Immune-Inspired Algorithm For Network Intrusion Detection

Ameur Bennaoui, Rabah Hachemani, Belkacem Kouninef
Institut National des Télécommunications et Technologies de l'Information et la Communication INTTIC, Oran Algérie
ameur_bennaoui@yahoo.fr, hachemani@yahoo.fr, bkoninef@ito.dz

ABSTRACT: *The central challenge with computer security is determining the difference between normal and potentially harmful activity. A promising solution is emerging in the form of Artificial Immune Systems (AIS). These include the theories regarding how the immune system responds to pathogenic material. This paper takes relatively new theory: the Danger theory and Dendritic cells, and explores the relevance of those to the application domain of security and evaluating on the Kdd'99 data.*

Keywords: Artificial immune systems, Dendritic cell Algorithm (DCA), Security, Intrusion detection

Received: 11 March 2010, Revised 31 March 2010, Accepted 4 April 2010

1. Introduction

Recently, Intrusion Detection Systems (IDS) have been used in monitoring attempts to break security, which provides important information for timely countermeasures. Intrusion detection is classified into two types: misuse intrusion detection and anomaly intrusion detection. Misuse intrusion detection uses well-defined patterns of the attack that exploit weaknesses in the system and application software to identify the intrusions. These patterns are encoded in advance and used to match against the user behavior to detect intrusion. Anomaly intrusion detection identifies deviations from the normal usage behavior patterns to identify the intrusion.

Artificial Immune Systems (AIS) are algorithms inspired by the behavior of the human immune system. The biological immune system tries to protect the body from the attack of any invading pathogens like bacteria and viruses. AIS have been applied to problems in computer security since their initial development in the mid-1990.

Artificial Immune Systems (AIS) have become an increasingly popular computational intelligence paradigm. Inspired by the mammalian immune system, AIS seek to use observed immune components and processes as metaphors to produce systems that encapsulate a number of desirable properties of the natural immune system. These systems are then applied to solve problems in a wide variety of domains [03]. There are a number of motivations for using the immune system as inspiration for data mining; these include recognition, diversity, memory, self-regulation, dynamic protection and learning

Currently, the majority of AIS encompasses two different types of immune inspired algorithms based on anomaly detection, namely negative selection (T-cell based), and Dendritic cell algorithm [6][1].

2. The Dendritic Cell Algorithm – DCA

A recent addition to the AIS family is the Dendritic Cell Algorithm (DCA) implemented by Greensmith et al. [6][10]. DCA is inspired by the function of the Dendritic Cells (DCs) of the innate immune system and uses principles of a key novel theory in immunology and termed as the danger theory described by Matzinger [9]. The danger theory, according to Matzinger suggests that the DCs are the first line of defense against invaders and the response is generated by the immune system upon the receipt of molecular information which indicates the presence of stress or damage in the body. The DCA has been discussed in details in [11][12][13] [14] [15]. The DCA is viewed for an extended application using mathematical perspective in [16]. *Thomas Stibor* et al have shown that by representing the signal processing phase of the algorithm using the dot product, it is shown that the signal processing element of the DCA is actually a collection of linear classifiers. In {17], *Anthony Kulis* and *Shahram Rahimi* found that the DCA is a highly effective tool in finding anomalies in data sets.

3. Algorithm Overview

Thus DCA is stated to have potential applications in many platforms.
In this section we provide an overview of the operation of the algorithm using a modified version of the one proposed by Yousof Al-Hammadi in [2]:

When viewed from a computational perspective, DCs are anomaly detector agents, which are responsible for data fusion and generating appropriate actions in response to the attack in the human body. In nature, DCs exist in one of three states: immature, semi-mature and mature. The initial maturation state of a DC is immature for sensing and processing three categories of input signals (see Table 1) and in response produces three output signals. The three input signals can influence the behavior of DCs' sensitivity.

Signal Name	Symbol	Définitions
Pathogen Associated Molecular Patterns (PAMP)	*S1=PAMP*	A signature of abnormal behavior. An increase in this signal is associated with a high confidence of abnormality.
Danger Signal	*S2=DS*	A measure of an attribute which increases in value to indicate an abnormality. Low values of this signal may not be anomalous, giving a high value a moderate confidence of indicating abnormality.
Safe Signal	*S3=SS*	A measure which increases value in conjunction with observed normal behavior. This is a confident indicator of normal, predictable or steady-state system behavior. This signal is used to counteract the effects of PAMPs and danger signals.

Table 1. Signals Definitions

The first two input signals are S1 and S2. S1 signal is derived from the detection of pathogens while S2 signal is generated from the unexpected cell death due to damage of the tissue cells. The third input signal is S3 which is molecules released as a result of normal cell death. During the immature lifespan stage, if the DC has collected majority of S3 signals, it will change state to a semi-mature state and suppress the activation of the immune system. Conversely, cells exposed to S1 and S2 signals transform into a mature state and can instruct the immune system to activate.

While in immature state, DCs capture the suspect entities (termed "antigen") and combine them with evidence of damage in the form of signals to provide information about how "dangerous" a particular protein is to the host body. Antigens collected by the semi-mature DCs are presented in a "safe" context while antigens presented by mature DCs are presented in a "dangerous" context. In terms of the algorithm, the DCA is a population based algorithm which performs anomaly detection based on the indication of abnormality of the system by aggregating and performing asynchronous correlation of signals with the suspect's antigen. Signal processing occurs within DCs of the immature state. Each DC in the immature state performs three functions as follows:

-To sample antigen by collecting antigen from an external source and transfers the antigens to its own antigen storage facility.

-To update input signals in which the DC collects values of all input signals present in the signal storage area.

-To calculate temporary output signal values from the received input signals, with the output values then added to form the cell's cumulative output signals.

The transformation from input to output signal per cell is performed using a simple weighted sum (Equation 1) described in detail in [6][5]. These weights determine the value of the output and derived from preliminary observation that defines the danger level of the input signals.

$$\mathrm{O}_{\mathrm{j}} = \sum (W_{ij} * S_i) .... \forall j \qquad (1)$$

Where:

- *Wij* is the signal weight
- *Si* is the input signal category (*S1=PS, S2=DS* and *S3=SS*)
- Oj is the output concentrations of one of the following signal:

□ *j*=1 costimulatory signal (csm)

□ *j* =2 a semi-mature DC output signal (semi)

□ *j* =3 mature DC output signal (mat)

In the algorithm, the signal values are assigned real valued numbers and the antigen are assigned as categorical values of the object to be classified. The algorithm has three different stages, the initialization stage, the data processing and the analysis stage.

In the initialization stage, the algorithm generates DCs population where each cell is assigned a random “migration” threshold. The input data forms the sorted antigen and signals (*S1, S2* and *S3*) with respect to the time and passed to the processing stage. Each DC performs an internal correlation between signals and antigen with respect to a specified time window determined by the migration threshold, signals and antigen. To cease data collection, a DC must have experienced signals, and in response to this express output signals. As the level of input signal experienced increases, the probability of the DC exceeding its lifespan also increases. The level of signal input is mapped as a cumulative O_1 value. Once O_1 exceeds a migration threshold value, the cell ceases signal and antigen collection and is removed from the population and enters the maturation stage. Upon removal from the population the cell is replaced by a new cell, to keep the population level static.

A high concentration of S_1 and S_2 increases the probability of immature cells to become mature cells while a high concentration of S3 imposes the immature cells to become semi-mature cells. Therefore, if $O_2 > O_3$, the DC is termed “semi-mature” cell. Antigen presented by a semi-mature cell is assigned a context value of zero. In contrast, $O_2 < O_3$ leads to a “mature” cell and antigen presented by a mature cell is assigned a context value of one. The detection of anomaly is based on having more mature cells than semi-mature cells in which the antigen in a mature context is detected. The pseudo code for the functioning of a single cell is presented in Algorithm 1.

```
input: Sorted antigen and signals (S1=PS,S2=DS,S3=SS)
output: Antigen and their context (0/1)
Initialize DC;
For each cell in DC population
{
    while CSM output signal (O1) < migration threshold
    {
        get antigen;
        store antigen;
        get signals;
        calculate interim output signals;
        update cumulative output signals;
    }
    if semi-mature output (O2) > mature output(O3)
        cell context is assigned as 0 ;
    else
        cell context is assigned as 1 ;
    kill cell;
    replace cell in population;
```

Algorithm 1. dendritic cell algorithm

The final stage involves calculating an anomaly coefficient per antigen type - termed the mature context antigen value (MCAV), once all antigens and signals are processed by the cell population, an analysis stage is performed. The derivation of the MCAV per antigen type in the range of zero to one is shown in Equation 2.

The closer this value is to *one*, the more likely the antigen type is to be anomalous. A threshold is applied to distinguish between anomalous and normal type of antigen.

$$\mathrm{MCAV}_x = \frac{Z_x}{Y_x} \tag{2}$$

Where:

MCAVx: is the MCAV coefficient for antigen type *x*.

Zx: is the number of mature context antigen presentations for antigen type *x* and *Yx* is the total number of antigen presented for antigen type *x*.

4. KDD dataset

KDD-99 Dataset The KDD-99 dataset is based on the 1998 DARPA initiative to provide designers of intrusion detection systems (IDS) with a benchmark on which to evaluate different methodologies [8]. To do so, a simulation is made of a factitious military network consisting of three target machines.

Additionally, there are three machines to spoof different IP addresses to generate traffic between different hosts. Finally, a sniffer is used to record all network traffic using the tcpdump format. Normal connections are designed to reflect traffic seen on military bases and attacks fall into one of four categories: Denial of Service (dos), User to Root (u2r); Remote to Local (r2l) and Probe(probe).

The KDD-99 data is composed of several components as seen in Table 1. Only the 10% KDD data is used for the evaluation of the intrusion detection system.

Dataset label	dos	probe	u2r	r2l	Total Attack	Total Normal
10% KDD	391458	4107	52	1126	396,744	97,277
Corrected (Test)	229853	4166	70	16347	250,436	60,593
Whole KDD	3883370	41102	52	1126	3,925,651	972,78

Table 2. Components of Kdd99 data

5. Preprocessing

In the experiments , we have utilized 10% of the complete whole of data KDD' 99, which corresponds to 494019 connections of Training data and 311029 connections of testing data, each connection is represented by 41 attributes Figure.1

```
5,tcp, smtp, SF, 959,337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,144,19
2,0.70,0.02,0.01,0.01,0.00,0.00,0.00,0.00,0.00,normal.

0,tcp,http,SF,54540,8314,0,0,0,2,0,1,1,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,118,
118,1.00,0.00,0.01,0.00,0.00,0.00,0.02,0.02,back.

0.tcp,http_443,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,114,2,1.00,1.00,0.00,0.00,0.02,0.06,0.00,255,2,
0.01,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune.
```

Figure 1. Example of kdd connections

In our experiments we will be interested only by knowing if a given connection is normal (don't present any attack) or not (two classes problem). For this case, we modified the data by grouping all the attacks to make only one class which we called "Abnormal".

The dendritic cell algorithm requires that its inputs are categorized into three types or group of signals: PAMP, Danger and safe (Table 3).

	Csm	Semi Mature	Mature
PAMP	4	0	4
DANGER	2	0	2
SAFE	6	1	-3

Table 3. DCA Weights used in our experiments

To adapt these connections to the inputs of this algorithm, we categorizes the attributes of each connection (41 attributes) in three type or three groups of signals, signals PAMP having large an effect. For that, we based on the distribution of values of each attribute in the two classes normal and abnormal in training data.

The figure 2 displays for each attribute the percentage of its values in the two classes (normal and abnormal)The attributes having a large distribution in the abnormal class, that is having a great effect so that connection assigned to the abnormal class, are considered as PAMP signals. And the attributes having a large distribution in the normal class are considered as Safe signals, and the attributes having a distribution in the class abnormal relatively higher than the distribution in the class normal are considered as danger signals, and the attributes having an equivalent distribution in the two classes, are considered as Danger and Safe at the same time.

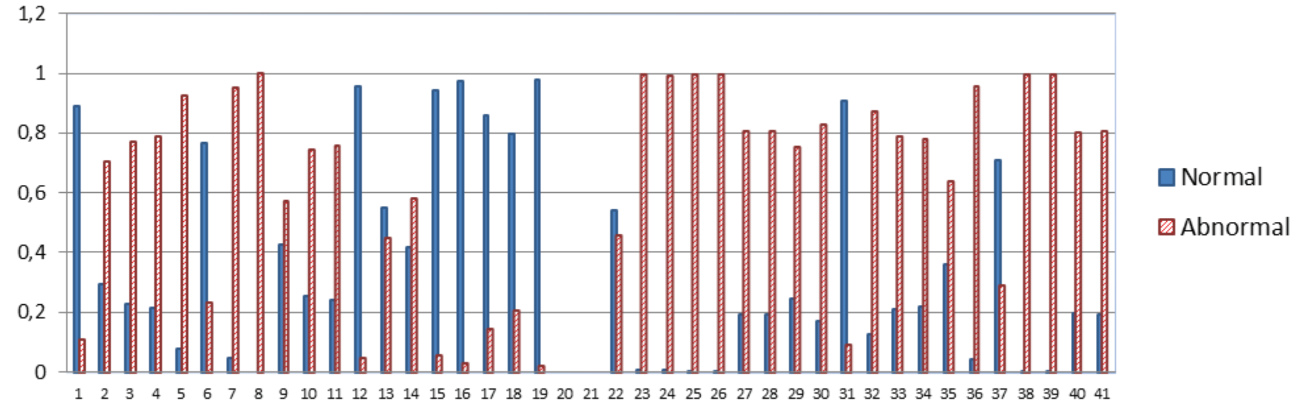

Figure 2. the values distribution of each attribute in the two classes (normal, abnormal)

According to the distribution we grouped the attributes as following:

PAMP = {7, 8, 23, 24, 25, 26, 36, 38, 39}

Danger ={ 2,3,4,9,10,11,13,14,22,27,28,29,30,32, 33,34, 35, 36, 37, 40, 41}

Safe = {1,2,3,4,5,6,9,10,11,12,13,14,15,16,17,18,19,22,31, 33, 34, 35, 37}

6. Results

To evaluate the performance of the DCA algorithm, two indices are used: Detection Rate (DR) and False Alarm Rate (FA), computed as follow:

$$DR = \frac{TP}{TP + FN} \tag{3}$$

$$FA = \frac{FP}{FP + TN} \tag{4}$$

Where

TP: (true positive) is the number of anomalous elements identified as anomalous

FN :(false negative) is the number of anomalous elements identified as normal

FP: (false positive) is the number of normal elements identified as anomalous

TN: (true negative) is the number of normal elements identified as normal.

Also we used the percent of correct classification (PCC) to evaluate the DCA for each attack type (DoS, R2L, U2R and Probe).PCC was calculated for each class using the following formula:

$$\text{PCC} = \frac{\text{the numbre of element correctly classified}}{\text{the total numbre of element}} \qquad (5)$$

In our experiment we used the following parameters

- Number of cell in the populations: 09
- Decision threshold for MACV: 0.7
- Number of presentation of antigen to the denteritic cell: 16(each time we select 3 random attributes from the 3 type of signals: PAMP, Safe and Danger)
- The weights used in the algorithm are:

Test data (10% Kdd)	
Detection Rate	False alarm Rate
90.39%	**04 .26%**

Table 4. Detection rate and false alarm rate for 10% of Kdd data
(two classes: normal and abnormal)

In table 4, we visualize that the DCA algorithm yields the highest detection rate and the lowest false alarm rate and more detail can be show in Table 5. DCA has a the better classification of the normal connection (100%) and the abnormal connections (90.38%)

The table 5 shows that connections Normal, Dos and Probing are well classified (100%, 96.46%, and 84.10%). What is not the case of connections R2L and U 2R (6.72%, 17.14%).

		Normal	**Abnormal**
Normal class (PCC=100%)	Normal	100% (60593)	0% (0)
	DoS (229855)	3.53% (8123)	96.46% (221732)
Abnormal class (PCC=90.38%)	R2L (16345)	93.20% (15235)	6.79% (1110)
	U2R (70)	82.85 % (58)	17.14 % (12)
	Probe (284)	15.89 % (662)	84.10% (3504)
PCC total	**92.26 % (286951)**		

Table 5. Confusions matrix relative to two classes of connections (normal, abnormal)

That is due to the fact that the proportions in the training data of the attacks U2R and R2L are very few (0.22% for U2R and 0.23% for R2L ,Therefore the study which will be carried concerning this class will be weak and by consequences of false classifications of connections.

The table 6 shows that the increase in cell's number in the population and antigen presentation has an effect proportional with the system performance. High number of cells in the population can reduces the incorrect classification rate (false alarm rate) Figure.3, and the augmentation in number of antigen presentation can reduces the incorrect classification rate (false alarm rate) and increases the correct classification (detection rate) Figure .4.

			Number of cells in population					
			1	4	7	10	13	16
Number of antigen presentation	AP = 1	DR	0,5463	0,4696	0,4204	0,3454	0,1209	0,1187
		FA	0,425	0,3216	0,2589	0,2002	0,019	0,0169
	AP = 4	DR	0,8019	0,8064	0,8147	0,7884	0,7822	0,7913
		FA	0,4314	0,332	0,2712	0,2134	0,1777	0,0333
	AP = 8	DR	0,8699	0,8772	0,8261	0,8517	0,8634	0,8711
		FA	0,3709	0,2668	0,1035	0,0331	0,0333	0,0327
	AP = 16	DR	0,9122	0,9117	0,9117	**0,9005**	**0,9029**	**0,9034**
		FA	0,3077	0,1158	0,0947	**0,0359**	**0,0362**	**0,0429**

Table 6. Number of cells and number of antigen presentation effect on the DCA performance

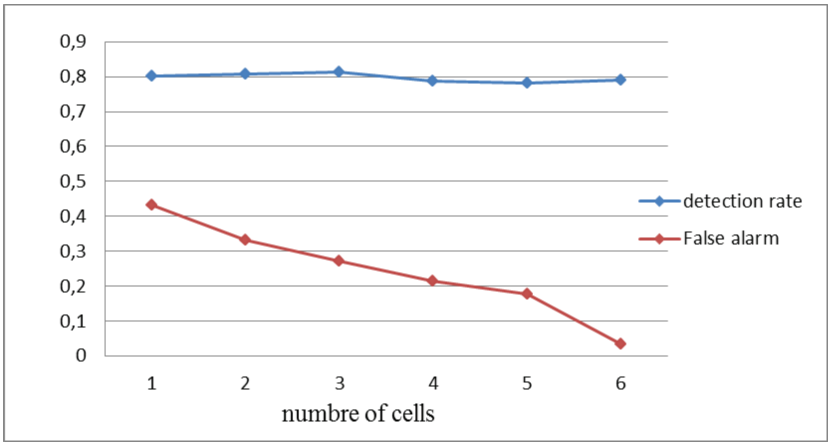

Figure 3. Effect of cells number (with antigen presentation = 4)

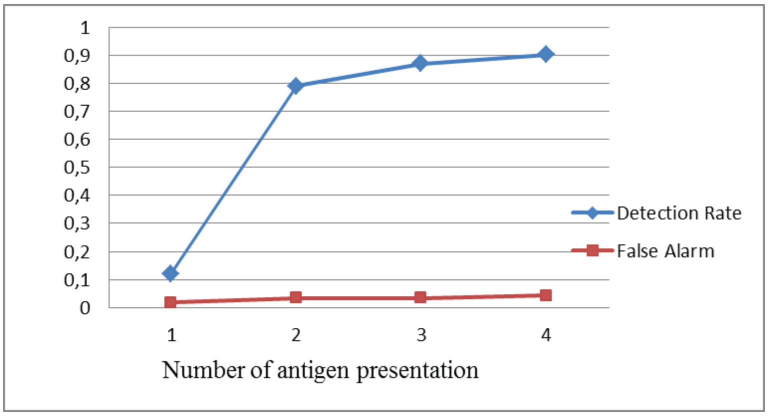

Figure 4. Effect of antigen presentation number (with number of cells = 16)

Finally we compared the performance of DCA (AP=16,cell number=13) with the studies of *Eskin* and *Günes* [7] [4],who used same the experimental data as those used in our study ,with the following methods :KNN (K nearest neighbor),SOM(self-organize map) and SVM(support vector machine)

The DCA(Table 7) algorithm present low false alarm rate compared to these methods, and present a detection rate similar to SOM and KNN therefore the DCA has a better performance compared to SOM and KNN and has a similar performance compared to SVM (low false alarm)

Method	Detection rate	False alarm rate
SVM	98%	10%
KNN	91%	8%
SOM	89%	4.6 %
DCA	90.29%	03 .62%

Table 7. DCA compared with others methods

7. Conclusions

In this paper, we tested a recent approach inspired from the workings of dendritic cells in the immune system and based on the correlation between environmental signals and the antigen presented to dendritic cells. The results obtained from the evaluation data Kdd'99 are acceptable: a detection rate of 90.26% and a false alerts rate 3.62%. These results are comparable with the most used classification methods (SVM, SOM and KNN). In future we plan not to just expand our study but like to explore the find the scalability of application of artificial immune system by introducing elegant algorithms.

References

[1] Aickelin, U., Bentley, P Cayzer, S.,.Kim, J., McLeod, J. (2003). Danger theory: The link between AIS and IDS. *In:* Proceedings of the 2nd International Conference on Artificial Immune Systems (ICARIS'03). Berlin, Heidelberg.

[2] Al-Hammadi, Yousof., Aickelin, Uwe., Greensmith, Julie (2010). .Performance Evaluation of DCA and SRC on a Single Bot Detection, *Journal of Information Assurance and Security* 5.

[3] Castro, L.N De., Timmis, J. I (2003). Artificial immune system as a Novel Soft Computing paradigm, Computing laboratory. University of Kent at Canterbury, *Soft Computing Journal*.

[4] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S. (2002).A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *In:* Applications of Data Mining in Computer Security. Kluwer.

[5] Greensmith, J. Aickelin, U. Cayzer, S. (2005). Introducing dendritic cells as a novel immune inspired algorithm for anomaly detection, *In*: Proceedings of the International Conference on Artificial Immune Systems (ICARIS05).

[6] Greensmith, J., Aickelin, U (2007). Dendritic cells for SYN scan detection, *In:* Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007) p.49-56.

[7] Günes Kayacık, H. Nur Zincir-heywood, A., Heywood, Malcolm I (2003). On the Capability of an SOM based Intrusion Detection system, *In*: IEEE-INNS International Joint Conference on Neural Networks.

[8] Knowledge discovery in databases DARPA archive. Task Description http://www.kdd.ics.uci.edu/databases/kddcup99/task.htm

[9] Matzinger, P. (2002). The danger model: A renewed sense of self. *Science* 296, 301-5.

[10] Greensmith, J. (2007). The dendritic cell algorithm, Ph.D thesis, The University of Nottingham, Nottingham, UK.

[11] Greensmith, J., Aickelin, U (2007). Dendritic cells for SYN scan detection, *In*: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007), p.49-56.

[12] Greensmith, J. Twycross, J., Aickelin, U. Dendritic cells for anomaly detection. *In:* IEEE Congress on Evolutionary Computation(CEC 2006), p. 664–671, 2006.

[13] Greensmith, J., Aickelin, U., Cayzer, S. (2005). Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection, *In*: ICARIS-05, LNCS 3627, p.153–167.

[14] McEwan, Chris., Hart, Emma (2009). Topological Constraints in the Evolution of Idiotypic Networks, Artificial Immune Systems, Lecture Notes in Computer Science, Springer. p. 252-263.

[15] Manzoor, Salman., Shafiq, Zubair S., Tabish, Momina S., Farooq, Muddassar (2009). A sense of 'Danger' for windows processes, Artificial Immune Systems, Lecture Notes in Computer Science, Springer. p. 220-233.

[16] Stibor, Thomas., Oates, Robert., Kendall, Graham., Garibaldi, Jonathan M. (2009). Geometrical insights into the dendritic cell algorithm, *In*: GECCO '09 Proceedings of the 11th Annual conference on Genetic and evolutionary computation. ACM New York, NY, USA

[17] Anthony Kulis, Shahram Rahimi (2010). Finding Danger Signals using Fuzzy Dendritic Cells, http://www.cs.siu.edu/~akulis/papers/ais.pdf (accessed on 11 March 2010)