Available online at www.sciencedirect.com

SciVerse ScienceDirect

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

Computers & Security

Swarm intelligence in intrusion detection: A survey

C. Kolias [a,b,*], G. Kambourakis [a,b], M. Maragoudakis [a,b]

[a] Laboratory of Information and Communication Systems Security, University of the Aegean, Samos GR-83200, Greece
[b] Department of Information and Communication Systems Engineering, University of the Aegean, Samos GR-83200, Greece

ARTICLE INFO

Article history:
Received 9 January 2011
Received in revised form
18 July 2011
Accepted 26 August 2011

Keywords:
Ant colony optimization
Ant colony clustering
Intrusion detection
Particle swarm optimization
Swarm intelligence
Survey

ABSTRACT

Intrusion Detection Systems (IDS) have nowadays become a necessary component of almost every security infrastructure. So far, many different approaches have been followed in order to increase the efficiency of IDS. Swarm Intelligence (SI), a relatively new bio-inspired family of methods, seeks inspiration in the behavior of swarms of insects or other animals. After applied in other fields with success SI started to gather the interest of researchers working in the field of intrusion detection. In this paper we explore the reasons that led to the application of SI in intrusion detection, and present SI methods that have been used for constructing IDS. A major contribution of this work is also a detailed comparison of several SI-based IDS in terms of efficiency. This gives a clear idea of which solution is more appropriate for each particular case.

1. Introduction

In the past years, numerous approaches have been proposed for computer systems protection from unauthorized use. Such approaches may involve symmetric and asymmetric encryption, include additional systems such as firewalls as well as sophisticated and complex security protocols. As the security mechanisms tend to evolve over time so do the methods adopted by the attackers. At the same time, new types of networks have made their appearance such as cellular networks, *Mobile Ad-Hoc Networks* (MANET) (Yang et al., 2004) and *Wireless Sensor Networks* (WSN) (Pathan et al., 2006). What is more, future implementations of 4G mobile networks (Fu et al., 2004) are expected to provide services for a large number of heterogeneous wireless access technologies. Nevertheless, each one of these networks has proven to carry its own security inefficiencies and vulnerabilities. As traditional approaches fail to fully counterattack intrusion attempts the need for an additional mechanism as the last line of defense has become a necessity. Thus, *Intrusion Detection Systems* (IDS) have quickly established themselves as one of the most basic components of every security infrastructure.

An IDS is a security system which is able to identify malevolent behavior (already finished or ongoing) against a protected network or computer. Without doubt, the construction of an efficient intrusion detection model is a challenging task. This is because an IDS must have a high attack *Detection Rate* (DR), with a low *False Alarm Rate* (FAR) at the same time. What might be even more challenging, is that an IDS must not be computational resource demanding and be intelligent enough in order to identify previously unseen attacks.

Since the appearance of the first IDS (Denning, 1987), a plethora of techniques has been proposed in order to boost their performance and effectiveness. It is only until recently though, that researchers sought inspiration in biology and

* *Corresponding author*. Department of Information and Communication Systems Engineering, University of the Aegean, Samos GR-83200, Greece. Tel.: +30 22730 82247; fax: +30 22730 82009.
E-mail addresses: kkolias@aegean.gr (C. Kolias), gkamb@aegean.gr (G. Kambourakis), mmarag@aegean.gr (M. Maragoudakis).
0167-4048/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2011.08.009

natural systems (Williamson, 2002). *Swarm Intelligence* (SI) as one of the many existing bio-inspired family of techniques, studies and emulates the behavior of swarms of animals for solving complex problems. Tasks such as nest organizing, seeking paths to food sources, or moving from one place to another as an organized unit have been analyzed and modeled. The IDS have applied these models for the execution of some critical procedures such as distinguishing between normal and abnormal behavior, tracing the source of an attack and for performance optimization. The motivation behind this is quite obvious: these natural systems possess a set of desirable characteristics that may immediately be inherited to the resulting IDS. For instance, a swarm of insects is able to complete complex tasks although it is based in a number of simple entities with very limited capabilities. Also, it is able to fulfill difficult undertakings even if its environment changes drastically, and function efficiently even if a small number of its population becomes extinct. Likewise, swarm based IDS are usually lightweight systems yet simple to implement, self-configurable, highly adaptive and extremely robust.

The clear advantages that SI approaches impose to the field of intrusion detection in conjunction with the ever increasing interest of both academia and industry in this field is the main driving force behind this work. This paper attempts to categorize and classify the work that has been done so far in the field of SI-based IDS. The taxonomy adopted is based primarily on the function of the natural swarm that acted as a source of inspiration for each one of the described SI-based IDS.

1.1. *Our contribution*

This work offers a comprehensive analysis of the internal mechanisms of numerous SI-based IDS. Although in the past, some works (Wu and Banzhaf, 2010) have touched upon a limited number of such systems, the current one is exhaustive and focuses solely on SI-inspired IDS. Another major contribution is the presentation of a detailed and constructive comparison of the efficiency of several SI-based IDS. By doing so, we attempt to highlight the possible beneficial impact and point out possible pitfalls of integrating SI techniques into IDS. A chart that indexes major SI-based IDS in chronological order with respect to relevant technologies is also contributed. Our work refers primarily to SI approaches or SI hybrid approaches. In this way, works that fall into the broader field of *Machine Learning* (ML) (Tesink, 2007; Haglund et al., 2000; Amini and Jalili, 2004; Dickerson and Dickerson, 2000) or adopt other biology inspired approaches (Kim, 2002; Jian et al., 2004) are considered out of scope. Also, this work concentrates on techniques and methodologies used for some core functionality of IDS such as supervised learning in terms of classification. Thus, SI approaches used for secondary functions or as preprocessing steps like *Feature Selection* (FS) or *Feature Reduction* (FR) (Sivagaminathan and Ramakrishnan, 2007; Gao et al., 2005b; Zainal et al., 2007), (although frequently applied in many IDS) have been intentionally neglected.

The remainder of this paper is organized as follows: The next section provides an introduction to both the concepts of intrusion detection and swarm intelligence. Section 3 gives an insight, categorizes and surveys several SI-based approaches used in intrusion detection. Section 4 compares major SI-based IDS. Finally, Section 5 concludes and provides suggestions for future research.

2. Relevant terms

2.1. *Intrusion detection*

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices (Scarfone and Mell, 2007). Systems that are assigned to perform all the procedures relevant to intrusion detection are called *Intrusion Detection Systems* (IDS). Although, there is a wide variety of mechanisms and frameworks that IDS systems employ, a generic architecture can be extracted. Usually, systems of this type are comprised of:

- A number of sensors which are responsible for gathering the appropriate data from the monitored system. Depending on the type of the IDS the sensors might be part of the system they protect or external.
- An analysis and configuration engine which is usually a centralized point that collects the data from the sensors and analyses them. This component might have to reconfigure the protected system accordingly if the results of the analysis indicate an intrusion during the response step. The response step might involve human interaction (e.g., the security administrator) or be fully automated.
- A report system that notifies the administrator for possible attacks.

In some IDS types (such as misuse detection IDS) a knowledge base which contains signatures of known attacks might also be present. This component is utilized by the analysis and configuration engine during a step known as the data analysis step and it must be frequently updated to include the signatures of the latest attacks. Finally, it is possible for a response engine to exist. The response engine might be able to take actions automatically or after specific command of the administrator. Fig. 1 depicts a high level architecture of a generic IDS that protects a network.

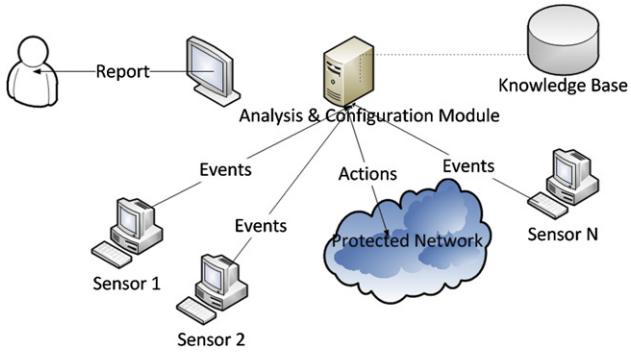

Fig. 1 – Architecture of a typical IDS.

It is possible to come across different classifications of the existing IDS based on different criteria. A first distinction can be made in terms of the location of the active sensing components of the IDS. Based on this aforementioned attribute, the IDS are usually classified into host-based and network-based. In host-based approaches the sensing components – or quite often the entire IDS per se – are installed on every host that requires protection. On the other hand, a network-based IDS monitors the network that contains the hosts of interest. This type of IDS is usually installed on multiple dedicated machines, which are possibly different from the protected hosts, and monitors the network traffic.

A more widespread categorization is based on the adopted data analysis approach. In this case, IDS may belong in one of the two main groups: *misuse detection* and *anomaly detection*. The first approach examines the activity of the entire infrastructure for patterns of misuses known beforehand, usually referred to as "attack identities". On the opposite, anomaly detection approaches analyze the behavior of the protected system over time toward extracting an approximate estimation of what behavior is considered normal (or legitimate). Any action that significantly deviates from that kind of behavior is considered an attack.

Beyond everything else, an IDS must be able to identify intrusions with high accuracy. At the same time it must not confuse legitimate actions that occur on a system with intrusive ones. These two criteria have been associated with two performance evaluation variables: (i) *Detection Rate* (DR), which is defined as the ratio of the number of correctly detected attacks to the total number of attacks, and (ii) the *False Alarm Rate* (FAR), or false positive rate, which is the ratio of the number of normal connections that are misclassified as attacks to the total number of normal connections. Normally, an IDS tries to maintain high detection rates while keeping false alarm rates as low as possible. Aside from these two basic criteria, Kim et al. identify a number of additional requirements for the realization of an effective IDS (Kim et al., 2007).

2.2. *Swarm Intelligence*

Nature has always been an inspiration to humans for complex problem solving. In the recent past, biology inspired approaches have made their appearance in a variety of research fields, ranging from engineering, computer science, economics, medicine and social sciences. Likewise, many biology inspired techniques have been proposed for intrusion detection. Swarm intelligence is one of them.

The term Swarm Intelligence (SI) was first introduced by Beni in the context of cellular robotics system (Beni and Wang, 1989). Methodologies, techniques and algorithms that this research field embraces draw their inspiration from the behavior of insects, birds and fishes, and their unique ability to solve complex tasks in the form of swarms, although the same thing would seem impossible in individual level. Indeed, single ants, bees or even birds and fishes appear to have very limited intelligence as individuals, but when they socially interact with each other and with their environment they seem to be able to accomplish hard tasks such as finding the shortest path to a food source, organizing their nest, synchronize their movement and travel as a single coherent entity with high speed etc. This achievement becomes even more significant if it is taken into account that they accomplish such tasks without the presence of a centralized authority (e.g., the queen of the hive) dictating any of this behavior. Applications of this can be found in NP-hard optimizations problems such as the traveling salesman, the quadratic assignment, scheduling, vehicle routing etc.

3. SI approaches in intrusion detection

Most IDS that will be examined in this section fall into the broad category of anomaly detection IDS. Bear in mind that systems of this type do not rely on a base of signatures of known attacks for their detection and thus are destined to recognize novel malicious behavior. Also, it is a common ground that intrusion detection problems in general and anomaly detection IDS in particular have to cope with huge volume and high dimensional datasets, the need for real time detection, and with diverse and constantly changing behavior. This is where computation intelligence comes into play and converges with the IDS realm. In a step known as training, a number of records that is already gathered from the sensing components of the system (in the form of network connection data or log file data) is fed to the analysis engine. After the training step the IDS goes online to protect the system in real time. A classification or clustering algorithm is applied in this component to categorize the behavior into normal or abnormal. So, in a sense, the intrusion detection problem is reduced to a classification or clustering problem. In this context researchers have always been seeking easy-to-implement methods that provide high quality results in a fast and efficient manner.

The unique characteristics of SI make it ideal for this purpose. More specifically, SI techniques aim at solving complex problems by the employment of multiple but simple agents without the need of any form of supervision to exist. Every agent collaborates with others toward finding the optimal solution. This happens via direct or indirect communications (interactions) while the agents constantly roam in the search space. In this respect, agents can be used for several hard tasks like finding classification rules for misuse detection, discover clusters for anomaly detection, keep track of intruder trails etc. Indeed, these self-organizing and distributed attributes are highly appreciable by offering the means to break down a difficult IDS problem into multiple simple ones assigned to agents. This potentially makes the IDS autonomous, highly adaptive, parallel, self-organizing and cost efficient. In the literature the efficiency of such systems is usually evaluated against one of the existing benchmarks that specifically target IDS (DARPA, 2008; Internet Exploration Shootout Dataset, 2008; KDD99, 2008; Unix User Dataset, 2008). This section thoroughly surveys SI-based approaches used in intrusion detection. The systems that are presented in this work are categorized primarily according to the adopted SI technique. The three main categories that accrue are: (a) IDS that make use of Ant Colony Optimization, (b) IDS that employ Particle Swarm Optimization and (c) IDS

that utilize Ant Colony Clustering. For each category, a brief introduction of the corresponding adopted SI technique is presented first. Each class may further be broken down into smaller subcategories leading to the following taxonomy scheme:

- ACO Oriented IDS Approaches
  - ACO for Detecting the Origin of an Attack
  - ACO for Induction of Classification Rules
- PSO Oriented IDS Approaches
  - PSO & Neural Network Hybrid Approaches
  - PSO & SVM Approaches
  - PSO & K-Means Approaches
  - PSO for Induction of Classification Rules
- ACC Oriented IDS Approaches
  - ACC & SOM Hybrid Approaches
  - ACC & SVM Hybrid Approaches

3.1. Ant colony optimization background

The foraging behavior of ants and more specifically their unique ability to find the shortest path from their nests to a food source has inspired the creation of perhaps the most successful algorithmic model which is known as *Ant Colony Optimization* (ACO). ACO portrays beneficial characteristics in environments with highly dynamic parameters.

Most ant species have very limited or no vision. At the same time they are deprived of speech or any other means of conventional communication. Nevertheless, ants seem to act in a strictly organized manner, which indicates that some sort of latent communication takes place. Indeed, experiments conducted to certain ant species prove that this communication occurs by depositing a substance called *pheromone* along the path they follow. In more detail, ants initially move randomly in order to locate a food source. As soon as they do so, ants carry food to their nest and deposit pheromone traces along the trail. Subsequently, ants decide on which of the available paths they shall follow based on the pheromone concentration deposited on each particular path. Paths with greater pheromone concentration have higher probability of being selected. Ants that follow the shortest path return to their nests earlier and pheromone on that path is reinforced with an additional amount sooner than the one in the longer path. Therefore, the selection among the paths is biased toward the shortest path.

Deneubourg et al. presented the double bridge experiment in which nest and food source were separated by a bridge of two branches of equal lengths (Deneubourg et al., 1990b). The authors noticed that the majority of ants will follow only one of the paths. Which one of the two, is randomly decided. Goss et al. extended the experiment by using paths of unequal lengths (Goss et al., 1989) showing that in all experiments the majority of the ants will eventually choose the shortest one as shown in Fig. 2. Dorigo et al. presented an algorithmic implementation of that behavior for solving minimum cost path problems on graphs known as *Simple Ant Colony Optimization* (SACO) (Dorigo and Stutzle, 2004), (Dorigo and Di Caro, 1999). In this model ants begin from a source node of a graph $G = (N, A)$ and try to reach a destination node following the shortest path. To each arc (i, j) of a graph an amount of artificial pheromone is deposited $\tau_{i,j}$. This information can be read and written by the ants to govern their movement to the next node. Specifically, the probability of an ant k located at a node i of choosing j as the next node to be visited is calculated as:

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{a}}{\sum_{l \in N_i^k} \tau_{il}^{a}} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases}$$

Where N_i^k of ant k when in node i contains all the nodes directly connected to i, except the predecessor of i. α is a parameter for controlling convergence speed. When the ant reaches its destination it has to return to the source. In this backward mode the ants deposit pheromone along the trail. Normally, the ant will attempt to follow the same route but if that route contains loops then it must eliminate them first, in order to avoid the problem of self-reinforcing loops. The new amount of pheromone in the arc (i, j) after ant k has traversed it in backward mode is calculated as:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^{k}$$

Pheromone trails evaporate over time. This mechanism can be seen as a way to avoid the problem of convergence to suboptimal paths, or a way to adapt to dynamic graph changes if they ever occur. Pheromone evaporation is simulated by applying the following equation to all arcs:

$$\tau_{ij} \leftarrow (1 - p)\tau_{ij},\ \forall (i, j) \in A$$

where $p \in (0, 1]$ is a constant.

3.2. ACO oriented IDS approaches

The AntNag algorithm was one of the first approaches that introduced the ACO into intrusion detection (Abadi and Jalali, 2006). The authors are motivated by the assumption that usually intruders unleash their attacks by taking advantage multiple vulnerabilities of the system. The AntNag algorithm perceives the set of all possible attack scenarios as a directed graph, called *Network Attack Graph* (NAG). Each edge represents an exploit and every complete path from an initial node to a target node corresponds to an attack scenario. The minimization analysis of this graph designates the minimum set of exploits that must be eliminated to assure that no attack scenario is feasible. This is actually an NP-hard problem as proven in the literature (Sheyner et al., 2002; Jha et al., 2002a,b). As a first step vulnerability scanning tools discover possible vulnerabilities of the system. These results along with other information (e.g., exploit templates, intruder's goal and connectivity between network hosts) are used to generate the NAG. Then based on that graph, a number of ants iteratively constructs a set of critical exploits by incrementally adding exploits until all attack scenarios are covered. At each construction step each ant chooses probabilistically an exploit (i.e., chooses an arc to move to) based on the amount of pheromone associated with that exploit. After that, the iteration-based solution is improved by local search. Finally, global updating rules modify the pheromone concentration on each trail. The effectiveness of this system seems to heavily

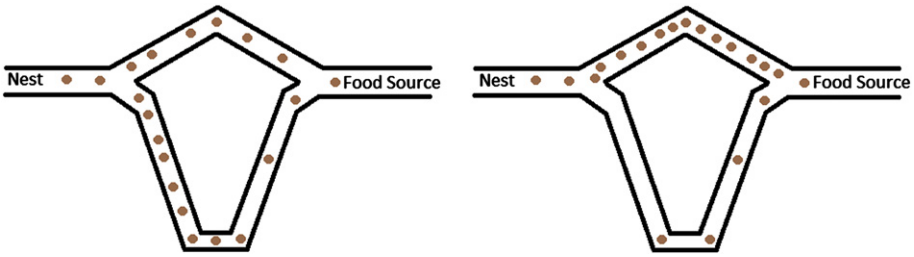

Fig. 2 – Extended double bridge experiment.

depend on how accurate the results of the vulnerability analysis are. Nevertheless, in real life scenarios and especially in newly deployed systems, not all vulnerabilities can be known beforehand. In addition, realistically it is expected the generated NAGs to be extremely large and complex.

Lianying and Fengyu proposed the separation of the IDS into independent detection units for increasing its performance and reducing misjudgment and misdetection rates (Lianying and Fengyu, 2006). The pheromone paradigm is adopted here to make these units communicate without having to directly exchange any information which would result in increasing the network load and creating possible security vulnerabilities. First off, the detection units analyze the behavior concerning a recourse they are assigned to and produce a suspicion degree value. If this value is greater than a threshold then they proceed to operations such as alerting, responding or recording. Otherwise, local information gathered by each one of these units is stored on a shared information database. This database can be perceived as the pheromone repository. The suspicion degree is summed up with the results of the other units and if the collective suspicion index of the system exceeds a threshold then this behavior is still perceived as intrusive. In other words, global system behavior emerges from local analysis and indirect communication of its autonomous units.

The rest of the approaches found in the literature can be organized into two major categories: (a) Those that use the ACO technique for locating the source of the attack, as part of the response step and (b) those that take advantage of the ACO for creating a set of rules that can classify network traffic as normal or into one of the attack classes.

3.2.1. *ACO for detecting the origin of an attack*

Fenet and Hassas proposed one of the first IDS architectures that make use of the ant colony metaphor to locate the source of an attack (Fenet and Hassas, 2001). Their system has been based on a number of mobile and static agents. The *pheromone server* is a static agent installed on each host meant to be protected. Among its others duties the pheromone server is in charge of spreading an alert-like message throughout the network in case of an intrusion. This message is perceived as of the ants' pheromone, and the pheromone server is in charge for its diffusion in a gradient pattern. The *watcher* is a static agent installed on each host which monitors processes of that host and its network connections. This means that the watcher is the core component of the detection part of the system. The *lymphocytes* are mobile agents that typically roam randomly through the network searching for pheromone traces. If pheromone trails are discovered they converge to the threatened machine and take the appropriate defensive actions. These actions depend on the type of the attack. Actually, lymphocytes are the core component of the response part of the system. In this case the ant colony analogy is only used as a part of the response system so that intrusions can be faced rapidly and more efficiently. The overall architecture leads to a fully distributed intrusion detection and response system.

IDReAM adopts a similar methodology for identifying and responding to network attacks (Foukia, 2005). The intrusion detection part adopts mechanisms from the human immune system, while the intrusion response module relies on the ACO paradigm. In this architecture, each node runs a *Mobile Agent* (MA) platform which hosts different types of mobile agents: *Intrusion Detection Agents* (IDA) and/or *Intrusion Response Agents* (IRA). IDAs move randomly on the network, then enter nodes and based on their local status they compute the *Suspicion Index* (SIn). If the SIn exceeds a specific threshold then the agent builds and diffuses the appropriate amount of pheromone. If the IRAs, which also traverse the network randomly, happen to track pheromone traces along their way, then they follow them back to their source where they initiate a response to the attack.

IDEAS migrates this approach on *Wireless Sensor Networks* (WSN) environment for locating the source of intrusions (Banerjee et al., 2005a,b). This system relies on agents embedded on each sensor that monitor their hosts, peers and network traffic for possible attack signatures. The network of sensors is presumed as a graph where other ant-like agents are placed on nodes randomly and traverse it. Ants move from their current node of the sensor network to the adjacent node that has the maximum number of violations represented as pheromone. Besides pheromone their movement is coordinated by mechanisms resembling the human social interaction as described by affective computing theory (Picard, 1997). Thus, the agents are characterized as *emotional ants*. In that way, the search becomes more accurate and efficient.

Chen et al. concentrated their efforts on a system that deals solely with *Denial of Service* (DoS) attacks (Chen et al., 2006). They proposed an IP trace-back approach for tracing the source of DoS attacks without relying solely on network routers to conduct the detection process. Their motivation was driven from the fact that conventional methods usually fail to trace the origin of attacks as intruders spoof the address of the network entity that generates the DoS traffic. According to their scheme, as a first step, the same amount of pheromone is set on each router and ants are positioned on the

victim node(s). Then ants will first read the topology information to discover routers in the same neighborhood and then calculate the probability to move to the next node with respect to traffic flow. The average amount of octets (network traffic) is used for pheromone calculation. In this way, the ants tend to favor routers with heavy traffic as their next node to move and the procedure is repeated until the boundary routers of the monitored network are reached. As in the case of real ants this creates a positive feedback loop which eventually forces most ants to converge to the same path.

Chang-Lung et al. describe an intrusive analysis model based on the design of honeypots and ant colony (Chang-Lung et al., 2009). The honeypot is a decoy system with many vulnerabilities aiming to attract the interest of potential intruders. Thus, conclusions can be extracted about the characteristics of attacks and the behavior of intruders before damage is done to the real system. In this model all network assets of the honeypot are associated with a pheromone value proportional to the significance of this resource. After intrusions or other malicious behavior the honeypot is configured in a way so that the amount of pheromone of each affected asset is increased. If attackers repeatedly attempt to compromise a resource then the concentration of pheromone will be higher. Next, the ACO is applied to trace the trail of attack and analyze the habits and interests of aggressors. Muraleedharan and Osadciw adopt a similar approach by integrating a honeypot architecture and the ACO algorithm in the sensor network realm this time (Muraleedharan and Osadciw, 2009). In this case a number of inexpensive nodes is actually used as a part of the IDS while it appears as a normal part of the sensor network. Tracking intruders is done in a similar way.

3.2.2. *ACO for Induction of Classification Rules*

Soroush et al. presented one of the pioneering works where ACO is used for intrusion detection unlike previous approaches where it was used for intrusion response (Soroush et al., 2006). Their proposed system is based on the classification Ant-Miner rule extracting algorithm (Parpinelli et al., 2002). Our pseudocode version of the Ant-Miner algorithm is given in the online resources of the manuscript (Swarm Intelligence in Intrusion Detection). A quick examination of the code indicates that this concept is very easy in implementation. Specifically, the authors adjusted the Ant-Miner algorithm to cope with high dimensional, high volume data, such as the ones analyzed for intrusion detection. Ant-Miner itself is inspired by the foraging behavior of ants in order to classify numerical data to one of some predefined classes. In particular, this algorithm utilizes ants to construct a set of candidate rules of the type:

if $(term_1 term_2 \ldots term_n)$ **then** $class_c$

In this case $term_i$ is formed by (a) an attribute of a record of the dataset, (b) an operator and (c) a value, e.g., IP = 182.123.0.2. The performance of the candidate rules is evaluated against a training set. Quality is measured by taking the confusion matrix of real and predicted instances, i.e. the number of true positives, false positives, false negatives and true negatives with respect to the training set. During the process the pheromone increases for the terms used for the construction of a rule proportional to the performance of the constructed rule. At the same time it decreases for all other terms (evaporation). Among the discovered rules the best one is selected and augmented to the discovered rules. This is done iteratively until a large base of rules is constructed which can be later on used in test sets as criteria for classifying network connections into intrusive or normal. Like all systems of this type this approach demands a pre-existing dataset to be used for training.

Junbing et al. also propose an Ant-Miner based classification system (Junbing et al., 2007). Its main contribution is the introduction of multiple ant colonies instead of a single one that the ant-miner normally employs. The authors noticed that the algorithm might be pushed back in the case where ants searching for best rules of a class B, have been mislead by the pheromone trails deposited at a prior time, by ants searching for rules of a class A. In this case each class is handled by different ant types organized into colonies. That is, each ant that belongs to a colony deposits a distinct type of pheromone which affects only the ants belonging to the same colony. Colonies are searched in parallel to finally discover one rule per colony. The rule with the best quality is selected and added to the rule set.

Fork is another IDS based on a variation of the Ant-Miner algorithm (Ramachandran et al., 2008). In this case the algorithm (and the IDS itself) is optimized to function under the constraints of ad-hoc networks. Due to the inherent limitation of these networks in terms of resources it is possible that some nodes may be unable to perform intrusion detection. Therefore, nodes may produce an intrusion detection task request and propagate it to the other nodes. Then the nodes compete according to an auctioning system for performing these tasks. The actual recognition of the intrusive network behavior is done by the winner nodes. The modifications on Ant-Miner which is responsible for this task include: (a) The priority assignment strategy: a method which identifies candidate solutions that may act as obstacles to the creation of rules and gives them priority. (b) Use of modularity: a method of forming clusters of similar pathways in the solution graph. Thus, terms that belong to the same cluster can be added without being evaluated by the heuristic function. (c) Use of attack thresholds: These modifications improve the processing time for the formation of more accurate rules.

Works of Abadeh et al. (Abadeh et al., 2008; Abadeh and Habibi, 2010) and Alipour et al. (Alipour et al., 2008) were among the first that combined genetic algorithms and ACO for the induction of accurate fuzzy classification rules. Fuzzy set theory (Zadeh, 1965) has been applied successfully in the past in the field of intrusion detection (Wang and Megalooikonomou, 2005) and has proven to provide very competitive DR and FAR percentages. The combination of Fuzzy set theory, Genetic Algorithms and SI is expected to boost the performance of an IDS. In this case, fuzzy *if-then* rules are coded as strings, with 5 linguistic values being represented by the following symbols: small (A_1), medium small (A_2), medium (A_3), medium large (A_4) and large (A_5). For instance, a rule which is coded as follows: (A_3, A_2, A_5, A_1), C_j, CF_j, can be translated as: if x_1 is medium and x_2 is medium small and x_3 is large and x_4 is small then the class is C_j with

certainty CF = CFj. For the most part their algorithm follows the flow of the Michigan algorithm (Ishibuchi and Nakashima, 1999; Ishibuchi et al., 1999), thus an initial population of fuzzy if-then rules is randomly generated. This population is then evaluated and in the process genetic operations take place so that a new population can be produced by generating new rules. At this point, the ant colony algorithm takes a fuzzy rule and modifies it by performing a number of predefined changes so that an improved version of the same rule is produced. The algorithm then continues as normal by replacing a pre-specified number of if-then rules with newly generated ones and finally stops according to some termination rules. In other words the authors added a local search step based on ACO to the Michigan algorithm. By doing so the entire (global) search capability of the algorithm is enhanced.

Agravat et al. noticed that when a fitness function is utilized, many rules of the same pattern are generated for similar set of data (Agravat et al., 2010). In this approach the algorithm stores all the generated high quality rules by the entire ant colony, instead of simply saving the best rule produced by each ant. Next, all rules are initially sorted with respect to their predictive accuracy in decreasing order and sorted again with respect on false positives this time, but in increasing order.

Summarizing, ACO inspired IDS in most cases utilize this technique as a response mechanism (usually for tracking the source of an intrusion) rather than a detection one. Actually, some works use the ACO approach for extracting classification rules. When possible, we have gathered experimental results for the approaches discussed here. These results are analytically presented and discussed further down in Section 4.

3.3. Particle swarm optimization background

Particle Swarm Optimization (PSO) seeks inspiration in the coordinated movement dynamics of groups of animals. Reynolds' studies in the bird flocking behavior (Flocks, 1987) indicate that the kinesiology of the entire flock is a result of the individual behavior of birds which simply follow 3 basic rules: (i) collision avoidance, which dictates individuals to avoid neighbor mates by readjusting their physical position, (ii) velocity matching, which dictates individuals to synchronize their speed with neighbor mates, (iii) flock centering, which dictates individuals to stay close to flockmates. Reynolds applied this model to simulate the aesthetics of the flock chorography with 3D computer experiments.

The sociologist Wilson, noticed that individual members of a swarm may profit from the discoveries and previous experience of other members of the swarm during tasks such as food discovery for instance (Wilson, 1975). In other words, a larger number of swarm members, increases the chances of locating a rich food source and the social information sharing among the swarm members offers an additional advantage.

Later, Kennedy and Eberhart introduced the term of Particle Swarm Optimization and their work was the main influence of the basic PSO model (Kennedy and Eberhart, 1995). According to this model a fitness function exists $f : \mathbb{R}^n \to \mathbb{R}$ which measures the quality of the current solution. A number S of particles (solutions) is placed randomly inside the hyperspace in the position $x_i \in \mathbb{R}^n$ each having a random velocity $v_i \in \mathbb{R}^n$. The particles move in the hyperspace and at each step evaluate their position according to the fitness function. Each particle in the swarm represents a possible solution. The basic update rule for the speed is:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_i - x_i) + c_2 r_2 (g - x_i)$$

Where ω is the inertia weight constant, c_1 and c_2 are the acceleration constants, r_1 and r_2 are random numbers, p_i is the personal best position of particle i, g is the global best position among all particles in the swarm, and x_i is the current position of particle i. Moreover, the update rule for the position is:

$$x_i(t+1) = x_i + v_i(t+1)$$

Two key features of this model are that (a) the speed (and therefore the next position) of each particle is calculated according to the findings of both that particle and the findings of the rest of the swarm and that (b) the global best solution is communicated among all particles of the swarm. Our pseudocode version of the Standard Particle Swarm Optimization algorithm is included in the online resources of the manuscript (Swarm Intelligence in Intrusion Detection). It is obvious that the algorithm is easy-to-implement. Readers may notice the obvious similarities PSO portray to Genetic Algorithms. Indeed, they both consider a fitness function that acts as a criterion for population reproduction and update their population using randomness. However, PSO does not incorporate genetic operators such as mutation and gene crossover. Furthermore, PSO retain a kind of memory, which is essential toward the convergence to an optimal solution.

3.4. PSO oriented IDS approaches

Dozier et al. presented a system that can be used as a part of an IDS to identify possible attacks that would otherwise go unnoticed, i.e. perceived as normal traffic (Dozier et al., 2004, 2007). The authors pose the question if it is more preferable to manually try to identify holes in the security system, or let potential intruders do that job. A module of the system namely *Red Teams* emulates the behavior of hackers. The Red Teams component employs PSO techniques in their intrusion methodology. The acquired results can dynamically help the IDS reconfigure on-the-fly in order to be more effective.

Since most of the PSO based IDS are hybrid anomaly detection systems, it is possible to categorize them according to the additional ML method that is employed. We distinguish (a) hybrid PSO-Neural Network Systems, (b) hybrid PSO-SVM Systems, (c) hybrid PSO-K-means Systems. Another category is comprised of IDS that employ PSO for the extraction of classification rules.

3.4.1. PSO & neural network hybrid approaches

Artificial Neural Networks (ANN) is one of the most popular soft computing techniques for data classification. Hence the largest volume of research has been done on the application of the ANN in the field of intrusion detection. PSO is a technique which is used extensively in combination with various types of ANN for improving the performance of the resulting system.

Michailidis et al. were among the first who merged the two aforementioned soft computing techniques to create an improved system for intrusion detection (Michailidis et al., 2008). Their work presents an integrated IDS implemented in Java. During the training phase the PSO is executed recursively to train the network. Specifically, each particle in the PSO corresponds to the synaptic weights of the network. The optimal synaptic weights are fed to ANN, which conducts the main part of the classification with improved efficiency, during the testing phase. Generally, systems of this type follow a similar two step approach. An ANN classifier is the system component that conducts the classification process underneath, while a PSO algorithm runs on top of it to improve critical parameters and train the synaptic weights. The input layer of the ANN is constructed by the *m* features of the monitored network connection attributes and the output layer is comprised of the normal and abnormal types. The particles are viewed as multidimensional vectors composed of the ANN parameters and the particle with the optimum adaptation values is searched globally. This can be easily seen in Fig. 3.

A *Wavelet Neural Network* (WNN) (Zhang and Benveniste, 1992) is a feedforward ANN based on wavelet analysis (Torrence and Compo, 1998). ANN of this type use a wavelet function on the hidden layer instead of the sigmoid one. The resulting systems may achieve higher learning speed and avoid the creation of local minima, therefore this type of NN has been used frequently in intrusion detection. Liu and Liu (Liu et al., 2009; Liu and Liu, 2009) noticed that PSO when used instead of the typical methods of connection weight adjustment (such as the *Gradient Descent* (GD) algorithm (Moller, 1993)), it becomes possible to avoid the oscillation effect in which the optimization is trapped in local minima. They applied these principles with two variations of PSO, namely *Quantum Particle Swarm Optimization* (QPSO) (Yang et al., 2004) and *Modified Quantum Particle Swarm Optimization* (MQPSO) respectively (as described in that work), to train a WNN.

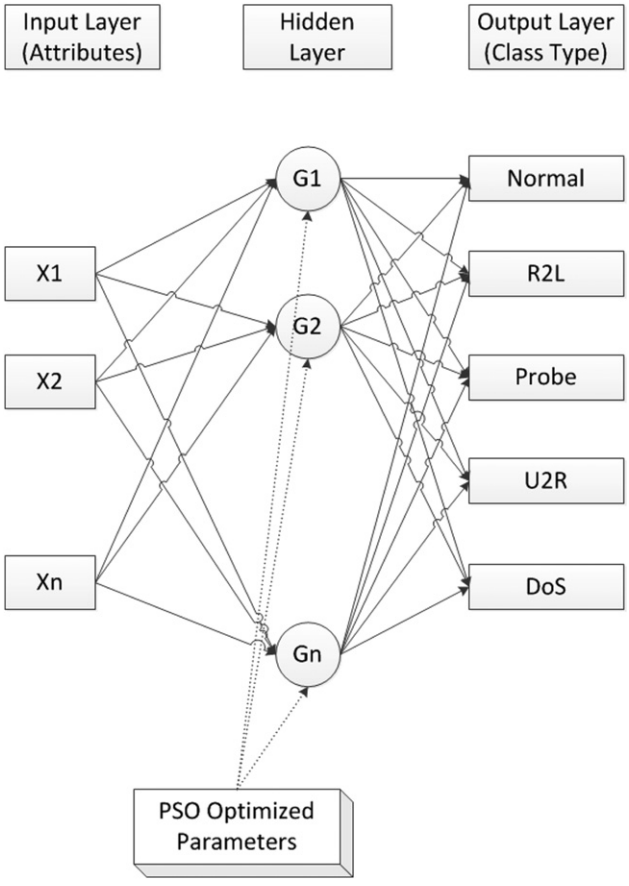

Fig. 3 – Generic hybrid PSO-ANN architecture.

Ma et al. (Ma et al., 2007) propose a similar infrastructure and uses both the *Conjugate Gradient* (CG) algorithm (Hestenes and Stiefel, 1952) and QPSO rather than relaying in one of them for parameter optimization. The QPSO has a better global searching ability compared to the CG, so it is preferable to be used in the initial steps of the training to quickly cover a larger portion of the search space. As the generations (iterations steps) proceed, the solution might be trapped. At that point CG is utilized to help QPSO escape this possible status. Ma and Liu (Ma and Liu, 2010) adopt principles of fuzzy set theory and integrate them on a WNN based IDS. The hybrid ANN is able to "fuzzily" describe fault characteristics of a state classified as "abnormal".

Radial Basis Function Neural Networks (RBF) (Orr, 1996) is a type of probabilistic Neural Network frequently adopted by IDS. A RBF may achieve classification faster because the classification process is based on the simple measure of the distance of the centers of the neurons from the inputs fed to it. This characteristic makes RBF a good candidate for network intrusion detection. Nevertheless, RBF requires certain parameters like the number of center and the variance of the RBF to be chosen manually. If the parameters are not optimal this will have an impact on the accuracy of the resulting classification. Systems such as (Ma et al., 2008b; Chen et al., 2009) use PSO as an extra step for RBF optimization and achieve better performance than standard RBF. This has been verified by experimental results included in the same paper. Tian and Liu (Tian and Liu, 2010) use the same logic to create a hybrid PSO-ANN system but also introduce an evolutionary mutation algorithm as an extra step in order to (a) protect PSO from trapping into local minima, (b) increase the diversity of the population, and (c) expand the scope of the search.

3.4.2. *PSO & SVM hybrid approaches*

Similarly to ANN, another technique frequently used in combination with PSO is *Support Vector Machines* (SVM) (Burges, 1998; Cortes and Vapnik, 1995). SVM is based on structural risk minimization of statistical learning theory and shows good learning ability and generalization skill in high dimensional or noisy datasets, two attributes highly appreciated in intrusion detection. However, one of the basic shortcomings of this technique is the difficulty to determine certain parameters so that the performance of the algorithm becomes optimal. Wang et al. were among the first who combined the two techniques (Wang et al., 2009). They used two different flavors of PSO the *Standard Particle Swarm Optimization* (SPSO) and *Binary Particle Swarm Optimization* (BPSO) (Kennedy and Eberhart, 1997) for seeking optimal SVM parameters and extracting a feature subset respectively. In the latter step each particle represents a solution that indicates which features

and parameter values should be kept. Finally, the results (selected features and parameter values) along with the training dataset are fed to the SVM classifier which executes normally to classify specific network behavior as intrusive or normal. In a similar way, Ma et al. (Ma et al., 2008a) propose a combinatorial BPSO-SVM technique where dataset features and the crucial SVM parameters are represented by each particle position. The choice of SVM parameters for the classification process and the selection of the optimum features happens simultaneously in one step instead of two. Then the classification process based on SVM is conducted which (given the inputs from the previous step) is much more accurate. Hybrid PSO-SVM systems are common in literature (Gao et al., 2005a, 2006; Srinoy and Rajabhat, 2007; Zhou et al., 2009; Tian and Liu, 2009; Liu et al., 2010).

3.4.3. PSO & K-means hybrid approaches

Xiao et al. (Xiao et al., 2006) combined the simplicity and good local search of the K-Means algorithm (MacQueen, 1967) with the PSO to create an IDS. According to this algorithm, each particle's position is the set of D dimensional centroids produced by the K-Means algorithm. Thus, each particle's position can be represented as an array:

$$\begin{bmatrix} Z_{11} & Z_{12} & \dots & Z_{1D} \\ Z_{21} & Z_{22} & \dots & Z_{2D} \\ \dots & \dots & \dots & \dots \\ Z_{k1} & Z_{k2} & \dots & Z_{kD} \end{bmatrix}$$

where D is the number of the dimensions of the dataset (therefore the centroids dimension) and k represents the number of clusters. Initially, data points are assigned to k clusters in a random manner. Then the centroids are calculated and the position of each particle is deduced. For each particle, the fitness function evaluates the position and if necessary the P_{best} and G_{best} values are updated along with that of velocity and position. Finally, the K-Means algorithm runs in order to optimize the new generation of particles. The algorithm converges to local optimum with very low probability and has high convergence speed. Yongzhong also proposes a similar PSO-K-Means hybrid system (Li et al., 2009).

3.4.4. PSO for induction of classification rules

Guolong et al. explored the efficiency of a novel rule-based IDS based on PSO (Guolong et al., 2007). According to the authors each particle is a network connection that represents a rule. Their algorithm recursively creates a particle population from a training dataset. Then, for each particle, it computes its fitness and updates the P_{best} and G_{best}, i.e. the velocity and the position values of that particle. When some criteria are met the G_{best} particle (the fittest rule) is inserted into the rule sets and at the same time the training data covered by this rule are deleted. The authors noticed that PSO cannot be directly applied to network intrusion datasets because in this case the attributes take distinct values. To overcome this limitation they also proposed a new coding scheme that maps distinct attribute values to non-negative integer values as well. Chang et al. (Zhao and Wang, 2009) achieve better detection rates by incorporating a more accurate fitness function to the system described above.

Summarizing PSO oriented IDS utilize this technique as an extra step of a conventional classification mechanism. Section 4 contains experimental results from several approaches discussed earlier. From the results it is obvious that the integration of the PSO algorithm can greatly improve the performance of the IDS.

3.5. Ant colony clustering background

Many ant species exhibit an interesting behavior concerning the organization of their nest. By simply observing their nest it is obvious that eggs, brood and food are not randomly scattered. On the contrary, they follow a strict organization into piles of homogenous or similar objects. Moreover, if the nest was messed by an external force then the ants will reconstruct these piles rapidly. This behavior is achieved while each ant appears to work autonomously without receiving any orders by ants placed higher in the hierarchy.

Based on these observations mathematical models have been constructed to simulate the clustering and sorting behavior of real ants. Deneubourg et al. constructed the basic model to describe this behavior and applied it in robotics (Deneubourg et al., 1990a). According to their model ant-like robots without communication abilities, hierarchical organization or any global mapping of their environment, move randomly on a two dimensional space and pick up objects in less dense areas. Being able to carry them they dispose them in locations where a large number of the same type of object exists. Thus, the probability of picking up or dropping objects is relevant to two factors: the density of objects in the immediate neighborhood and the similarity of objects. More specifically, the probability for an unloaded ant-like robot to pick up an object o_j is calculated as:

$$p_{pick}(o_i) = \left(\frac{k^+}{k^+ + f}\right)^2$$

Where f is an estimation of the spaces in the neighborhood that are occupied by objects of the same type, and k^+ is a constant. When there is a small number of objects in the neighborhood then $f << k^+$ and p_{pickup} tends to 1 and as a result the objects will likely be picked up. On the other hand, the probability for a loaded ant-like robot to drop the object if the robot is located on an empty cell is calculated as:

$$p_{putdown}(o_i) = \left(\frac{f}{k^- + f}\right)^2$$

In case where many objects are observed in the immediate neighborhood then $f >> k^-$ and $p_{putdown}$ tends to 1, which in turn means that the object will most likely be dropped. The model assumes that each ant-like robot has a short term memory of m steps that records what is met in each of the last m time steps. Since the robot moves randomly in space, this sampling provides an estimation of the type of objects that exist in the immediate neighborhood. For example, for a memory of 5 steps at time t the memory string could have been "_AA_B" indicating that the robot met 2 objects of type A and 1 object of type B. Thus $f_A = 2/5$ and $f_B = 1/5$.

Lumer and Faieta generalized the aforementioned model for clustering multidimensional datasets (Lumer and Faieta, 1994). The algorithm scatters the multidimensional records of the dataset in a theoretical two dimensional grid. At each

iteration of the algorithm the elements are rearranged in such a way so that similar elements are grouped together to form compact clusters (ideally one for each class in the dataset). This is done in theoretical level and no changes are in the order of the records in the dataset. According to the LF model, the probability of picking an element i, is defined as:

$$P_{pick}(i) = \left(\frac{k_p}{k_p + f(i)}\right)^2$$

Where k_p is a constant and $f(i)$ is the local estimation of the density of elements in a small surrounding area defined as a square of d nodes. Likewise, the probability of dropping a carried item is calculated by:

$$P_{drop}(i) = \begin{cases} 2f(i) \text{ if } f(i) < k_d \\ 1 \text{ otherwise} \end{cases}$$

The density dependent function $f(i)$ for an element i, at a particular grid location, is defined as:

$$f(i) = \begin{cases} \frac{1}{d^2}\sum_j (1 - d(i,j)/a) \text{ if } f(i) < k_d \\ 0 \text{ otherwise} \end{cases}$$

In the expression above, $d(i,j)$ measures the dissimilarity between all elements in the local area that surrounds node i and α scales the dissimilarities. Since the elements are vectors, d measures dissimilarities by calculating the Euclidian distance between the elements in nodes i and j. The normalizing term d^2 equals the total number of sites in the local area of interest, thus $f(i)$ may only take its maximum value if all the neighborhood is occupied by identical elements. The algorithm described above can lead to the construction of clusters of similar objects from an initial randomly scattered state. Fig. 4 visually depicts this process. This achievement is of paramount importance for any IDS. Based on the assumption that intrusive activity happens rarer than legitimate one, datasets that contain low level network traffic can be analyzed in order to form clusters that represent different types of attacks or normal activity respectively. The LF algorithm and subsequent variations of it were also utilized with great success in a number of other applications such as text document classification (Vizine et al., 2005) to name one. As always this paragraph focuses solely on the application of this family of algorithms in intrusion detection, neglecting the rest of the potential applications. Our pseudocode version for the Lumer-Faieta algorithm (commonly referred as LF algorithm) is included in the online resources of the manuscript (Swarm Intelligence in Intrusion Detection). From the code one can understand that the algorithm has very low complexity of implementation.

3.6. ACC oriented IDS approaches

Ramos and Abraham were two of the first researchers who attempted to introduce the LF algorithm described above into the intrusion detection realm (Ramos and Abraham, 2005). Similarly, to the LF algorithm their model called ANTIDS was based on a number of ant-like agents that pick up and drop items with a certain probability to form clusters. In this case, instead of having agents exploring the terrain randomly seeking objects or clusters, they suggested that it would be more efficient to have agents guided by pheromone traces. Moreover, the computation of average object similarities which is dictated by the LF algorithm is avoided since it is blind to the actual number of objects present in a given neighborhood. According to the authors this strategy (a) allows ants to find clusters of objects in an adaptive way, (b) eliminates the need of short term memory of the agents thus making the algorithm less resource demanding, (c) it accelerates the algorithm in finding optimal solutions since the ants tend to move to areas of higher interest.

Tsang and Kwong noticed that data used in cases of intrusion detection analysis are typically large in volume as regards to instances and high dimensional as regards to features (Tsang and Kwong, 2005a, 2006). The original LF algorithm suffers from two problems: (a) many homogenous clusters are formed and thus it is difficult to be merged when they are spatially separated into a large search space, (b) the density of similarity measures, favors cluster formation in locally dense regions but discriminates dissimilar objects intensively. In other words, elements of type A close to compact clusters of elements of type B tend to remain isolated. Under these circumstances, the authors proposed a variation of the LF algorithm to deal with these two inefficiencies. In their version of the model called ACCM, a combined measurement of local regional entropy and average similarity was used in order for the model to identify clusters into coarse, compact, and incorrectly merged ones. Further improvements were introduced to boost the efficiency of the algorithm such as two different types of pheromones for guiding the ant-like agents toward clusters (for object deposition) and toward isolated objects (for object pick up) respectively. Based on this classification algorithm the authors proposed an integrated multi-agent IDS architecture for industrial control systems later on (Tsang and Kwong, 2005b).

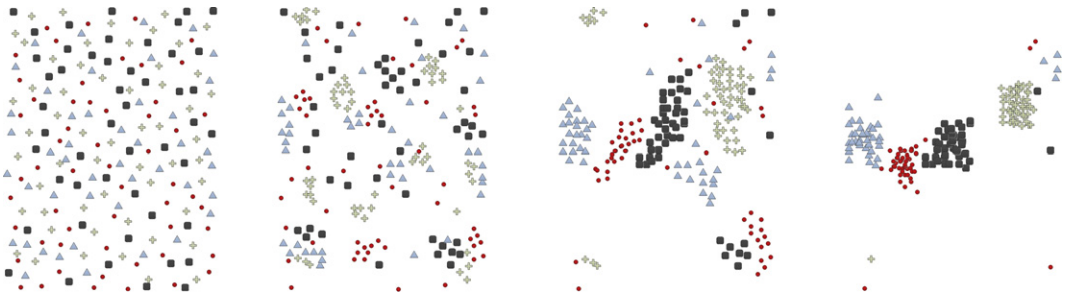

Fig. 4 – Arangement of data into 4 clusters on the grid after (a) 0 (b) 10,000 (c) 50,000 (d) 130,000 iterations.

Rajeswari et al. (Rajeswari et al., 2008) noticed that connection attributes that belong to User to Root (U2R) and Remote to Local (R2L) attack classes (for more details see Section 4) have very close resemblance with the ones found in normal traffic. Because of this, most of the existing IDS approaches suffer from low DR in these two classes specifically. The authors proposed a hybrid multilevel IDS. In the first level the enhanced version of C4.5 (Quinlan, 1993) algorithm is used to classify connection record found in the dataset into DoS, Probe and "Others". The class of "Others" contains U2R, R2L and normal connections. In the second level the classic ACC algorithm splits the data into two clusters. The resulting clusters represent normal and abnormal traffic. The cluster with abnormal connections can be distinguished easily it is typically much smaller in size. Finally, on the third level the C4.5 algorithm classifies the abnormal traffic. By splitting the classification processes into multiple levels the resulting IDS achieves competitive DR and FR rates.

As it is clear, many ACC-based IDS relay to this algorithm for achieving clustering of the data. A sub-categorization is possible to be achieved by taking into account the auxiliary techniques applied for improving the results. In this case, we distinguish: (a) hybrid ACC-SOM and (b) hybrid ACC-SVM systems.

3.6.1. *ACC & SOM hybrid approaches*

Feng et al. followed a similar approach to the LF model although in their case the neighborhood is perceived as circular area around the ant, and the pick and drop probabilities are calculated based on non-linear functions (Feng et al., 2006, 2005). This remark can assist in solving linear inseparable problems. After the clustering step, the procedure where the formatted clusters are labeled is initiated. The actual labeling is done by calculating the maximum quantities difference and the labeling clusters threshold. These variables depend on the result of the ant clustering algorithm rather than the user-controlled parameter selection which is the standard practice, therefore are more accurate. Finally, live detection is possible by calculating the *a-posteriori* probability with the help of the Bayes theorem. This makes the detection procedure more accurate since it is independent of cluster centers. Later on, Feng et al. (Feng et al., 2007a,b) fused the algorithm described above with a variation of the *Self Organizing Maps* (SOM) (Kohonen, 1995) neural network model. *Dynamic Self-Organizing Maps* (DSOM) (Alahakoon et al., 2000) was added as an extra step before the main ant colony clustering process. Rather than placing the input data randomly on a 2-D grid, DSOM is used to represent the input data. As an extra step the ants move the objects in the output layer of DSOM and normally form clusters. This additional step increases the efficiency of cluster formation process. Fig. 5 depicts this concept.

3.6.2. *ACC & SVM hybrid approaches*

Zhang and Feng presented a hybrid framework (Zhang and Feng, 2009) which combines SVM (Cortes and Vapnik, 1995) and ant colony clustering for increasing the performance of IDS. Typical SVM techniques when used for clustering in intrusion detection, map the network data as data points in a multidimensional space. SVMs create hyperplanes between two classes of objects (i.e. data points that correspond to normal and abnormal network traffic). The best hyperplane is the one with the maximum distance between marginal points of the two classes. Also, SVMs require an initial training set of labeled data to be provided. Active training (Duan et al., 2007) is a technique for decreasing the necessary amount of this training data. This is a multistep process where, for each iteration, only some of the training data are chosen and the hyperplane is modified gradually. The authors incorporate ant colony clustering as a selection technique of the data points used for training at each step. The active training algorithm is extended by adding an extra step of cluster creation around marginal points and then the selection is made from the data points of these clusters.

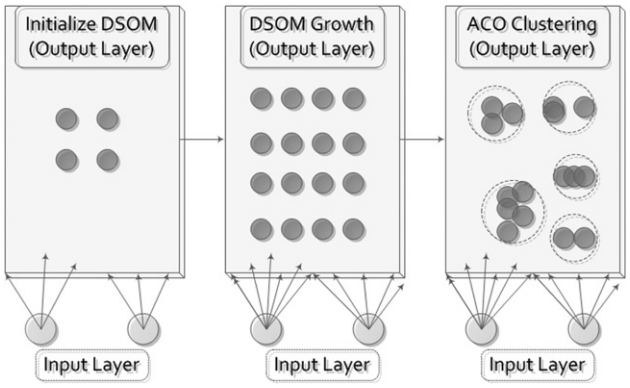

Fig. 5 – Concept of hybrid DSOM and ACC IDS.

An intrusion detection model based on the combination of SVMs and ant colony clustering can also be found in the literature (Srinoy, 2006). According to the authors ACC should have a more active role which is to refine the clusters initially produced with SVMs. This is in contrast to approaches, where ACC is only used to reduce the amount of training data for the SVM (Zhang and Feng, 2009). The utilized ACC algorithm in this case is based on the fuzzy ants concept (Kanade and Hall, 2003). This algorithm is a variation of the original LF algorithm where ants pick up and drop objects on a two dimensional board to initially form heaps, each positioned on a single cell, rather than directly forming clusters. As a second stage of this algorithm the ants pick up and drop the entire heaps formed in the first stage to construct clusters. ACC in most cases acts as the basic clustering mechanism of the IDS. The experimental results given in the next section demonstrate that ACC leads to solid IDS with very competitive DR and FAR.

The approaches that were presented above are arranged in chronological order in Fig. 6.

4. Evaluation

4.1. *Experimental tools and methodologies*

Most IDS conduct their efficiency tests against a set of records which may contain connections from the network, records from log files, call sequences of the operating system or user command sequences. Depending on the specific needs of each

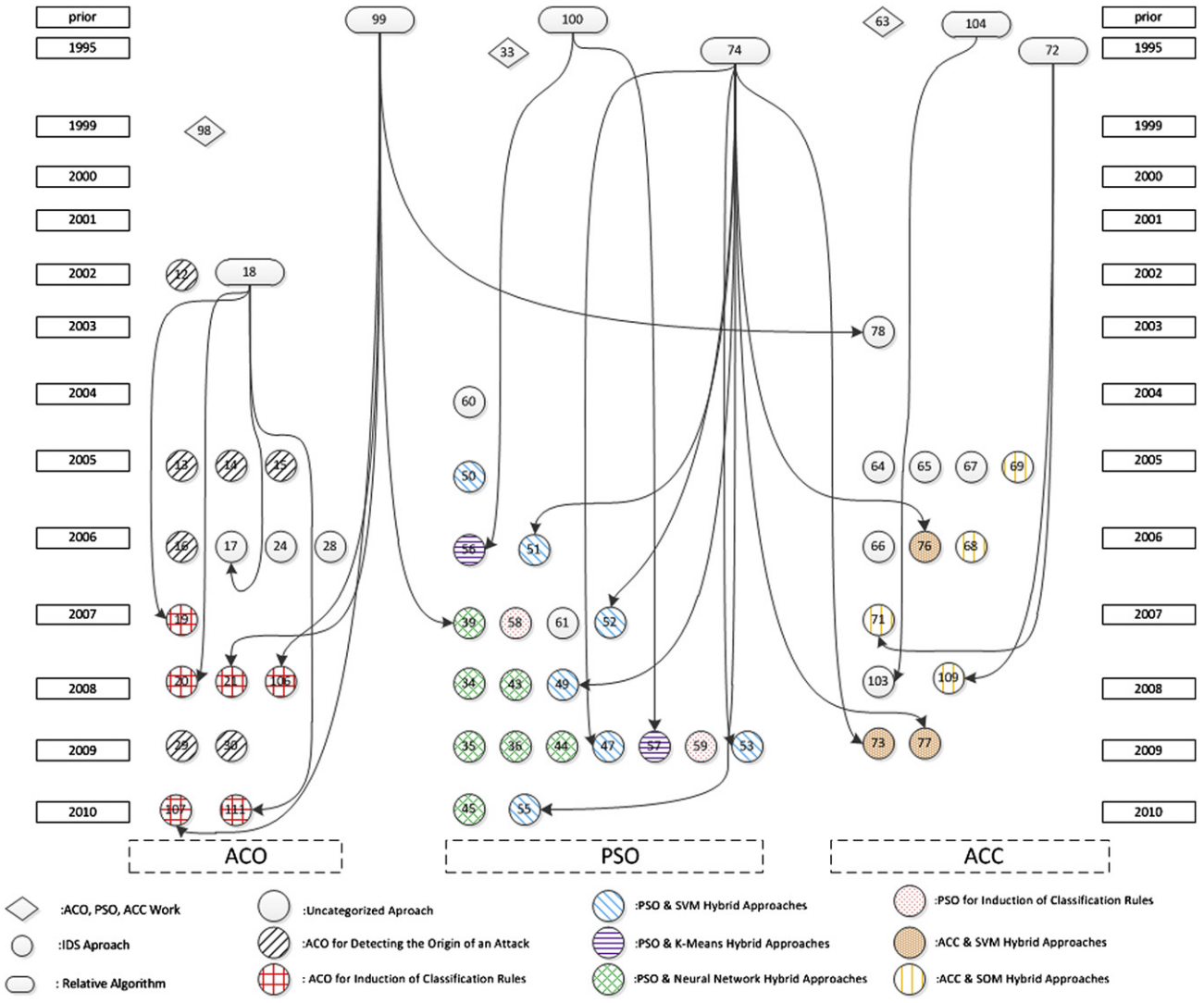

Fig. 6 – Major SI-based IDS approaches in chronological order. IDS are organized according the SI technique adopted by the system. Arrows indicate other ML methods that possibly influenced each IDS.

particular IDS this dataset might be a custom one (i.e. created specifically for the purposes of the evaluation) or in most of the cases it might be one of the pre-existing datasets that are used as benchmarks for intrusion detection (DARPA, 2008; Internet Exploration Shootout Dataset, 2008; KDD99, 2008; Unix User Dataset, 2008). The most commonly used dataset is KDD99 (KDD99, 2008). It was created by the DARPA Intrusion Detection Evaluation Project (Lippmann et al., 2000) on behalf of the MIT Lincoln Labs. For the purposes of the project a simulated LAN which operated as if it was a true U.S. Air Force LAN was created. The simulated LAN consisted of 3 target machines running different operating systems and services and 3 other machines that generated traffic. A dedicated component recorded the traffic and logged the TCP/IP connections into a TCP dump file. During the period of data collection which lasted 7 weeks the LAN would become target to a number of different types of attacks. The resulting dataset was split in training and testing sets. The training set is composed of 4,940,000 network connections each one containing 41 quantitative and qualitative attributes. There are 24 types of attacks (such as buffer overflow, rootkit, neptune, satan, smurf etc.) classified by experts into one of 4 categories. Each record in the dataset is labeled as normal traffic or one of the following 4 classes of intrusive behavior:

1. *DoS attacks*: where use of a specific service is denied to legitimated users.
2. *Probe attacks*: where information about the system is revealed to non authorized entities.
3. *User to Remote (U2R) attacks*: where access to administrator account types is gained by non authorized entities.
4. *Remote to Local (R2L) attacks*: where access to hosts is gained by non authorized entities.

This dataset was used during the International Knowledge Discovery and Data Mining Tools competition in 1999 and since then it has become the de-facto standard benchmark for intrusion detection.

Despite its wide adoption by the academic community, KDD99 has become the target of criticism (McHugh, 2000;

Mahoney and Chan, 2003; Sabhnani and Serpen, 2004). One of the points of concern is its large size especially that of the training set which makes it virtually impossible for an IDS to use it for its training phase. A second point that raises serious objections is the unrealistic distribution of normal versus intrusive records in the test set (with the later to be over 80% of the entire set), which under normal occasions is very rare. Moreover, the unrealistic distribution expands to the attacks types themselves with some of them (like DoS type attacks) making their appearance much more frequently than others (like U2R and R2L). Finally, in their majority, U2R and R2L classes contained in the test set, are comprised by attacks not seen before in the training set.

For the above reasons, it is quite common for a minimized version of the KDD99 dataset to be used. This version contains only 10% of the total records. The training set contains 494,021 connections, where 97,278 correspond to normal traffic and the rest belong to one of 22 attack types. The testing set contains 311,029 connections with 37 attack types 17 of which are new attack types that were not seen in the training set. Also, in many cases, the researchers create a custom-tailored version of the dataset.

4.2. *Discussion*

In Section 3 we described how different SI approaches have been applied to IDS. Certain conclusions can be extracted about the success of such undergoing which will be summarized here. Table 1 presents a comparison of some of the previously described approaches. The original winner entry (Elkan, 1999) of the KDD99 cup is also included in the comparisons. The winner entry might be considered outdated by today's standards but it was included mainly for giving a common base of comparison. All experiments contained in the table were conducted upon the KDD99 test set, after the system was trained with the 10% subset of the KDD99 training set. Note that the table contains only these works that were immediately comparable to each other (for instance were tested upon the same dataset and not a custom one) and the authors provided the corresponding experimental results. As explained earlier, KDD99 dataset has been the target of criticism. The results included in the comparison should not be used to extract conclusions about the efficiency of these systems in real life situations. The ultimate goal of this comparison is to highlight strong points and inefficiencies of these IDS and in a more abstract level the fitness of each adopted SI approach. The table contains the detection rates for each class separately as well as the overall detection rates. The winner entry of KDD99 cup performs very poorly in U2R and R2L attack classes since these two types contain a big number of attacks not seen in the training set. From the comparison table it is obvious that the SI systems in all given cases greatly outperform the winner entry in the U2R class and in almost all cases in the R2L class. Better results are also shown for the rest of the classes and for all of the discussed approaches. This confirms that the SI systems are highly adaptive and are able to detect novel attack types easier.

The majority of the IDS that make use of ACO, utilize this mechanism for procedures done after the attack has taken place, rather than for procedures relevant to intrusion detection itself. More specifically, many approaches employ ACO for tracing the source of an intrusion and in some cases even responding to that intrusion at its source (Fenet and Hassas, 2001; Foukia, 2005; Banerjee et al., 2005a; Banerjee et al., 2005b; Chen et al., 2006). One can notice that the majority of these systems are implemented using the mobile agents technology. Mobile agents have the privilege of relocating and in a sense they exist virtually anywhere on the network. ACO introduced many advantages to the IDS architecture. For example, ACO dictates the movement of the agents in a more dynamic manner depending on the severity and the location of the threat. Moreover, it reduces the amount of data exchanged by making use of asynchronous communication techniques like pheromone traces and finally it leads to fully distributed IDS architectures.

Fewer proposals on the other hand saw the potential of ACO in the improvement of the classification process. It is

Table 1 – Performance comparison of several SI-based IDS[a].

ML type	Winner KDD99	NN based techniques				SVM based techniques			Classification rules			None			
SI type		PSO				PSO			PSO	ACO		ACC			
	(Elkan, 1999)	(Chen et al., 2009)	(Flocks, 1987)	(Ma et al., 2008b)	(Ma et al., 2007)	(Liu et al., 2010)	(Zhou et al., 2009)	(Wang et al., 2009)	(Zhao and Wang, 2009)	(Alipour et al., 2008)	(Abadeh and Habibi, 2010)	(Ramos and Abraham, 2005)	(Tsang and Kwong, 2005a)	(Tsang and Kwong, 2005b)	(Feng et al., 2006)
Normal	94.5	N/A	96.88	N/A	N/A	N/A	N/A	N/A	N/A	98.5	96	99.64	98.5	98.8	99.1
Probe	83.3	88.86	92.20	N/A	N/A	86.48	N/A	N/A	N/A	82.5	86.25	98.29	86.9	87.5	97.18
DoS	97.1	92.57	97.74	N/A	N/A	88.48	N/A	N/A	N/A	98.5	98.83	99.98	97.5	97.3	99.35
U2R	13.2	91.14	52.86	N/A	N/A	85.52	N/A	N/A	N/A	76.3	72.8	64	27.2	30.7	63
R2L	8.4	94.29	8.30	N/A	N/A	84.53	N/A	N/A	N/A	89	33.45	99.47	11.0	12.6	97.79
DR	90.9	N/A	N/A	96.77	97.3	N/A	97.26	99.84	92.2	95.5	94.33	N/A	92.25	N/A	N/A
FAR	N/A	N/A	0.61	8.01	4.89	N/A	N/A	N/A	3.97	0.0018	N/A	N/A	1.5	N/A	N/A

a IDS are organized by the adopted SI technique as well as the auxiliary Machine Learning technique. Results include the DR for each attack class, the overall DR and the FAR.

noticeable that all these approaches apply ACO for classification rules extraction and all of them relay on a modification of the Ant-Miner algorithm which uses the pheromone concept for finding more quickly good quality classification rules. This has an impact not only in the overall speed of the algorithm but also in the number of rules which normally is reduced meaning that the detection rate is increased. The validity of the statements above is proven by experimental results presented in the corresponding works (Soroush et al., 2006; Junbing et al., 2007). Note that the experimental results given in these two works are not immediately comparable because they use different percentage of the training and testing datasets. Nevertheless, since both approaches are based on variations of the same algorithm it can be argued that the modifications the authors applied in (Soroush et al., 2006) managed to increase the detection rate of the system only by 0.1%. At the same time it reduced the false alarm rate by 1.4% comparing to the original Ant-Miner algorithm. Also, the multiple colonies concept (Junbing et al., 2007) increased the overall system detection rate by 6.02% but did not manage to reduce significantly the false alarm rate (it was reduced only by 0.08%). Generally, the Ant-Miner algorithm was originally tested on datasets with distinct record values and it is not optimized for datasets like KDD99. The level of improvement in detection rate shown in (Wu and Banzhaf, 2010) should be considered as a step in the right direction since it made the algorithm competitive to the rest of machine learning solutions.

PSO in intrusion detection is rarely used as the exclusive method for classification. The main reason behind this is that PSO shows a great tendency to converge to a suboptimal solution on early stages of its execution and naturally it is outperformed. It is clear from the above that the majority of the relative research treats this technique as a supplementary step to some other machine learning classifier which conducts the main part of the classification. By taking advantage of the two basic characteristics of PSO, which are the simplicity of the implementation and the fast discovery of a good solution, all hybrid systems of this type are able to present improvements on their detection rates.

One basic point to be taken into account is that the use of PSO has significantly boosted the performance of all the machine learning techniques in which it was applied. More specifically, the approach described in (Chen et al., 2009) managed to achieve detection rates improved by 7.15% for Probe class, 7.43% for DoS class, 4.28% for U2R type attacks and 6.29% for R2L when compared to a conventional RBF detection system. Other systems such as (Ma et al., 2008b) and (Zhou et al., 2009) improve their overall detection rates by 5.52% and 3.59% respectively which proves that PSO can have a positive impact on the performance of the IDS system. It is safe to say that the use of PSO into an existing machine learning based IDS is expected to enhance the system's DR accuracy by a magnitude greater than 3%. The PSO classification rules based approaches (Guolong et al., 2007) achieve lower detection rates which is expected since as stated in (Zhao and Wang, 2009) this method in general has limited descriptive power (it is confined by few operators only such as "AND", "OR", "NOT") and leads to poorer results.

Unlike most ACO approaches (where the ACO technique is utilized mostly for response) and PSO approaches (where the PSO technique is utilized for improving performance of other classification algorithms), most ACC methods, rely solely on an ACC algorithm for the classification process. Systems (Ramos and Abraham, 2005; Tsang and Kwong, 2005a,b 2006; Feng et al., 2006) are based on some variation of the basic LF algorithm for clustering the records into different classes. What is more interesting is that these systems provide some of the most solid results with the detection rate for each class to be constantly kept on high levels. The resulting systems perform extremely well for the R2L class and with the exception of two cases (Tsang and Kwong, 2005a,b), all other systems achieve detections rates over 97%. The importance of this remark gains added value if we consider that most traditional approaches perform extremely poorly in that particular class (the winner of KDD99 achieved only 8.4%). It is safe to say that ACC produces some of the best results for the R2L class among all machine learning approaches used for intrusion detection. Although all ACC approaches perform extremely well for Normal class (99.1% on average), Probe and DoS, their performance is moderately poorer for the U2R attack class. A possible explanation to this phenomenon could be attributed to the fact that ACC does not suffer from sparseness in data since density can be effectively treated through the pheromone trail criterion.

5. Conclusions and future directions

As explained in the previous section, SI techniques have established themselves as a solid option for any contemporary IDS. Nevertheless, many aspects exist that remain unexplored. First of all the contemporary approaches seem to fail to take advantage of the full potential of ACO for the detection part (i.e. the main classification process). Most existing systems that rely on ACO for intrusion detection adopt some sort of rules extraction technique which has proven (as already explained) to have an upper limit in its potential. The low complexity of ACO algorithm establishes it as a major candidate for the creation of fast, robust and adaptive IDS. The combination of ACO with some other machine learning technique is expected to lead to highly adaptive IDS. On the other hand, PSO based IDS have been extensively studied in combination with other ML techniques constantly providing solid DR rates. Unfortunately, with the incorporation of multiple techniques the computation requirements are expected to increase. A very interesting study would be that of the time required for training and live detection of those systems. As far as ACC-based systems, they seem to provide excellent detection rates in all but one attack classes (U2R) as can be seen from the table in previous section. Since the work done so far with hybrid ACC/Machine Learning approaches is minimal, it would be interesting to study the effects in DR and FAR when ACC is combined with another machine learning classifier. A good idea is one approach constantly provides good results for that particular attack class, such as ANN or SVM.

Another potential field of excel of SI-based approaches which seems to be neglected so far, is the creation of

distributed IDS. Since most of the SI algorithms relay or could be implemented with the help of agents it is obvious that highly distributed architectures could be created easily. Parallel computing methods could increase the training speed and training quality of the IDS, increase the system's accuracy and potentially be deployed for protecting ad-hoc network architectures. The success of such concept has been already investigated and proved by very little works in literature (Janakiraman and Vasudevan, 2009). Moreover, the proliferation of mobile and ad-hoc/sensor networks that make use of devices with limited computational power, complexity and computational requirements is an aspect that should be taken into serious consideration. Furthermore, it is obvious from Section 3 that many of the SI-based IDS incorporate additional algorithms or internally allow very high number of iterations (through setting of specific parameters) in an effort to boost the system's detection rates. Both of these factors are expected to have a negative impact on the requirements of the system in terms of computational resources. It is therefore necessary to provide a more standard complexity analysis alongside the metrics that correspond to the detection accuracy of the IDS.

REFERENCES

Abadeh MS, Habibi J. A hybridization of evolutionary fuzzy systems and ant colony optimization for intrusion detection. The ISC International Journal of Information Security 2010; 2(1):33–46.

Abadeh MS, Habibi J, Soroush E. Induction of fuzzy classification systems via evolutionary ACO-based Algorithms. International Journal of Simulation, Systems, Science, Technology 2008;9(3).

Abadi M, Jalali S. An ant colony optimization algorithm for network vulnerability analysis. Iranian Journal for Electrical and Electronic Engineering; 2006:106–20.

Agravat D, Vaishnav U, Swadas PB. Modified ant miner for intrusion detection. In: Proceedings of the Second International Conference on Machine Learning and Computing 2010. p. 228–232.

Alahakoon D, Halgamuge SK, Srinivasan B. Dynamic self-organizing maps with controlled growth for knowledge discovery. IEEE Transactions on Neural Networks 2000;11(3):601–14.

Alipour H, Khosrowshahi E, Esmaeili M, Nourhossein M. ACO-FCR: applying ACO-based algorithms to induct FCR. In: Proceedings of the World Congress on Engineering (IWCE) 2008. p. 12–17.

Amini M, Jalili R. Network-based intrusion detection using unsupervised adaptive resonance theory (ART). In: Proceedings of the 4th Conference on Engineering of Intelligent Systems (EIS 2004) 2004.

Banerjee S, Grosan C, Abraham A. IDEAS: Intrusion Detection Based on Emotional Ants for Sensors. In: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications 2005a. p. 344–349.

Banerjee S, Grosan C, Abraham A, Mahanti PK. Intrusion detection in sensor networks using emotional ants. International Journal of Applied Science and Computations 2005b;12(3):152–73.

Beni G, Wang J. Swarm intelligence in cellular robotics systems. In: Proceedings of NATO Advanced Workshop on Robots and Biological System 1989. p. 703–712.

Burges CJC. A tutorial on support vector machines for pattern recognition. Knowledge Discovery and Data Mining 1998;2(2): 121–67.

Chang-Lung T, Chun-Chi T, Chin-Chuan H. Intrusive behavior analysis based on honey pot tracking and ant algorithm analysis. In: Proceedings of the 43rd Annual 2009 International Carnahan Conference on Security Technology 2009. p. 248–252.

Chen M-C, Chiang B, Jeng C, Yang CR, Lai GH. Tracing denial of service origin: ant colony approach. Applications of Evolutionary Computing; 2006.

Chen ZF, Qian PD, Chen ZF. Application of PSO-RBF neural network in network intrusion detection. In: Proceedings of the 3rd International Symposium on Intelligent Information Technology Application 2009. p. 362–364.

Cortes C, Vapnik V. Support vector networks. Machine Learning 1995;20:273–97.

The DARPA-Lincoln Dataset. Retrieved January 26, 2008, from http://www.ll.mit.edu/IST/ideval/data/data_index.html.

Deneubourg JL, Goss S, Franks N, Sendova Franks A, Detrain C, Chretien L. The dynamics of collective sorting robot-like ants and ant-like robots. In: Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats. 1990a. p.356–363.

Deneubourg JL, Aron S, Goss S, Pasteels J-M. The self-organizing exploratory pattern of the Argentine ant. Journal of Insect Behavior 1990b;3(1):159–68.

Denning D. An intrusion detection model. IEEE Transactions of Software Engineering 1987;13(2):222–32.

Dickerson JE, Dickerson JA. Fuzzy network profiling for intrusion detection. In: Proceedings of the 19th International Conference of the North American on Fuzzy Information Processing Society (NAFIPS). 2000. p.301–306.

Dorigo M, Di Caro G. The ant colony optimization meta-heuristic. New Ideas in Optimization; 1999:11–32.

Dorigo M, Stutzle T. Ant colony optimization. MIT Press; 2004.

Dozier G, Brown D, Hurley J, Cain K. Vulnerability analysis of AIS-based intrusion detection systems via genetic and particle swarm red teams. In: Proceedings of the Congress on Evolutionary Computation (CEC2004). 2004. p. 111–116.

Dozier G, Brown D, Hou H, Hurley J. Vulnerability analysis of immunity-based intrusion detection systems using genetic and evolutionary hackers. Applied Soft Computing 2007;7(2): 547–53.

Duan D, Chen S, Yang W. Intrusion detection system based on support vector machine and active learning. Computer: Engineering; 2007.

Elkan C. Results of the KDD'99 classifier learning contest. SIGKDD. Explor. Newsl 1999;1(2):63–4.

Fenet S, Hassas S. A distributed intrusion detection and response system based on mobile autonomous agents using social insects communication paradigm. In: Proceedings of the First International Workshop on Security of Mobile Multiagent Systems (SEMAS). 2001. p. 41–58.

Feng Y, Wu ZF, Wu KG, Xiong ZY, Zhou Y. An unsupervised anomaly intrusion detection algorithm based on swarm intelligence. In: the Proceedings of the Fourth International Conference on Machine Learning and Cybernetics. 2005. p. 3965–3969.

Feng Y, Zhong J, Ye CY, Wu ZF. Clustering based on self-organizing ant colony networks with application to intrusion detection. In: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA '06). 2006. p.1077–1080.

Feng Y, Zhong J, Xiong Z, Ye CY, Wu KG. Intrusion detection classifier based on dynamic SOM and swarm intelligence clustering. Advances in Congnitive Neurodynamics ICCN; 2007a:969–74.

Feng Y, Zhong Z, Xiong Z-Y, Ye C-X, Wu K-G. Network anomaly detection based on DSOM and ACO clustering. Advances in Neural Networks; 2007b:947–55.
RG Reynolds. Flocks, herds, and schools: a distributed behavioral model. Computer Graphics 1987;21(4):25–34.
Foukia N. IDReAM: Intrusion Detection and Response executed with Agent Mobility. In: Proceedings of The International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05). 2005. p. 264–270.
Fu X, Hogrefe D, Narayanan S, Soltwisch R. QoS and security in 4G networks. In: Proceedings of the 1st CIC/IEEE Global Mobile Congress (GMC) .2004. p. 117–122.
Gao H, Wang X, Yang H. Swarm intelligence and SVM based network intrusion feature selection and detection. Technical Report. Shanghai: College of Information Science and Engineering, East China University of Science and Technology; 2005a.
Gao HH, Yang HH, Wang XY. Ant colony optimization based network intrusion feature selection and detection. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics. 2005b. p.3871–3875.
Gao H, Yang H, Wang X. Selection and detection of network intrusion feature based on BPSO-SVM. Technical Report. Shanghai: College of Information Science and Engineering, East China University of Science and Technology; 2006.
Goss S, Aron S, Deneubourg JL, Pasteels JM. Self-organized shortcuts in the Argentine ant. Naturwissenschaften 1989; 76(12):579–81.
Guolong C, Qingliang C, Wenzhong G. A PSO-based approach to rule learning in network intrusion detection. Fuzzy Information and Engineering; 2007:666–73.
Haglund AJ, Hatanen K, Sorvari AS. A computer host-based user anomaly detection system using the self-organizing map. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN'00). 2000. p.411–416.
Hestenes MR, Stiefel E. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards 1952;49(6):409–36.
The Internet Exploration Shootout Dataset. Retrieved January 26, 2008, from http://ivpr.cs.uml.edu/shootout/network.html.
Ishibuchi H, Nakashima T. Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes. IEEE Transactions on Industrial Electronics 1999;46(6):1057–68.
Ishibuchi H, Nakashima T, Muratam T. Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems. IEEE Transactions on Systems. Man and Cybernetics 1999;21(5):61–8.
Janakiraman S, Vasudevan V. ACO based distributed intrusion detection system. International Journal of Digital Content Technology and Its Applications 2009;3(1):66–72.
Jha S, Sheyner O, Wing JM. Minimization and reliability analysis of attack graphs. Technical Report. USA: School of Computer Science, Carnegie Mellon University; 2002a .
Jha S, Sheyner O, Wing MJ. Two formal analyses of attack graphs. In: Proceedings of the 15th IEEE Computer Security Foundations Workshop. 2002b. p. 49–63.
Jian G, Da-Xin L, Bin-Ge C. An induction learning approach for building intrusion detection models using genetic algorithms. In: Proceedings of the Fifth World Congress on Intelligent Control and Automation (WCICA). 2004. p.4339–4342.
Junbing H, Dongyang L, Chuan C. An improved ant-based classifier for intrusion detection. In: Proceedings of the Third International Conference on Natural Computation (ICNC 2007). 2007. p. 819–823.
Kanade PM, Hall LO. Fuzzy ants as a clustering concept. In: Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society. 2003. p. 227–232.
The KDD99 Dataset. Retrieved January 26, 2008, from http://kdd.ics.uci.edu/databases/kddcup99/task.html.
Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings of the IEEE International Joint Conference on Neural Networks. 1995. p. 1942–1948.
Kennedy J, Eberhart R. A discrete binary version of the particle swarm algorithm. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. 1997. p. 4104–4108.
Kim J, Bentley PJ, Aickelin U, Greensmith J, Tedesco G, Twycross J. Immune system approaches to intrusion detection – a review. Natural Computing 2007;6(4):413–66.
Kim JW. Integrating artificial immune algorithms for intrusion detection. PhD Thesis. University College London 2002.
Kohonen T. Self-Organizing Maps. Berlin Germany:Springer-Verlang. 1995
Li Y, Yang G, Xu J, Zhao B. Anomaly detection for clustering algorithm based on particle swarm optimization. Journal of Jiangsu University of Science and Technology(Natural Science Edition); 2009.
Lianying Z, Fengyu L. A Swarm-Intelligence-based intrusion detection technique. IJCSNS International Journal of Computer Science and Network Security 2006;6(7):146–50.
Lippmann R, Haines JW, Fried JD, Korba J, Das K. The 1999 DARPA off-line intrusion detection evaluation. Computer Networks 2000;34(4):579–95.
Liu L, Liu Y. MQPSO based on wavelet neural network for network anomaly detection. In: Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom '09). 2009. p. 1–5.
Liu Y, Ma R, Lin X. Network anomal detection wavelet neural network based on QPSO. Journal of Liaoning Technical University(Natural Science); 2009.
Liu H, Jian Y, Liu S. A new intelligent intrusion detection method based on attribute reduction and parameters optimization of SVM. In: Proceedings of the Second International Workshop on Education Technology and Computer Science (ETCS). 2010. p.202–205.
Lumer R, Faieta B. Diversity and adaptation in populations of clustering ants. In Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats. 1994. p. 501–508.
Ma R-H, Liu Y. Wavelet fuzzy neural network based on modified QPSO for network anomaly detection. Applied Mechanics and Materials 2010;20-23:1378–84.
Ma R, Liu Y, Lin X. Hybrid QPSO based wavelet neural networks for network anomaly detection. In: Proceedings of the Second Workshop on Digital Media and its Application in Museum and Heritages. 2007. p. 442–447.
Ma J, Liu X, Liu S. A new intrusion detection method based on BPSO-SVM. In: Proceedings of the International Symposium on Computational Intelligence and Design, 2008a. p. 473–477.
Ma R, Liu Y, Lin X, Wang Z. Network anomaly detection using RBF neural network with hybrid QPSO. In: Proceedings of the IEEE International Conference on Networking, Sensing and Control 2008b. p. 1284–1287.
MacQueen JB. Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1967. p. 281–297.
Mahoney M, Chan PK. An Analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. Recent Advances in Intrusion Detection 2003. p. 220–237.
McHugh J. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations

as performed by Lincoln Laboratory. ACM Transactions on Information and System Security (TISSEC) 2000;3(4):262–94.
Michailidis E, Katsikas SK, Georgopoulos E. Intrusion detection using evolutionary neural networks. In: Proceedings of the Panhellenic conference on informatics 2008 (PCI 2008). p. 8–12, 2008.
Moller MF. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 1993;6(4):525–33.
Muraleedharan R, Osadciw LA. An intrusion detection framework for sensor networks using honeypot and Swarm Intelligence. In: Proceedings of the 6th Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous '09) 2009. p. 1–2.
Orr M. Introduction to radial basis function networks. Technical report. Institute for adaptive and neural computation Edinburg: Edinburgh University; 1996.
Parpinelli RS, Lopes HS, Freitas AA. Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation 2002;6(4):321–32.
Pathan ASK, Hyung-Woo L, Choong Seon H. Security in wireless sensor networks: issues and challenges. In: Proceedings of The 8th International Conference on Advanced Communication Technology (ICACT) 2006. pp. 1048.
Picard RW. Affective computing. , Cambridge, MA: MIT Press; 1997.
Quinlan JR. C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann; 1993.
Rajeswari LR, Kannan A, Baskaran R. An escalated approach to ant colony clustering algorithm for intrusion detection system. Distributed Computing and Networking; 2008: 393–400.
Ramachandran C, Misra S, Obaidat MS. FORK: a novel two-pronged strategy for an agent-based intrusion detection scheme in ad-hoc networks. Computer Communications 2008; 31(16):3855–69.
Ramos V, Abraham A, ANTIDS: Self organized ant based clustering model for intrusion detection system. In: Proceedings of The Fourth IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST'05) 2005. p. 977–986.
Sabhnani M, Serpen G. Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set. Journal of Intelligent Data Analysis 2004;8(4):403–15.
Scarfone K, Mell P. Guide to intrusion detection and prevention systems (IDPS). Technical report. NIST: National Institute of Standards and Technology. U.S. Department of Commerce; 2007.
Sheyner O, Haines J, Jha S, Lippmann R, Wing JM. Automated generation and analysis of attack graphs. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy 2002. p. 273–284.
Sivagaminathan RK, Ramakrishnan S. A hybrid approach for feature subset selection using neural networks and ant colony optimization. Expert Systems with Applications 2007;33(1): 49–60.
Soroush E, Saniee Abadeh M, Habibi JA. Boosting ant-colony optimization algorithm for computer intrusion detection. In: Proceedings of The IEEE 20th International Symposium on Frontiers in Networking with Applications 2006.
Srinoy S, Rajabhat S. Intelligence system approach for computer network security. In: Proceedings of the Fourth IASTED Asian Conference on Communication Systems and Networks 2007. p. 89–95.
Srinoy S. An adaptive IDS model based on swarm intelligence and support vector machine. In: Proceedings of the International Symposium on Communications and Information Technologies 2006. p. 584–589.
Swarm Intelligence in Intrusion Detection: A Survey (Online Material), http://www.icsd.aegean.gr/postgraduates/kkolias/swarm-intelligence-in-intrusion-detection/online-resources.pdf
Tesink S. Improving intrusion detection system through machine learning. Technical Report. ILK Research Group. Tilburg University; 2007.
Tian W, Liu J. Intrusion detection quantitative analysis with support vector regression and particle swarm optimization algorithm. In: Proceedings of International Conference on the Wireless Networks and Information Systems, 2009 (WNIS '09). p. 133–136.
Tian W, Liu J. A new network intrusion detection identification model research. In: Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR) 2010. p. 9–12.
Torrence C, Compo G. A practical guide to wavelet analysis. Bulletin of the American Meteorological Society 1998;79(1): 61–78.
Tsang W, Kwong S. Unsupervised anomaly intrusion detection using ant colony clustering model. In: Proceedings of the 4th IEEE International Workshop on Soft Computing as Transdiciplinary Science and Technology 2005. p. 223–232.
Tsang CH, Kwong S. Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction. In: Proceedings of the IEEE International Conference on Industrial Technology 2005 (ICIT 2005). p.51–56.
Tsang W, Kwong S. Ant colony clustering and feature extraction for anomaly intrusion detection. Swarm Intelligence in Data Mining; 2006:101–21.
The Unix User Dataset. Retrieved January 26, 2008, from http://kdd.ics.uci.edu/databases/UNIX_user_data/UNIX_user_data.htm.
Vizine AL, de Castro LN, Gudwin RR Text document classification using swarm intelligence. In: Proceedings of the International Conference on Integration of Knowledge Intensive Multi-Agent Systems 2005. p.134–139.
Wang Q, Megalooikonomou V. A clustering algorithm for intrusion detection. In: Proceedings of the SPIE Conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005. p.31–38.
Wang J, Hong X, Ren R, Li T. A real-time intrusion detection system based on PSO-SVM. In: Proceedings of the International Workshop on Information Security and Application 2009 (IWISA 2009). p. 319–321.
Williamson M. Biologically inspired approaches to computer security. Technical Report. Bristol: HP Laboratories; 2002.
Wilson EO. Sociobiology: the new synthesis. Belknap Press; 1975.
Wu SX, Banzhaf W. The use of computational intelligence in intrusion detection systems: a review. Applied Soft Computing 2010;10(1):1–35.
Xiao L, Shao Z, Liu G. K-means algorithm based on particle swarm optimization algorithm for anomaly intrusion detection. In: Proceedings of The Sixth World Congress on Intelligent Control and Automation 2006 (WCICA2006). p. 5854–5858.
Yang H, Luo H, Ye F, Lu S, Zhang L. Security in mobile ad hoc networks: challenges and solutions. IEEE Wireless Communications 2004;11(1):38–47.
Yang S, Wang M, Licheng J. A quantum particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation 2004 (CEC2004). p. 320–324.
Zadeh LA. Fuzzy sets. Inf. Control 8 1965. p. 338–353.
Zainal A, Maarof MA, Shamsuddin SM. Feature selection using rough-dpso in anomaly intrusion detection In: Proceedings of the Conference on Computational Science and its Application (ICCSA) 2007. p. 512–524.
Zhang Q, Benveniste A. Wavelet networks. IEEE Transactions on Neural Networks 1992;3(6):889–98.

Zhang Q, Feng W. Network intrusion detection by support vectors and ant colony. In: Proceedings of the 2009 International Workshop on Information Security and Application 2009. p. 639–642.
Zhao C, Wang W. An improved PSO-Based rule extraction algorithm for intrusion detection. In: Proceedings of International Conference on the Computational Intelligence and Natural Computing 2009 (CINC '09). p.56–58.
Zhou T, Li Y, Li J. Research on intrusion detection of SVM based on PSO. In: Proceedings of the International Conference on Machine Learning and Cybernetics 2009. p. 1205–1209.

Constantinos Kolias holds a Diploma in Computer Science from Technological Educational Institute of Athens, Greece and MSc in Information and Communication System Security. He is currently a Ph.D. candidate, supervised by Dr. G. Kambourakis, at the Department of Information and Communication Systems Engineering, University of the Aegean, Greece. His primary research interests lie in the field of: Intrusion Detection, WiMax Security, UMTS Security, RFID Security, Ubiquitous Computing, Pervasive Applications Development, User Adaptive Applications Development.

Georgios Kambourakis received the Diploma in Applied Informatics from the Athens University of Economics and Business and the Ph.D. in Information and Communication Systems Engineering from the Department of Information and Communications Systems Engineering of the University of Aegean. He also holds a M.Ed. from the Hellenic Open University. Currently, Dr. Kambourakis is a Lecturer at the Department of Information and Communication Systems Engineering of the University of the Aegean, Greece. His main research interests are in the fields of mobile and wireless networks security and privacy, VoIP security and mLearning. He has been involved in several national and EU funded R&D projects in the areas of Information and Communication Systems Security. He is a reviewer of several IEEE and other international journals and has served as a technical program committee member in numerous conferences.

Manolis Maragoudakis holds a PhD from the Department of Electrical and Computer Engineering , University of Patras and a diploma in Computer Science from the Computer Science Department, University of Crete. The thesis was entitled "Reasoning under uncertainty in dialogue and other natural language systems using Bayesian network techniques". He is currently a lecturer at the Department of Information and Communication Systems Engineering at the University of the Aegean with "Data Mining" as a field of expertiese. Furthermore, he is the Departmental Coordinator for the Programme: LLP/Erasmus within the University of the Aegean. Manolis Maragoudakis is a reviewer for "IEEE Transactions on Knowledge and Data Engineering", "Knowledge-Based Systems" and "International Journal of Artificial Intelligence Tools". He has actively supported a plethora of Artificial Intelligence and Data Mining conferences. Since 2001, is a member of the Hellenic Artificial Intelligence Society. His research interests focuses on the following thematic areas: Data Mining, Privacy Preserving Data Mining, Machine Learning, User Modeling, Semantic Web, Data Bases, Bayesian Networks.