ANTIDS: Self Organized Ant-Based Clustering Model for Intrusion Detection System

Vitorino Ramos and Ajith Abraham*

CVRM-IST, Technical University of Lisbon (IST), Lisbon, Portugal
vitorino.ramos@alfa.ist.utl.pt
*School of Computer Science and Engineering, Chung-Ang University, S. Korea
ajith.abraham@ieee.org

Abstract: Security of computers and the networks that connect them is increasingly becoming of great significance. Computer security is defined as the protection of computing systems against threats to confidentiality, integrity, and availability. Due to the fact that it is almost difficult for a system administrator to recognize and manually intervene to stop an attack, there is an increasing recognition that Intrusion Detection Systems (IDS) should have a lot to earn on following its basic principles on the behavior of complex natural systems, namely in what refers to self-organization, allowing for a real distributed and collective perception of this phenomena. Having that aim in mind, the present work presents a self-organized ANT colony based Intrusion Detection System (ANTIDS) to detect intrusions in a network infrastructure. The performance is compared among conventional soft computing paradigms like Decision Trees (DT), Support Vector Machines (SVM) and Linear Genetic Programming (LGP) to model fast, online and efficient intrusion detection systems.

1. Introduction

The process of monitoring the events occurring in a computer system or network and analyzing them for sign of intrusions is known as Intrusion detection. It is impossible in practice, and even if possible, extremely difficult and expensive, to write a completely secure system. Transition to such a system for use in the entire world would be an equally difficult task. An Intrusion Detection System (IDS) is a program that analyzes what happens or has happened during an execution and tries to find indications that the computer has been misused. An Intrusion detection system does not eliminate the use of preventive mechanism but it works as the last defensive mechanism in securing the system. Data mining approaches for intrusion detection were first implemented in mining audit data for automated models for intrusion detection [2]. On the other hand, and due to the fact that it is more and more improper for a system administrator to recognize and manually intervene to stop an attack (an option only possible in small scale networks) without harming too much the integrity of the overall system, there is an increasing recognition that ID systems should have a lot to earn on following its basic principles on the behavior of complex natural systems, namely in what refers to self-organization. Due to their nature, self-organizing complex adaptive systems typically are comprised of a large number of frequently similar components (e.g. agents) or events. Through their process, a pattern at the global-level of a system emerges solely from numerous interactions among the lower-level components [18][22]. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern, which, as in many real world problems is not easily accessible or possible to be found. Stigmergy [21][20], a kind of indirect communication and learning by the environment found in social insects is a well know example of self-organization, providing not only vital clues in order to understand how the components can interact to produce a complex pattern, as can pinpoint simple biological non-linear rules and methods to achieve improved artificial intelligent adaptive categorization systems, critical for collective perception and recognition. In fact, their distributed bottom-up emergent nature, along with their massively implicit parallel properties and the fact that there is no need of a global top-down hierarchical supervisor, makes them ideal candidates for IDS and to be embedded on-line in complex large scale computer network infrastructures, where traditional security mechanisms demonstrate severe weaknesses. Some works have already been presented along these lines. Recently, *Foukia* et al [12] designed an IR (Intrusion Response) system cooperating with an IDS using mobile agents distributed throughout the network, based on stigmergic properties. In other works [7][10], detection was based on artificial immune systems [6] where ID agents map the functionalities of the natural immune system to distinguish between normal and abnormal events (respectively "self" and "non self" in the immune system). This paper introduces a self-organized ANT colony based Intrusion Detection System (ANTIDS) to detect intrusions and compares its performance with Linear Genetic Programming (LGP), Support Vector Machines (SVM) and Decision Trees (DT). The rest of the paper is organized as follows. The technical details of the ANT colony algorithm are presented in Section 2 followed by the importance of attribute or feature reduction and experiment results in Section 3. Some conclusions are also provided towards the end.

2. Bio-Inspired Self-Organized Ant-Based Clustering

Data Mining is precisely one of those problems in which real ants can suggest very interesting heuristics for computer scientists [18], namely for clustering purposes. One of the first studies using the metaphor of ant colonies related to the above clustering domain is due to *Deneubourg* [5], where a population of ant-like agents randomly moving onto a 2D grid are allowed to move basic objects so as to cluster them. This method was then further generalized by *Lumer* and *Faieta* [16] (here after *LF* algorithm), applying it to exploratory data analysis, for the first time. However, the last work entitled "Exploratory Database Analysis via Self-Organization", according to [3], was never published due to commercial applications. More recently, *Ramos* et al. [20][21][22] presented a novel strategy (*ACLUSTER*) to tackle unsupervised clustering as well as data retrieval problems, avoiding not only short-term memory based strategies, as well as the use of several artificial ant types (using different speeds), present in those approaches proposed initially by *Lumer* [16]. Other works in this area include those by *Monmarché* et al. [17], *Ramos*, *Merelo* et al. [18][20][21][22], *Handl* and *Dorigo* [13], *Ramos* and *Abraham* [19].

2.1 Distributed, Collaborative and Stigmergic Clustering

The swarm intelligence algorithm fully uses agents that stochastically move around the classification "habitat" following pheromone concentrations. That is, instead of trying to solve some disparities in the basic *LF* algorithm by adding different ant casts, short-term memories and behavioral switches, which are computationally intensive, representing simultaneously a potential and difficult complex parameter tuning, it was our intention to follow real ant-like behaviors as possible (some other features will be incorporated, as the use of different response thresholds to task-associated stimulus intensities, discussed later). In that sense, bio-inspired spatial transition probabilities are incorporated into the system, avoiding randomly moving agents, which tend the distributed algorithm to explore regions manifestly without interest (e.g., regions without any type of object clusters), being generally, this type of exploration, counterproductive and time consuming. Since this type of transition probabilities depend on the spatial distribution of pheromone across the environment, the behavior reproduced is also a stigmergic one [5][21]. Moreover, the strategy not only allows to guide ants to find clusters of objects in an adaptive way (if, by any reason, one cluster disappears, pheromone tends to evaporate on that location), as the use of embodied short-term memories is avoided (since this transition probabilities tends also to increase pheromone in specific locations, where more objects are present). As we shall see, the distribution of the pheromone represents the memory of the recent history of the swarm, and in a sense it contains information which the individual ants are unable to hold or transmit. There is no direct communication between the organisms but a type of indirect communication through the pheromonal field. In fact, ants are not allowed to have any memory and the individual's spatial knowledge is restricted to local information about the whole colony pheromone density. In order to design this behavior, one simple model was adopted [4], and extended (as in [20][22]) due to specific constraints of the present proposal. As described in [7], the state of an individual ant can be expressed by its position *r*, and orientation θ. It is then sufficient to specify a transition probability from one place and orientation (r, θ) to the next (r_*, θ_*) an instant later. The response function can effectively be translated into a two-parameter transition rule between the cells by use of a pheromone weighting function (Eq. 1).

$$W(\sigma) = \left(1 + \frac{\sigma}{1 + \delta\sigma}\right)^{\beta} \qquad (1) \qquad\qquad P_{ik} = \frac{W(\sigma_i)w(\Delta_i)}{\sum_{j/k} W(\sigma_j)w(\Delta_j)} \qquad (2)$$

This equation measures the relative probabilities of moving to a cite *r* (in our context, to a grid location) with pheromone density $\sigma(r)$. The parameter β is associated with the osmotropotaxic sensitivity (a kind of instantaneous pheromonal gradient following), and on the other hand, $1/\delta$ is the sensory capacity, which describes the fact that each ant's ability to sense pheromone decreases somewhat at high concentrations. In addition to the former equation, there is a weighting factor $w(\Delta\theta)$, where $\Delta\theta$ is the change in direction at each time step, i.e. measures the magnitude of the difference in orientation. As an additional condition, each individual leaves a constant amount η of pheromone at the cell in which it is located at every time step t. This pheromone decays at each time step at a rate k. Then, the normalised transition probabilities on the lattice to go from cell k to cell i are given by P_{ik} [4] (Eq. 2), where the notation j/k indicates the sum over all the pixels j which are in the local neighborhood of k. Finally, Δ_i measures the magnitude of the difference in orientation for the previous direction at time t-1.

2.2 Picking and Dropping Data Objects

In order to model the behavior of ants associated to different tasks, as dropping and picking up objects, other works [20] suggest the use of combinations of different response thresholds. As we have seen before, there are two major factors that should influence any local action taken by the ant-like agent: the number of objects in his neighborhood, and their similarity (including the hypothetical object carried by one ant). *Lumer* and *Faieta* [16], use an average similarity, mixing distances between objects with their number, incorporating it simultaneously into a response threshold function. Instead, we recommend the use of combinations of two independent response threshold functions, each associated with a different environmental factor (or, stimuli intensity), that is, the number of objects in the area, and their similarity. Moreover, the computation of average similarities are avoided in the present algorithm, since this strategy can be somehow blind to the number of objects present in one specific neighborhood. In fact, in *Lumer* and *Faieta*'s work [16], there is an hypothetical chance of having the same average similarity value, respectively having one or, more objects present in that region. But, experimental evidences and observation in some types of ant colonies can provide us with a different answer. After *Wilson* (*The Insect Societies*, Cambridge Press, 1971), it is known that minors and majors in the polymorphic species of ants *Genus Pheidole*, have different response thresholds to task-associated stimulus intensities (i.e., division of labor). Recently, and inspired by this experimental evidence, *Bonabeau* et al. [3], proposed a family of response threshold functions in order to model this behavior. According to it, every individual has a response threshold θ for every task. Individuals engage in task performance when the level of the task-associated stimuli s, exceeds their thresholds. Author's defined s as the intensity of a stimulus associated with a particular task, i.e. s can be a number of encounters, a chemical concentration, or any quantitative cue sensed by individuals. One family of response functions $T_\theta(s)$ (the probability of performing the task as a function of stimulus intensity s), that satisfy this requirement is given by (Eq. 3) [3], where n>1 determines the steepness of the threshold (normally n=2, but similar results can be obtained with other values of n>1). Now, at $s = \theta$, this probability is exactly ½. Therefore, individuals with a lower value of θ are likely to respond to a lower level of stimulus. In order to take account on the number of objects present in one neighborhood, Eq. 13, was used (where, n now stands for the number of objects present in one neighborhood, and $\theta = 5$), defining χ (Eq. 4) as the response threshold associated to the number of items present in a 3 x 3 region around r (one specific grid location).

$$T_\theta(s) = \frac{s^n}{s^n + \theta^n} \quad (3) \qquad \chi = \frac{n^2}{n^2 + 5^2} \quad (4)$$

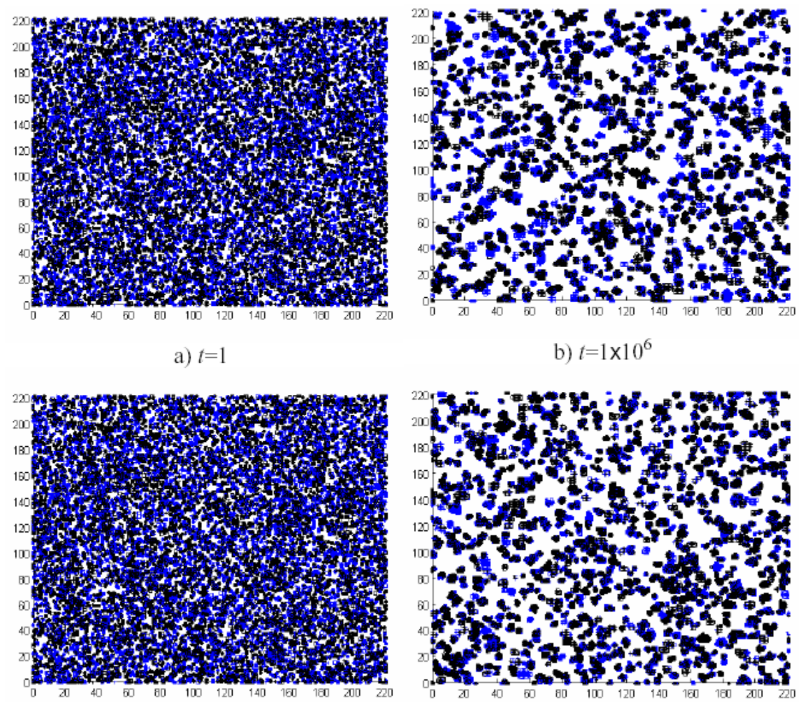
$$\delta = \left(\frac{k_1}{k_1 + d}\right)^2 \quad (5) \qquad \varepsilon = \left(\frac{d}{k_2 + d}\right)^2 \quad (6)$$

$$P_p = (1-\chi).\varepsilon \quad (7) \qquad P_d = \chi.\delta \quad (8)$$

Now, in order to take account on the hypothetical similarity between objects, and in each ant action due to this factor, a *Euclidean* normalized distance d is computed within all the pairs of objects present in that 3 x 3 region around r. Being a and b, a pair of objects, and $f_a(i)$, $f_b(i)$ their respective feature vectors (being each object defined by F features), then $d = (1/d_{max}).[(1/F).\Sigma_{i=1,F}(f_a(i)-f_b(i))^2]^{½}$. Clearly, this distance d reaches its maximum (=1, since d is normalized by d_{max}) when two objects are maximally different, and d=0 when they are equally defined by the same F features. Moreover, δ and ε (Eqs. 5,6), are respectively defined as the response threshold functions associated to the similarity of objects, in case of dropping an object (Eq. 5), and picking it up (Eq. 6), at site r. Finally, in every action taken by an agent, and in order to deal, and represent different stimulus intensities (number of items and their similarity), present at each site in the environment visited by one ant, the strategy uses a composition of the above defined response threshold functions (Eqs. 4,5 and 6). Several composed probabilities were analyzed [20] and used as test functions in one preliminary test. The best results were achieved with the test function #1 (Eqs. 7,8), achieving a high classification rate (out of 4 different functions were used, as well the *LF* algorithm [16]; for comparison reasons – see [20][21]). Alternatively, the system can also be robust feeding the data continuously as proved in past works [19].

3. Attribute Deduction, Experiment Setup and Results

Complex relationships exist between features, which are difficult for humans to discover. IDS must therefore reduce the amount of data to be processed. This is very important if real-time detection is desired [14]. In this research, feature selection is done based on the contribution the input variables made to the construction of the decision tree. Feature importance is determined by the role of each input variable either as a main splitter or as a surrogate. Surrogate splitters are defined as back-up rules that closely mimic the action of primary splitting rules. Attack types fall into four main categories: DoS: Denial of Service , R2L: Unauthorized Access from a Remote Machine; U2R: Unauthorized Access to Local Super User (root) and Probing. Our experiments had three conventional phases namely input feature reduction, training phase and testing phase for DT, SVM and LGP. The 41 features are labeled in order as *A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, AA, AB, AC, AD, AF, AG, AH, AI,* *AJ, AK, AL, AM, AN, AO* and the class label is named as *AP*. This data set has five different classes namely *Normal, DoS, R2L, U2R* and *Probes.* The training and test comprises of 5092 and 6890 records respectively [15]. All the training data were scaled to (0-1). The decision tree approach helped to reduce the 12 variable data set with *C, E, F, L, W, X, Y, AB, AE, AF, AG* and *AI* as variables [14]. Using the original and reduced data sets, we performed a 5-class classification. The normal data belongs to class 1, probe belongs to class 2, denial of service belongs to class 3, user to super user belongs to class 4, remote to local belongs to class 5. The ANTIDS experimental setup took however a different path, since the algorithm's underneath principle is based on an unsupervised process.



a) t=1 b) t=1x10^6

c) t=1 d) t=1x10^6

Figure 1 (a - b). Ant-like clustering using the full data set (with 41 features) and (c- d) the reduced data set (12 features), at t=1and t=1x106. For the purpose of reader transparency, in all diagrams class 1 to 5 was respectively represented by ■, □, ○, ●, and finally by + (training samples in blue; testing samples in black).

In our case, each object (each ID sample) manipulated by the artificial ant colony is represent by a feature vector composed of 41 elements or 12 elements, depending if we instead use the reduced data set . Based on self-organizing these objects in a non-parametrical toroidal 2-D space, the unsupervised clustering proceeds for t = 1 x 10^6 time steps. After the unsupervised and self-organized clustering process is finished (Fig. 1) - with 11982 samples x 41 (12) features each –, the first 5092 samples are used as markers or reference points (blue markers in Fig. 1), and via k-NNR nearest neighbor rule classification [11,13] the remaining 6890 samples (black markers in Fig. 1) are classified (we used k = 3 neighbors; k must be always an odd number). In order to do so, for each sample i = 6890, …,11982, we computed their first k = 3 marker neighbors on this non-parametric toroidal 2-D space. An algorithm to find the first k neighbors in a toroidal space can however be largely tricky. Our idea was to use 8 virtual spaces (windows) around the one we see in Fig. 1, copying to each one of them, all the respective reference markers to be used in the normal k-NNR and finally computing the geographical vicinity in this large virtual space for each one of the testing samples only present in the central window. The majority of those marker label values, considered for each still unclassified sample, give them the respective final classification result. Two experiment setups were then tested and the results are depicted in Tables 1 and 2.

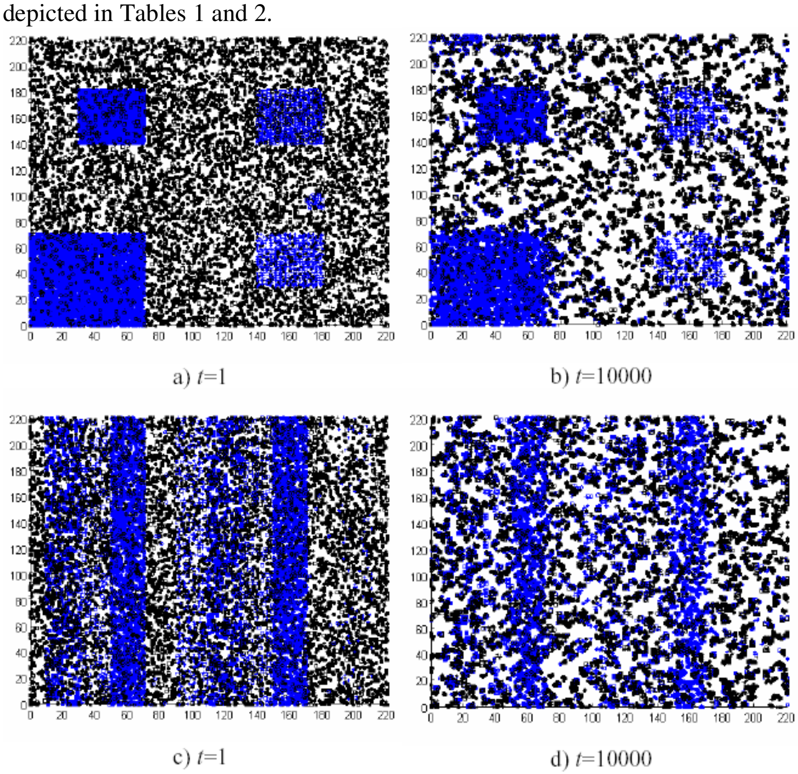


Figure 2. Random allocation of marker items into specified zones.

The first one (ANTIDS-a) used self-organization in order to cluster all the 11982 samples at once (5092 training + 6890 testing samples). In ANTIDS-b, however, we process and treat parts and streams of data independently, set after set. In this framework, we use always the full markers set (5092 training samples) plus only a part of the testing set (1000 samples each time). That is, we had to make six runs with 1000 testing samples and the 7th final one with 890 testing samples. For any of these cases, the final self- organized stigmergic map [21] achieved is highly robust. If for example, we use k = 1 in the final k-NNR classification, which normally is not prudent, we also arrive at similar recognition rates.

Attack type	Classification accuracy on test data set (%)				
	ANTIDS-*a*	ANTIDS-*b*	DT	SVM	LGP
Normal	70.52	99.64	99.64	99.64	**99.73**
Probe	71.73	98.29	99.86	98.57	**99.89**
DOS	83.39	**99.97**	96.83	99.92	99.95
U2R	0.00	64.00	**68.00**	40.00	64.00
R2L	10.47	**99.47**	84.19	33.92	**99.47**

Table 1. Performance comparison using full data set

Attack type	Classification accuracy on test data set (%)				
	ANTIDS-*a*	ANTIDS-*b*	DT	SVM	LGP
Normal	69.40	99.73	**100.00**	99.75	99.97
Probe	60.07	99.86	97.71	98.20	**99.93**
DOS	84.31	**99.97**	85.34	98.89	99.96
U2R	47.62	68.00	64.00	59.00	**68.26**
R2L	87.63	99.47	95.56	56.00	**99.98**

Table 2. Performance comparison using reduced data set

Other experiments included the initial random allocation of marker items into specified zones (Figure 2). In a) marker samples (training data set for the subsequent k-NNR classification; in blue) are randomly allocated in 5 box zones corresponding to samples from class 1 to 5 (class 1 in the bottom-left corner; other classes distributed clockwise), while testing samples are allocated everywhere in this toroidal classification space. These marker samples, however, can now be translated to new places if any ant-like agent wishes to do so, in order to proceed the unsupervised clustering. In c) marker samples are randomly allocated into 10 vertical stripes (2 for each class). In b-d) corresponding results at t=10000. Both strategies however, leaded to meager recognition rates in the interval [40-72%], for each class, after finally using k-NNR (k=1,3).

4. Conclusions

As depicted in Tables 1 and 2, ANTIDS approach has several limitations in what refers to the final recognition rate, obtaining optimal results only for some cases. This is in part due to several reasons. First, the large number of samples used in the present study forces *ACLUSTER* algorithm to be run on a large toroidal space.

Empirical studies [21][22] show that the optimal classification "habitat" area should be in the order of 4 times the number of objects, while the number of ants should be in the order of 1/10 of the number of objects. Rather, it's by large preferably to process and treat parts and streams of data independently, set after set (check ANTIDS-b results). Second, and still in the present case, the data is poorly uniformly distributed between all the five classes. In fact, while some classes like class 3 (DOS) are represented by a sum of 3000 training and 4200 testing samples, a total of 7200 items (around 60% of our entire data set), other classes like class 4 (U2R) are merely represented by a sum of 27 training and 25 testing samples, a total of 52 items (0.4% of our entire data set). In fact, the probability of one ant to encounter a class 3 sample in the toroidal classification space, process and treat it, is 120 times bigger than to find one from class 4. This fact *per se*, can bias a lot our final results, since any self-organizing mechanism depends a lot on a self-sufficient critical mass. However, the self-organizing ID system has 4 major advantages in what refers to a comparison to their counterpart paradigms (DT, SVM, LGP), namely: (1) Classification can be processed online and in real time due to their distributed nature, as proved before with swarms nourished with continuous streams of data [19]. (2) ANTIDS can deal with new classes whenever it's needed without the need of retraining. In fact, stigmergy is often associated with flexibility: when the environment changes because of an external perturbation, the insects respond appropriately to that perturbation, as if it were a modification of the environment caused by the colony's activities. In other words, the colony can collectively respond to the perturbation with individuals exhibiting the same behavior. When it comes to artificial agents, this type of flexibility is priceless: it means that the agents can respond to a perturbation without being reprogrammed to deal with that particular instability. (3) ANTIDS algorithm can work either in unsupervised or supervised mode (adding the final *k*-NNR classification and using training samples as markers), and finally (4) The self-organizing nature of ANTIDS makes them an ideal candidate for distributed IDS.

Acknowledgements

Authors would like to thank *Joaquim Castelo Ramos* for his time on running part of the several experiments.

References

[1] Abraham A., "Evolutionary Computation in Intelligent Web Management", Evolutionary Computing in Data Mining, Ghosh A. et al. (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag Germany, pp. 189-210, 2004.

[2] Barbara D., Couto J., Jajodia S. and Wu N., ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection, *SIGMOD Record*, 30(4), pp. 15-24, 2001.

[3] Bonabeau, E., Dorigo, M. and Théraulaz, G., *Swarm Intelligence: From Natural to Artificial Systems*, Santa Fe Institute in the Sciences of the Complexity, Oxford Univ. Press, New York, 1999.

[4] Chialvo, D.R. and Millonas M.M., How Swarms Build Cognitive Maps, *The Biology of Intelligent Autonomous Agents*, 144, NATO ASI Series, pp. 439-450, 1995.

[5] Denebourg J.-L. et al, "The Dynamic of Collective Sorting Robot-like Ants and Ant-like Robots", in *1^{st} Conf. on Simulation of Adaptive Behavior: From Animal to Animats*, Cambridge, MA: MIT Press, pp. 356-365, 1991.

[6] Dasgupta, D. (Eds.), *Artificial Immune Systems and their Applications*, Springer-Verlag, Berlin, 1999.

[7] Dasarathy B.V. (Ed.), *Nearest Neighbour (NN) Norms: NN Pattern Classification Techniques*, IEEE Comp. Soc. Press, 1991.

[8] Denning D., "An Intrusion-Detection Model", *IEEE Transactions on Software Engineering*, Vol. SE-13, no. 2, pp.222-232, 1987.

[9] Duda R.O., Hart P.E*., Pattern Classification and Scene Analysis*, John Wiley & Sons, NewYork, 1973.

[10] Foukia, N., Hassas, S., Fenet, S. and Albuquerque, P., "Combining Immune Systems and Social Insect Metaphors: A Paradigm for Distributed Intrusion Detection and Response System", in *Proc. of Mobile Agents for Telecommunication Applications*, 5th International Workshop, MATA 2003, Marrakech, Morocco, 2003.

[11] Foukia, N., Di-Marzo, G., "Self-Organizing Architecture for Intrusion Detection and Response Systems", in *Proc. of HP-OVUA 2003 Workshop*, Geneva, 2003.

[12] Foukia, N., Hassas, S., Fenet, S. and Hulaas, J., "An Intrusion Response Scheme: Tracking the Source Using the Stigmergy Paradigm", in *Proc. of Security of Mobile Multi-agent Systems Workshop* (SEMAS- 2002), Bologna, Italy, 2002.

[13] Handl J. and Dorigo M., "On the Performance of Ant-based Clustering", in *Proc. of the 3rd Int. Conf. on Hybrid Intelligent Systems*, IOS Press, Australia, 2003.

[14] Chebrolu S., Abraham A. and Thomas J., Feature Deduction and Ensemble Design of Intrusion Detection Systems, Computers and Security, Elsevier Science, 2005 http://dx.doi.org/10.1016/j.cose.2004.09.008.

[15] *KDD Cup 1999* Intrusion detection data set: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup.data_10_percent.gz> .

[16] Lumer E.D. and Faieta B., "Diversity and Adaptation in Populations of Clustering Ants", in *Procs. Of SAB´94 – 3rd Conf. on Simulation of Adaptive Behavior: From Animal to Animats*, The MIT Press/Bradford Books, Cambridge, MA., 1994.

[17] Monmarché N. et al, On Improving Clustering in Numerical Databases with Artificial Ants", in *ECAL'99 Proc. - European Conf. on Artificial Life,* pp. 626-635, 1999.

[18] Ramos, V., Abraham, A., Evolving a Stigmergic Self-Organized Data-Mining, *Fourth International Conference on Intelligent Systems, Design and Applications, ISDA-04*, Hungary, ISBN 9637154302, pp. 725-730, 2004.

[19] Ramos, V. and Abraham A., Swarms on Continuous Data, in *CEC'03 – Proc. Congress on Evolutionary Computation,* IEEE Press, Australia, pp. 1370-1375, 2003.

[20] Ramos V., Pina, P. D Muge, F., Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies, in *Frontiers in Artificial Intelligence and Applications, Soft Computing Systems - Design, Management and Applications*, IOS Press, Vol. 87, pp. 500-509, 2002.

[21] Ramos V. and Merelo J.J., Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning, *Procs. of AEB´02 - 1st Spanish Conf. on Evolutionary and Bio-Inspired Algorithms*, Spain, pp. 284-293, 2002.

[22] Ramos V. and Almeida F., Artificial Ant Colonies in Digital Image Habitats - A Mass Behaviour Effect Study on Pattern Recognition, *From Ant Colonies to Artificial Ants - Procs. of the 2nd Int. Work. on Ant Algorithms*, Belgium, pp. 113-116, 2000.