Data Mining Techniques for (Network) Intrusion Detection Systems

Theodoros Lappas and Konstantinos Pelechrinis
Department of Computer Science and Engineering
UC Riverside, Riverside CA 92521
{*tlappas,kpele*}*@cs.ucr.edu*

***Abstract*—In Information Security, intrusion detection is the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource. Intrusion detection does not, in general, include prevention of intrusions. In this paper, we are mostly focused on data mining techniques that are being used for such purposes. We debate on the advantages and disadvantages of these techniques. Finally we present a new idea on how data mining can aid IDSs.**

General Terms
Security, Data mining
Keywords
Denial of Service, Data mining, IDS, Network security

I. INTRODUCTION

One of the main challenges in the security management of large-scale high-speed networks is the detection of suspicious anomalies in network traffic patterns due to Distributed Denial of Service (DDoS) attacks or worm propagation [1][2] A secure network must provide the following:

- *Data confidentiality:* Data that are being transferred through the network should be accessible only to those that have been properly authorized.
- *Data integrity:* Data should maintain their integrity from the moment they are transmitted to the moment they are actually received . No corruption or data loss is accepted either from random events or malicious activity.
- *Data availability:* The network should be resilient to Denial of Service attacks.

The first threat for a computer network system was realised in 1988 when 23-year old Robert Morris launched the first worm, which overid over 6000 PCs of the ARPANET network. On February 7th, 2000 the first DoS attacks of great volume where launched, targeting the computer systems of large companies like Yahoo!, eBay, Amazon, CNN, ZDnet and Dadet. More details on these attacks can be found at [3].

These threats and others that are likely to appear in the future have lead to the design and development of Intrusion Detection Systems. According to webopedia [4] an intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system.

The rest of this work is a survey of data mining techniques that have been applied to IDSs and is organized as follows: In section 2 we present a short taxonomy of IDSs. In section 3 we debate on the drawbacks of standard IDSs. Section 4 offers a brief introduction to data mining and section 5 illustrates how data mining can be used to enhance IDSs. In section 6 we talk about the various data mining techniques that have been employed in IDSs by various researchers. Section 7 presents existing IDSs (presented either in academia or in the market) that use data mining techniques. In secton 8, we give our own proposal on how data mining can be used to aid IDSs, while in section 9 we conclude our work.

II. IDS TAXONOMY

The goal of an IDS is to detect malicious traffic. In order to accomplish this, the IDS monitors all incoming and outgoing traffic. There are several approaches on the implementetion of an IDS. Among those, two are the most popular:

Anomaly detection: This technique is based on the detection of traffic anomalies. The deviation of the monitored traffic from the normal profile is measured . Various different implementations of this technique have been proposed, based on the metrics used for measuring traffic profile deviation.

Misuse/Signature detection: This technique looks for patterns and signatures of already known attacks in the network traffic. A constantly updated database is usually used to store the signatures of known attacks. The way this technique deals with intrusion detection resembles the way that anti-virus software operates.

A more extended taxonomy on IDS can be found in [5]. Figure 1 shows a taxonomy of Intrusion Detection Systems.

More details and information on the various IDS systems and the way they work can be found in [6][7][8][9].

III. DRAWBACKS OF IDSs

Intrusion Detection Systems (IDS) have become a standard component in security infrastructures as they allow network administrators to detect policy violations. These policy violations

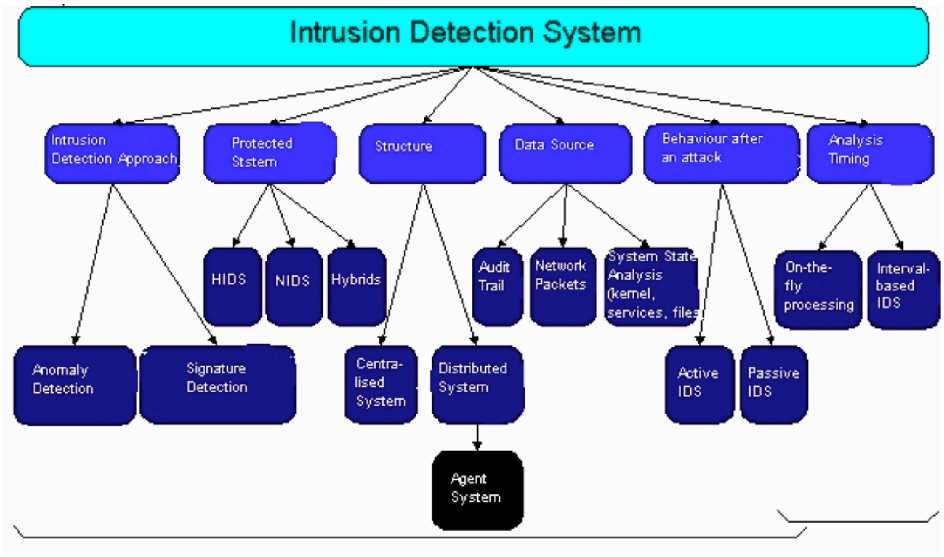

Fig. 1. An Intrusion Detection Systems' taxonomy.

range from external attackers trying to gain unauthorized access to insiders abusing their access.

Current IDS have a number of significant drawbacks:

- Current IDS are usually tuned to detect known service-level network attacks. This leaves them vulnerable to original and novel malicious attacks.
- ***Data overload:*** Another aspect which does not relate directly to misuse detection but is extremely important is how much data an analyst can efficiently analyze. That amount of data he needs to look at seems to be growing rapidly. Depending on the intrusion detection tools employed by a company and its size there is the possibility for logs to reach millions of records per day.
- ***False positives:*** A common complaint is the amount of false positives an IDS will generate. A false positive occurs when normal attack is mistakenly classified as malicious and treated accordingly.
- ***False negatives:*** This is the case where an IDS does not generate an alert when an intrusion is actually taking place. (Classification of malicious traffic as normal)

Data mining can help improve intrusion detection by addressing each and every one of the above mentioned problems.

IV. DATA MINING. WHAT IS IT?

Data mining (DM), also called **Knowledge-Discovery and Data Mining**, is the process of automatically searching large volumes of data for patterns using association rules [see fig 2]. It is a fairly recent topic in computer science but utilizes many older computational techniques from statistics, information retrieval, machine learning and pattern recognition.

Here are a few specific things that data mining might contribute to an intrusion detection project:

- Remove normal activity from alarm data to allow analysts to focus on real attacks
- Identify false alarm generators and "bad" sensor signatures
- Find anomalous activity that uncovers a real attack
- Identify long, ongoing patterns (different IP address, same activity)

To accomplish these tasks, data miners employ one or more of the following techniques:

- Data summarization with statistics, including finding outliers
- Visualization: presenting a graphical summary of the data
- Clustering of the data into natural categories
- Association rule discovery: defining normal activity and enabling the discovery of anomalies
- Classification: predicting the category to which a particular record belongs

V. DATA MINING AND IDS

Data mining techniques can be differentiated by their different model functions and representation, preference criterion, and algorithms [10].

The main function of the model that we are interested in is classification, as normal, or malicious, or as a particular type of attack [11][12].

We are also interested in link and sequence analysis [13][14][15]. Additionally, data mining systems provide the means to easily perform data summarization and visualization, aiding the security analyst in identifying areas of concern [16].

The models must be represented in some form. Common representations for data mining techniques include rules, decision trees, linear and non-linear functions (including neural nets), instance-based examples, and probability models [10].

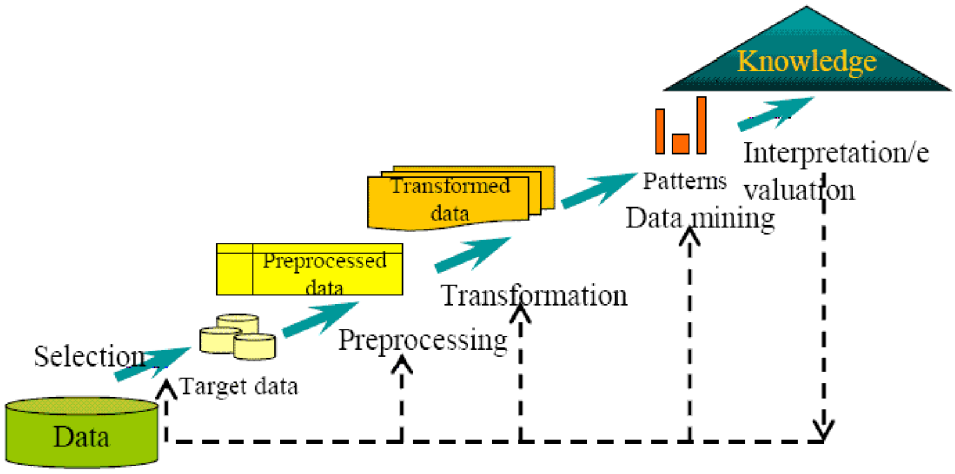

Fig. 2. The transition from raw data to valuable knowledge.

A. *Off Line Processing*

The use of data mining techniques in IDSs, usually implies analysis of the collected data in an offline environment. There are important advantages in performing intrusion detection in an offline environment, in addition to the real-time detection tasks typically employed.

Below we present the most important of these advantages:

- In off-line analysis, it is assumed that all connections have already finished and, therefore, we can compute all the features and check the detection rules one by one [14].
- The estimation and detection process is generally very demanding and, therefore, the problem cannot be addressed in an online environment because of the various the real-time constraints [17][16]. Many real-time IDSs will start to drop packets when flooded with data faster than they can process it.
- An offline environment provides the ability to transfer logs from remote sites to a central site for analysis during off-peak times.

B. *Data Mining and Real Time IDSs*

Even though offline processing has a number of significant advantages, data mining techniques can also be used to enhance IDSs in real time. Lee *et al.* [18] were one of the first to address important and challenging issues of accuracy, efficiency, and usability of real-time IDSs. They implemented feature extraction and construction algorithms for labeled audit data. They developed several anomaly detection algorithms. In the paper, the authors explore the use of information-theoretic measures, i.e., entropy, conditional entropy, relative entropy, information gain, and information cost to capture intrinsic characteristics of normal data and use such measures to guide the process of building and evaluating anomaly detection models. They also develop efficient approaches that use statistics on packet header values for network anomaly detection.

A real-time IDS, called "Judge", was also developed to test and evaluate the use of those techniques. A serious limitation of their approaches (as well as with most existing IDSs) is that they only do intrusion detection at the network or system level.

However, with the rapid growth of e-Commerce and e-Government applications, there is an urgent need to do intrusion and fraud detection at the application-level. This is because many attacks may focus on applications that have no effect on the underlying network or system activities.

C. *Multisensor Correlation*

The use of multiple sensors to collect data by various sources has been presented by numerous researchers as a way to increase the performance of an IDS.

- Lee *et al.*[18], state that using multiple sensors for ID should increase the accuracy of IDSs.
- Kumar [12] states that, "Correlation of information from different sources has allowed additional information to be inferred that may be difficult to obtain directly."
- Lee *et al.* note that, "an IDS should consist of multiple cooperative lightweight subsystems that each monitor a separate part (such as an access point) of the entire environment."
- Dickerson and Dickerson [19] also explore a possible implementation of such a mechanism. Their architecture consists of three layers:
  - A set of Data Collectors (packet collectors)
  - A set of Data Processors
  - A Threat analyzer that utilizes fuzzy logic and basically performs a risk assessment of the collected data.
- Honig *et al.* [20] propose a model similar to the one by Dickerson and Dickerson [19] and also has components

for feature extraction, model generation and distribution, data labeling, visualization, and forensic analysis.

- Helmer *et al.* [21] state that the use of a data warehouse facilitates the handling of the accumulated data and allows distributed attacks to be more easily detected, providing administrators with additional tools for doing auditing and forensics.

D. Evaluation Datasets

To test the effectiveness of data mining techniques in IDSs, the use of established and appropriate datasets is required. The DARPA datasets, available from the Lincoln Laboratory at MIT (http://www.ll.mit.edu/IST/ideval), are the most popular and widely used.

The Information Systems Technology Group (IST) of the MIT Lincoln Laboratory, has collected and distributed the first standard corpora for evaluation of computer network intrusion detection systems. They have also coordinated, along with the Air Force Research Laboratory, the first formal, repeatable, and statistically-significant evaluations of intrusion detection systems. Such evaluation efforts have been carried out in 1998 and 1999. These evaluations measure probability of detection and probability of false-alarm for each system under test.

These evaluations are contributing significantly to the intrusion detection research field by providing direction for research efforts and an objective calibration of the current technical state-of-the-art. The evaluation is designed to be simple, to focus on core technology issues, and to encourage the widest possible participation by eliminating security and privacy concerns, and by providing data types that are used commonly by the majority of intrusion detection systems.

VI. Survey of Applied Techniques

In this section we present a survey of data mining techniques that have been applied to IDSs by various research groups.

A. Feature Selection

*"**Feature selection**, also known as **subset selection** or **variable selection**, is a process commonly used in machine learning, wherein a subset of the features available from the data is selected for application of a learning algorithm. Feature selection is necessary either because it is computationally infeasible to use all available features, or because of problems of estimation when limited data samples (but a large number of features) are present." - Wikipedia*

Feature selection from the available data is vital to the effectiveness of the methods employed. Researchers apply various analysis procedures to the accumulated data, in order to select the set of features that they think maximizes the effectiveness of their data mining techniques. Table I contains some examples of the features selected. Each of these features offers a valuable piece of information to the System. Extracted features can be ranked with respect to their contribution and utilized accordingly.

% of same service to same host	# different services accessed
% on same host to same service	# establishment errors
average duration / all services	# FIN flags
average duration /current host	# ICMP packets
average duration / current service	# keys with outside hosts
bytes transfered / all services	# new keys
bytes transfered / current host	# other errors
bytes transfered / current service	# packets to all services
Destination bytes	# RST flags
Destination IP	# SYN flags
Destination port	# to certain services
Duplicate ACK rate	# to privileged services
Duration	# to the same host
Hole rate	# to the same service
Land packet	# to unprivileged services
Protocol	# total connections
Resent rate	# unique keys
Source bytes	# urgent
Source IP	% control packets
Source port	% data packets
TCP Flags	wrong data packet size rate
Timestamp	variance of packet count to keys

TABLE I
A TABLE OF FEATURES THAT HAVE BEEN EXTRACTED IN THE PROCESS OF APPLYING DATA MINING TECHNIQUES TO IDSS

B. Machine Learning

Machine Learning is the study of computer algorithms that improve automatically through experience. Applications range from data mining programs that discover general rules in large data sets, to information filtering systems that automatically learn users' interests. In contrast to statistical techniques, machine learning techniques are well suited to learning patterns with no a priori knowledge of what those patterns may be. Clustering and Classification are probably the two most popular machine learning problems. Techniques that address both of these problems have been applied to IDSs.

1) Classification Techniques: In a classification task in machine learning, the task is to take each instance of a dataset and assign it to a particular class. A classification based IDS attempts to classify all traffic as either normal or malicious. The challenge in this is to minimize the number of false positives (classification of normal traffic as malicious) and false negatives (classification of malicious traffic as normal).

Five general categories of techniques have been tried to perform classification for intrusion detection purposes:

*a) **Inductive Rule Generation**:* The RIPPER System is probably the most popular representative of this classification mechanism. RIPPER [22], is a rule learning program. RIPPER is fast and is known to generate concise rule sets. It is very stable and has shown to be consistently one of the best algorithms in past experiments [23]. The system is a set of association rules and frequent patterns than can be applied to the network traffic to classify it properly. One of the attractive features of this ap-

proach is that the generated rule set is easy to understand, hence a security analyst can verify it.

Another attractive property of this process is that multiple rule sets may be generated and used with a meta-classifier (Lee et Stolfo) [13] [15] [24] [14] [25].

Lee et Stolfo used the RIPPER system and proposed a framework that employs data mining techniques for intrusion detection. This framework consists of classification, association rules, and frequency episodes algorithms, that can be used to (automatically) construct detection models. They suggested that the association rules and frequent episodes algorithms can be effectively used to compute the consistent patterns from audit data.

Helmer *et al.* [21] duplicated Lee and Stolfo's approach and enhanced it by proposing the feature vector representation and verifying its correctness with additional experiments.

Warrender *et al.* [26] also used RIPPER to produce inductive rules and addressed issues that may arise if the mechanism was to be applied to an on-line system.

*b) **Genetic Algorithms**:* Genetic algorithms were originally introduced in the field of computational biology. Since then, they have been applied in various fields with promising results. Fairly recently, researchers have tried to integrate these algorithms with IDSs.

- The REGAL System [27][28] is a concept learning system based on a distributed genetic algorithm that learns First Order Logic multi-modal concept descriptions. REGAL uses a relational database to handle the learning examples that are represented as relational tuples.
- Dasgupta and Gonzalez [29] used a genetic algorithm, however they were examining host-based, not network-based IDSs. Instead of running the algorithm directly on the feature set, they used it only for the meta-learning step, on labeled vectors of statistical classifiers. Each of the statistical classifiers was a 2-bit binary encoding of the abnormality of a particular feature, ranging from normal to dangerous.
- Chittur [30] applied a genetic algorithm and used a decision tree to represent the data. They used the "Detection rate minus the false positive rate" as their preference criterion to distinguish among the data.
- Crosbie and Spafford [31] also used a genetic algorithm for sparse trees to detect anomalies. They attempted to minimize the occurrence of false positives by utilizing human input in a feedback loop.

*c) **Fuzzy Logic**:* **Fuzzy logic** is derived from fuzzy set theory dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic. It can be thought of as "the application side of fuzzy set theory dealing with well thought out real world expert values for a complex problem" [32].

In Dickerson and Dickerson 2000 [19] the authors classify the data based on various statistical metrics. They then create and apply fuzzy logic rules to these portions of data to classify them as normal or malicious. They found that the approach is particularly effective against scans and probes.

An enhancement of the fuzzy data mining approach has also been applied by Florez *et al.* [33] The authors use fuzzy data mining techniques to extract patterns that represent normal behavior for intrusion detection. They describe a variety of modifications that they have made to the data mining algorithms in order to improve accuracy and efficiency. They use sets of fuzzy association rules that are mined from network audit data as models of "normal behavior." To detect anomalous behavior, they generate fuzzy association rules from new audit data and compute the similarity with sets mined from "normal" data. If the similarity values are below a threshold value, an alarm is issued. They describe an algorithm for computing fuzzy association rules based on Borgelt's prefix trees, modifications to the computation of support and confidence of fuzzy rules, a new method for computing the similarity of two fuzzy rule sets, and feature selection and optimization with genetic algorithms. Experiments showed that their approach not only reduces the number of rules, but also increases the accuracy of the system.

Luo [34] also attempted classification of the data using Fuzzy logic rules. He demonstrated that the integration of fuzzy logic with association rules and frequency episodes generates more abstract and flexible patterns for anomaly detection. He also added a normalization step to the procedure for mining fuzzy association rules by Kuok, Fu, and Wong [35] in order to prevent one data instance from contributing more than others. He modified the procedure of Mannila and Toivonen [36] for mining frequency episodes to learn fuzzy frequency episodes. His approach utilizes fuzzy association rules and fuzzy frequency episodes to extract patterns for temporal statistical measurements at a higher level than the data level. Finally he presented the first real-time intrusion detection method that uses fuzzy episode rules.

*d) **Neural Networks**:* The application of neural networks for IDSs has been investigated by a number of researchers. Neural networks provide a solution to the problem of modeling the users' behavior in anomaly detection because they do not require any explicit user model. Neural networks for intrusion detection were first introduced as an alternative to statistical techniques in the IDES intrusion detection expert system to model [37]. In particular, the typical sequence of commands executed by each user is learned.

Numerous projects have used neural nets for intrusion detection using data from individual hosts, such as BSM data [11][38][39].

McHugh *et al* [40] have pointed out that advanced research issues on IDSs should involve the use of pattern recognition and learning by example approaches for the following two main reasons:

- The capability of learning by example allows the system to detect new types of intrusion.
- With earning by example approaches, attack "signatures" can be extracted automatically from labeled traffic data.

This basically eliminates the subjectivity and other problems introduced by the presence of the human factor.

A different approach to anomaly detection based on neural networks is proposed by Lee *et al.* While previous works have addressed the anomaly detection problem by analyzing the audit records produced by the operating system, in this approach, anomalies are detected by looking at the usage of network protocols.

Ghosh *et al.* [11] found that a "well trained, pure feed-forward, back propagation neural network" performed comparably to a basic signature matching system. The training set is made up of strings of events captured by the Base Security Module (BSM) that is part of many operating systems. If the training set is made up of strings related to normal behavior, neural networks act as an anomaly detector. On the other hand, if strings captured during an attack session are included in the training set, the network model can be modified to act as a misuse detection system.

Training sets made up of traffic data instead of audit records have also been used for misuse detection by Cannady [41] and Bonifacio [42]. Traffic features at different levels of abstraction have been used, from packet data to very high level features, such as the security level of the source and destination machines, occurrences of suspicious strings. For each neural network model, different numbers of output nodes have been selected according to the attack classification used. In some cases one network for each attack class has been used. A common characteristic shared by all the pattern recognition approaches reviewed is the use of a feature space made up of features related to information at different abstraction levels.

Didaci *et al* [43] present an ensemble variation of the neural networks approach. First, a decision is made by individual classifiers on the basis of partial information, e.g. intrinsic information, knowledge of past events, etc. Then the decisions are combined by means of suitable decision combination functions. Performances of ensemble approaches are usually higher than those of individual pattern classifiers.

*e) **Immunological based techniques**:* Hofmeyr and Forrest [44] present an interesting technique based on immunological concepts. They define the set of connections from normal traffic as the "self", then generate a large number of "non-self" examples: connections that are not part of the normal traffic on a machine. These examples are generated using a byte oriented hash and permutation. They can then compare incoming connections using the r-contiguous bits match rule. If a connection matches one of the examples, it is assumed to be in non-self and marked as anomalous.

Dasgupta and Gonzalez [45] used a similar approach. The authors generated a set of fuzzy rules using a genetic algorithm. They found that while this approach was not as accurate as a nearest neighbor match with the self-set, it was significantly more efficient.

Fan also used a similar approach in [46]. He found that injecting artificial anomalies into the dataset significantly increased detection of malicious anomalies, including those that had never been seen before.

*f) **Support Vector Machine**:* **Support vector machines (SVMs)** are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. SVMs attempt to separate data into multiple classes (two in the basic case) though the use of a hyper-plane.

Eskin *et al.* [47], and Honig *et al.* [20] used an SVM in addition to their clustering methods for unsupervised learning. The achieved performance was comparable to or better than both of their clustering methods.

Mukkamala, Sung, *et al.* [48][49] used a more conventional SVM approach. They used five SVMs, one to identify normal traffic, and one to identify each of the four types of malicious activity in the KDD Cup dataset. Every SVM performed with better than 99% accuracy, even using seven different variations of the feature set. As the best accuracy they could achieve with a neural network (with a much longer training time) was 87.07%, they concluded that SVMs are superior to neural nets in both accuracy and speed.

2) Clustering Techniques: **Data clustering** is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. Machine learning typically regards data clustering as a form of unsupervised learning. Clustering is useful in intrusion detection as malicious activity should cluster together, separating itself from non-malicious activity. Clustering provides some significant advantages over the classification techniques already discussed, in that it does not require the use of a labeled data set for training.

Frank [50] breaks clustering techniques into five areas: hierarchical, statistical, exemplar, distance, and conceptual clustering, each of which has different ways of determining cluster membership and representation.

Portnoy *et al* [51] present a method for detecting intrusions based on feature vectors collected from the network, without being given any information about classifications of these vectors. They designed a system that implemented this method, and it was able to detect a large number of intrusions while keeping the false positive rate reasonably low. There are two primary advantages of this system over signature based classifiers or learning algorithms that require labeled data in their training sets. The first is that no manual classification of training data needs to be done. The second is that we do not have to be aware of new types of intrusions in order for the system to be able to detect them. All that is required is that the data conform to several assumptions. The system tries to automatically determine which data instances fall into the normal class

and which ones are intrusions. Even though the detection rate of the system they implemented is not as high as of those using algorithms relying on labeled data, they claim it is still very useful. Since no prior classification is required on the training data, and no knowledge is needed about new attacks, the process of training and creating new cluster sets can be automated. In practice, this would mean periodically collecting raw data from the network, extracting feature values from it, and training on the resulting set of feature vectors. This will help detect new and yet unknown attacks.

Eskin *et al.* [47], and Chan *et al.* [52] have applied fixed-width and k-nearest neighbor clustering techniques to connection logs looking for outliers, which represent anomalies in the network traffic. Bloedorn *et al.* [16] use a similar approach utilizing k-means clustering.

Marin *et al.* [53] employed a hybrid approach that begins with the application of expert rules to reduce the dimensionality of the data, followed by an initial clustering of the data and subsequent refinement of the cluster locations using a competitive network called Learning Vector Quantization. Since Learning Vector Quantization is a nearest neighbor classifier, they classified a new record presented to the network that lies outside a specified distance as a masquerader. Thus, this system does not require anomalous records to be included in the training set. The authors were able to achieve classification rates, in some cases near 80% with misclassification rates less than 20%.

Staniford *et al.* [54] use "simulated annealing" to cluster events (anomalous packets) together, such that connections from coordinated port scans should cluster together. By using simulated annealing they reduce the run time from polynomial to linear.

Marchette [55] used clustering to project high dimensionality data into a lower dimensional space where it could be more easily modeled using a mixture model.

Sequeira and Zaki [56] also note the difficulty in determining the number of clusters in advance, and created the "Dynamic Clustering" method to cluster similar user activity together, creating the proper number of clusters as it proceeds.

Intrusion data are usually scarce and difficult to collect. Yeung *et al.*[57] propose to solve this problem using a novelty detection approach. In particular, they propose to take a nonparametric density estimation approach based on Parzen-window estimators with Gaussian kernels to build an intrusion detection system using normal data only. To facilitate comparison, they have tested their system on the KDD Cup 1999 dataset. Their system compares favorably with the KDD Cup winner which is based on an ensemble of decision trees with bagged boosting, as their system uses no intrusion data at all and much less normal data for training.

Leung and Leckie [58] propose a new approach in unsupervised anomaly detection in the application of network intrusion detection. This new algorithm, called "fpMAFIA", is a density-based and grid-based high dimensional clustering algorithm for large data sets. It has the advantage that it can produce clusters of any arbitrary shapes and cover over 95% of the data set with appropriate values of parameters. The authors provided a detailed complexity analysis and showed that it scales linearly with the number of records in the data set. They evaluated the accuracy of the new approach and showed that it achieves a reasonable detection rate while maintaining a low positive rate.

C. Statistical Techniques

Statistical techniques, also known as "top-down" learning, are employed when we have some idea as to the relationship were looking for and can employ mathematics to aid our search.

Three basic classes of statistical techniques are linear, non-linear (such as a regression-curve), and decision trees [59]. Statistics also includes more complicated techniques, such as Markov models and Bayes estimators. Statistical patterns can be calculated with respect to different time windows, such as day of the week, day of the month, month of the year, etc. [50], or on a per-host, or per-service basis [14][60].

Denning (1987) [61] described how to use statistical measures to detect anomalies, as well as some of the problems and their solutions in such an approach. The five statistical measures that she described were the operational model, the mean and standard deviation model, the multivariate model, the Markov process model, and the time series model.

Javitz and Valdes [62] provide more details on the individual statistical measures used in ID. They also provide formulas for calculating informative statistic metrics.

Staniford *et al.* [54] uses a similar approach by employing a Bayes network to calculate the conditional probabilities of various connection features with respect to other connection features. These probabilities are then used to determine how anomalous each connection is.

Mahoney and Chan [63] combined the output of five specific probability measures to determine how anomalous each connection was. In [63] they generate a set of rules for normal traffic where each rule retains the percentage of records in the training stream that support it. When a record is detected that violates a given rule, its anomaly score is the sum of each rules support value times the time since that rule was last violated.

Sinclair *et al.* [64] describe how they used Quinlan's ID3 algorithm to build a decision tree to classify network connection data. Bloedorn *et al.*[16] and Barbara *et al.* [65] also use decision tree-based methods.

1) Hidden Markov Models: Much work has been done or proposed involving Markovian models. For instance, the generalized Markov chain may improve the accuracy of detecting statistical anomalies. Unfortunately, it has been noted that these are complex and time consuming to construct [12], however their use may be more feasible in a high-power off-line environment.

A **hidden Markov model (HMM)** is a statistical model where the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine

the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis, for example for pattern recognition applications. A HMM can be considered as the simplest dynamic Bayesian network.

Ourston *et al.* [66] describe a novel approach using Hidden Markov Models (HMM) to detect complex Internet attacks. These attacks consist of several steps that may occur over an extended period of time. Within each step, specific actions may be interchangeable. A perpetrator may deliberately use a choice of actions within a step to mask the intrusion. In other cases, alternate action sequences may be random (due to noise) or because of lack of experience on the part of the perpetrator. For an intrusion detection system to be effective against complex Internet attacks, it must be capable of dealing with the ambiguities described above. The authors describe research results concerning the use of HMMs as a defense against complex Internet attacks. They describe why HMMs are particularly useful when there is an order to the actions constituting the attack (that is, for the case where one action must precede or follow another action in order to be effective). Because of this property, they show that HMMs are well suited to address the multi-step attack problem. In a direct comparison with two other classic techniques, decision trees and neural nets, the authors show that HMMs perform generally better than decision trees and substantially better than neural networks in detecting these complex intrusions.

Lane [67] trained a Hidden Markov Model on the same data that he used to train an instance-based learner. He notes that the Hidden Markov Model "assumes that the state variables are hidden and correspond to phenomena that are, perhaps, fundamentally unobservable," and as such, should perform well in modeling user actions. He concluded that the HMM and the instance-based learner mentioned above, trained using the same data, performed comparably.

Warrender *et al.* [26] applied a Hidden Markov Model to system call data. They noted that best performance was obtained by using a number of states corresponding to the number of system calls used by an application.

Even though HMMs have been shown to improve the accuracy of IDSs they are also accompanied by high complexity. They require a lot of resources and are often considered to be too time-consuming for practical purposes.

D. Other techniques

Numerous other data mining techniques, that do not really fit into any of the above-mentioned categories, have been applied to IDSs. In this section, we present the most important of these techniques.

1) NIDS with Random Forests: Many Network Intrusion Detection Systems are rule-based systems the performances of which highly depends on their rule sets. Unfortunately, due to the huge volume of network traffic, coding the rules by security experts becomes difficult and time-consuming. Since data mining techniques can build intrusion detection models adaptively, data mining-based NIDSs have significant advantages over rule-based NIDSs.

Jiong Zhang et Zulkernine [68] apply one of the efficient data mining algorithms called random forests for network intrusion detection. The NIDS can be employed to detect intrusions online. They discuss approaches for handling imbalanced intrusions, selecting features, and optimizing the parameters of random forests. To increase the detection rate of the minority intrusions, they build the balanced dataset by over-sampling the minority classes and down-sampling the majority classes. Random forests can build patterns more efficiently over the balanced dataset, which is much smaller than the original one. Experiments have shown that the approach can reduce the time to build patterns dramatically and increase the detection rate of the minority intrusions. The authors report experimental results over the KDD99 datasets. The results show that the proposed approach provides better performance compared to the best results from the KDD99 contest.

2) Exploiting the infrastracture: V. Mittal and G. Vigna [69] presented a novel approach to detect attacks against the routing infrastructure. The approach uses a set of sensors that analyze routing traffic in different locations within a network. An algorithm to automatically generate both the detection signatures and the inter-sensor messages needed to verify the state of the routing infrastructure has been devised for the case of the RIP distance-vector routing protocol. The approach described here has a number of advantages.

First, the implementation of their approach does not require any modification to routers and routing protocols. Most current approaches require routers and routing protocols be changed. The high cost of replacing routers and the risk of large-scale service disruption due to possible routing protocol incompatibility has resulted in some inertia in the deployment of these approaches. This approach, on the other hand, is deployable and provides a preliminary solution to detecting attacks against the routing infrastructure.

Second, the detection process does not use the computational resources of routers. There might be additional load on the routers from having to forward the traffic generated by sensors. However, this additional load should not be as much as it would be if a router had to perform public-key decryption of every routing update that it received, which is what most current schemes require.

Third, the approach supports the automatic generation of intrusion detection signatures, which is a human-intensive and error-prone task.

3) Other approaches: Numerous other techniques have been suggested that could be used to mine security information from network connection logs. Here we will look at some techniques that have been successfully applied for intrusion detection.

A technique that has been successfully applied to misuse detection systems is colored Petri nets [12]. In IDIOT, the colored Petri nets were created by hand.

Bass [70], suggested that the Dempster-Shafer Method, or Generalized EPT Method may be useful as combinatorial methods in a system employing multiple approaches, or in fusing the data from multiple sources.

Lane [67] identified numerous techniques from the signal processing and pattern recognition communities such as spectral analysis, principle component analysis, linear regression, linear predictive coding, self-similarity, neural networks, and nearest-neighbor matching.

Based on the observation that an intrusion scenario might be represented as a planning activity, Cuppens et al [71] suggest a model to recognize intrusion scenarios and malicious intentions. This model does not follow previous proposals that require to explicitly specify a library of intrusion scenarios. Instead, their approach is based on specification of elementary attacks and intrusion objectives. They then show how to derive correlation relations between two attacks or between an attack and an intrusion objective. Detection of complex intrusion scenario is obtained by combining these binary correlation relations. They also suggest using abduction to recognize intrusion scenarios when some steps in these scenarios are not detected. They then define the notion of anti correlation that is useful to recognize a sequence of correlated attacks that does no longer enable the intruder to achieve an intrusion objective. This may be used to eliminate a category of false positives that correspond to false attacks.

E. Ensemble Approaches

"In reality there are many different types of intrusions, and different detectors are needed to detect them." [Axelsson]

One way to improve certain properties, such as accuracy, of a data mining system is to use a multiplicity of techniques and correlate the results together. The combined use of numerous data mining methods is known as an ensemble approach, and the process of learning the correlation between these ensemble techniques is known by names such as multistrategy learning, or meta-learning.

Lee and Stolfo [14][15][24] state that if one method or technique fails to detect an attack, then another should detect it. They propose the use of a mechanism that consists of multiple classifiers, in order to improve the effectiveness of the IDS.

Axelsson [72][73] proposes the modeling and analysis of both possible types of traffic (normal and malicious). Having results from both patterns can help improve the overall performance of the system.

Lee [15][24] proposes that combining multiple models leads to the synthesis of a much more flexible system, one that has the ability to adapt to new demands and face new challenges because of its diverse nature.

Researchers at the Columbia University IDS lab have applied meta-classification both to improve accuracy and efficiency, as well as to make data mining based IDS systems more adaptable.

Lee and Stolfo [14] used meta-classification to improve both accuracy and efficiency (by running high cost classifiers only when necessary), and combined the results using boolean logic. The classifiers are produced using cost factors that quantify the expense of examining any particular feature in terms of processing time, versus the cost of responding to an alert or missing an intrusion [74][46].

Didaci *et al.* [43] applied a meta-classification approach. The authors applied three different classification methods - the majority voting rule, the average rule, and the "belief" function - to the outputs of three distinct neural nets. The Neural nets had previously been trained on different features sets from the KDD tcpdump data. They found that these multistrategy techniques, particularly the belief function, performed better than all three neural nets individually.

Crosbie and Spafford [31] use an ensemble of "autonomous agents" to determine the threat level presented by network activity.

F. Predictive Analysis

Ideally, a data-mining based IDS will do more than just detect intrusions that have already happened: we would like it to provide some degree of predictive analysis.

Lee [15][24] notes in that "a typical attack session can be split into three phases: a learning phase, a standard attack phase, and an innovative attack phase." Given that, we should be able to predict standard and innovative attacks to some degree based on prior activity.

Another area that predictive analysis may be useful is in early detection of worms. Typically, retrospective analysis of worms such as Code Red have shown similar activity of the worms a number of weeks before its widespread outbreak. Additionally, statistical trending should be able to detect the start of a worm's characteristic exponential curve before the infection rate begins increasing steeply, at least for traditional worms such as Code Red or Nimda. Unfortunately, fast infection rate worms, such as the SQL Slammer worm, will most likely have completed their exponential growth before the connection data can be fused and mined.

VII. Existing Systems

In this section, we present some of the implemented systems that apply data mining techniques in the field of Intrusion Detection.

- **ISOA (Information Security Officer's Assistant) [75]:** ISOA is a system for monitoring security relevant behavior in computer networks. ISOA serves as the central point for real-time collection and analysis of audit information. When an anomalous situation is identified, associated indicators are triggered. ISOA automates analysis of audit trails, allowing indications and warnings of security threats to be generated in a timely manner so that threats can be countered. ISOA allows a single designated workstation to perform automated security monitoring, analysis and warning

- **Distributed Intrusion Detection System (DIDS) [76]:** A risk intrusion detection system that aggregates audit reports from a collection of hosts on a single network. Unique to DIDS is its ability to track a user as he establishes connections across the network.
- **EMERALD (SRI) [77]:** EMERALD is a software-based solution that utilizes lightweight sensors distributed over a network or series of networks for real-time detection of anomalous or suspicious activity. EMERALD sensors monitor activity both on host servers and network traffic streams, and empower system defenders with the capacity to detect and ultimately thwart cyber attacks across large networks By using highly distributed surveillance and response monitors, EMERALD provides a wide range of information security coverage, real-time monitoring and response, protection of informational assets. EMERALD implements an enterprise-wide analysis to correlate the activity reports produced across asset of monitored domains. EMERALD offers protection from network-wide threats such as Internet worm-like attacks, attacks repeated against common network services across domains, or coordinated attacks from multiple domains against a single domain.
- **The MINDS System [78]:** The Minnesota Intrusion Detection System (MINDS), uses data mining techniques to automatically detect attacks against computer networks and systems. While the long-term objective of MINDS is to address all aspects of intrusion detection, the system currently focuses on two specific issues:
  - An unsupervised anomaly detection technique that assigns a score to each network connection that reflects how anomalous the connection is, and
  - An association pattern analysis that summarizes those network connections that are ranked highly anomalous by the anomaly detection module.

  Experimental results on live network traffic at the University of Minnesota show that the applied anomaly detection techniques are very promising and are successful in automatically detecting several novel intrusions that could not be identified using popular signature-based tools such as SNORT. Furthermore, given the very high volume of connections observed per unit time, association pattern based summarization of novel attacks is quite useful in enabling a security analyst to understand and characterize emerging threats.
- **The IDDM project [79]:** The IDDM (Intrusion Detection using Data Mining) project is a project that uses data mining techniques in order to describe the data on a network and analyze them for further deviation in observed traffic. IDDM utilizes meta-mining to achieve its goals. The goal is to track and understand changes in the network traffic over time. IDDM uses association rules in order to observe network traffic. Two different snapshots of the association rules -created in two different timestamps- are compared in order to see which rules have remained the same, have been changed, been added and which have been eliminated.

  The system uses agents that apply association rule mining on raw network packets. Attributes that are taken into consideration are :
  - Packet type (protocol)
  - Source/Destination Port
  - Packet Size
  - TCP flags

  Analysis on a stable network should produce the following results between the 2 snapshots:
  - A small number of added/deleted rules
  - A Fairly large number of unchanged rules
  - A Small to medium number of changed rules

  Two more systems that use the basic meta-mining concept behind IDDM can be found at :
  - http://www1.cs.columbia.edu/jam/itoprojsubmitted99.html - The MADAM ID System is part of the larger JAM project. It is held at the department of Computer Science at the University of Columbia and is led by Prof. Salvatolre Stolfo [23].
  - In Iowa State University researchers Helmer *et al* [80] use agents in order to collect low lever information and correlate them at a higher level.
- **IDSs in the Open Market:** Various systems that employ data mining techniques have already been released as parts of commercial security packages. Some of the most popular of these systems are:
  - RealSecure SiteProtector [81]
  - Symantec ManHunt [82]
  - nSecure nPatrol,
  - Dshield [83]
  - MyNetWatchman [84]

VIII. An Idea of Our Own

In this section we propose a data mining technique that could potentially prove to be beneficial to IDSs. The idea is to use biclustering as a tool to analyze network traffic and enhance IDSs. Bi-clustering is the problem of finding a partition of the vectors and a subset of the dimensions such that the projections along those directions of the vectors in each cluster are close to one another. The problem requires the clustering of the vectors and the dimensions simultaneously. The clusters produced by this process are called *biclusters*. Biclustering measures the similarity across a subset of the experiments, when the testing conditions are heterogeneous. Biclusters may overlap, revealing the role of features in multiple objects and the relations between different objects. The easiest way to approach the problem is by representing the data in a matrix form. Each row represents an object (e.g a traffic trace generated by a process/user) and each column represents a feature (e.g. the Destination Port). Biclustering is now reduced to the problem of finding a subset

	Feature A	Feature B	Feature C	Feature D	Feature E
Process 1	A1	B2	C3	D2	E2
Process 2	A3	B3	C2	D1	E1
Process 3	A1	B2	C3	D2	E2
Process 4	A1	B2	C3	D3	E2
Process 5	A1	B2	C1	D2	E2
Process 6	A1	B2	C2	D2	E2
Process 7	A3	B1	C1	D1	E1

TABLE II
BICLUSTERING TECHNIQUE

of the rows and a subset of the columns such that the submatrix induced has the property that each row reads the same string.

Example:

In Table II the rows represents processes (or more accurately the traces they produced) and the columns represent selected features a process trace can have. For simplicity, we consider that each feature has 3 possible discrete values (e.g. feature A can only take values from the set [A1, A2, A3])

By applying Biclustering to the above matrix we find the following 2 clusters:

- {(Process 1, Process 3, Process 4) (A, B, C, E)}
- {(Process 1, Process 5, Process 6) (B, D, E) }

The first cluster shows that Process 1, 4 and 5 always have the same values for features A, B, C and E. The second cluster shows that Process 1, 3 and 4 always have the same values for features B, D and E.

If we know in advance that the processes in the matrix are malicious, these process can give us the characteristic feature set for malicious traces. The set can be then used to classify new data collected from the network. Even if know nothing about the processes, this process will not only cluster them, but also show cluster their feature sets. The obtained biclusters could be an effective way to summarize and separate similar processes and analyze them as a group.

In general, biclustering can provide valuable knowledge on the relationships between processes and features. More information on biclustering can be found in [85][86][87].

IX. CONCLUSIONS

This paper has presented a survey of the various data mining techniques that have been proposed towards the enhancement of IDSs. We have shown the ways in which data mining has been known to aid the process of Intrusion Detection and the ways in which the various techniques have been applied and evaluated by researchers. Finally, in the last section, we proposed a data mining approach that we feel can contribute significantly in the attempt to create better and more effective Intrusion Detection Systems.

REFERENCES

[1] Christos Douligeris, Aikaterini Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art" ,*Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 44, Issue 5 , pp: 643 - 666, 2004.*

[2] Z. Chen, L. Gao, K. Kwiat, Modeling the spread of active worms,*Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 3, pp. 1890 1900, 2003.*

[3] http://www.securityfocus.com/news/2445

[4] http://www.webopedia.com

[5] Mithcell Rowton, Introduction to Network SecurityIntrusion Detection,*December 2005.*

[6] Biswanath Mukherjee, L.Todd Heberlein, Karl N.Levitt, "Network Intrusion Detection",*IEEE, June 1994.*

[7] Presentation on Intrusion Detection Systems, Arian Mavriqi.

[8] Intrusion Detection Methodologies Demystified, *Enterasys Networks TM.*

[9] Protocol Analysis VS Pattern matching in Network and Host IDS, 3^{rd} *Generation Intrusion Detection Technology from Network ICE.*

[10] Fayyad, U. M., G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Communications of the ACM 39 (11),November 1996, 2734.*

[11] Ghosh, A. K., A. Schwartzbard, and M. Schatz,"Learning program behavior profiles for intrusion detection", *In Proc. 1st USENIX, 9-12 April, 1999*

[12] Kumar, S., "Classification and Detection of Computer Intrusion", *PhD. thesis, 1995, Purdue Univ., West Lafayette, IN.*

[13] Lee, W. and S. J. Stolfo, "Data mining approaches for intrusion detection", *In Proc. of the 7th USENIX Security Symp., San Antonio, TX. USENIX, 1998.*

[14] W. Lee, S.J.Stolfo *et al*, "A data mining and CIDF based approach for detecting novel and distributed intrusions", *Proc. of Third International Workshop on Recent Advancesin Intrusion Detection (RAID 2000), Toulouse, France.*

[15] Lee, W., S. J. Stolfo, and K. W. Mok, "A data mining framework for building intrusion detection models," *In Proc. of the 1999 IEEE Symp. on Security and Privacy, Oakland, CA, pp. 120132. IEEE Computer Society Press, 9-12 May 1999*

[16] Eric Bloedorn *et al*, "Data Mining for Network Intrusion Detection: How to Get Started," *Technical paper, 2001.*

[17] Singh, S. and S. Kandula, "Argus - a distributed network-intrusion detection system," *Undergraduate Thesis, Indian Institute of Technology, May 2001.*

[18] Lee, W. and D. Xiang, "Information-theoretic measures for anomaly detection", *In Proc. of the 2001 IEEE Symp. on Security and Privacy, Oakland, CA, pp. 130143. IEEE Computer Society Press, May 2001*

[19] Dickerson, J. E. and J. A. Dickerson, "Fuzzy network profiling for intrusion detection", *In Proc. of NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society, Atlanta, pp. 301306. North American Fuzzy Information Processing Society (NAFIPS), July 2000.*

[20] Honig, A., A. Howard, E. Eskin, and S. J. Stolfo, "Adaptive model generation: An architecture for the deployment of data mining based intrusion detection systems", *In D. Barbar and S. Jajodia (Eds.), Data Mining for Security Applications. Boston: Kluwer Academic Publishers, May 2002.*

[21] Helmer, G., J. Wong, V. Honavar, and L. Miller, "Automated discovery of concise predictive rules for intrusion detection", *Technical Report 99-01, Iowa State Univ., Ames, IA, January, 1999.*

[22] Cohen, W. W., "Fast effective rule induction", *In A. Prieditis and S. Russell (Eds.), Proc. of the 12th International Conference on Machine Learning, Tahoe City, CA, pp. 115123. Morgan Kaufmann, 9-12 July, 1995.*

[23] S. Stolfo, A. L. Prodromidis and P. K. Chan, "JAM: Java Agents for Meta-Learning over Distributed Databases", *in Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, AAAI Press, Menlo Park, 1997.*

[24] Lee, W., S. J. Stolfo, and K. W. Mok, " Mining in a data-flow environment: Experience in network intrusion detection," *In S. Chaudhuri and D. Madigan (Eds.), Proc. of the Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99), San Diego, CA, pp. 114124. ACM, 12-15 August 1999.*

[25] Lee, W., S. J. Stolfo, and K. W. Mok, "Adaptive intrusion detection: A data mining approach," *Artificial Intelligence Review 14 (6), 533567, 2000.*

[26] Warrender, C., S. Forrest, and B. A. Pearlmutter, "Detecting intrusions using system calls: Alternative data models", *In Proc. of the 1999 IEEE Symp. on Security and Privacy, Oakland, CA, pp. 133145. IEEE Computer Society Press, 1999.*

[27] Neri, F., "Comparing local search with respect to genetic evolution to detect intrusion in computer networks", *In Proc. of the 2000 Congress on*

Evolutionary Computation CEC00, La Jolla, CA, pp. 238243. IEEE Press, 16-19 July, 2000.

[28] Neri, F., "Mining TCP/IP traffic for network intrusion detection", *In R. L. de M'antaras and E. Plaza (Eds.), Proc. of Machine Learning: ECML 2000, 11th European Conference on Machine Learning, Volume 1810 of Lecture Notes in Computer Science, Barcelona, Spain, pp. 313322. Springer, May 31- June 2, 2000.*

[29] Dasgupta, D. and F. A. Gonzalez, "An intelligent decision support system for intrusion detection and response", . *In Proc. of International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security (MMM-ACNS), St.Petersburg. Springer-Verlag, 21-23 May, 2001.*

[30] Chittur, A., "Model generation for an intrusion detection system using genetic algorithms", *High School Honors Thesis, Ossining High School. In cooperation with Columbia Univ, 2001.*

[31] Crosbie, M. and E. H. Spafford, "Active defense of a computer system using autonomous agents", *Technical Report CSD-TR- 95-008, Purdue Univ., West Lafayette, IN, 15 February 1995.*

[32] G. J. Klir, "Fuzzy arithmetic with requisite constraints", *Fuzzy Sets and Systems, 91:165175, 1997.*

[33] G. Florez, SM. Bridges, Vaughn RB, "An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection", *Annual Meeting of The North American Fuzzy Information Processing Society Proceedings, 2002.*

[34] Luo, J., "Integrating fuzzy logic with data mining methods for intrusion detection", *Masters thesis, Mississippi State Univ., 1999.*

[35] Chan Man Kuok, Ada Wai-Chee Fu, Man Hon Wong, "Mining Fuzzy Association Rules in Databases", *SIGMOD Record 27(1): 41-46 (1998).*

[36] Heikki Mannila and Hannu Toivone, "Discovering Generalized Episodes Using Minimal Occurrences", *In Proceedings of the Second Int'l Conference on Knowledge Discovery and Data Mining, 1996.*

[37] Debar, H., Becker, M., and Siboni, D., "A Neural Network Component for an Intrusion Detection System", *IEEE Computer Society Symposium on Research in Security and Privacy, Los Alamitos, CA, pp. 240–250, Oakland, CA, May 1992.*

[38] Ryan, J., M.-J. Lin, and R. Miikkulainen, "Intrusion detection with neural networks", *In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), Advances in Neural Information Processing Systems, Volume 10, Cambridge, MA. The MIT Press, 1998.*

[39] Endler, D., "Intrusion detection applying machine learning to Solaris audit data", *In Proc. of the 1998 Annual Computer Security Applications Conference (ACSAC98), Scottsdale, AZ, pp. 268279. IEEE Computer Society Press, 1998.*

[40] McHugh J., "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory", *ACM Trans. Information System Security 3 (4), 262294, 2000.*

[41] Cannady, J., "Applying CMAC-based On-line Learning to Intrusion Detection", *In Proceedings of the 2000 IEEE/INNS Joint International Conference on Neural Networks, July 2000.*

[42] Bonifacio, J.M, Cansian, A.M., de Carvalho, A., & Moreira, E., "Neural Networks Applied in Intrusion Detection", *In Proceedings of the International Joint Conference on Neural Networks, 1998.*

[43] Didaci, L., G. Giacinto, and F. Roli, "Ensemble learning for intrusion detection in computer networks", *Proc. of AI*IA, Workshop on "Apprendimento automatico: metodi e applicazioni", Sept 11, 2002, Siena, Italy.*

[44] Hofmeyr, S. A. and S. Forrest, "Immunizing computer networks: Getting all the machines in your network to fight the hacker disease", *In Proc. of the 1999 IEEE Symp. on Security and Privacy, Oakland, CA. IEEE Computer Society Press, 1999.*

[45] Dasgupta, D. and F. Gonzalez, "An immunity-based technique to characterize intrusions in computer networks", *IEEE Trans. Evol. Comput. 6 (3), 1081-1088, June 2002.*

[46] Fan, W., "Cost-Sensitive, Scalable and Adaptive Learning Using Ensemble-based Methods", *Ph. D. thesis, Columbia Univ., 2001.*

[47] Eskin, E., A. Arnold, M. Preraua, L. Portnoy, and S. J. Stolfo, "A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data", *In D. Barbar and S. Jajodia (Eds.), Data Mining for Security Applications. Boston: Kluwer Academic Publishers, May 2002.*

[48] Mukkamala, S., A. H. Sung, "Identifying key variables for intrusion detection using neural networks", *Proceedings of* 15^{th} *International Conference on Computer Communications, pp. 1132-1138, 2002.*

[49] Mukkamala, S. and A. H. Sung, "Identifying significant features for network forensic analysis using artificial intelligent techniques", . *International Journal of Digital Evidence 1 (4), 1-17, 2003.*

[50] Frank, J., "Artificial intelligence and intrusion detection: Current and future directions", *In Proc. of the 17th National Computer Security Conference, Baltimore, MD. National Institute of Standards and Technology (NIST), 1994.*

[51] Portnoy, L., E. Eskin, and S. J. Stolfo, "Intrusion detection with unlabeled data using clustering", *In Proc. of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001), Philadelphia. ACM, 5-8 November, 2001.*

[52] Chan, P. K., M. V. Mahoney, and M. H. Arshad, "Managing Cyber Threats: Issues, Approaches and Challenges", *Chapter Learning Rules and Clusters for Anomaly Detection in Network Traffic. Kluwer, 2003.*

[53] Marin, J. A., D. Ragsdale, and J. Surdu, "A hybrid approach to profile creation and intrusion detection", *In Proc. of DARPA Information Survivability Conference and Exposition, Anaheim, CA. IEEE Computer Society, 12-14 June, 2001.*

[54] Staniford, S., J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans", *Journal of Computer Security 10 (1-2), 105-136, 2002.*

[55] Marchette D., "A Statistical method for profiling network traffic", *In First USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, CA, pp.119-128, USENIX, 9-12 April, 1999.*

[56] Sequeira, K. and M. Zaki, "Admit: Anomaly-based data mining for intrusions", *In Proc. of the 8th ACM SIGKDD International conf. on Knowledge Discovery and Data mining, Edmonton, Alberta, Canada, pp. 386-395. ACM Press, 2002.*

[57] Yeung, D.-Y. and C. Chow, "Parzen-window network intrusion detectors", *In Proc. of the Sixteenth International Conference on Pattern Recognition, Volume 4, Quebec City, Canada, pp. 385388. IEEE Computer Society, 11-15 August, 2002.*

[58] Leung, K. and Leckie, C., "Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters", *In Proc. Twenty-Eighth Australasian Computer Science Conference (ACSC2005), Newcastle, Australia, 1-3 February 2005, pp. 333-342, 2005.*

[59] Carbone, P. L., "Data mining or knowledge discovery in databases: An overview", *In Data Management Handbook. New York: Auerbach Publications, 1997.*

[60] Krugel, C., T. Toth, and E. Kirda, "Service specific anomaly detection for network intrusion detection", *In Proc. of the 2002 ACM Symp. on Applied Computing (SAC2002), Madrid, Spain, pp. 201-208. ACM Press, 2002.*

[61] Denning, D. E., "An intrusion-detection model", *IEEE Transactions on Software Engineering 13 (2), 222-232, February, 1987.*

[62] Javitz, H. S. and A. Valdes, "The NIDES statistical component: Description and justification", *Technical report, SRI International, March 1993.*

[63] Mahoney, M. V. and P. K. Chan, "Learning non stationary models of normal network traffic for detecting novel attacks", *In Proc. of the 8th ACM SIGKDD International Conf. on Knowledge Discovery and Data mining, Edmonton, Alberta, Canada, pp. 376385. ACM Press, 2002.*

[64] Sinclair, C., L. Pierce, and S. Matzner, "An application of machine learning to network intrusion detection", *In Proc. 15th Annual Computer Security Applications Conference (ACSAC 99), Phoenix, pp. 371-377. IEEE Computer Society, 6-10 December, 1999.*

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

[66] Ourston *et al.*, "Applications of Hidden Markov Models to Detecting Multi-stage Network Attacks", *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS03).*

[67] Lane, T. D., "Machine Learning Techniques for the computer security domain of anomaly detection", *Ph. D. thesis, Purdue Univ., West Lafayette, IN, August, 2000.*

[68] J. Zhang and M. Zulkernine, "Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection", *Symposium on Network Security and Information Assurance Proc. of the IEEE International Conference on Communications (ICC), 6 pages, Istanbul, Turkey, June 2006.*

[69] V. Mittal and G. Vigna, "Sensor-based intrusion detection for intra-domain distance-vector routing", *In Proceedings of the CCS, pages 127 - 137. ACM, 2002.*

[70] Tim Bass, "IDS and multisensor data fusion", *Communications of the ACM, April 2000.*

[71] F. Cuppens, F. Autrel, A. Miege, and S. Benferhat, "Correlation in an intrusion detection process", *In S Ecurit e des Communications sur Internet (SECI'02), Sep. 2002.*

[72] Axelsson, S., "The base-rate fallacy and the difficulty of intrusion detection", *ACM Trans. Information and System Security 3 (3), 186205, August 2000.*

[73] Axelsson, S., "A preliminary attempt to apply detection and estimation theory to intrusion detection", *Technical Report 00-4, Chalmers Univ. of Technology, Goteborg, Sweden, 2000.*

[74] Fan, W., W. Lee, S. J. Stolfo, and M. Miller, "A multiple model cost-sensitive approach for intrusion detection", *In R. L. de M'antaras and E. Plaza (Eds.), Proc. of Machine Learning: ECML 2000, 11th European Conference on Machine Learning, Volume 1810 of Lecture Notes in Computer Science, Barcelona, Spain, pp. 142153. Springer, 31 May - 2 June, 2000.*

[75] Winkler, J. R., Landry, L. C., "Intrusion and anomaly detection, ISOA update", *In Proceedings of the 15th National Computer Security Conference, pages 272-281, Oct. 1992.*

[76] Snapp, S. R., Smaha, S. E., Grance, T., Teal, D. M., "The DIDS (Distributed Intrusion Detection System) Prototype", *In Proceedings of the USENIX Summer 1992 Technical Conference, pages 227-233, June 1992.*

[77] Porras, A. and Neumann, P. G., "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances", *In Proceedings of the National Information Systems Security Conference, October 1997.*

[78] Levent Ertoz and Eric Eilertson and Aleksandar Lazarevic and Pang-Ning Tan and Vipin Kumar and Jaideep Srivastava and Paul Dokas, "MINDS - Minnesota Intrusion Detection System", *Next Generation Data Mining, MIT Press, 2004.*

[79] Tamas Abraham, "IDDM: Intrusion Detection using Data Mining techniques", *Information Technology Division Electronics and Surveillance Research Laboratory, May, 2001.*

[80] Helmer G.G, Wong.J.S.K, Honavar V, Miller L., Inteligent Agents for Intrusion Detection", *Proccedings of the IEEE Information Technology Conference, Syracuse, USA.*

[81] http://www.afina.com.ve/download/docs/iss/iss_real%20secure.pdf

[82] http://www.softwarespectrum.com/business/TAAP_Library/Symantec_docs/Manhunt_Fact_Sheet.pdf

[83] http://www.dshield.org/

[84] http://www.mynetwatchman.com/

[85] Yang,J., Wang,H., Wang,W., and Yu,P., "Enhanced biclustering on expression data", *In Proceedings of the 3rd IEEE Conference on Bioinformatics and Bioengineering (BIBE), pp. 321-327, 2003.*

[86] Q. Sheng, Y. Moreau, and B. De Moor, "Biclustering microarray data by Gibbs sampling", *Bioinformatics, 19(Suppl. 2):II196–II205, 2003.*

[87] Mishra N, Ron D., Swaminathan R., "A New Conceptual Clustering Framework", *Journal on Machine Learning, Volume 56, Numbers 1-3 / July, 2004, pages 115-151,Springer Netherlands.*