J Ambient Intell Human Comput (2010) 1:133–145
DOI 10.1007/s12652-010-0012-4

ORIGINAL RESEARCH

Soft computing in intrusion detection: the state of the art

Chet Langin · Shahram Rahimi

Received: 21 September 2009 / Accepted: 6 April 2010 / Published online: 21 April 2010

Abstract The state of the art is explored in using soft computing (SC) methods for network intrusion detection, including the examination of efforts in ten specific areas of SC as well as consecutive, ensemble, and hybrid combinations. Numerous comparisons of these methods are listed followed by a recommendation for future research. This paper can be used as a reference of strategies, and as a resource for planning future research.

Keywords Artificial immune systems (AIS) · Artificial neural networks (ANN) · Bayes reasoning · Decision trees (DT) · Dempster–Shafer (D–S) · Ensemble combinations · Evolutionary computing (EC) · Feature selection · Fuzzy reasoning · Hidden Markov model (HMM) · Hybrid combinations · Intrusion detection systems (IDS) · Self-organizing maps (SOM) · Soft computing · State of the art · Swarm intelligence

1 Introduction

Heartland payment systems disclosed a network security breach on January 20, 2009, which exposed information on credit card transactions of potentially over 100 million credit cards, which not only placed a tremendous liability on Heartland, and not only impacted some 175,000 merchants and their banks involved in the financial transactions, but also jeopardized the credit and financial transactions of all of the customers involved (Acohido 2009).

The Heartland incident is just one of the latest in a plague of information security breaches, from major corporations and government agencies, to small businesses and homes.

Intrusion detection became a paramount issue when the Morris Worm of 1988 disrupted computer operations on a wide scale. Other notable widespread infections were Code Red in 2001 and the SQL Slammer in 2003. These infections caused many thousands of hours of lost time in the work force, plus thousands of more work hours in restoring computers and network activity. The total cost of patching and/or rebuilding computers, restoring and maintaining network activity, and providing information security, as well as the indirect cost of lost work time as the result of outages, since 1983 is easily in the billions of dollars.

An intrusion is often defined as anything that compromises the confidentiality, integrity, or availability (Abraham and Jain 2004) (CIA) of a resource, including insider abuse. Some researchers add the compromise of authentication and accountability to this list. Somayaji et al. (1997), studying artificial immunology, rolled it all up into one concept: the compromise of *survivability*. A network appliance which attempts to find intrusions is called an *intrusion Detection System*, or *IDS*. The most widely-used IDS appears to be the free Snort IDS (Kohlenberg et al. 2007), which uses rules to evaluate network packets. Snort was introduced in November, 1998, (Kohlenberg et al. 2007), as APE on Linux as a packet sniffing misuse detector. Snort became available at Packet Storm on December 22, 1998.

C. Langin (✉)
Information Technology, Southern Illinois University
Carbondale, Carbondale, USA
e-mail: clangin@siu.edu

S. Rahimi
Department of Computer Science, Southern Illinois University
Carbondale, Carbondale, USA
e-mail: rahimi@cs.siu.edu

Often the IDS rules look for malware signatures. Rule-based intrusion detection is also called *misuse detection* because the network security officer knows the symptoms of misuse and looks for this evidence. An obvious disadvantage of this is that the officer must first know what constitutes evidence of misuse. Another disadvantage is that experience indicates that in practice, for each verifiable intrusion, this type of IDS also typically produces many thousands of false positives, situations where the alerts are either false or else do not indicate enough evidence to take action.

Even if the resources needed to properly administer an IDS are provided, some newer malware is designed to deceive rule-based intrusion detection, anyway. In reference to current Internet background traffic, Pang et al. (2004) notes, "The gross features of this new breed of traffic are that it is complex in structure, highly automated, frequently malicious, potentially adversarial, and mutates at a rapid pace."

This introduction will continue by showing general time lines of the progression of soft computing (SC) and intrusion detection with subsections on SC components in Sect. 1.1, SC in Sect. 1.2, and intrusion detection in Sect. 1.3.

After this introduction, related works will be provided in Sect. 2 followed by related issues in Sect. 3. Alphabetical listings of SC methods are given in primary SC ID components in Sect. 4 and combined strategies in Sect. 5. The conclusion is in Sect. 6.

1.1 Soft computing components

The term *soft computing* came into existence formally in the early-1990s (Zadeh 1994b), but some SC components predate this time. Bayes probability, for example, was published in 1763 (Bayes 1763), and McCulloch's artificial neural network (ANN) was published in 1943 (McCulloch and Pitts 1943). Likewise, with fuzzy sets (Zadeh 1965), the first usage preceded the *soft computing* labeling.

Soft computing is called *soft* in order to contrast it with hard computing, i.e., exactness. Some characteristics of SC include probability, such as with Bayes reasoning; randomness, such as the initial randomness of nodes in ANN; inexactness, such as in fuzzy sets; and biological attributes, such as evolutionary computing. Soft computing has similarities with artificial intelligence (AI), but AI researchers have traditionally used hard computing.

An additional occasional characteristic of SC is emergence. Many refer to this SC characteristic as being a black box, described as a device where something goes in and something comes out, but what happens inside cannot be seen. SC is not really a black box—researchers write the code for SC software and know exactly what is inside the box. However, results emerge from this code in ways which often cannot readily be understood.

1.2 Soft computing

The first formal use of the term *soft computing* appears to be with the formation of the Berkeley initiative in SC (BISC) (Zadeh 1994b), which was conceived in October 1990, and instituted on 13 March 1991.

Zadeh (1994a) described SC as employing methods of reasoning that are approximate rather than exact. Zadeh noted, "The principal aim of SC is to achieve tractability, robustness, low solution cost, and high MIQ through the exploitation of the tolerance for imprecision and uncertainty." MIQ stands for Machine IQ. Zadeh said the principal constituents of SC are fuzzy logic (FL), ANN, and probabilistic reasoning (PR), with the latter subsuming belief networks, genetic algorithms, parts of learning theory, and chaotic systems."

Zadeh (1998b) summarized SC as consisting of fuzzy logic (FL), ANN, probabilistic reasoning (PR), and evolutionary computing (EC).

The term *soft computing* was in general use at least by 1996, for example at the NATO Advanced Study Institute (ASI) conference on "soft computing and its applications," Manavgat, Antalya, Turkey, August 21–31. Kaynak et al. (1998) described SC as having the characteristics of *approximation* and *dispositionality*. They said, "The tolerance for *imprecision* and *uncertainty* is exploited to achieve *tractability*, *lower cost*, high Machine IQ (MIQ), and *economy* of communication." They said SC includes fuzzy logic (FL), neurocomputing (NC), genetic computing (GC), and probabilistic reasoning (PR), as well as computing with words (CW) and information granulation.

Zadeh (1998a) further described SC as "Aimed at an accommodation with the pervasive imprecision of the real world", as well as to "exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness, low solution cost and better rapport with reality", and he said "the role model for SC is the human mind." He said types of SC include fuzzy logic (FL), neurocomputing (NC), genetic computing (GC) and probabilistic computing (PC) (chaotic systems, belief networks, and parts of learning theory). He said SC is really about hybrid systems, such as neuro-fuzzy, fuzzy genetic, neuro-genetic, and neuro-fuzzy-genetic.

Although every packet on the Internet is created for a specific reason, the security analyst cannot know this reason for each of the billions of packets created by every router, switch, appliance, program, and user that exists in the world that may be sent in the analyst's direction. This background Internet noise, from an analysts point of view, contains imprecision, uncertainty, and partial truths. The analyst also cannot examine every single packet, meaning that decisions must be made from incomplete information. SC thus provides coping mechanisms for tasks which are otherwise impossible.

1.3 Intrusion detection

Intrusion detection is the process of identifying and responding to malicious activity targeted at computing and networking sources (Amoroso (1999).

See Fig. 1 for an illustration of some kinds of intrusions. The top of the figure represents the Internet cloud from which network traffic passes through a firewall and an IDS into a local area network (LAN). The IDS is shown in line to signify that it could also be used as an intrusion prevention system (IPS). The exclamation points (!) near Part *a* of this figure show a coordinated attack coming from the Internet, such as could be part of a botnet. This type of attack could involve up to millions of packets from thousands of unique off-site IP addresses. The exclamation point near Part *b* represents a single malware packet which could get through to damage particular kinds of operating systems. The exclamation point near Part *c* represents an internal computer which could be part of a botnet or which could represent an insider attack.

Applied intrusion detection became more methodized when techniques for an intrusion detection system (IDS) were proposed in a paper by Denning (1986). One of the first intrusion detection systems developed in the late 1980s was called intrusion detection expert system (IDES) (Lunt 1990). IDES was an expert system which also included statistical analysis of user behaviors. While other IDS research was being done at this time, these works are considered by many to be the seminal publications in the field.

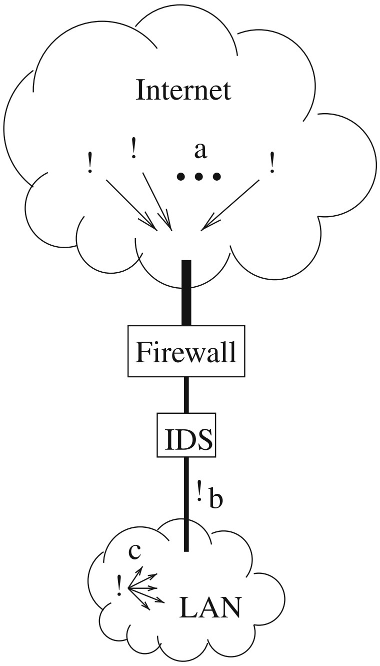

Fig. 1 Some types of intrusions

The concepts of misuse and anomaly detection were established at this time, although they were not labeled together as such until apparently 1994 (Kumar and Spafford 1994). Misuse detection was mentioned earlier in Sect. 1 in reference to a rule-based IDS such as Snort. An anomaly intrusion was defined as being based on anomalous behavior (Kumar and Spafford 1994). Extensions of these ideas based on statistics and/or expert systems form the basis of many modern IDSs. Statistics, as well as expert systems, is generally not considered to be SC for the purposes of this paper. Anomaly and misuse detection were considered by many authors for many years to be the only two major kinds of intrusion detection.

Ilgun et al. (1995) noted four types of intrusion detection: Threshold, anomaly, rule-based, and model-based. Anomaly detection is discussed above, and rule-based detection is the same as misuse detection, which is also discussed above. Threshold detection is simply summary statistics: for example, an alarm goes off because someone enters the wrong password ten times in a row. Model-based intrusion detection is defined in this paper as being a scenario model at a higher level of abstraction then audit records. Model-based detection can be used with other types of detection. For example, a model can be created describing steps required to acquire administrative access to a computer. Then, misuse detection can be used to see if anyone attempts to take these steps.

In reference to anomaly detection, Lee and Heinbuch (2001) noted, “These systems were confounded by two difficulties, one practical and the other theoretical. The practical difficulty is that nominal usage has high variability and changes over time. To meet this challenge, systems had a fairly loose threshold for tolerance of anomalous behavior, and were designed to learn new nominal statistics as they worked. This solution to the practical limitations of statistical anomaly detectors led to the theoretical difficulties: intruders could work below the threshold of tolerance and ‘teach’ the systems to recognize increasingly abnormal patterns as normal.” Newsome et al. (2006) explains how to thwart conjunction and Bayes learners.

In reference to misuse detection, Lee and Heinbuch (2001) noted, “Today, nearly all practical IDS are signature based. The performance of these systems is limited by the signature database they work from. Many known attacks can be easily modified to present many different signatures; if all variations are not in the database, even known attacks may be missed. Completely novel attacks cannot be present in the database, and will nearly always be missed.”

Formal proofs confirm the problem. Using misuse to mean all kinds of badness with a scale of 0 (normal) to 1 (misuse), Helman et al. (1992) showed that expert systems are NP-Hard. Me (1998) used a genetic algorithm to manipulate vectors based on event counts, and said the problem was NP Complete.

Cohen (1987) noted that the determination of a virus was undecidable: “In order to determine that a program ‘P’ is a virus, it must be determined that P infects other programs. This is undecidable since P could invoke the decision procedure ‘D’ and infect other programs if and only if D determines that P is not a virus.” He also noted that tracing exact information flow requires NP-Complete time. Cohen concluded that “this leaves us with imprecise techniques.”

2 Related works

Lunt (1993) wrote a survey of intrusion detection techniques at that time. Garcia and Copeland (2000) reported that SC tools were beginning to be used in intrusion detection, appearing to be the first time that the term *soft computing* was used in a paper about intrusion detection.

While not about intrusion detection, Bonissone (2000) gave a nice summary and history of SC. He broke SC into knowledge-driven reasoning systems such as probabilistic and fuzzy computing, and data-driven search and optimization approaches such as neuro and evolutionary computing. An excellent illustration was provided of hybrid systems. He described real-world problems appropriate for SC as being ill-defined, difficult to model, and possessing large solution spaces.

Biermann et al. (2001) compared IDSs with a table of approaches, and for each approach whether it was anomaly or misuse detection, the type of date analyzed, the amount of data, the origin of data, the accuracy, the completeness, and whether it detects these types of attacks: known, unknown, masquerade, DoS, malicious use, leakage, break in, and penetration of security. The approaches listed were statistical, predictive pattern, neural networks, sequence matching and learning, expert systems, keystroke monitoring, model-based, state transitional analysis, and pattern matching.

Noel et al. (2002) gave an IDS summary while emphasizing data mining techniques. Mukkamala et al. (2004a) presented architectures and perspectives on designing IDSs. Lazarevic et al. (2005) wrote a comprehensive 60-page survey of IDSs with 248 references. Zanero (2008) summarized, in 163 pages, his research on unsupervised learning algorithms for intrusion detection.

3 Related issues

A couple of miscellaneous issues are covered in this section which will aid in understanding later concepts: data sources and models.

3.1 Data sources

Although some IDSs analyze raw packets, most of the research IDSs preprocess data into vectors, and it is these data vectors which are analyzed in an effort to detect intrusions. A couple of noted data sources for vectors follow.

Lunt (1990) proposed over 50 vector elements, including user measures, target system measures, and remote host measures, taken from system calls, command names and arguments, and audit data. See that paper for the list of elements.

A data set was produced in 1998 by MIT’s Lincoln Laboratory under Defense Advanced Research Projects Agency (DARPA) sponsorship for the evaluations of IDSs which were submitted by six research groups (Lippmann et al. 2000b). This DARPA Data Set simulated traffic similar to a small Air Force base with hundreds of users (secretaries, programmers, workers, managers, system administrators, and attackers) on thousands of hosts. More than 300 instances of 38 different attacks were launched against UNIX victim hosts (Linux, SunOS, and Solaris) to produce 7 weeks of training data and 2 weeks of test data. There were four types of attacks: probe, Denial of Service (DoS), Remote to Local (R2L), and User to Root (U2R). A similar off-line intrusion detection evaluation was done the next year, in 1999 (Lippmann et al. 2000a), when Windows NT hosts were added. The tcpdump files of the 1998 DARPA Data Set were preprocessed into vectors of 41 features per network connection for use in the 1999 ACM International Knowledge Discovery and Data Mining Tools Competition (KDD’99) (Kayacik and Zincir-Heywood 2006). All of the variations of this data set are referred to in this paper collectively as the DARPA Data Set. Most of the tests and comparisons referred to in this paper were accomplished on this data.

3.2 Models

Lunt (1993) covered models fairly extensively, also referring to them as being scenarios which are specified in terms of user behavior. A model presents a hypothesis in which the IDS attempts to predict the next action until a conclusion is reached. She said advantages to this are that more data can be processed because the focus can be narrowed to the relevant data, and that more intuitive explanations of what is being detected can be generated because the events are related to the defined intrusion scenario.

Lippmann et al. (2000a) illustrated a model of possible paths of User to Root (U2R) attacks. Scott (2004) described using Bayesian logic for designing intrusion detection models, including modeling user commands, bursts of malicious network activity, and other network-level behavior.

Kayacik and Zincir-Heywood (2006) developed a visual self-organizing map (SOM) to build topological models of known attacks for forensic analysis. SOM was used also by Langin et al. (2008) in a model of detecting malignant network traffic which was later called a *Vulture Fest*.

4 Primary SC ID components

The state of the art includes many different variations of types, sub-types, and combinations with no single solution covering the entire realm of possible intrusions. Since each of these multiple research ideas can be covered only briefly, see the individual papers for detailed information.

This section describes ten general SC methods in alphabetical order with illustrative papers of how these methods have been implemented as IDSs.

See Sect. 5 for how these methods have been used in combination with each other for IDS. Figure 2 represents which of these individual components have been used together in other notable research. The thicker the line connecting the methods, the more often those methods have been used together.

An artificial immune system (AIS) IDS simulates the human immune system by creating a randomized set of *foreign* criteria. In production, if any of the input matches, or has a close match, to any of the criteria, then an alert is triggered. A human expert follows up on the alert to see if it is for an actual intrusion. Danger Theory considers the context of the environment in addition to differentiating between self and non-self.

Dasgupta and Nino (2009) gave a comprehensive introduction and update of AIS including an extensive section on security.

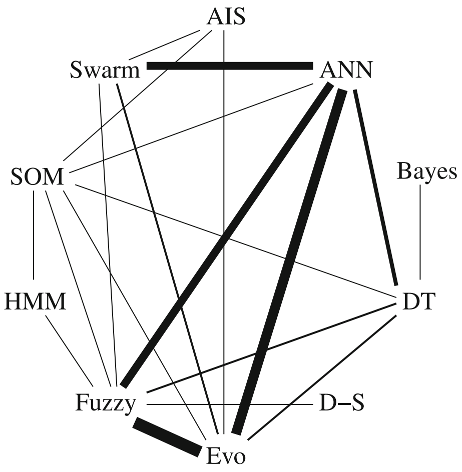

Fig. 2 Examples of methods that have been used together

Artificial neural network as used in this paper refers to a multi-layer perceptron. A SOM is technically also a type of ANN, but the data structure is different enough that SOMs in this paper are always labeled as such and are covered separately in another section. Some researchers refer to ANN as being a *connectionist* approach.

An ANN is typically a classifier in a graph structure which simulates biological neural networks. The knowledge engineer determines the structure of the ANN, which is then trained, usually with supervision. A human expert then decides if a follow up to an alert is appropriate to see if an intrusion has actually occurred.

Herrero et al. (2009) presented the MObile-Visualization Hybrid IDS (MOVIH-IDS) incorporating an ANN which could be trained without supervision and which could provide visual results of a network sniffer on a mobile device, such as a phone.

Bayes reasoning is a classifier which given the probability of an intrusion, and the probability of a particular alert based on that intrusion, can provide the posterior probability that the intrusion has occurred based on the alert going off. Also, given that any particular alarm type might indicate more than one kind of an intrusion, and that an intrusion type can trigger more than one kind of an alert, Bayes reasoning can provide the posterior probability that a particular type of intrusion has occurred based on which alerts have gone off.

The prior probabilities could be determined by a human expert or by a computational intelligence and stored in a table for later use by the IDS. However, since there are thousands of kinds of intrusions and symptoms, determining and updating these prior probabilities would require a lot of resources. Then, a human expert would decide which posterior probabilities should trigger alerts and which of these alerts should warrant follow ups to see if any intrusions had actually occurred.

Mahoney and Chan (2002) provided an excellent explanation of Bayes odds in Packet Header Anomaly Detection (PHAD) and Application Layer Anomaly Detection (ALAD). For example,

$$P(attack|x) = P(x|attack)P(attack)/P(x)$$

Newsome et al. (2006) explained how to thwart conjunction and Bayes learners with two types of malicious training: (1) red herring attacks to create false classifications; and (2) inseparability attacks to blur distinctions between classes.

Insider threats were investigated by ELICIT, which used a Bayesian Network to process 16 terabytes of raw packets over 13 months (Maloof and Stephens 2007). Sample indications of an insider threat are sensitive search terms, printing to a non-local printer, anomalous browsing activity, and retrieving documents outside of one's social network. ELICIT created a threat score based on these and other indications.

A decision tree (DT) is a classifier that uses a tree graph in order to provide the best match for the input along with the probability that the input falls into that class. A human expert determines what classes and probabilities trigger alerts, and what alerts warrant follow ups to see if any intrusions actually occurred.

Peddabachigari et al. (2007) presented an approach with DT and Support Vector Machines (SVM). Data from the DARPA Data Set passed through the DT to generate node information which was passed through the SVM. An ensemble was then made from DT, SVM, and DT–SVM.

Dempster–Shafer (D–S) theory of evidence was initiated by Dempster (1967) and refined by Shafer (1976). D–S is a classifier similar to probability, but uses the term *belief*, instead. D–S has two distinguishing characteristics over Bayes probability: (1) In addition to the belief of success or failure, D–S adds a third possibility of ignorance; and (2) beliefs from different sources, even differing beliefs, can be combined. The idea was explained in Wang et al. (2004) who used it to fuse sensor outputs. Another deployment of D–S could be to fuse the results of different methods in an ensemble. A disadvantage of D–S is that it can sometimes give a non-intuitive result. A human expert must decide what level of belief report warrants a follow up to see if an actual intrusion has occurred.

An opinion triangle illustrates a method based on D–S in Svensson and Josang (2001) which applies subjective logic to correlate multiple sensors to reduce alerts.

Network output from MARS, Sniffers, Snoop, and Wireshark was fused by D–S in the Unix environment by Sultan (2009), who reported that the detection rate went up by 20% while the false positive rate went down 51%.

Evolutionary computation (EC) in this paper includes Genetic Algorithms (GA), evolution strategies, evolutionary programming, and Genetic Programming (GP). The intrusion detection scenario as a whole is evolutionary: The attackers try a new method which the defenders attempt to block, resulting in the attackers developing yet another new method which the defenders attempt to block, resulting in a seemingly never ending cycle of attack and defense.

Lin and Wang (2008) proposed a new genetic clustering algorithm for intrusion detection which obtained the optimal number of clusters from the DARPA Data Set and had high performance rates.

Fuzzy reasoning is a classifier that helps to cope with inexact descriptions of intrusions. Network indications of a P2P botnet might be, for example, that a local computer has contacts with a large number of external IP addresses, the packet size entropy is high, a wide range of destination ports is used with many high-numbered ports, and the UDP ratio is high.

Degrees of attack guilt were outlined by Noel et al. (2002) which can be used for Fuzzy Inference: Absolute innocence, probable innocence, possible innocence, possible guilt, probable guilt, and provable guilt.

Fuzzy IDS (FIDS) was proposed by Tillapart et al. (2002) as a framework for network intrusions, including SYN and UDP floods, Ping of Death, E-mail Bomb, FTP and Telnet password guessing, and port scanning. Numerous example rules are provided in the paper based on network packet data.

Su et al. (2009) proposed a method of incremental mining so that fuzzy association rules can be implemented in a real-time network IDS. They used features from packet headers, comparing two rule sets, one mined online and the other mined from training data, rendering a decision every two seconds on large-scale attack types, such as Denial of Service (DoS).

A Hidden Markov model (HMM) is a partially hidden Markov Chain, named after Andrey Markov, 1856–1922. An HMM contains a graph created by a human expert and determines if the input follows the graph in a way that indicates a possible intrusion. If so, an alert is triggered and the probability of that intrusion is given. The expert then decides if this alert warrants a follow up to see if an actual intrusion occurred.

Markov chain models can be created of some kinds of network attacks. Here is a general example of some types of activities which may occur in a network attack: probing for vulnerabilities; exploiting a vulnerability; obtaining user access; obtaining administrative access; installing malicious software; fixing the vulnerability to keep other attackers out; setting up keylogging; using one computer to attack other computers; and, reporting back to the controller.

Not all of the above actions may be visible to an intrusion detection system, but the detection of some of them can indicate the probability of an attack.

Ourston et al. (2004) presented an excellent graphical explanation of HMM for network intrusions based on sensor alarms and compared HMM to DT and ANN, preferring HMM.

Zhengdao et al. (2008) analyzed data from the Sun Basic Security Module (BSM) with a host-based Hidden semi-Markov Model (HsMM). In a semi-Markov system, the probability of a state changing varies depending upon the amount of time spent in a state.

A SOM is a type of ANN, but its structure is different enough that SOMs are placed here in their own subsection of this paper. Both ANNs and SOMs have nodes (*neurons*), but in ANNs the nodes are connected in such a way that data is manipulated as it conceptually flows through the node structure resulting in an answer, whereas in SOMs the nodes represent clusters in space which are conceptually pulled like rubber over the data resulting in a visual display of the data.

SOMs can be used to gain insight by representing multidimensional data into a smaller dimensional space, like shining light on a three-dimensional object to create a two-dimensional shadow. Hexagonal SOM maps have been used often to display high dimensional data in a more human readable format.

SOMs primarily cluster the data, but they can also classify data by finding the most similar node, called the *best matching unit* (BMU). A visual display is usually made, but is not necessary for clustering and classification. Since SOMs are self-trained, the meanings of the resulting maps are not always obvious. Kayacik and Zincir-Heywood (2006) described a method of labeling nodes in a U-matrix hex map, which used a grey-scale scheme to show the relative distances between nodes in order to highlight clusters of nodes.

Langin et al. (2009) reported a self-trained SOM that was in production and that had discovered feral malware by using network firewall log entries.

Swarm Intelligence can be characterized by emergent behavior from swarming activity. Two kinds discussed in this paper are ant colony optimization (ACO), proposed by Colorni et al. (1991), and particle swarm optimization (PSO), proposed by Eberhart and Kennedy (1995) (the *particles* were based on bird flocking and fish schooling).

ACO imitates ant pheromone trails. Ramos and Abraham (2005) created ANTIDS (Ant IDS) which examined the DARPA Data Set and compared ANTIDS with DT, SVM, and LGP and giving four advantages of ants: (1) classification can be done in real time; (2) new classes can be handled without retraining; (3) learning can be either supervised or non-supervised; and, (4) the self-organizing nature makes it ideal for distributed IDS.

Tao et al. (2009) proposed a fish swarm algorithm for intrusion detection which was tested with the DARPA Data Set. Each *fish* goes towards the best better visible fish if it is not too crowded at that location, else it goes towards the center of the visible fish if this is a better and not too crowded, else it examines random locations to see if they are better, else it stays put.

5 Combined strategies

Table 1 shows the most common combinations of SC components which are mentioned in this paper. Combined strategies can be consecutive, parallel, or hybrid. A parallel strategy is typically called an ensemble, and should have a fusion method at the end to combine multiple results into a single output. Hybrid methods dynamically interact with each other in some way, such as looping back and forth multiple times or by having one method used internally by another method.

Table 1 Most common of representative soft computing intrusion detection combined strategies

Paper	DT	Fuzzy	Evo	ANN	Swarm
Dhanalakshmi and Babu (2008)		•	•		
Marin-Blazquez and Perez (2008)		•	•		
Chen et al. (2006)				•	•
Chen et al. (2006)			•	•	
Katar (2006)	•			•	
Chen et al. (2005b)				•	•
Chen et al. (2005b)			•	•	•
Abraham et al. (2004)	•	•	•	•	
Abraham and Jain (2004)	•	•	•	•	
Chavan et al. (2004)		•		•	
Shah et al. (2004)		•		•	
Copeland and Garcia (2001)		•	•	•	
Bridges and Vaughn (2000)		•	•		

Each of these types of combined strategies is covered in a subsection below, plus an additional subsection describes even more complex structures, such as a hybrid system which may be part of an ensemble.

5.1 Consecutive combinations

A consecutive combination uses methods in order, first one, and then the next. Figure 3 illustrates this by showing how Method 1 is used followed by Method 2.

Foukia et al. (2003) proposed a consecutive combination in an IDS with an immunology method for detection and artificial ants for response and simulated this on a 20-host network using Starlogo.

Fuzzy C-Means (FCM) clustering was followed by Dempster–Shafer for decision making in a proposal by Chou and Yen (2007) using the DARPA Data Set.

Dhanalakshmi and Babu (2008) created fuzzy sets from the input network packet data, defined membership functions for fuzzy variables, and then applied a genetic algorithm to identify the best rules.

An AIS was used by Powers and He (2008) for anomaly detection on the DARPA Data Set followed by a SOM for classification of the anomalous activity.

5.2 Ensemble combinations

An ensemble combination has methods which are run in parallel with an additional method at the end which provides a single output from multiple potential outputs. Figure 4 illustrates an ensemble.

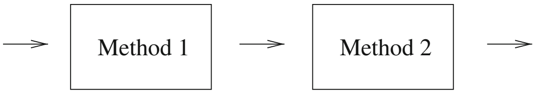

Fig. 3 A consecutive strategy

Mukkamala et al. (2001) suggested that a single output can be determined by averaging, voting, or by using the maximum value.

Mukkamala et al. (2003) tested ANNs on the DARPA Data Set with Resilient Back Propagation (RBP), Scaled Conjugate Gradient algorithm (SCG), and One-Step-Secant Algorithm (OSS) in an ensemble with a support vector machine (SVM). In order to fuse the results, let the four interim results be a_n, b_n, c_n, and d_n for each of the four parallel methods. Then, let x_n be the final result which is to be determined. Find x_n such that $|a_n - x_n| + |b_n - x_n| + |c_n - x_n| + |d_n - x_n|$ is minimized.

Abraham and Jain (2004) proposed an ensemble containing a DT, linear genetic programming (LGP), and fuzzy rules to analyze the DARPA Data Set. An ANN was also considered as part of the system.

An ensemble of a SOM for Anomaly Detection and a J48 DT for misuse detection was proposed by Depren et al. (2005) for the DARPA Data Set. The final output was positive if either of these two was positive.

Katar (2006) proposed an ensemble of Naive Bayes, ANN, and C4.5 DT on three separate sets of data input from the DARPA Data Set with the multiple fusion methods of Bayes Average, Bayes Product, Dempster–Shafer, Recognition, Substitution, and Rejection rates (RSR), the Predictive Rate Method (PRM), and Rogova's Class Level Method.

5.3 Hybrid combinations

A hybrid combination is an offspring of two different parents which implies an interaction of some sort as opposed to being consecutive or parallel. A hybrid strategy can loop back and forth multiple times between methods or can embed one method within another method. See Fig. 5 for an illustration of a loopy hybrid.

The first hybrid strategy appears to have been the dual usage of immunology and genetic algorithms (GA) in a system by Somayaji et al. (1997), who gives a nice summary of immunology. Lymphocytes include B-cells and T-cells which are selected at random as an ongoing process. They adapt (evolve) with GA when something foreign is found in order to identify it better. Some 10^9 lymphocytes look for potentially 10^{16} foreign patterns. The process is a *cell*, the computer an *organ*, and the network an *organism* with a variety of possible types of data inputs.

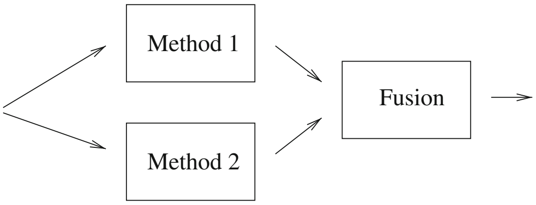

Fig. 4 An ensemble strategy

An Evolving Fuzzy Neural Network (EFuNN) IDS was tested by Chavan et al. (2004) on the DARPA Data Set.

Bridges and Vaughn (2000) discussed a prototype IDS using fuzzy data mining and genetic algorithms (GA) on network or audit data. GA fine-tuned the fuzzy sets by adjusting two parameters which defined functions in the fuzzy system: Where membership started and where membership was 1.

Chen et al. (2005b) proposed a flexible neural tree (FNT) using the DARPA Data Set in which the node structure was first optimized by genetic programming (GP) and then the node weights and function parameters were fine tuned with PSO. If this did not result in a satisfactory solution, then the GP and PSO processes were repeated.

Two hybrids were studied by Chen et al. (2006) using the DARPA Data Set. The first one was an ANN which was trained by the estimation of distribution algorithm (EDA), an evolutionary method, while the second one was an ANN which was trained by PSO.

Ensafi et al. (2008) proposed a hybrid swarm fuzzy K-means (SFK-means) approach with the DARPA Data Set in which each particle contained a constant number of clusters, resulting in a particle with the best fitness of clusters. The clusters in this particle were then used to classify anomalies when a distance was beyond a threshold.

An Enhanced Swarm Intelligence Clustering (ESIC) method was proposed by Feng et al. (2008) to choose the center of the radial basis function (RBF) in an RBF neural network (RBFNN) using the DARPA Data Set.

The classical Hidden Markov Model (HMM) was improved by Li et al. (2008) in the fuzzy HMM (FHMM) where fuzzy similarity measures replaced probabilistic measures using the DARPA Data Set.

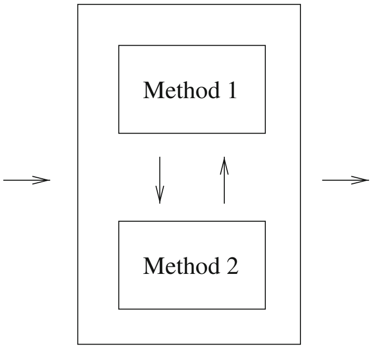

Fig. 5 A hybrid strategy

A quantum-behaved particle Swarm optimization (QPSO), combined with the Gradient Descent (GD) algorithm, has been used to train a radial basis function (RBF) ANN (Ma et al. 2008), also using the DARPA Data Set.

Marin-Blazquez and Perez (2008) applied hedged linguistic fuzzy classifiers to the XCS classifier system, which uses a genetic algorithm (GA), to allow for interpretation (using the DARPA Data Set).

An excellent description of evolutionary neural networks was given by Michailidis et al. (2008) who proposed an ANN trained with PSO. They noted that evolutionary algorithms (EA) can perform various ANN tasks, including weight training, architecture design, learning rule adaptation, input feature selection, and connection weight initialization. A couple of noted advantages of ANN over other methods, such as back propagation, are avoidance of trapping in local minima, and the fitness function does not have to be differentiable or continuous. They used the DARPA Data Set.

5.4 Multiple combinations

A combined strategy was explored by Copeland and Garcia (2001) starting with a SOM to group TCP flags, followed by a Learning Vector Quantizer (LVQ), a type of ANN, to characterize connection types, followed by a network handshake-watching fuzzy inference system (FIS) which was fine-tuned with a genetic algorithm (GA) with the end goal of detecting anomalies in network traffic.

Cho (2002) introduced a combined structure using host data from Sun's Basic Security Module (BSM) beginning with a SOM that fed multiple Hidden Markov Models (HMM), and ending with a decision made with fuzzy logic.

A light weight soft computing IDS (SCIDS) was proposed by Abraham et al. (2004) using the DARPA Data Set that started with a DT to reduce features that were fed into an ensemble of another DT, Linear Genetic Programs (LGP), and a fuzzy classifier. This was expanded by Abraham et al. (2007a) into a Distributed SCIDS (D-SCIDS).

6 Comparisons and conclusion

Many of the researchers performed direct comparisons between various SC methods and combinations of methods. A summary of these comparisons is made in the next subsection, followed by concluding remarks.

6.1 Comparisons

No solution is best for detecting all of the different attack types, and so many variables are involved that even a method with good accuracy may not be the best solution because it takes up too many resources. The intrusion detection environment is like an athletic track meet with many different events. An athlete, for example, who is good at the shot put may not do well in distance running; one who can high jump may not be good at the discuss throw; and, an excellent sprinter may not be able to handle the hurdles. An IDS, like a winning track team, needs many diversified skill sets.

The remainder of this subsection reports on comparisons of methods by other researchers.

The best performer of several ANN topologies consisted of two intermediary layers 126-20-5-1 with back propagation in research by (Bonifacio et al. 1998) which studied network traffic including sources, destinations, ports, and a security level assigned to services.

Ghosh et al. (1999a) tested equality matching (opposite of immunocomputing), a back propagation ANN, and a recurrent Elman Network on the DARPA Data Set, with Elman working the best. Ghosh and Schwartzbard (1999b) stated in a similar work that "applying recurrent, time delay neural networks to program-based anomaly detection has proved to be more successful than using back propagation networks for the same purpose."

Mukkamala et al. (2001) utilized host-based user activities to compare support vectors machines (SVM) with ANN and found several advantages to SVM, including better insight of the training process, a dynamic number of vectors, and faster training time. A disadvantage of SVM is that it only provides binary classifications. Weights were assigned to host-based user commands, such as 1 to exit and 10 to chown, and the average and highest weights for a user were determined. Web related data was also examined, such as the number of 404 errors.

Mukkamala et al. (2004a) performed more testing on the DARPA Data Set and added multivariate adaptive regression splines (MARS) and linear genetic programming (LGP) to the mix. SVM outperformed MARS and ANN in convergence and scalability, but LGP gave the overall best performance accuracy. In Mukkamala et al. (2004b) on the DARPA Data Set, resilient back propagation (RBP) outperformed other ANNs in terms of accuracy, although sometimes the ANNs did not properly converge. LGP outperformed the SVMs and RBP in accuracy with the expense of time. Chen et al. (2005a) using the DARPA Data Set also preferred SVM over ANN, noting that, in addition to other reasons, SVM had fewer parameters to set than ANN.

Ourston et al. (2004) reported a performance advantage of Hidden Markov Models (HMM) over DT and ANN based on network sensor alarms.

Shah et al. (2004) compared an evolving fuzzy neural network (EFuNN) with ANN using Snort for the training.

EFuNN was preferred because accuracy was comparable to that of ANN but training time was in seconds while the ANN training time was in minutes. Another advantage of EFuNN was its easy interpretability from using simple if-then rules.

Chen et al. (2005b) compared a hybrid FNT with PSO and evolutionary algorithm with a hybrid ANN-PSO using the DARPA Data Set. The FNT hybrid outperformed the ANN hybrid in accuracy in most categories.

ANTIDS, an ACO IDS analyzing the DARPA Data Set, performed well when compared to DT, support vector machines (SVM), and linear genetic programming (LGP) (Ramos and Abraham 2005). Four advantages of ACO were given: (1) Classification can be processed online and in real time; (2) new classes can be handled without retraining; (3) training can be either supervised on unsupervised; and, (4) this method is ideal for distributed IDS.

Chen et al. (2006) compared an estimation of distribution algorithm (EDA) ANN with a PSO ANN, and a DT using the DARPA Data Set. The EDA-ANN outperformed the others in accuracy in most categories followed by the PSO-ANN.

Naive Bayes outperformed Bayesian Networks in identifying Internet Relay Chat (IRC) botnet traffic (Livadas et al. 2006) on wireless network traffic. Wireless network information was examined such as the maximum initial congestion window, who initiated the flow, percentage of packets pushed in a flow, variance of packet inter-arrival time for flow, and variance of bytes-per-packet for flow.

While reporting on D-SCIDS using the DARPA Data Set, Abraham et al. (2007a) noted that LGP was ideal because it could be manipulated at the machine code level, but that fuzzy classification with rule generation based on partition of overlapping areas provided 100% accuracy on test data for all attack types when using 41 attributes.

Three different kinds of genetic programming were tested for IDS on the DARPA Data Set by Abraham et al. (2007b): Linear genetic programming (LGP), multi-expression programming (MEP), and gene expression programming (GEP). MEP outperformed LGP in three attack classes, while LGP outperformed MEP in two attack classes. GEP also obtained good results for all of the classes.

Four types of k-Nearest Neighbors (k-NN) were tested for intrusion detection on the DARPA Data Set: k-NN, fuzzy k-NN, evidence-theoretic k-NN, and fuzzy belief k-NN, with the last one performing best (Chou and Yen 2007). An ensemble approach was recommended to reduce the false positive rate.

Mukkamala et al. (2007) reported that LGP outperformed SVM and MARS with a 100% detection rate on the DARPA Data Set looking for stealthy probes.

A comparison between Bayesian Networks (BN) and C4.5 DT on the DARPA Data Set showed mixed results on accuracy depending upon the attack type, but also showed that using only 10 of 41 attributes saved training time, and also generally maintained or exceeded the accuracy of using all 41 attributes (Wang et al. 2008).

Yang et al. (2008) compared three types of cellular neural network (CNN), which can be viewed as a hybrid of cellular automaton and ANN: A Tabu Search (TS) CNN, a genetic algorithm (GA) CNN, and a simulated annealing (SA) CNN. The TS CNN performed best in terms of detection and false positive rates. This comparison used the DARPA Data Set.

7 Conclusion

A rule-based packet sniffing IDS is necessary but not sufficient in detecting the wide variety of intrusions which are possible. SC offers a plethora of impressive methods for intrusion detection with many variations and sub-variations. Comprehensive direct comparisons of the methods based on available research is not currently feasible because of the many different kinds of intrusions, criteria, and variables which were involved in setting up test environments. As general possible indications, reports indicated a swarm method better than an evolutionary method, and an evolutionary method was reported as outperforming SVM, which was reported as exceeding some ANN methods. Some ANN methods were reported as outperforming DT and AIS methods. However, not all of the authors used the same variations of methods. For example, many different types of ANN were evaluated. The statistical relevance of the results of these tests is also not known.

However, some results stand out. SVM excels over ANN for binary classifications. DT stand out for feature reduction. A self-trained SOM has been in production with positive detections of feral malware. Fuzzy classification with rule generation based on partition of overlapping areas provided 100 percent detection on test data, as did LGP on test data looking for stealthy probes. Ensemble combinations clearly stand out as winners because of their multifaceted approaches.

Future research along the lines of Fig. 6 is recommended, with a paradigm of a track meet with many parallel strategies needed against many different challenges, as opposed to a sharpshooting event with a *silver bullet*. Data can arrive from multiple sources, such as from firewall logs, packet sniffers, audit logs, and routers. This data can be preprocessed, and normalized, if appropriate, in different ways at once with some of the data being combined from different sources. Feature reduction can occur

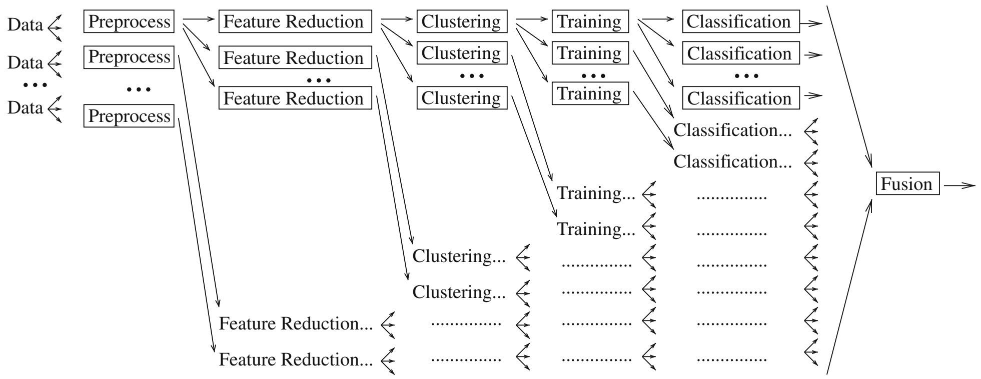

Fig. 6 Potential complexities of future intrusion detection strategies

by DT and other methods. Clustering can be done in parallel by evolutionary computing (EC), fuzzy reasoning, Swarm intelligence, SOM, and other methods. Training can be accomplished with many different techniques, as can classification with many different methods in parallel, and the output should be fused at the end. Any of these components can have hybridized subcomponents. Attack scenarios are constantly changing and the best defense is a diverse and adaptable system.

Zadeh (1998a) said, "SC is a consortium of methodologies: Synergistic and complementary—not competitive." This is a rich area for future intrusion detection research.

References

Abraham A, Jain R (2004) Soft computing models for network intrusion detection systems. http://arxiv.org/ftp/cs/papers/0405/0405046.pdf. Accessed 15 May 2008

Abraham A, Jain R, Sanyal S, Han SY (2004) Scids: a soft computing intrusion detection system. In: 6th international workshop on distributed computing (IWDC 2004). Springer, Berlin, pp 252–257

Abraham A, Jain R, Thomas J, Han SY (2007a) D-scids: distributed soft computing intrusion detection system. J Network Comput Appl 30:81–98

Abraham A, Grosan C, Martin-Vide C (2007b) Evolutionary design of intrusion detection programs. Int J Network Security 4(3):328–339

Acohido B (2009) Hackers breach heartland payment credit card system. http://www.usatoday.com/money/perfi/credit/2009-01-20-heartland-credit-card-security-breach_N.htm. Accessed 11 March 2009

Amoroso EG (1999) Intrusion detection: an introduction to internet surveillance, correlation, trace back, traps, and response. Intrusion.Net Books, NJ

Bayes T (1763) An essay towards solving a problem in the doctrine of chances. Philos Trans Roy Soc Lond 53:370–418

Biermann E, Cloete E, Venter LM (2001) A comparison of intrusion detection systems. Comput Security 20:676–683

Bonifacio JM, Cansian AM, de Carvalho ACPLF, Moreira ES (1998) Neural networks applied in intrusion detection system. In: The IEEE international joint conference, pp 205–210

Bonissone PP (2000) Hybrid soft computing systems: Where are we going?, http://www.cs.berkeley.edu/nikraves/bisc/Present/Fall0/Pieroecai2000v4.pdf (5/7/08)

Bridges SM, Vaughn RB (2000) Fuzzy data mining and genetic algorithms applied to intrusion detection. In: National information systems security conference, vol. 1. 16–19 October, pp 13–26

Chavan S, Shah K, Dave N, Mukherjee S (2004) Adaptive neuro-fuzzy intrusion detection systems. In: IEEE international conference on information technology: coding and computing (ITCC'04). IEEE Computer Society Press, Los Alamitos, CA, pp 70–74

Chen W-H, Hsu S-H, Shen H-P (2005a) Application of svm and ann for intrusion detection. Comput Oper Res 32(10):2617–2634

Chen Y, Abraham A, Yang J (2005b) Feature deduction and intrusion detection using flexible neural trees. In: Second IEEE International Symposium on Neural Networks (ISNN 2005)

Chen Y, Zhang Y, Abraham A (2006) Estimation of distribution algorithm for optimization of neural networks for intrusion detection system. In: Rutkowski L, Tadeusiewicz R, Zadeh LA, Zurada J (eds) Artificial intelligence and soft computing—ICAISC 2006. Springer, New York

Cho S-B (2002) Incorporating soft computing techniques into a probabilistic intrusion detection system. IEEE Trans Syst Man Cybernet 32(2):154

Chou T-S, Yen KK (2007) Fuzzy belief k-nearest neighbors anomaly detection of user to root and remote to local attacks. In: The 2007 IEEE workshop on information assurance, United States Military Academy, West Point, NY, pp 207–213

Cohen F (1987) Computer viruses: theory and experiments. Comput Security 6(1):22–35

Colorni A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: European conference on artificial life, Elsevier Publishing, Paris, France, pp 134–142

Copeland JA, Garcia RC (2001) Real-time anomaly detection using soft computing techniques. In: IEEE Southeast Conference 2001

Dasgupta D, Nino LF (2009) Immunological computation. CRC Press, Boca Raton

Dempster A (1967) Upper and lower probabilities induced by a multivalued mapping. Ann Math Stat 38(2):325–339

Denning DE (1986) An intrusion-detection model. IEEE Trans Software Eng 13(2):118–131

Depren O, Topallar M, Anarim E, Ciliz MK (2005) An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks. Exp Syst Appl 29(4):713–722

Dhanalakshmi Y, Ramesh Babu I (2008) Intrusion detection using data mining along fuzzy logic and genetic algorithms. Int J Comput Sci Security 8(2):27–32

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Sixth international symposium on micro machine and human science. IEEE Service Center, Piscataway

Ensafi R, Dehghanzadeh S, Mohammad R, Akbarzadeh T (2008) Optimizing fuzzy k-means for network anomaly detection using pso. In: ACS/IEEE international conference on computer systems and applications, Doha, Qatar

Feng Y, Wu Z-f, Zhong J, Ye C-x, Wu K-g (2008) An enhanced swarm intelligence clustering-based rbf neural network detection classifier. In: Fourth international conference on intelligent computing, Springer, Shanghai, China, pp 526–533

Foukia N, Hassas S, Fenet S, Albuquerque P (2003) Combining immune systems and social insect metaphors: a paradigm for distributed intrusion detection and response system. In: Mobile agents for telecommunications applications, 5th international workshop, MATA, Marrakech, Morocco

Garcia RC, Copeland JA (2000) Soft computing tools to detect and characterize anomalous network behavior. In: IEEE Southeast conference 2000

Ghosh AK, Schwartzbard A, Schatz M (1999) Learning program behavior profiles for intrusion detection. In: Workshop on intrusion detection and network monitoring, Santa Clara, CA, USENIX

Ghosh AK, Schwartzbard A (1999) A study in using neural networks for anomaly and misuse detection. In: Usenix security symposium, Washington, DC

Gunes Kayacik H, Nur Zincir-Heywood A (2006) Using self-organizing maps to build an attack map for forensic analysis. In: ACM international conference on privacy, security, and trust (PST 2006), pp 285–293

Helman P, Liepins G, Richards W (1992) Foundations of intrusion detection. In: The IEEE computer security foundations workshop V. IEEE Press, New York

Herrero A, Corchado E, Pellicer MA, Abraham A (2009) Movih-ids: a mobile-visualization hybrid intrusion detection system. Neurocomputing 72:2775–2784

Ilgun K, Kemmerer RA, Porras PA (1995) State transition analysis: a rule-based intrusion detection approach. IEEE Trans Software Eng 21(3):181–199

Katar C (2006) Combining multiple techniques for intrusion detection. Int J Comput Sci Network Security 6(2B):208–218

Kaynak O, Zadeh LA, Turksen B, Rudas IJ (1998) Computational Intelligence: soft computing and fuzzy-neuro integration with applications, volume 162 of series F: computers and systems sciences. Springer, New York

Kohlenberg T, Alder R Jr, Carter EF, (Skip), Foster JC, Jonkman M, Marty R, Poor M (2007) Snort IDS and IPS Toolkit. Open Source Security. Syngress

Kumar S, Spafford EH (1994) An application of pattern matching in intrusion detection. Technical report, Purdue University

Langin C, Zhou H, Gupta B, Rahimi S, Sayeh MR (2009) A self-organizing map and its modeling for discovering malignant network traffic. In: 2009 IEEE symposium on computational intelligence in Cyber Security, Nashville, TN, USA

Langin C, Zhou H, Rahimi S (2008) A model to use denied internet traffic to indirectly discover internal network security problems. In: The first IEEE international workshop on information and data assurance, Austin, Texas, USA

Lazarevic A, Kumar V, Srivastava J (2005) Intrusion detection: as urvey. In Kumar V, Srivastava, J, Lazarevic A (eds) Managing cyber threats, Springer, New York, pp 19–78

Lee SC, Heinbuch DV (2001) Training a neural-network based intrusion detector to recognize novel attacks. IEEE Trans Syst Man Cybernet A 31:294–299

Li Y, Ge Y, Jing X, Bo Z (2008) A new intrusion detection method based on fuzzy hmm. In: 3rd IEEE conference on industrial electronics and applications, Singapore

Lin C-C, Wang M-S (2008) Genetic-clustering algorithm for intrusion detection system. Int J Inform Comput Security 2(2):218–234

Lippmann R, Haines JW, Fried DJ, Korba J, Das K (2000a) Analysis and results of the 1999 darpa off-line intrusion detection evaluation. In: Debar H, Me L, Wu SF (eds) Recent advances in intrusion detection, third International Workshop (RAID). Springer, Toulouse, France, pp 162–182

Lippmann RP, Fried DJ, Graf i, Haines JW, Kendall KR, McClung D, Weber D, Webster SE, Wyschogrod D, Cunningham RK, Zissman MA (2000b) Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation. In: The 2000 DARPA information survivability conference and exposition (DISCEX), vol 2. IEEE Computer Society Press, Los Alamitos, CA, pp 12–26

Livadas C, Walsh B, Lapsley D, Strayer T (2006) Using machine learning techniques to identify botnet traffic. In: Second IEEE LCN workshop on network security (WNS), Tampa, FL, USA

Lunt TF (1990) Ides: an intelligent system for detecting intruders. In: Computer security, threat and countermeasures, Italy

Lunt TF (1993) A survey of intrusion detection techniques. Comput Security 12:405–418

Ma R, Liu Y, Lin X, Wang Z (2008) Network anomaly detection using rbf neural network with hybrid qpso. In: IEEE international conference on networking, sensing and control (ICNSC 2008), Sanya

Mahoney MV, Chan PK (2002) Learning nonstationary models of normal network traffic for detecting novel attacks. In: 8th ACM SIGKDD international conference on knowledge discovery and data mining. ACM Press, pp 376–385

Maloof MA, Stephens GD (2007) Elicit: a system for detecting insiders who violate need-to-know. In: Kruegel C, Lippmann R, Clark A (eds) Recent advances in intrusion detection. In: 10th international symposium, RAID 2007, volume 4637 of Lecture Notes in Computer Science, Springer, Gold Coast, Australia, pp 146–166

Marin-Blazquez J, Martinez Perez G (2008) Intrusion detection using a linguistic hedged fuzzy-xcs. Soft Comput Fusion Found Methodolog Appl 13(3):273–290

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115–133

Me L (1998) A genetic algorithm as an alternative tool for security audit trails analysis. In: Recent advances in intrusion detection (RAID'98)

Michailidis E, Katsikas SK, Georgopoulos E (2008) Intrusion detection using evolutionary neural networks. In: Panhellenic conference on informatics (PCI 2008), pp 8–12

Mukkamala S, Janoski G, Sung A (2001) Monitoring systsem security using neural networks and support vector machines. In: International workshop on hybrid intelligent systems, pp 121–138

Mukkamala S, Sung A, Abraham A (2004a) Designing intrusion detection systems: architectures and perspectives. In: The international engineering consortium (IEC) annual review of communications, vol 57, pp 1229–1241

Mukkamala S, Sung A, Abraham A (2007) Hybrid multi-agent framework for detection of stealthy probes. Appl Soft Comput J 7(3):631–641

Mukkamala S, Sung AH, Abraham A (2003) Intrusion detection using ensemble of soft computing paradigms. In: Third international conference on intelligent systems design and applications, advances in soft computing. Springer, New York, pp 239–248

Mukkamala S, Sung AH, Abraham A (2004b) Modeling intrusion detection systems using linear genetic programming approach. In: 17th international conference on industrial and engineering applications of artificial intelligence and expert systems, volume 3029 of Lecture Notes in Computer Science. Springer, New York, pp 633–642

Newsome J, Karp B, Song D (2006) Paragraph: thwarting signature learning by training maliciously. In: Zamboni D, Kruegel C (eds) Recent advances in intrusion detection, 9th international symposium, RAID 2006, volume 4219 of Lecture Notes in Computer Science. Springer, Hamburg, Germany, pp 81–105

Noel S, Wijesekera D, Youman C (2002) Modern intrusion detection, data mining, and degrees of attack guilt. In: Barbara D, Jajodia S (eds) Applications of data mining in computer security, advances in information security. Kluwer, Dordrecht

Ourston D, Matzner S, Stump W, Hopkins B (2004) Coordinated internet attacks: responding to attack complexity. J Comput Security 12:165–190

Pang R, Yegneswaran V, Barford P, Paxson V, Peterson L (2004) Characteristics of internet background radiation. In: Proceedings of ACM IMC, NY

Peddabachigari S, Abraham A, Grosan C, Thomas J (2007) Modeling intrusion detection system using hybrid intelligent systems. J Network Comput Appl 30(1):114–132

Powers ST, He J (2008) A hybrid artificial immune system and self organizing map for network intrusion detection. Inform Sci 178(15):3024–3042

Ramos V, Abraham A (2005) Antids: self organized ant-based clustering model for intrusion detection system. In: The Fourth IEEE international workshop on soft computing as transdisciplinary science and technology (WSTST'05), Springer, New York, pp 977–986

Scott SL (2004) A bayesian paradigm for designing intrusion detection systems. Comput Stat Data Anal 45(1):69–83

Shafer G (1976) A mathematical theory of evidence. Princeton University Press, Princeton

Shah K, Dave N, Chavan S, Mukherjee S, Abraham A, Sanyal S (2004) Adaptive neuro-fuzzy intrusion detection system. In: IEEE international conference on ITCC'04, vol 1. pp 70–74

Somayaji A, Hofmeyr S, Forrest S (1997) Principles of a computer immune system. New security paradigms workshop, Langdale, Cumbria, UK

Su M-Y, Yu G-J, Lin C-Y (2009) A real-time network intrusion detection system for large-scale attacks based on an incremental mining approach. Comput Security 75:301–309

Sultan Z (2009) Multiple simultaneous threat detection in unix environment. Int J Comput Sci Network Security 9(2):65–75

Svensson H, Josang A (2001) Correlation of intrusion alarms with subjective logic. In: The sixth nordic workshop on secure IT systems (NordSec 2001), Copenhagen, Denmark

Tao L, Yuan-bin H, Ai-ling Q, Xin-Tan C (2009) Feature optimization based on artificial fish-swarm algorithm in intrusion detection. In: 2009 international conference on networks, security, wireless communications and trusted computing, Hube, Wuhan, pp 542–545

Tillapart P, Thumthawatworn T, Santiprabhob P (2002) Fuzzy intrusion detection system. Assump Univ J Technol (AU J.T.) 6(2):109–114

Wang W, Gombault S, Guyet T (2008) Towards fast detecting intrusions: using key attributes of network traffic. In: The third international conference on internet monitoring and protection, IEEE Press, New York, pp 86–91

Wang Y, Yang H, Wang X, Zhang R (2004) Distributed intrusion detection system based on data fusion method. In: The 5th world congress on intelligent control and automation, IEEE, Hangzhou, PR China, pp 4331–4334

Yang Z, Karahoca A, Yang N, Aydin N (2008) Network intrusion detection by using cellular neural network with tabu search. In: Bio-inspired learning and intelligent systems for security, 2008. BLISS'08

Zadeh LA (1965) Fuzzy sets. Inform Control 9:338–353

Zadeh LA (1994a) Fuzzy logic, neural networks, and soft computing. Commun ACM 37(3):77–84

Zadeh LA (1994b) History; bisc during 90's, http://www-bisc.cs.berkeley.edu/BISCProgram/History.htm. Accessed 7July 2008

Zadeh LA (1998a) Roles of soft computing and fuzzy logic in the conception, design and deployment of information/intelligent systems. In: Kaynak O, Zadeh LA, Turksen B, Rudas IJ (eds) Computational intelligence: soft computing and fuzzy-neuro integration with applications, vol 162. Springer, New York

Zadeh LA (1998b) Some reflections on soft computing, granular computing and their roles in the conception, design and utilitzation of information/intelligent systems. Soft Comput Fusion Found Method Appl 2(1):23–25

Zanero S (2008) Unsupervised learning algorithms for intrusion detection. PhD thesis, Politecnico di Milano

Zhengdao Z, Zhumiao P, Zhiping Z (2008) The study of intrusion prediction based on hsmm. In: IEEE Asia-Pacific services computing conference (APSCC 2008). Yilan, Taiwan