UMF}(\tilde{A}) = \sum_{x\in X} \bar{u}(x)/x$ and $\text{LMF}(\tilde{A}) = \sum_{x\in X} \underline{u}(x)/x$. In other words, FOU($\tilde{A}$) is bounded by UMF($\tilde{A}$) and LMF($\tilde{A}$).

Centroid of a type-1 fuzzy set $A = \sum_{x\in X}\mu_A(x)/x$ is defined as

$$C_A = \frac{\sum_{x\in X} x\mu_A(x)}{\sum_{x\in X}\mu_A(x)} \tag{4}$$

Using Zadeh's Extension Principle [23], and knowing that the secondary grades in interval type-2 fuzzy sets are all equal to 1, the centroid of interval type-2 fuzzy set $\tilde{A}$ is defined as

$$C_{\tilde{A}} = \sum_{u_1\in J_{x_1}} \cdots \sum_{u_N\in J_{x_N}} \frac{1}{\sum_{i=1}^{N} x_i u_i / \sum_{i=1}^{N} u_i} \tag{5}$$

which is an interval that can be simply represented as $C_{\tilde{A}} = [c_l, c_r]$. Although yet no closed form formula for calculating c_l and c_r exists, however, Karnik and Mendel have proposed an iterative algorithm – called *KM Algorithm* – for calculating the exact bounds [29,30]. Moreover, in [31,32], Mendel and Wu have proved that the end points of the centroid of an interval type-2 fuzzy set are bounded – i.e. $\underline{c_l} \le c_l \le \overline{c_l}$, and $\underline{c_r} \le c_r \le \overline{c_r}$. Toward the aim, since LMF($\tilde{A}$) and UMF($\tilde{A}$) are type-1 fuzzy sets, we can compute the centroids $C_{LMF(\tilde{A})}$ and $C_{UMF(\tilde{A})}$ using (4). So the bounds would be given as

$$\overline{c_l} = \min\{C_{LMF(\tilde{A})}, C_{UMF(\tilde{A})}\} \tag{6}$$

$$\underline{c_r} = \max\{C_{LMF(\tilde{A})}, C_{UMF(\tilde{A})}\} \tag{7}$$

$$\underline{c_l} = \overline{c_l} - \frac{\sum_{i=1}^{N}(\bar{u}(x_i) - \underline{u}(x_i))}{\sum_{i=1}^{N}\bar{u}(x_i)\sum_{i=1}^{N}\underline{u}(x_i)} \times \frac{\sum_{i=1}^{N}\underline{u}(x_i)(x_i - x_1)\sum_{i=1}^{N}\bar{u}(x_i)(x_N - x_i)}{\sum_{i=1}^{N}\underline{u}(x_i)(x_i - x_1) + \sum_{i=1}^{N}\bar{u}(x_i)(x_N - x_i)} \tag{8}$$

$$\overline{c_r} = \underline{c_r} + \frac{\sum_{i=1}^{N}(\bar{u}(x_i) - \underline{u}(x_i))}{\sum_{i=1}^{N}\bar{u}(x_i)\sum_{i=1}^{N}\underline{u}(x_i)} \times \frac{\sum_{i=1}^{N}\bar{u}(x_i)(x_i - x_1)\sum_{i=1}^{N}\underline{u}(x_i)(x_N - x_i)}{\sum_{i=1}^{N}\bar{u}(x_i)(x_i - x_1) + \sum_{i=1}^{N}\underline{u}(x_i)(x_N - x_i)} \tag{9}$$

Valuable resources for detailed discussions on interval type-2 fuzzy sets would be [26,33–39].

6. Interval type-2 fuzzy set based algorithm

There are some system parameters that their changes may have indication of an attack. In the sequel, for simplicity, we suppose to have *M* system parameters, *M* artificial lymphocytes, each of which checks a system parameter, and a shared vector *V*. In reality, for security policies, we engineered artificial lymphocytes that redundantly check all the parameters and, each of them, have a private vector *V*. The designed independent-artificial lymphocytes are to monitor the system parameters $F = \{f_1, \ldots, f_M\}$ and report their changes. We show the changes of the system parameters as a vector $V = [v_1, \ldots, v_M]$, where v_i, $i \in \{1, \ldots, M\}$ indicates the percentage of the changes in the parameter f_i with respect to the latest measurement; be aware that the elements of the vector *V* do not necessarily update simultaneously.

We have surveyed a number of experts on their intuitions on how far the changes in different system parameters may be an indication of an attack. Using the *person MF approach* discussed in [28] that enable modeling intra and inter uncertainties about the effects of changes in the system parameters, primarily we asked each expert to provide a broad-brush FOU for each parameter in the domain of $X = [0, 100]$ – it is argued that it is more natural and feasible than asking for a crisp number [28]. Aggregating all equally weighted experts' FOUs on each parameter through mathematical operation of union and in order to get to a parsimonious one, using the advice of [28] and filling the results of the union, we reached to an *interval type-2 fuzzy map* (IT2FM) of each parameter. So, using (3) the IT2FM of the system parameter f_i would be extensively shown as

$$IT2FM\,(f_i) = \sum_{x}[1/[\bar{u}_{f_i}(x), \underline{u}_{f_i}(x)]]/x \tag{10}$$

that we simply show as

$$\begin{aligned} IT2FM(f_i) &= \sum_{x}[\underline{u}_{f_i}(x), \bar{u}_{f_i}(x)]/x \\ &= \sum_{x}\tilde{\mu}_{f_i}(x)/x \end{aligned} \tag{11}$$

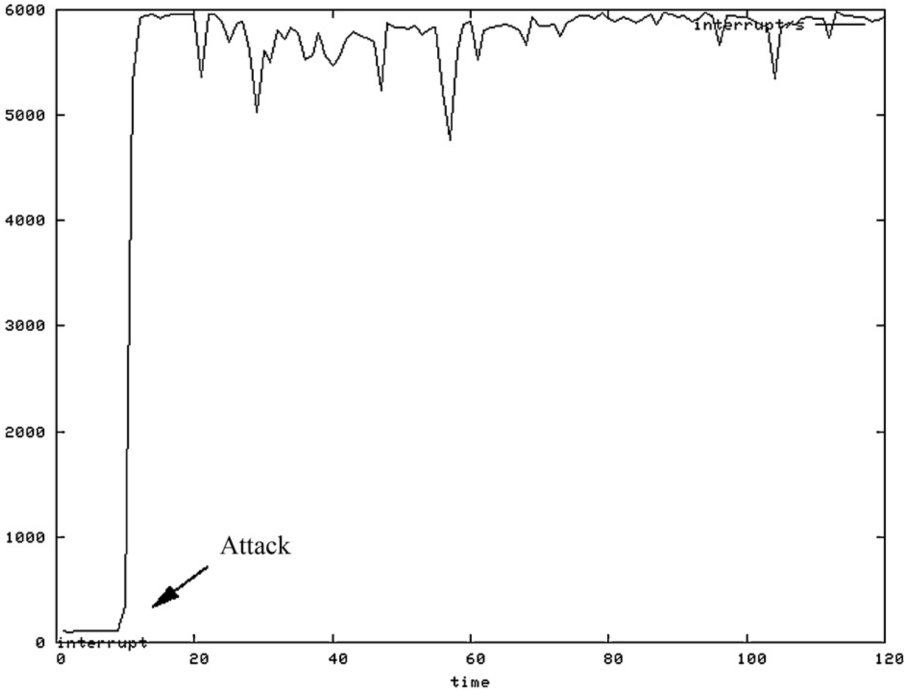

Fig. 2. Interrupts per second during a DDOS attack without legitimate traffic.

where $\tilde{\mu}_{f_i}(x) = [\underline{u}_{f_i}(x), \bar{u}_{f_i}(x)] \subseteq [0, 1]$ represent the uncertain attack-indication of x percent change in the system parameter f_i. We also assumed that $\tilde{\mu}_{f_i}(x) = \tilde{\mu}_{f_i}(0)$, $x < 0$ and $\tilde{\mu}_{f_i}(x) = \tilde{\mu}_{f_i}(100)$, $x > 100$. Fig. 1 shows an example of driving IT2FM of the system parameter f_i through person MF approach.

In order to capture the effective role of each factor f_i in the process of intrusion detection, we have used the vector $Weight = \{w_1, \ldots, w_M\}$ that $w_i \in [0, 1]$ corresponds to the importance of the factor f_i in determining if the system is under attack. Initially, all the weights are set to one, indicating they are all equally highly important in determining the status of the system. The weights will be tuned in each round of the algorithm whereas the closer the weight to the unity, the more effective and important the factor is in the process of detection.

To decide if a change of a system parameter needs a special attention, based on the experts recommendations, we devised three non-necessarily pair-wise disjoint regions in [0, 1], namely $White = [\underline{r}_1, \bar{r}_1]$, $Yellow = [\underline{r}_2, \bar{r}_2]$ and $Red = [\underline{r}_3, \bar{r}_3]$. The new measurement of the system parameter f_i cause v_i to be refreshed, then we find the region to which $\tilde{\mu}_{f_i}(v_i)$ is more close by calculating their

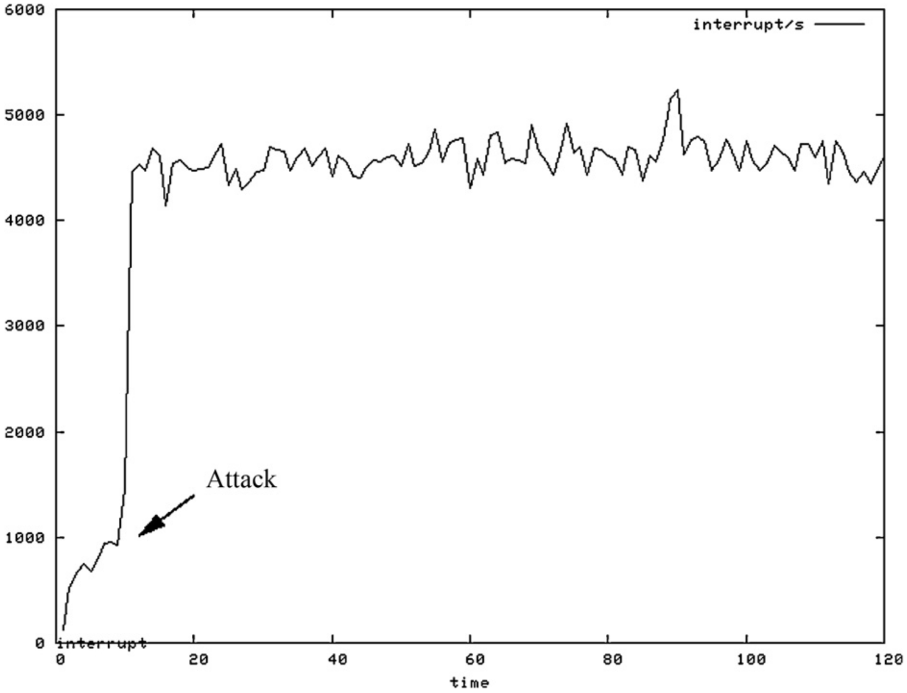

Fig. 3. Interrupts per second during a DDOS attack under moderate traffic.

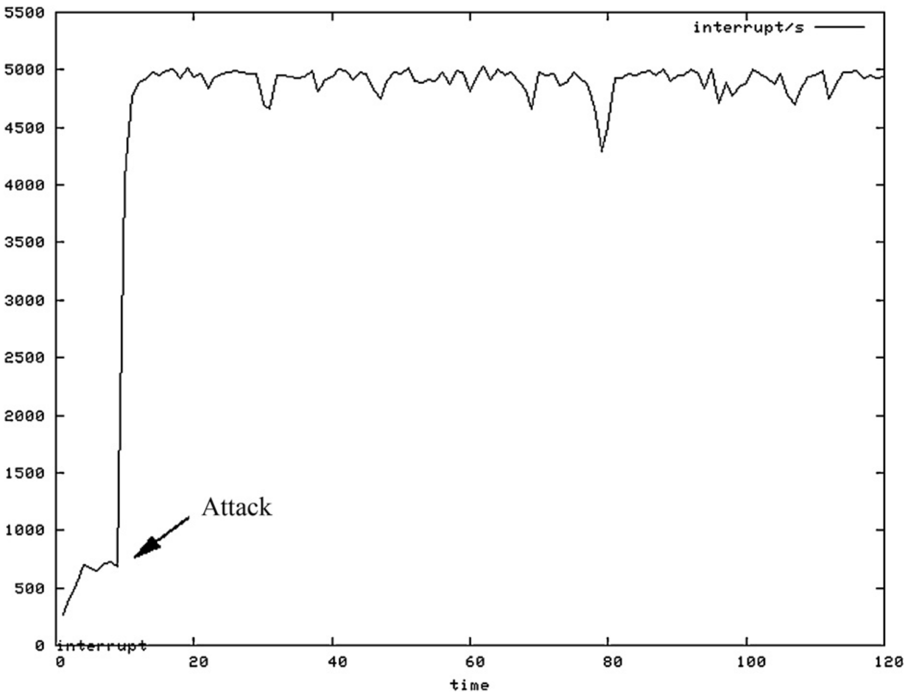

Fig. 4. Interrupts per second during a DDOS attack under intense traffic.

distance, i.e. $\tilde{\mu}_{f_i}(v_i) \approx [\underline{r}_i, \bar{r}_i]$ where

$$d(\tilde{\mu}_{f_i}(v_i), [\underline{r}_i, \bar{r}_i]) = Inf(d(\tilde{\mu}_{f_i}(v_i), White), d(\tilde{\mu}_{f_i}(v_i), Yellow), d(\tilde{\mu}_{f_i}(v_i), Red)) \quad (12)$$

The reasonable distance function used [40] is,

$$d([\underline{a}, \bar{a}], [\underline{b}, \bar{b}]) = Max(|\underline{a} - \underline{b}|, |\bar{a} - \bar{b}|) \quad (13)$$

$\tilde{\mu}_{f_i}(v_i)$ being *White*, indicates that changes of the system parameter f_i has not been abnormal, whereas in the case of $\tilde{\mu}_{f_i}(v_i) = Yellow$ a warning will be issued and the corresponding agent will measure the f_i instantly and will *add* the new registered changes to v_i instead of replacing it. If it was backed to the *White* position, then the monitoring of the parameter will be set to normal. However, if it is locked for a long time to the *Yellow* position, or positioned in the *Red* region, a command will be issued to all other agents whose system parameters had not been in *Red* region, to measure their corresponding parameters instantly and refresh their related cell in *V*, based on which the *Interval Type-2 Fuzzy Map* of the System – *IT2FM(sys)* – will be built. Initially, *IT2FM(sys)* is set to null, i.e. *IT2FM(sys)*= $\sum_x 0/x$. When the system is put in the *Red* position and after *V* is refreshed the map will be built iteratively as

$$\tilde{\mu}_{sys}(v_i) \leftarrow Hull(\tilde{\mu}_{sys}(v_i) \cup (\tilde{\mu}_{f_i}(v_i) \times w_i)) \quad (14)$$

where $Hull([\underline{a}, \bar{a}] \cup [\underline{b}, \bar{b}]) = [Min(\underline{a}, \underline{b}), Max(\bar{a}, \bar{b})]$ and $i \in \{1, \ldots, M\}$. To make the final decision, i.e. whether there has been an attack or not, using (6)–(9) we calculate the boundaries of the centroid of *IT2FM(sys)* and compare the result with the predefined granules *G* = {*Non-Attack*, *Suspicious-Non-Attack*, *Non-Decidable*, *Suspicious-Attack*, *Attack*}. The granules are defined through first surveying the experts using the same method of *person MF approach*, aggregating all the equally weighted individual *person MFs* and then filling them and finally drawing their centroid using the KM-Algorithm [29,30]. More precisely, we can show *G* as

$$G = \{g_i | i = 1, \ldots, 5\}, \quad g_i \triangleq [\underline{cg}_i, \bar{c}g_i] \quad (15)$$

To draw the decision, we calculate the distance of the centroid bounds of *IT2FM*(*sys*), i.e. $C'_{IT2FM(sys)} = [[\underline{c}_l, \bar{c}_l], [\underline{c}_r, \bar{c}_r]]$ with each and every granules in *G* to find the minimum distance; the granule with the minimum distance to the centroid bound indicate the status of the system, more formally,

$$System\_Status \approx g_l, \quad g_l \in G \quad (16)$$

given

$$\begin{aligned} d(C'_{IT2FM(sys)}, g_l) &= \inf_i (d(C'_{IT2FM(sys)}, g_i)) \\ &= \inf_i (Max(d([\underline{c}_l, \bar{c}_r], g_i), d([\bar{c}_l, \underline{c}_r], g_i))) \end{aligned} \quad (17)$$

If the system is found to be under attack, the weights of all the factors involving in the process that were measured to be in the red region will be increased by $\alpha \in [0, 1]$ while those in yellow by $\alpha * 0.5$. However while the system determined to be *Suspicious-Attack* the factors found in the red and yellow regions will be increased by $\beta = \alpha * 0.5$ and $b * 0.5$ respectively. In any case afterwards the weight vector will be normalized. In the case of wrong detection, subtraction will take place and hence, the weight of misleading factors will be reduced. Such, the learning phase takes effect and the most important factors that can be relied for determining system status will be nominated.

7. Testing

Being engineered, the ability of the proposed AIS to recognize attacks without human intervention was tested. All tests were made on a dual Intel® PIII® server with 1.5Gb of RAM on which Gentoo Linux [41] was installed; running Apache 2.0.59 with PHP 5.2.5, Mysql 5.0.40 and OpenSSH 4.7. Several types of attacks were tested:

(a) Port scanning (NMAP);
(b) Remote buffer overflow;
(c) Distributed Denial of Service (DDOS);
(d) Dictionary-attack against SSH authentication.

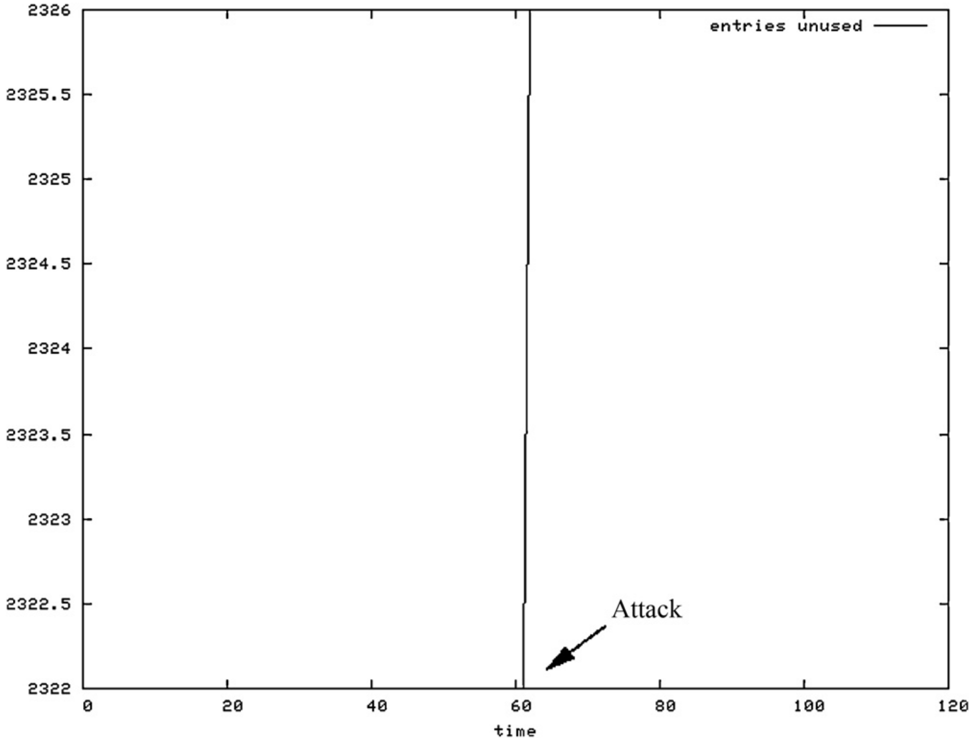

Fig. 5. Unused cache entries during a buffer overflow without legitimate traffic.

These types of attacks were chosen because (a) testings were based on the observation that all significant network attacks are preceded by preparatory small-scale intrusions meant to collect information for bypassing firewalls, login policies and access controls, (b) such attacks are the most common, and (c) each one of them have very different performance fingerprint, e.g. a port scanning such as NMAP has very light performance fingerprint, whereas a DDOS attack has not.

In order to simulate a real-world situation, a fake institutional website was implemented and the system was tested with different amounts of traffic:

(1) *None*– no one, or a small number of users, surf the website. This profile has been used in order to provide baseline data.
(2) *Moderate*– an average number of users surf the website.
(3) *Intense*– a really high amount of users surf the website, making a huge number of self requests. This profile has been used in

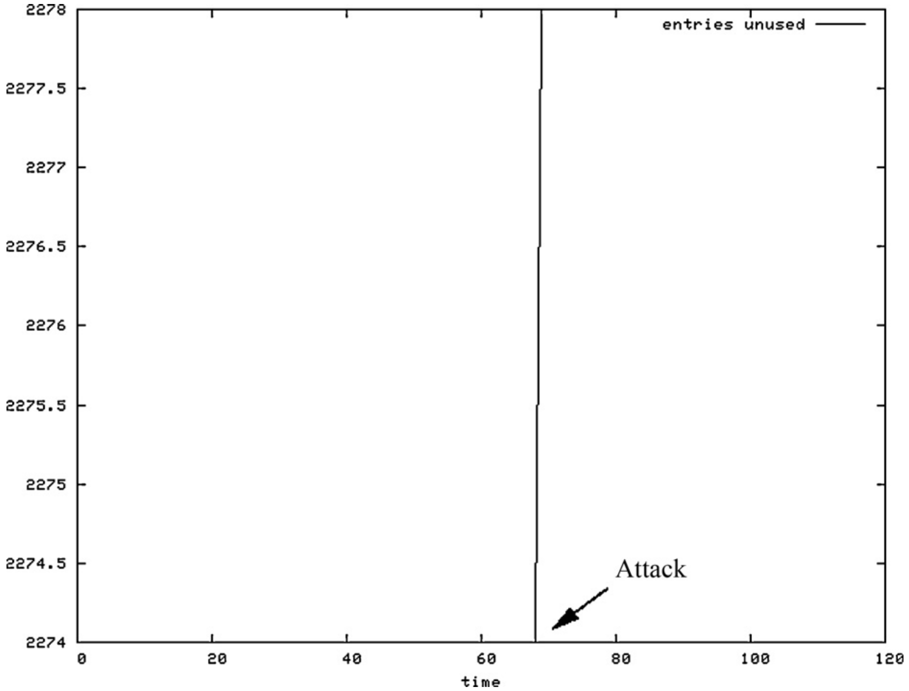

Fig. 6. Unused cache entries during a buffer overflow attack under moderate traffic

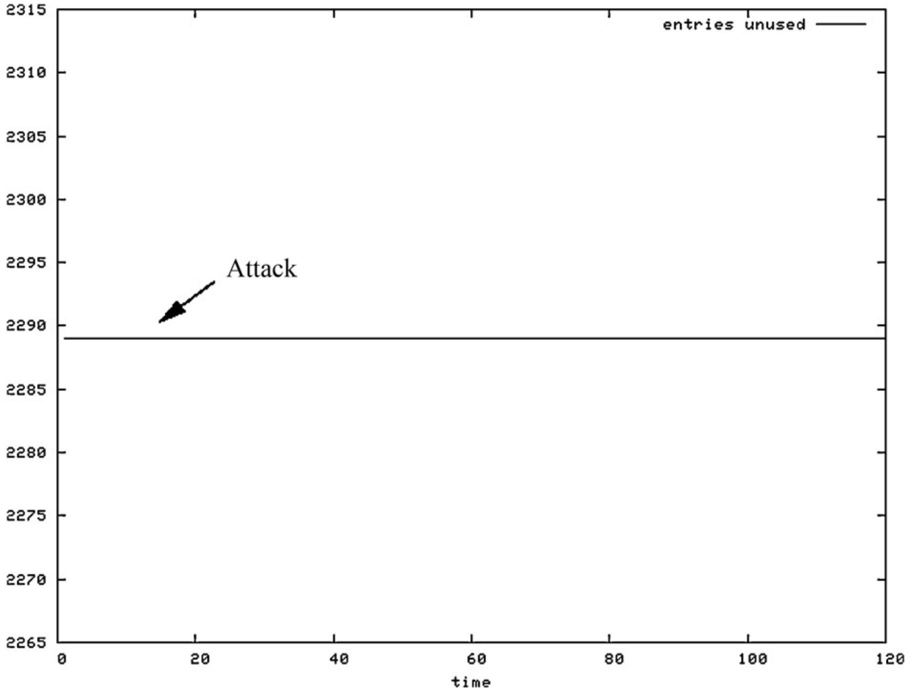

Fig. 7. Unused cache entries during a buffer overflow attack under intense traffic.

order to evaluate the behavior of the system under severe stress condition.

The population of digital cells used for the testing is as follows:

- 3-10 killer T-cells,
- 5-10 helper T-cells,
- 3-10 memory B cells.

Even if the test server was not very powerful, it could handle a reasonable amount of processes because the most expensive agent operations were highly optimized. The traffic has been simulated with JMeter [42], a stress testing tool for web applications provided by the Apache software foundation.

As mentioned in the previous section, our AIS monitors the system parameters in order to identify unusual patterns that may be related to unwanted behaviors. For example, Figs. 2–4 show the behavior of a system parameter during a DDOS attack. Analyzing the values of this parameter, an unusual pattern in the number of interrupt requests can be recognized.

Figs. 2–4 show clearly that DDOS attack is rather easy to spot, because it largely affects the system performance. Unfortunately, as can be seen in Figs. 5–7, not all kinds of attack are always so easy to recognize. Such figures represent the unused cache entries during a buffer overflow attack while a small, average, and large number of users, respectively, surf the website. In the first two cases – Figs. 5 and 6, low and moderate workload – the artificial immune system is able to recognize the anomaly. In the last case – Fig. 7, high workload – the artificial immune system fails. It is easy to see that a high workload situation may introduce an excessive level of background noise, decreasing the ability of the system to recognize anomalous behaviors. Such situations affect negatively the performance of the artificial immune system, enhancing the risk of false positive.

Preliminary testing show that attacks with strong fingerprint – DDOS, buffer overflow, and brute force attack – can be more, or less easily recognized independent of the workload of the system. Furthermore, first experiments show that recognizing signatures' attacks in the presence of many net-surfing users can be increasingly difficult for specific kinds of attack. This is the case of buffer overflow attack, for which it is easy to see that high workload situations can introduce an excessive level of background noise affecting negatively the measurements of system parameters.

In other cases, recognizing signatures' attacks in the presence, or absence, of many net-surfing users can be almost impossible. This is the case of the port scanning attack. This attack is recognized with a low level of percentage because it does not significantly affect the system parameters, and its feeble fingerprint can be easily mistaken for legitimate traffic or regular background noise.

8. Conclusions and future works

The suggested artificial immune system is an intrusion detection system based on the idea of equipping servers with the technologically equivalence of an acquired immune system. To this end, our AIS monitors and analyzes a set of system parameters to check for anomalous behaviors. Although the development of the type-2 fuzzy set based algorithm is at its first release, the preliminary testing revealed that the proposed AIS is able to quickly detect anomalous behaviors previously encountered, deny proliferation of foreign processes by killing dangerous processes before they will widely used, and recognize attacks with strong fingerprint such as denial of service, dictionary attacks and buffer overflow attacks. Moreover, the learning phase helps the AIS to recognize the most important factors involving in specific kinds of attack.

On the other hand, preliminary testing has shown that in presence of many net-surfing users recognizing attacks with a feeble fingerprint can be increasingly difficult, because a high workload situation may introduce an excessive level of background noise, enhancing the risk of false positive. This is the case of port scanning attacks, where in order to avoid a large number of false positives, we have to lower the sensibility of the system, affecting the recognition phase.

It should be stressed that yet, no single method, biological or artificial, can achieve one hundred percent precision. For these reasons, the suggested system is not meant to replace firewalls, login policies, or antivirus because it cannot blocks every kind of attack.

The proposed artificial immune system should be used in conjunction with other complementing technologies either biologically inspired or not.

Our future works will be devoted to extend and compare actual interval type-2 fuzzy set based algorithm with a general type-2 fuzzy set based one.

References

[1] D. Dasgupta, Advances in artificial immune systems, IEEE Computational Intelligence Magazine 1 (4) (2006) 40–49.

[2] S. Hofmeyr, A. Somayaji, S. Forrest, Intrusion detection using sequences of system calls, Journal of Computer Security 6 (3) (1998) 151–180.

[3] D. Dasgupta, Immune-based intrusion detection system: a general framework, in: Proceedings of the 22nd National Information Systems Security Conference, 1999.

[4] A.O. Tarakanov, V.A. Skormin, S.P. Sokolova, Immunocomputing: Principles and Applications, Springer-Verlag, New York, 2003.

[5] S. Forrest, S. Hofmeyr, A. Somayaji, T. Longstaff, A sense of self for UNIX processes, in: Proceedings of the IEEE Symposium on Research in Security and Privacy, 1996.

[6] S. Forrest, M.R. Glickman, Revisiting LISYS: parameters and normal behavior, in: Proceedings of the Congress on Evolutionary Computation, 2002, pp. 1045–1050.

[7] J.M. Mendel, R.I. John, Type-2 fuzzy sets made simple, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 117–127.

[8] P. D'haeseleer, S. Forrest, P. Helman, An immunological approach to change detection: algorithms, analysis and implication, in: Proceedings of the IEEE Symposium on Computer Security and Privacy, 1996.

[9] S. Forrest, S. Hofmeyr, A. Somayaji, Computer immunology, Communication of ACM 40 (10) (1997) 88–96.

[10] C. Warrender, S. Forrest, B. Pearlmutter, Detecting intrusions using system calls: alternative data models 1999, in: Proceedings off the IEEE Symposium on Security and Privacy, 1999, pp. 133–145.

[11] S. Hofmeyr, S. Forrest, Architecture for an artificial immune system, Evolutionary Computation 8 (4) (2000) 443–473.

[12] S. Hofmeyr, An immunological model of distributed detection and its application to computer security, PhD Thesis, University of New Mexico, 1999.

[13] J. Balthrop, S. Forrest, M. Glickman, Revisiting lisys: parameters and normal behavior, in: Proceedings of the Congress on Evolutionary Computation, 2002, pp. 1045–1050.

[14] R. Horn, C. Johnson, Matrix Analisys, Cambridge University Press, 1986.

[15] U. Aickelin, S. Cayzer, The danger theory and its application to artificial immune systems, in: Proceedings of 1st International Conference on Artificial Immune Systems, 2002.

[16] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, J. McLeod, Danger theory: the link between AIS and IDS? LCNS 2787, 2003.

[17] C. Anderson, P. Matzinger, Danger: the view from the bottom of the cliff, Seminars in Immunology 12 (3) (2000) 231–238.

[18] A. Pagnoni, A. Visconti, An innate immune system for the protection of computer networks, in: Proceedings of the 4th International Symposium on Information and Communication Technologies, 2005, pp. 63–68.

[19] F. Gonzalez, D. Dasgupta, An immunogenetic technique to detect anomalies in network traffic, in: Proceedings of the International Conference Genetic and Evolutionary Computation, 2002.

[20] J. Kim, P. Bentley, The human immune system and network intrusion detection, in: Proceedings of 7th European Congress on Intelligent Techniques and Soft Computing, 1999.

[21] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter, Molecular Biology of the Cell, 5th ed., Garland Science Publishing, London, 2007.

[22] L.R. de Castro, J. Timmis, Artificial Immune Systems: A New Computational Intelligence Paradigm, Springer-Verlag, 2002.

[23] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-I, Information Science 8 (1975) 199–249.

[24] M. Mizamoto, K. Tanaka, Some properties of fuzzy set of type-2, Information and Control 31 (1976) 312–340.

[25] J.M. Mendel, R.I. John, Footprint of uncertainty and its importance to type-2 fuzzy sets, in: Proceedings of the 6th IASTED International Conference Artificial Intelligence and Soft Computing, 2002, pp. 587–592.

[26] J.M. Mendel, R.I. John, F. Liu, Interval type-2 fuzzy logic systems made simple, IEEE Transactions on Fuzzy Systems 14 (6) (2006) 808–821.

[27] J.M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice-Hall, Upper Saddle River, NJ, 2001.

[28] J.M. Mendel, Computing with words and its relationships with fuzzistics, Information Sciences 177 (2007) 988–1006.

[29] N.N. Karnik, J.M. Mendel, An introduction to type-2 fuzzy logic systems, USC Report, University of Southern California, 1998.

[30] N.N. Karnik, J.M. Mendel, Centroid of a type-2 fuzzy set, Information Sciences 132 (2001) 195–220.

[31] H. Wu, J.M. Mendel, Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 10 (5) (2002) 622–639.

[32] J.M. Mendel, H. Wu, Centroid uncertainty bounds for interval type-2 fuzzy sets: forward and inverse problems, in: Proceedings of IEEE International Conference on Fuzzy Systems, 2004, pp. 947–952.

[33] H. Bustinc, P. Burillo, Mathematical analysis of interval-valued fuzzy relations: application to approximate reasoning, Fuzzy Sets Systems 113 (2) (2000) 205–219.

[34] Q. Liang, J.M. Mendel, Interval type-2 fuzzy logic systems: theory and design, IEEE Transactions on Fuzzy Systems 8 (5) (2000) 535–550.

[35] I.B. Türkeşen, Interval valued fuzzy sets based on normal forms, Fuzzy Sets and Systems 20 (2) (1986) 191–218.

[36] I.B. Türkeşen, Interval valued fuzzy sets and 'Compensatory AND', Fuzzy Sets and Systems 51 (3) (1992) 295–307.

[37] I.B. Türkeşen, Interval valued fuzzy sets and fuzzy connectives, Journal of Interval Computations 4 (1993) 125–142.

[38] I.B. Türkeşen, Non-specificity and interval valued fuzzy sets, Fuzzy Sets and Systems 80 (1) (1996) 87–100.

[39] I.B. Türkeşen, Belief, plausibility and probability in interval-valued type 2 fuzzy sets, International Journal of Intelligent Systems 19 (2004) 681–699.

[40] E.R. Moore, Interval Analysis, Prentice-Hall, Englewood Cliff, NJ, 1966.

[41] Gentoo Linux, available at: http://www.gentoo.org/, April 2008.

[42] Apache JMeter, available at: http://jakarta.apache.org/jmeter/, April 2008.