60-510 SURVEY REPORT
ASSOCIATION AND CLASSIFICATION RULE MINING FOR NETWORK INTRUSION DETECTION

Student name: S. S. Ahmedur Rahman

rahma21@uwindsor.ca

Instructor: Dr. R. Frost

Fall 2006

SCHOOL OF COMPUTER SCIENCE

UNIVERSITY OF WINDSOR

Abstract

A network intrusion detection system is an important component in protecting or securing a network. Intrusion detection systems look for unusual or suspicious activities that deviate from normal behavior. However, sometimes normal operation produces traffic that looks like an “attack signature”, which results in false alarms. When the network grows or becomes complex, the rate of false alarm increases. Sensor devices have played some roles in monitoring network traffic. This report provides a comprehensive review of intrusion detection systems and data mining techniques specially association and classification rules. A main focus of the report is a sources of methods for reducing the volume of false alarms with association and classification rule techniques. This can be done by maintaining network sensor profiles and understanding sensor behaviors.

Keywords: Data mining, network intrusion detection, false alarm, algorithm, association rule, classification rule, intrusion detection filters, sensor behavior.

Table of Contents

1. INTRODUCTION:

Security of computer networks has become the most crucial issue nowadays. Traditionally, we consider the firewall as the first line of defense, but the unsophisticated firewall policy cannot meet the requirements of some organizations, which are highly sensitive to security [XQJ01]. Many studies and research has been done already in this field and still this field is getting a lot of attention from the researchers and professionals. There are many methods and models like ADAM [BCJ+01], DHP [LXY03], LERAD [MC03], ENTROPY [Yo03] already present today to resolve this problem, but we need an efficient and smart model which can detect any intrusions to the network as well as predict and handle new types of intrusions. We have many models [De87] [SGF+02] [SZ02] [KTK02] which are efficient at catching previously known intrusions but when the term "smart" comes then we need a technology that can detect new intrusions based on previously known intrusions. That is why nowadays "*Data Mining*" has become a vital part in network intrusion detection. The intrusion detection model is an old concept which was first introduced by Denning in [De87]. But data mining is a newer concept in the network intrusion detection model. Research in this area started as early as 1998/ 1999, [LSM99a] [LSM99b] [LS98] [LP99]. But within these few years this field has received a great deal of attention among researchers and many studies have been done on this topic: how data mining concept can be used efficiently to improve the performance of the network detection model [Le02][Yo03][MC03][LXY03] [NC03] [BCL02] [BK03] [LS98] [LSC+01a] [LSC+01b]. To understand this more clearly we need to know what is "*Data Mining*" and what is "*Network Intrusion Detection*" separately then we can concentrate on how they can be merged together.

Data Mining is knowledge discovery in databases. Data mining techniques help us to discover the hidden patterns in the database automatically. The environment for data mining is a huge database or a data warehouse source. Then it looks at all data, builds relationships among data and makes a prediction. As a definition we can say that Data

mining is the *automated* process of extracting *hidden predictive* information from a large database [63]. In other words, data mining is *automated statistical analysis*.
For example, suppose in case of a financial organization before they issue any credit or loan to any customer, they need to know if the customer will return the loan or will be a defaulter. The company has a large database that contains all information about all of their previous and current customers. From there the data mining technique can find some rules that: previously those customers who were under 20 years of age, and had income under certain level, had a bad history. With this rule and the current information of the new customer, a data mining model will predict the future behavior of a new customer and how possible it is to recover loans granted to these customers.

Currently, with the growing size of the World Wide Web, the network is also becoming more complex and new intrusions or attacks are coming out everyday. To detect an intrusion the systems use sensors to monitor the network traffic and raise alarms when they match saved patterns [LCE+01][CG00][MCH00]. Security analysts decide whether it is a false alarm or a true alarm, then respond accordingly.

If the network is small and signatures are kept up to date, then an analyst can observe all alarms and can determine the type of attack, if it is a new or old type of attack. But as the network grows and becomes more complex, the human analyst will be overwhelmed with all alarms produced by the system daily and here comes the question: is it possible to automate the process so that it can detect new intrusions without the help of a human analyst? The answer is yes, using Data Mining.

This report focuses on the work done in the field of Data Mining Based Network Intrusion Detection, specially using Association and Classification rule. The rest of the report is organized as follows: Chapter 2 discusses data mining based intrusion detection systems. Chapter 3 discusses the association rule and classification rule algorithms used in data mining based network intrusion detection. This is done in chronological order. Section 3.1 discusses the early research before 2000, section 3.2 discusses later researches, which are between 2001 and 2002 and finally section 3.3 discusses the

algorithms from 2003 to now. Finally the last part of this report consists of conclusions, acknowledgements, annotations and bibliography.

2. INTRODUCTION TO THE DATA MINING BASED NETWORK INTRUSION DETECTION MODEL:

As network-based computer systems play increasingly vital roles in modern society, they have become the target of intrusions by our enemies and criminals. In addition to intrusion prevention techniques, such as user authentication (e.g. using password), avoiding programming errors and information protection (e.g., encryption), intrusion detection is often used as another wall to protect computer systems [LSM99a]. At this point data mining will give the system extra strength against intrusions. Intrusion detection techniques can be categorized into two: 1) *Misuse detection system*, which use "signatures" to detect attacks whose behaviors are well understood and 2) *Anomaly detection system*, which use some kind of statistical, deviation from the normal user behavior pattern or data mining analysis to do the job [QW02] [BCJ+01] [LSM99a] [LSM99b] [Le02] [BCV01] [BR01]. To understand the data mining perspective in a network intrusion detection system (NIDS) we need to focus on NIDS without data mining first, then we can see how data mining techniques can be added to an existing system to make it an anomaly detection system from a misuse detection system. But it is also true that an anomaly-based intrusion detection system is not very efficient if it is not combined with a misuse intrusion detection system [LJ99] [LJ00]. Current commercial ID systems such as Real Secure and Computer Misuse Detection System (CMDS) have distributed architectures using rule-based detection, statistical-anomaly detection or both [Ba00]. We can see the practical example of this at the Columbia IDS project [SLC+01] or even in the model by Lee et. al [LCE+01] which is a combined misuse and anomaly detection model.

The rest of this chapter is categorized as follows. Section 2.1 discusses systems with data mining anomaly detection models and 2.2 reviews the association rule and classification rules, 2.3 about the data mining at alarm level and 2.4 reviews the chapter.

2.1. Data Mining Based NIDS:

All anomaly-based intrusion detection systems are mainly based on statistical analysis or data mining. In this method we have a repository of normal or innocent behavior patterns. Any new incoming event is compared with those normal profiles and the system determines any deviation from the normal profile. Then it determines if it can be flagged as an intrusion or not. Here two scenarios may occur: 1) Anomalous (normal but seems to be suspicious) activities that are not intrusive are flagged as intrusive. This will result in *false positive* and 2) Intrusive activities that are not flagged as intrusive will result in *false negatives*, which is more serious situation. [BK03].

In data-mining based NIDS we need a database, preferably a large database because data mining is scalable to large database, which will contain all innocent behavior or signatures. Then data-mining techniques can be applied on this database to determine the correlations and associations among the data to determine new rules. So through the next sections we will focus on source of these audit data, processing them and methods of data mining based NIDS.

2.1.1. Source of Audit Data:

The main source of audit data is network sensors. A network sensor can be internal or external. Internal sensors are mainly used to determine any attack to a certain host. External sensors are used to monitor many hosts of a network. These sensors monitor the network traffic and record those data.

Data records consist of many attributes. When doing data mining for intrusion detection one could use data at the level of TCPDUMP [LS98] or at the alarm level [MCH00]. In

both types of data we will find fields for source IP address, destination IP address, source port number, destination port number, transfer protocol (TCP, UDP etc.), date/ time and traffic duration. These base attributes give a good description of the individual connection or alarm [BCH+01]. For example, sensors are a kind of software that is installed in the host computer or in the network gateway computer. It scans ports, measures bandwidth of the connection and if there is any large deviation then it identifies it as an attack and as a result it can stop access to a certain port or can forward the matter to a system administrator. An example of this kind of software is BSM (Basic Security Module), which comes as a part of Solaris UNIX Operating System. The same kind of opportunity is available also for the Windows Operating System, which is called BAM (Basic Auditing Module), which was developed at Columbia University. Another example of a sensor system is the Network Flight Recorder (NFR), a commercial network monitoring system that includes a network packet sniffing engine with a high level language for interpreting network packet data [SLC+01].

Now let us come to the point of what this audit data looks like. When a network sensor gathers data about a connection, it gets all the data in an unformatted manner. Suppose a sensor that detects that a connection is being established or someone is accessing the network through a port, it gets data about source IP address, destination IP address, source port number, destination port number, transfer protocol (TCP, UDP etc.), date/ time and traffic duration etc. All these data remain in the same place together in such a manner that a human will not understand the data by looking into them. They are in binary format and they come as TCP/IP header data. The following figure shows the format of TCP/IP header and the next figure shows the result of such a TCP DUMP data [ZC02]. So, software like BSM, BAM, NFR [SLC+01] organize them and show us in such a way that we understand and can see the field attributes like source IP, destination IP, duration etc. Before organizing them we call them raw audit data.

|<----------32 bit---------->|

|<---------- 16 bit---------->|<----------16 bit---------->|

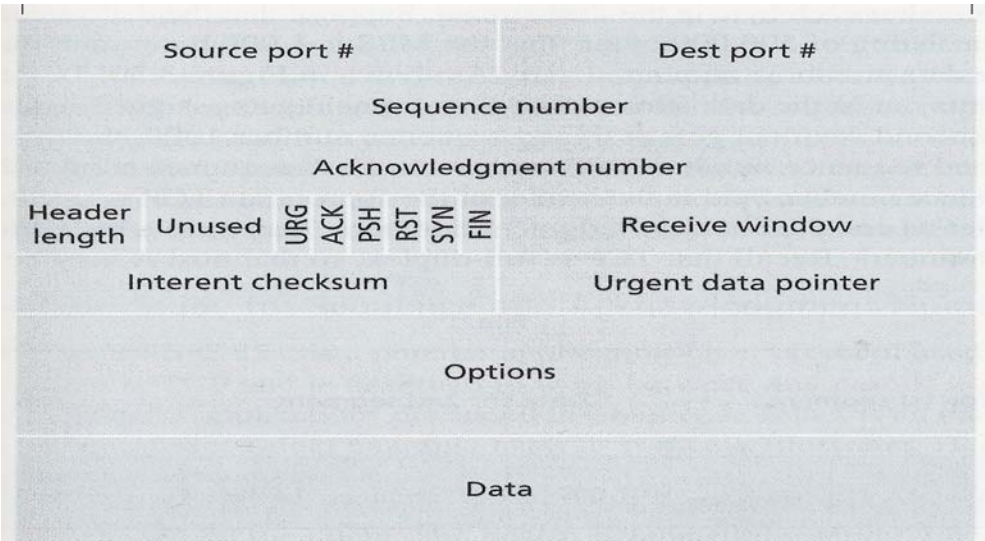

Figure 1: TCP Header Format/ Format of raw audit binary data

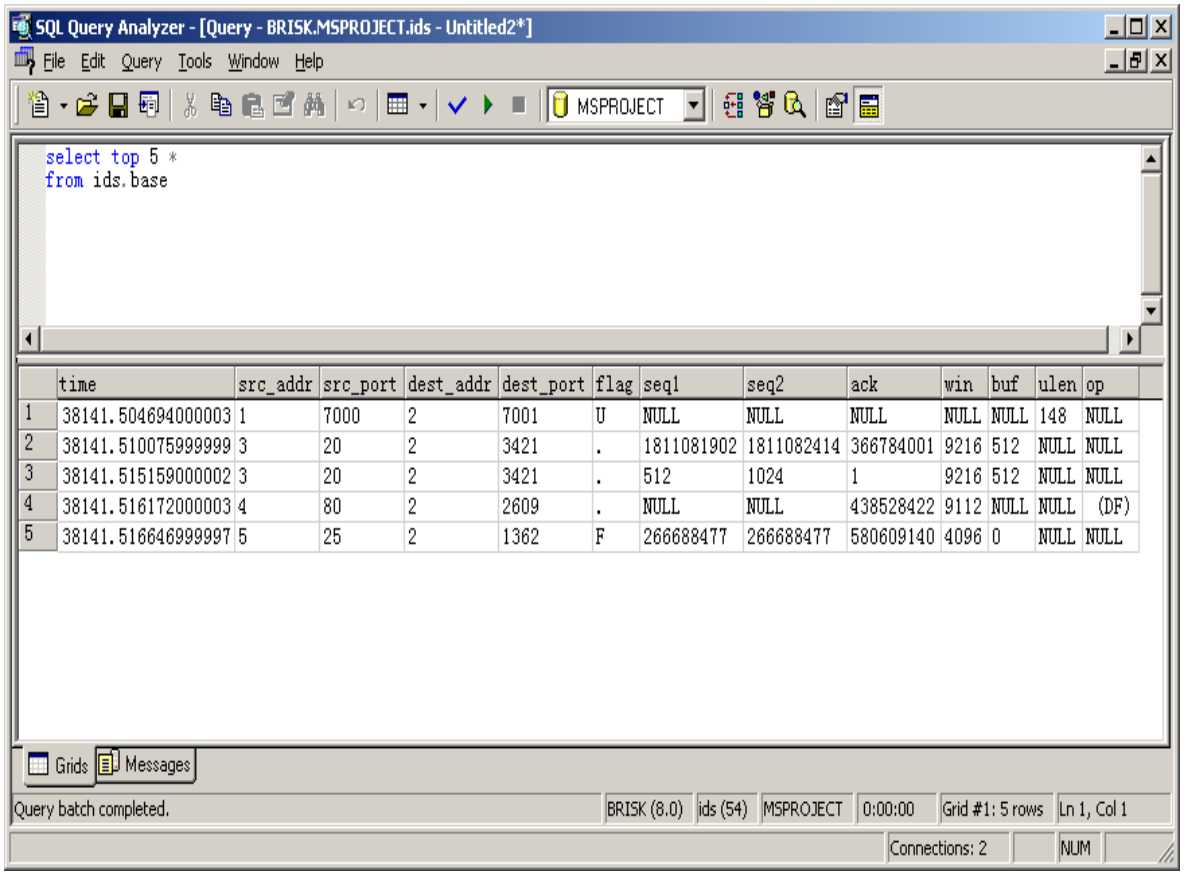

Figure 2: Sample TCP DUMP Data ([Z02] page: 11)

2.1.2. Processing of audit data and feature construction:

Audit data collected from network sensors, or from other sources, are raw and are in binary format. Before using them we need to process them and need to generate rules from them. The basic and primary concept here is that we build a database of audit data and we have some previously known rules. With the help of these two and with the association rule algorithm (association rule is introduced in the next section) we generate new rule sets for possible attacks. Then we apply those rules to new incoming events to detect new or unknown attacks.

Before applying any data mining rules we need to pre-process the sensor generated raw binary audit data. This work is done by TCPDUMP or BSM [LSM99a] [LCE+01]. To preprocess those raw audit data the Columbia team used BAM (Basic Auditing Model)

instead of BSM (Basic Security Model), which was their own creation. Pre-processing means that inputs are raw audit data and output will be those audit data in a format which will be organized with attributes like for source IP address, destination IP address, source port number, destination port number, transfer protocol (TCP, UDP etc.), date/ time and traffic duration.

The next step is to apply some data-mining algorithm to these pre-processed data. These algorithms are generally classification rule (Section 2.2.1), association rule (2.2.2) and frequent episodes algorithms [MTV97]. Some of the research has focused on one specific rule [CG00], some have focused on a combination of two or three rules [BCJ+01] or some have focused on some advanced modified rule such as the artificial anomaly generation rule [LSC+01]. Through the next several subsections, we will briefly discuss about some of these rules (Classification and Association Rules).

2.2. Methods of Data Mining in NIDS

2.2.1. Classification Rule:

Intrusion detection can be thought of as a classification problem: we wish to classify each audit record into one of a discrete set of possible categories, normal or a particular kind of intrusion.

Given a set of records, where one of the features is the class label (i.e., the concept), classification algorithms can compute a model that uses the most discriminating feature values to describe each concept. For example, consider the telnet connection records shown in Figure 3. Here, *hot* is the count of accesses to system directories, creation and execution of programs, etc, *compromised* is the count of file/path "not found" errors, and "Jump to" instructions, etc. RIPPER (a standard rule based machine learning algorithm

developed at ATT research) [LS00], a classification rule learning program, generates rules for classifying the telnet connections and some of the rules are displayed in Figure 4.

label	service	flag	hot	failed_logins	compromised	root_shell	su	duration	...
normal	telnet	SF	0	0	0	0	0	10.2	...
normal	telnet	SF	0	0	0	3	1	2.1	...
guess	telnet	SF	0	6	0	0	0	26.2	...
normal	telnet	SF	0	0	0	0	0	126.2	...
overflow	telnet	SF	3	0	2	1	0	92.5	...
normal	telnet	SF	0	0	0	0	0	2.1	...
guess	telnet	SF	0	5	0	0	0	13.9	...
overflow	telnet	SF	3	0	2	1	0	92.5	...
normal	telnet	SF	0	0	0	0	0	1248	...
...	...	...	...	...	...	...	...	...	...

Figure 3: Telnet Records ([LS00] page: 232)

RIPPER rule	Meaning
guess :- failed_logins >= 5.	If number of failed logins is greater than 5, then this telnet connection is "guess", a guessing password attack.
overflow :- hot = 3, compromised = 2, root_shell = 1.	If the number of hot indicators is 3, the number of compromised conditions is 2, and a root shell is obtained, then this telnet connection is a buffer overflow attack.
...	...
normal :- true.	If none of the above, then this connection is "normal".

Figure 4: Example RIPPER Rules from Telnet Records ([LS00] page: 233)

Here, we see that RIPPER indeed selects the unique feature values in identifying the intrusions. These rules can be first inspected and edited by security experts, and then be incorporated into misuse detection systems.

The accuracy of a classification model depends directly on the set of features provided in the training data. For example, if the features *hot*, *compromised* and *root shell* were removed from the records in Figure 3, RIPPER would not be able to produce accurate rules to identify buffer overflow connections. Thus, selecting the right set of system features is a critical step when formulating the classification tasks [LSM99a] [LS98].
Now let's use a generalized practical example.

Outlook	**Temp(F)**	**Humidity(%)**	**Windy?**	**Class**
sunny	75	70	true	Play

sunny	80	90	true	Don't Play
sunny	85	85	false	Don't Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
overcast	72	90	true	Play
overcast	83	78	false	Play
overcast	64	65	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play
rain	65	70	true	Don't Play
rain	75	80	false	Play
rain	68	80	false	Play
rain	70	96	false	Play

Figure 5: Training data set for classification rule learning. ([JHK+01] page: 6)

Suppose we have two classes: Play and Don't Play. We want to observe other attributes and want to conclude our decision among these two classes. At first we take the class Play. Then we relate it with the first attribute "outlook". "Outlook" can have three values which are "sunny", "overcast" and "rain". From the dataset we can see that if the outlook is overcast then it is always "play", so we can make it as non-expandable leaf, as there is no chance that this condition will be violated. Then if the outlook is "sunny" then we can see that "play" occurs two times and "don't play" occurs three times, so we make "don't play" as expandable leaf. At last when outlook is "rain" we see from the dataset that "don't play" occurs two times and "play" occurs three times, so we make "play as expandable leaf. Then we look at another attribute "humidity". We can see that when the outlook is sunny, "don't play" occurs 3 times and all those times "humidity" is greater than 75% (90%, 85%, 95%) and two times when "play" occurs "humidity" is less than 75% (70%, 70%). So when the humidity is less than 75% we make "play" as non-expandable leaf and when "humidity" is greater than 75% we make "don't play" as non-expandable leaf. This is how we classify each condition into a class until it becomes a non-expandable leaf. The steps and demonstration of this rule is as follows:

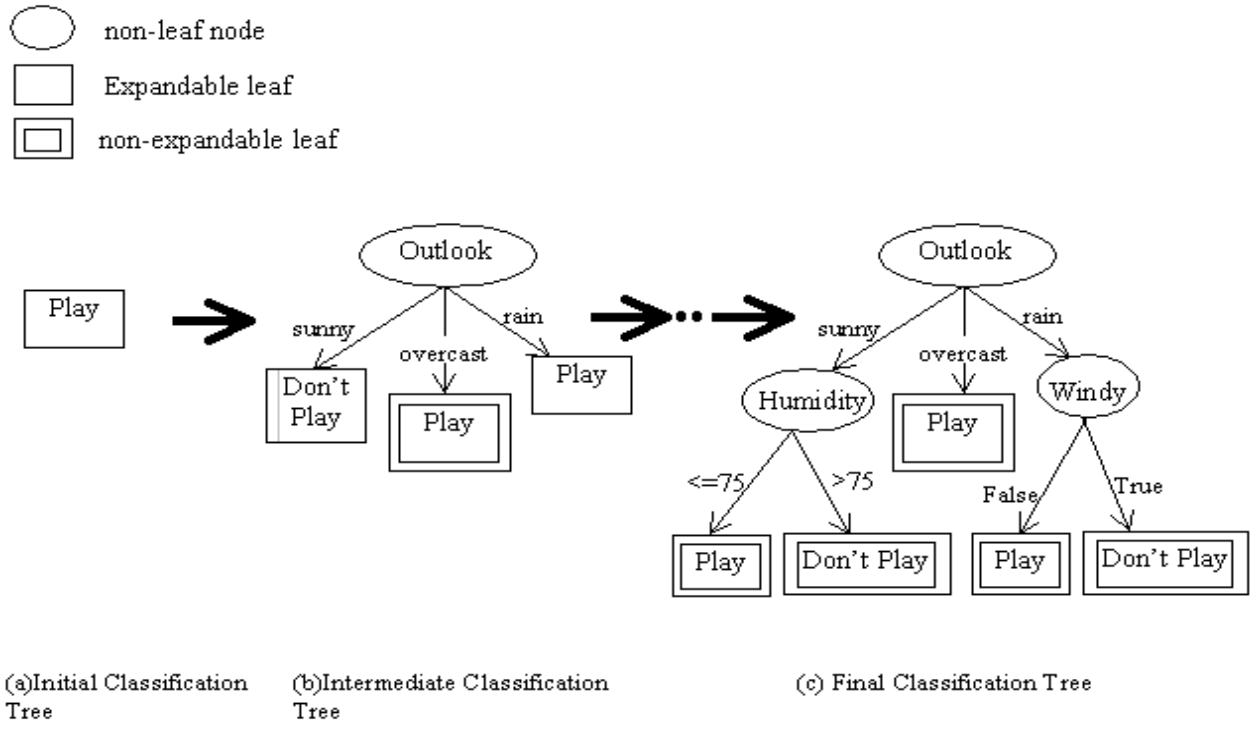

Figure 6: Demonstration of classification rule (Hunt's method) ([JHK+01] page: 6)

2.2.2. Association Rules:

An association rule is mainly a mathematical rule which is found useful in data mining based NIDS. Each and every model developed so far in this context is related somehow with an association rule.

In the database, the association between data items means that we can infer that particular data item is in existence because of the appearance of some data items in a transaction. The purpose of mining association rules is to dig out all of the latent association rules between the data items [LXY03].

Let us discuss the association rule in a mathematical form. Here are some standard definitions of association rule related terms from [D03] on page: 166:

Given a set of items $I = \{I_1, I_2, ..., I_m\}$ and a database of transactions $D = \{t_1, t_2, ..., t_n\}$ where $t_i = \{I_{i1}, I_{i2}, ..., I_{ik}\}$ and $I_{ij} \in I$, an association rule is an implication of the form $X \rightarrow Y$ where $X, Y \subset I$ are sets of items called *itemsets* and $X \cap Y = \varnothing$.

The *support* for an association rule $X \rightarrow Y$ is the percentage of transactions in the database that contain $X \cup Y$.

The *confidence or strength* for an association rule $X \rightarrow Y$ is the ratio of the number of transactions that contain $X \cup Y$ to the number of transactions that contain X.

Let us demonstrate the association rule with a practical example. Suppose in a library many customers who buy a book also buy a pencil leads to an association rule *book→pencil*. There are also two parameters involved with this: *support* and *confidence*. The association rule book→pencil has 30% *support* means that 30% of all customers buy both items. The rule has 10% *confidence* means that 10% of customers who a buy book (antecedent) also buy a pencil (rule consequent), meaning that this rule has 10% accuracy.

Let us discuss another example of how to derive association rules. For example, in a set of library transactions, the following records were found:

Transaction No	Items Purchased
1	Book, Pencil
2	Book, Pencil, Paper, Magazine
3	Book, Paper, Magazine

Figure 7: Sample library transactions

A standard algorithm for association rule mining is the Apriori algorithm [ISA93], which works as follows. Suppose a user specifies minSupport is 2. First we will collect the transactional database items and create an itemset Candidate 1 (C1), then we will eliminate all of them from C1 which do not have support greater than or equal to the minSupport. We will call the second one as frequent patterns with one element in their set, L1. After creating L1 we need to create C2. To create C2 we will join L1 with itself (Apriori-gen way), and will eliminate all itemsets which do not have at least minSupport. We will continue this process until we get a candidate set Cn or Ln empty. After that we will create a large itemset which will be the union of all L (i.e. L=L1 U L2 U ……). Now let's see the process with sample data. From the sample table given above, we compute the support of each itemset by scanning the database table to find that for example Book:3 meaning that Book has support of 3.

C1 = {Book:3, Pencil:2, Paper:2, Magazine:2}

L1 = {Book, Pencil, Paper, Magazine}, Since all have support >= minSupport 2 in the database.

Joining L1 by L1 (Apriori join way) gives C2 and scanning the database table gives the appeneded supports:

C2 = {(Book, Pencil):2, (Book, Paper):2, (Book, Magazine):2,
(Pencil, Paper) :1, (Pencil, Magazine) :1,
(Paper, Magazine) :2}

Support of (Book → Pencil): 2. √ (OK)
(Book → Paper): 2. √ (OK)
Support of (Book → Magazine): 2. √ (OK)
(Pencil → Paper): 1 < 2. X (Need to eliminate)
(Pencil → Magazine): 1 < 2. X (Need to eliminate)
(Paper → Magazine): 2. √ (OK)

So, L2 = {(Book, Pencil), (Book, Paper), (Book, Magazine), (Paper, Magazine)}

In the same way,

C3 = {(Book, Pencil, Paper):1, (Book, Pencil, Magazine):1, (Book, Paper, Magazine):2}

Among itemsets in C3, only (Book, Paper, Magazine) has support up to minSupport.

So, L3 = {(Book, Paper, Magazine)}

Then C4 = {}

L = L1 U L2 U L3

= {Book, Pencil, Paper, Magazine,

(Book, Pencil), (Book, Paper), (Book, Magazine), (Paper, Magazine)

(Book, Paper, Magazine)}.

Here L defines all frequent or large pattern from which association rules are generated. From every frequent pattern, e.g., (Book, Paper, Magazine) rules are generated from all of its itemsets and only rules with confidence greater than or equal to the minimum confidence provided by the user are retained, while the rest are pruned. Rules from the frequent pattern (Book, Paper, Magazine) are (Book, Paper) => Magazine, (Book, Magazine) => Paper, (Paper, Magazine) => Book and the confidence of the rule (Paper, Magazine) => Book from the database table is the cardinality of the rule divided by the cardinality of the antecedent equals 2/3 or 75%. Thus, this rule is retained if the minimum confidence is 50%.

Generally, in data-mining based NIDS we create a database of non-intrusive events and then apply association rule technique into that dataset to find out all other rules or events when there will be no intrusions. This will find all hidden normal behavior. Then these rules will be compared with any incoming data itemsets to determine if it is an intrusion or not. The most critical factor here is that we have to set a minimum threshold for minimum support and confidence level. Later when discussing the algorithms, we will see it in details.

2.3. Data Mining for NIDS at the Alarm level:

As mentioned before when doing data mining for intrusion detection one could use data at the level of TCPDUMP or at the alarm level [BCH+01]. In this section we will focus on applying data mining at alarm level.

So far we have seen that network traffic data are collected from various sensors, then they are passed to classifiers where we applied data mining (association rule, artificial anomaly, frequent episodes rules etc.) to train out classifier, so that in real time when new data will come it will compare those data with existing rules and will label them as anomaly or intrusion or unknown attack. After labeling those data it will generate an alarm to the analysts. We also know that sometimes these alarms may be false alarms.

As the number of sensors and sensor generated data increases, the number of false alarms also increases at a great rate. Some research has focused on how we can reduce the false alarm rate using data mining techniques.

There are two groups of researchers focused on reducing the rate of false alarms using data filters. One group [BCH+01] [BTS+01] used data filters before data are sent to the classifier to be trained. Another group [CG00] used data filters on the output (alarms) of an existing intrusion detection system. Both are described briefly in this part and will be covered in detail in the algorithm part.

The first group conducted their research and experiment in the MITRE Corporation at VA, USA. In their experiment, network traffic is analyzed by a variety of sensors. This sensor data was pulled periodically to a central server for conditioning and input to a relational database. In their sensor log table, upwards of 95% of the traffic fit the profile of an IP mapping activity. That means, a single source IP was attempting a connection to hundreds or even thousands of destination IPs.

Before security specialists can start providing input to the data mining effort, this traffic must be filtered. At MITRE, this preliminary filter was called HOMER (Heuristic for Obvious Mapping Episode Recognition). The heuristic operates on aggregations by source IP, destination port, and protocol and then check to see if a certain threshold of destination IPs were hit within a time window. If the threshold is crossed, an incident is generated and logged to the database. The reduction obtained by HOMER was significant. For example, for the period of Sep. 18 to Sep. 23, 2000, MITRE network sensors

generated 4,707,323 alarms. After HOMER there were 2,824,559 – *a reduction of 40%.* Thus, HOMER provides one other important function. One more situation may occur. Suppose within an organizational intranet there may be thousands of connection requests from one host to another or to many hosts, which is normal. HOMER handles this situation by keeping some safe source IPs, which are within the organization [BCH+01] [BTS+01].

The second group of researchers' goal was to identify patterns of false alarms coming from intrusion detection systems. Their approach was to develop custom filters that reduce the false alarm stream based on known "normal behavior" in a particular environment. They used commercial intrusion detection systems, but filtered out produced alarms that fit a pattern caused by normal operation at that site.

Their idea was that a sequence of operations that are normal in a particular environment may contains operations that look like a potential intrusion. However, the complete sequence is unlikely to be duplicated in an intrusion. So alarms that are part of the complete sequence can be ignored. For example, suppose "ABCDEF" is a normal sequence of operation. Now an operation "CDE" had been done and the IDS has generated an alarm for the operation "CDE". According to their assumption, as the later operation ("CDE") is completely a part of a normal operation ("ABCDEF"), the alarm produced by this operation will most probably be a false alarm and can be ignored. To find such sequence they have applied a *Frequent Episode Rules* algorithm, which is a data mining algorithm [CG00]. The algorithm will be discussed in detail, in the algorithm section.

2.4. Data Mining Based NIDS in a Nutshell:

Various data mining techniques have been applied to intrusion detection because it has the advantage of discovering useful knowledge that describes user's or program's behavior from large audit data sets. Statistics, artificial neural network, rule learning,

outlier detection scheme are some of the data mining techniques widely used for anomaly and misuse detections [HC03]. Moreover entropy based [BCL02] [Yo03], graph based [NC03] intrusion detection systems are also in use.

Data mining has been applied in two ways in NIDS. One is at TCPDUMP level, which has been investigated more frequently and has been developed and improved by many researchers. The second way is at the alarm level. Their main goal was to reduce the false alarm rate. In normal basic data mining based NIDS we can see that network traffic data are coming from various sensors. For pre-processing those raw binary data someone can use BSM/ BAM/ NFR [LCE+01]. According to the researchers in [LCE+01] "After preprocessing those data we get those data in a formatted manner with sourceIP, destinationIP, sourcePort etc. Then we have a classifier, where we apply data mining techniques (association rule, frequent episode rule, clustering etc.) to train the classifier with huge amount of previously known normal data; so that it can dig out all correlation among the normal datasets and specify them as normal. Then this trained classifier is placed into real environment and any incoming dataset is compared with those normal datasets in our database. If there is certain deviation we generate a flag. At the same time we keep the training process of classifier on, so that it will be capable of detecting more new attacks."

3. ALGORITHMS IN DATA MINING BASED NETWORK INTRUSION DETECTION:

In this section we will discuss some important algorithms in this field. We have divided the research and algorithms into their chronological order. Since data mining came into play of network intrusion detection very recently, the early research in this area is no later than 1997 or 1998. But within this very short time data mining was able to get enough attention among researchers, and the research in this field is still expanding. The chronological order followed in this report is as follow:

- Early research: Up to 2000.
- Later research: From 2001 to 2002.
- Recent and Current research: From 2003 to now

3.1. Early Research:

3.1.1. Developing Custom Intrusion Detection Filters Using Data Mining:

Typical Intrusion Detection Systems match signatures of incoming traffic with authorized signatures and hence detect intrusion, which is basically a misuse detection system. But the signature of an intrusion also can be matched with authorized use, resulting is false alarm. The approach in this model [CG00] development is to develop custom filters that reduce the false alarm based on known "normal behavior" in a particular environment.

A lot of normal behaviors are triggered as attacks by false alarms. The main concern here is that, if we can recognize normal patterns and identify them in an alarm stream then we can easily filter them out and decrease the rate of false alarm greatly. The difficulty with this approach is building these filters and defining normal patterns. It also needs a considerable human effort. To reduce this effort Clifton and Gengo [CG00] have tried to use data mining techniques in their approach.

Clifton and Gengo have developed filters based on sequences of alarms. The idea is that, a sequence of operations that are normal in an environment may not seem to be normal in another environment. So a normal operation can be suspicious in another environment and that operation would be flagged as an anomaly in that environment. But practically that operation was not an anomaly rather it was an innocent operation. This misinterpretation can cause a false alarm. But it is unlikely that a complete sequence of normal operations will be duplicated in an intrusion. So, alarms that are part of complete normal sequence can be ignored. They have used *"Frequent Episodes"* [MTV97] to identify such frequently occurring sequences of alarms. An *episode* is a sequence of

alarms that occurs within a specific time window. And a *frequent episode* is one that recurs in many windows. Sometimes there may be interleaving operations between frequent episodes which are not related. It is difficult to detect frequent episodes in the presence of these unrelated operations and data mining techniques come into effect in this point to detect the most frequent episodes efficiently and automatically.

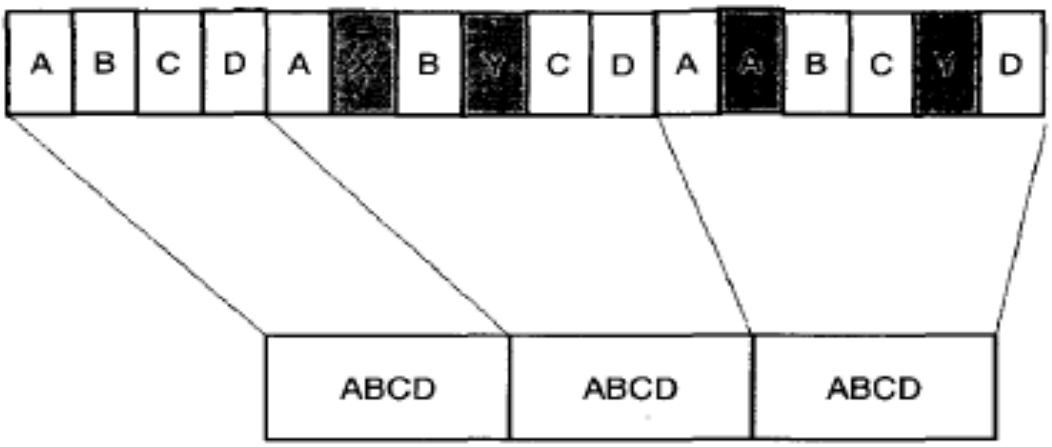

Figure 8: Sample frequent episodes with interleaving operation or noise (black) ([CG00] page: 2)

The main goal in this report is to identify sequences of alarms caused by normal operation. Frequent episodes are sequences of alarms that occur often. These frequent episodes are so important because of two things:

- A common sequence of alarms cannot be an intrusion. Because attackers will not try the same thing again and again unless they have not been detected. Normal operations are done more often and a frequent episode is the result of a normal operation.
- Analyzing frequent sequences and reduction of them from the list of possible attacks will be the largest reduction in the false alarm stream because there are always more normal operations than suspicious operations.

In their experiments [CG00] they have analyzed over one million intrusive alarms gathered from seven sensors in a network. The time period of their experiment was 2 weeks. They have loaded the logs into a relational database. The basic schema of the log is like Log(Event, FromIP, ToIP, time). Then they have used an algorithm called *"Query Flocks"*, which is basically a generalization of the association rule mining algorithm to detect the frequent sequences. Query Flocks algorithm gives some more flexibility in handling complex patterns than Frequent Episodes algorithm.

Their main focus was to augment data mining technology into an existing intrusion detection system. They didn't give any new algorithm but they proposed a model using the previously known frequent episode rule algorithm. Their proposed model was as follows:

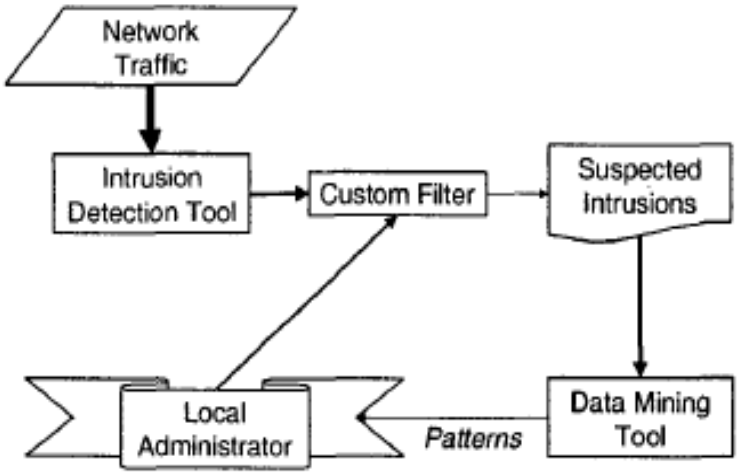

Figure 9: Developing custom filters with data mining ([CG00] page: 2)

In this model they used an existing commercial network intrusion detection system (they did not mention any specific system rather they mentioned that any existing Misuse Detection System would work) which collects network connection records data from sensors and performs basic operations and as an output the system generates alarms. Before that time data mining yet didn't come into play within the intrusion detection system and all systems were misuse systems based on signature detection. So, it tried to

find a match with a normal signature, if not found then it flagged it as alarm. As a result there were a lot of false alarms.

Then they passed those alarms through custom filters. In the filters using frequent episode rules they found out which are completely part of a normal sequence. Then alarms generated for those frequent patterns were ignored and only the rest of the suspicious intrusion alarms were passed to a data mining tool (Which is based on the Query Flocks algorithm, which is a generalization of association rule mining) [TUA+98], where it can use any data mining technique to find new patterns of intrusions, or in other words then the whole system can also perform anomaly detection.

In this research [CG00] they basically didn't develop any new model. They focused mainly on false alarm filters using previously implemented algorithms (Frequent Episodes and Query Flocks). They mainly showed that these algorithms can be used in the reduction of false alarms.

The main strength in their approach is that their concept of using filters can be easily implemented in any existing intrusion detection system. They also kept the data mining tool separate, which can be augmented with any existing system. One more thing is that the data mining tool will not have to handle a lot of inputs.

The main limitation in their model is that they used the misuse detection system and the data mining tools separately in two places but now there are anomaly based intrusion detection system like ADAM [BCJ+01] which can perform both at the same time.

3.1.2. A Data Mining Analysis of RTID Alarms:

In their research [MCZ+00] they tried to improve anomaly detection on IBM's Network Operation Center (NOC) which uses misuse detection. Generally IDS generates a lot of alarms everyday. Here they developed an algorithm where the system would learn to

detect anomalies based on its experience from the history of its alarms. Mainly they are applying data mining techniques (association rule) on the alarm context history.

Algorithm for anomaly detection:

This is the algorithm they have developed which can detect anomalies based on association rules. The algorithm is as follows:

```
LOOP
        B = read_next_burst_of_alarms
        IF (∃s ∈ F : s = B) THEN
            check rules(R), report frequent(B)
        ELSE
            M = most_specific_supported_itemsets(B, F)
            D = B − ∪_{m_i ∈ M} m_i
            check_rules(R), report_infrequent(B, M, D)
        END
END
```

Figure 10: Algorithm for anomaly Detection ([MCZ+00] page: 574)

Firstly, the continuous stream of alarms was partitioned into the bursts of alarms. The algorithm is closely related with association and correlation [Section 2.2.2.].

In this algorithm, at the first line, the burst of alarms is taken into B. Then next we check if this burst of alarms is a known frequent dataset or alarm burst. If so, then we report this burst of alarm as an innocent alarm set.

At the else part we identify the frequent alarm set supported by the burst and keep it in M. So, M contains the alarm set which is a closely subset of burst of alarm. Now we can say that M contains the alarm set which should have been innocent dataset, because frequent

datasets cannot be true alarms. But in real there are some more alarms which is not in the frequent dataset and we keep those in D and report D as infrequent and a kind of anomaly.

Another type of anomaly is unexpected absence of alarms. Suppose we generally know when these alarms come some other alarms also come along always. But in a certain situation if the first alarms came the some other is missing then it is an unexpected situation and will be reported as anomaly.

For example, let us take a reference string of a burst of alarms,

B = 1, 2, 3, 4, 5, 6

And some frequent alarm sequences,

F = 1, 2, 3, 4

= 10, 12, 22, 24

= 100, 200, 5, 9

Now we can see that our burst is not matched against any frequent sequences, so, we go to the else part and

M = 1, 2, 3, 4 (frequent set which is a sub set of B)

D = 5, 6 (those extra which are not in frequent set but are in burst)

Now suppose, B = 1, 2, 4

From the frequent set we know that 1, 2, 3, 4 always comes together. But in this case 1, 2, 4 has come but 3 has not come which is unexpected and is reported as anomaly. This finding of the absent 3 is done by association rule.

Sensor Profiling:

As each sensor has its own history of alarm behavior and they also vary by alarm types, alarm rates, distribution of alarm over day-of-week, time-of-week, so, here they tried to generate each sensor's own behavior table and later they applied it to customer market segmentation field.

In this experiment they used 27 NetRanger sensors for one month time and collected roughly 12 x 10^6 alarms, corresponding to about 2GB of data. Each alarm was described in its own type, priority level, time stamp, source and destination IP address and port and customer and sensor ID.

They were able to cluster those sensors based on these data. It showed that one group of sensors looked at internet traffic, some sensors of a particular client grouped together suggesting a very abnormal network and some sensors made a group with other sensors that seem to need improvement by tuning. The last group showed that this strategy could be successfully applied in servicing the faulty sensors.

The algorithm they used to cluster those sensors is very simple. At each step the algorithm decides whether to assign a new vector (Sensor profile) to an existing cluster or to create a new cluster. And this process continues until there is no change in existing clusters. This algorithm is a global measure called the *Condorset Criterion.*

3.2. Later Research:

3.2.1. ADAM: A Testbed for exploring the Use of Data Mining in Intrusion Detection:

ADAM [BCJ+01] was one of the most important research in this field at the time of 2001 and 2002. A lot of researches have been done on improvement of this algorithm later on.

ADAM uses a combination of association rules and classification to detect any attack in a TCP Dump audit trails. First, ADAM collects normal, known frequent datasets by mining into this model. Secondly, it runs an on-line algorithm to find last frequent connections and compare them with the known mined data and discards those which seem to be normal. With the suspicious ones it then uses a classifier which is previously trained to

classify the suspicious connections as a known type of attack, unknown type of attack or a false alarm.

There are total two phases in this experiment model. In the first phase they trained the classifier. This phase takes place only once offline before using the system. In the second phase they use the trained classifier to detect intrusions. The detailed algorithm with figure is described in the following page.

Phase 1:

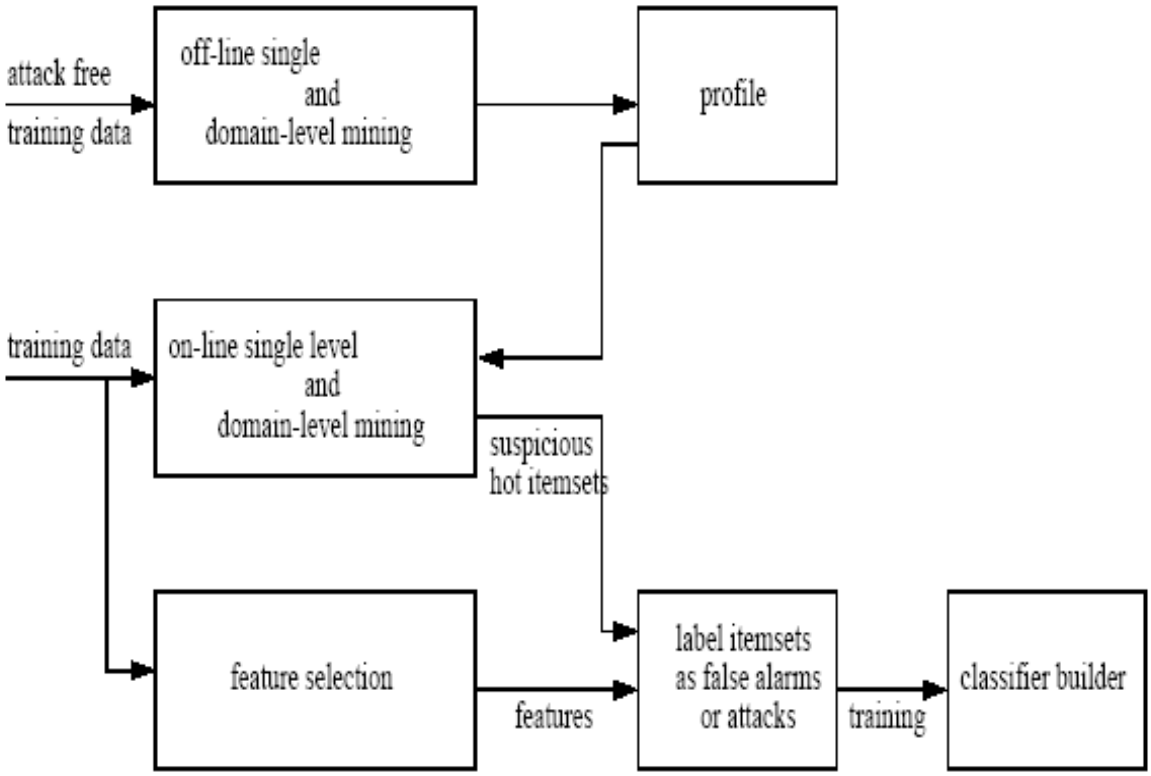

Figure 11: The training phase of ADAM ([BCJ+01] page: 5)

In the first step, a database of attack-free normal frequent itemsets, that have a minimum support, are created. This database serves as a profile, which is compared later against frequent datasets found. The profile database is populated with frequent itemsets with a specific format for the attack-free portion of the data. The algorithm used in this step can be a conventional association data mining algorithm although they used a customized algorithm for better speed. So, in the first step they are creating a profile of normal behavior. This profile mainly consists of network connection data which are normal, i.e. this profile contains set or combination of source IP, source port, destination IP, destination port, duration, timestamp, flag value etc which are normal.

In the second step they again use the training data, the normal profile and an on-line algorithm for association rule, whose output consists of frequent itemsets that may be attacks. The suspicious itemsets along with a set of features extracted from the data by a feature selection module are used as training data for the classifier which is based on decision tree. Now let's look at how the dynamic on-line association rule algorithm works. This algorithm is driven by a sliding window of tunable size. The algorithm outputs the itemsets that have received strong support with the profile within the specific window size time. They compare any datasets with profile database, if there is match then it is normal. Otherwise they leave a counter that will track the support of the itemset. If the support exceeds a certain threshold then the itemset is flagged as suspicious. Then the classifier labels that suspicious dataset as known attack, false alarm or unknown attack.

Phase 2:

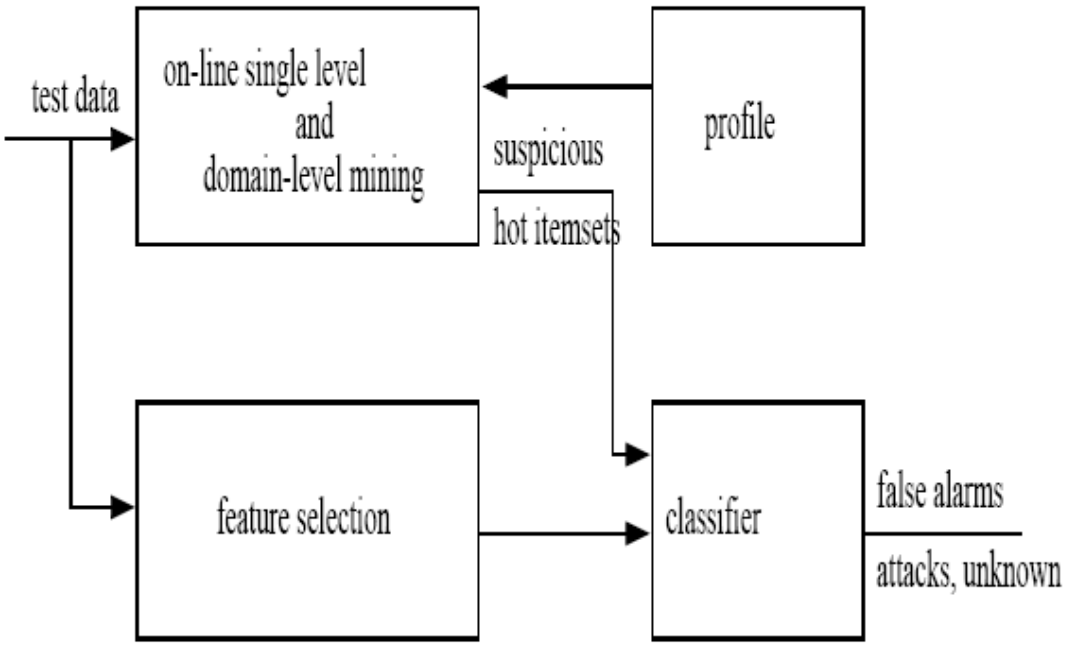

Figure 12: Discovering intrusion with ADAM (Phase 2) ([BCJ+01] page: 6)

In this phase the classifier is already trained and can categorize any attack as known, unknown or false alarm. Here also the same dynamic on-line algorithm is used to produce suspicious data and along with feature selection module, profile those suspicious data are sent to the trained classifier. Classifier then outputs which kind of attack it is. If it is false alarm then the classifier excludes those from the attack list and does not sent those to system monitoring officer.

So, as a conclusion we can say that this paper has shown a very effective way to use data-mining technique in that time. The main deficiency in the approach is that they used only association rules and as a result their classifier generated a lot of rules, among them many were redundant. They don't have any mechanism to avoid those redundant and irrelevant rules. *For example, suppose a rule is (A,B) → C, which means that is A and B occurs then C will occur. It already confirms that if B occurs then C will occur. But this algorithm will compute B → C also as a different rule, which means that this algorithm generates unnecessary extra rules.* But later, a lot of studies have been done on this

approach and many researchers introduced various measures (like interestingness, *I*) [MC03] [LSF+00] into their consideration and improved this model.

3.2.2. A Framework for Constructing Features and models for Intrusion Detection Systems (MADAM ID):

MADAMID is one of the well known IDS in this field. In this paper [LS00] their aim was to develop a more systematic and automated approach for building IDS. They have developed a set of tools that can be applied to variety of audit data [section 2.1.1] sources to generate intrusion detection models. The central theme of MADAMID approach is to apply data mining programs to the extensively gathered audit data to compute models that accurately capture the actual behavior or patterns of intrusions and normal activities. The main components of MADAMID framework include learning classifiers [Section 2.2.1] and meta-classifiers, association rules [Section 2.2.2] for link analysis and frequent episodes for sequence analysis. The process of applying MADAMID is as follows:

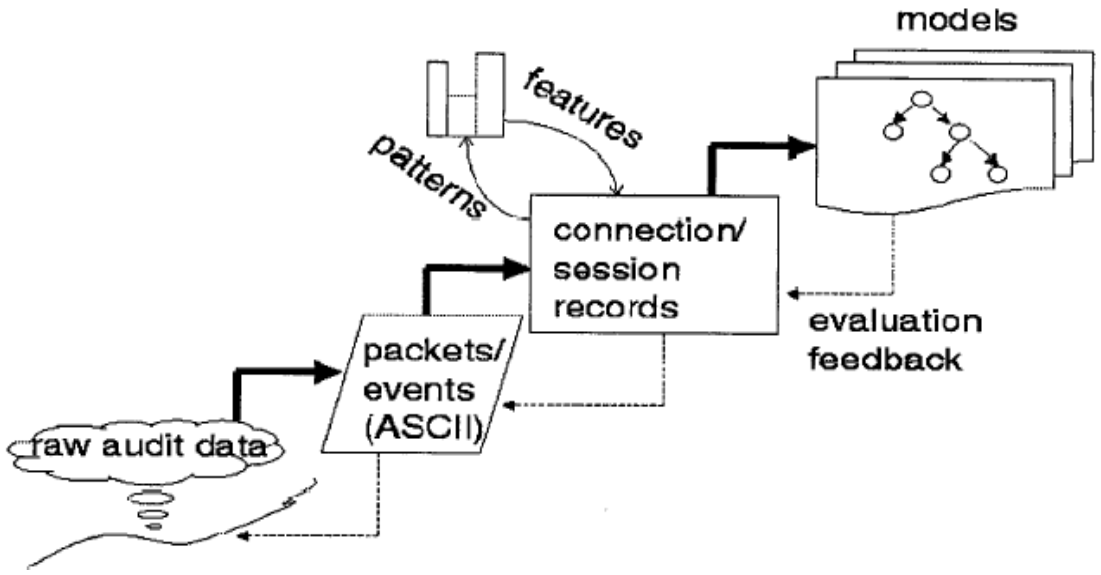

Figure 13: MADAMID workflow ([LS00] page: 231)

In the first step, raw audit data are gathered in binary format. Then they are processed into ASCII network packet information. For example, initially they were some bytes in 0 and 1. Then we convert those values into ASCII format, so that we can easily understand then. Suppose first 16 bit number indicates the source port number, so we convert the first 16 bit binary number into hex or decimal so that we can understand the source port number. After decoding all packet header information we then summarize them into connection records containing some basic features like service, duration etc. Various data mining programs like association rule, frequent episode rule are then applied into those connection records and as an output they got some derived features and then these derived features are used as rule in models. For example, suppose in connection records we have got that from the same source IP many packets are trying to access into many destination IPs but with same port. In packets/ event records step all these information were discrete and they are brought together on the basis of a certain time window (which was 5 minutes in their experiments) into the connection/ session records step. Then after applying data mining rules (association/ frequent episodes/ classification) into these records we come to know a feature that if above condition happens then, this might be an anomaly or attack and we get a rule describing this situation from this step. Then at last this rule is applied into the model. Since all these data mining methods (association rule, frequent episode rule, classification rule) are described with example in section 2.2.1 and feature construction is described in 2.1.2, they are not discussed here again.

Currently MADAM ID produces misuse detection models for network and host systems as well as anomaly detection models for users. The main strength in their paper is that they have focused on efficiency and automated the process of feature constructions. Their limitation is that their system is currently off-line and they are studying how to convert it into real time IDS because effective intrusion detection system should be real time system to minimize the security compromise. Another limitation in this model is that it computes only the frequent patterns of connection records. But many intrusions like those that embed all activities within a single connection do not have frequent patterns in connection data. These type of intrusions might go undetected in their model.

3.3. Recent and Current Research:

3.3.1. Learning Rules for Anomaly Detection (LERAD):

In this research [MC03] they have presented an efficient algorithm called LERAD (Learning Rules for Anomaly Detection). They presented it as an alternative of ADAM [BCJ+01] algorithm. The main difference between ADAM algorithm and LERAD algorithm is that in ADAM produces all possible association rules and relations and as a result the rate of false alarm is also very high. On the other hand LERAD produces fewer selected rules, which are free of redundancy and the false alarm rate is also lower than ADAM algorithm.

In this algorithm, we explain the algorithm along with an example rather than describing the theory. Suppose **S** is the sample of training data sets.

Port	Word1	Word2	Word3
80	GET	/	HTTP/1.0
80	GET	/index.html	HTTP/1.0
25	HELO	pascal	

Figure 14: Sample Training Dataset ([MC03] page: 602)

Here Port, Word1, Word2 and Word3 are four attributes.

Step 1: In this step we generate all possible rules and relations from the sample. The algorithm is as follows:

Repeat L times
    Randomly pick two instances S_1 and S_2 from S
    Set $A = \{a: S_1[a] = S_2[a]\}$ (*matching attributes*)
    For $m = 1$ to M and A not empty do
        Randomly remove a from A
        If $m = 1$ then create rule r_i = "$a = S_1[a]$"
        Else add $S_1[a] = a$ to r_i's antecedent
    Add r_i to rule set R

Figure 15: LERAD Algorithm ([MC03] page: 602)

In the first line of the algorithm, we randomly pick two instances S1 and S2 from sample S. So, S1 = {80, GET, /, HTTP/1.0} and

S2 = {80, GET, /index.html, HTTP/1.0}

Then we match the attributes of S1 and S2 in the second line. The matched attributes are (Port, Word1, Word3).

Then in the third line we start a loop. The loop goes 1 to M. Suppose in this case let M=4. Now we enter into the loop in the next line. Here we randomly choose Word1 as **a** from the list of attributes **A** and remove it from **A**. So now **a=Word1** and **A={Port, Word3}**. For the first time m=1 and we go inside the *if* statement and create a rule **r1: Word1=GET.** We add this rule to the ruleset.

The second time m=2 and attribute set "A" is not empty. So again we go inside the loop. This time we randomly remove another attribute "Port" as "a". So now, **a=Port, A={Word3}**. This time we go to the *else* part as m is not equal to 1. S1[Port] = 80 and we add this as antecedent of rule2 r2. So,

r2: if Port = 80 then Word1 = GET. We add this to the ruleset.

Third time still m < 4 and A is not empty. We randomly choose the only one attribute left Word3 and remove it from A. Now **a=Word3** and **A={ }**. Then we go to the *else* part. S1[Word3] = HTTP/1.0. We add this as antecedent of r3. **r3: if Port = 80 and Word3 = HTTP/1.0 then Word1 = GET.** We add this to the ruleset.

Fourth time m=4 and also A is empty. So we break the loop. This is how the algorithm generates rules and the whole process continues until generation of all rules. So finally in this step our ruleset is:

R = {

r1: Word1 = GET,

r2: if Port = 80 then Word1 = GET

r3: if Port = 80 and Word3 = HTTP/1.0 then Word1 = GET

}

Step 2: In this step we order those rules in decreasing order and remove any redundant rules. To sort these rules, we use a score of *n/r*, where n is the number of training instances satisfying the antecedent and r is the number of allowed value. The algorithm is as followed:

Update the consequents in R over S
Sort R by decreasing n/r
For each rule R_i in R in decreasing order of r/n
    Mark the values predicted by R_i
    If no new values can be marked, remove R_i

Figure 16: LERAD algorithm (Part 2) ([MC03] page: 602)

For example, in our case, after training over S and sorting by n/r these become:

- r2: if Port = 80 then Word1 = GET (n/r = 2/1)
- r3: if Port = 80 and Word3 = HTTP/1.0 then Word1 = GET (n/r = 2/1)
- r1: Word1 = GET or HELO (n/r = 3/2)

As an explanation let's see how n and r are selected of R3. Number of instances where word1 = GET or HELO is 3 and the allowed values are both of them which is 2. Let's see another example. For R2, the number of instances where the rule is matched is 2 (1st and 2nd rows of table) and the allowed value here is only GET, so r = 1. Here the arbitrary value of R2 ND R3 as same and anyone could be on the 1st place.

Removing redundant rule: R2 marks the two GET values in S. R3 would mark the same two values and no new values, so we remove it. R1 marks the HELO in the third instance in addition to the previously marked values, so we keep this rule.

So, we can see that a lot of algorithms are coming in recent days to improve the ADAM algorithm which was introduced in later research. All of them are assuming ADAM as an ideal and they are focusing on how to improve it.

3.3.2. Entropy-Based Intrusion Detection:

In this paper [Yo03] the author briefly describes two models- ADAM [BCJ+01] based data mining and Entropy based data mining [Yo03]. Then he compared both systems and showed us the advantages of his Entropy based system over the ADAM system.

A typical intrusion detection approach in ADAM is as follows:
1) First build a profile of normal or intrusion free behaviors of computer systems and networks. 2) Then the anomalous deviations from this normal behavior are considered as potential intrusions. The ADAM algorithm is used in the first step to mine association rules from the database. It finds all association rules that have support greater than user specified minimum support. Thus the ADAM algorithm finds common and typical event sequences as the system profile.

Although this [BCJ+01] is the mostly used and well studied algorithm, it always requires a very careful selection of minimum support parameter. *For example we have a database and there exists rule "A" which has 100 supporting data items with 200 contradictory data items and another rule "B" with 98 supporting data items with no contradictory data items. Now if we select the MinSupport to 100, then it will only select rule "A". To get the rule "B" we will have to set a smaller MinSupport value.*

So the limitations of the ADAM system are that it requires a careful selection of MinSupport. Because to analyze intrusions at large network or large e-commerce site the MinSupport has to be large enough to eliminate noise. Again to analyze intrusions at a small site, MinSupport has to be small enough to catch the traffic sized trails. One more point is that result of ADAM involves association rules with contradictions. So most of the time based of selection of MinSupport the result is noisy and requires post processing of results in order to use them in intrusion detection system. This is why Barbara used *classification rule learning system* in their "*ADAM*" system [BCJ+01].

So the author in this paper has proposed the use of "Entropy based Data Mining" method for intrusion detection in alternative of APRIORI based system. Mainly entropy based system is called more precisely as "*Graph Based Induction* ***(GBI)***" method for rule finding. The algorithm is as follows:

In this algorithm there are four sections and three steps. Four sections are: Extracted sub-graphs, Input graph, Contract input graph, Enumerate pair graph and select pair. Initially extracted sub-graph remains empty but as the algorithm runs it gets nodes representing association rules. In the example below we assumed that we already have two rules or edges: A: 4 ← 2 and

B: 1 ← 3

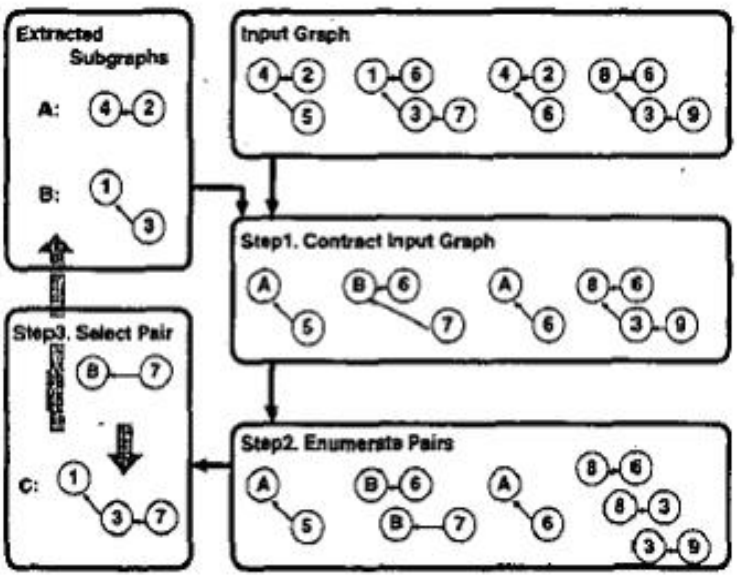

Figure 17: Workflow of Entropy based Intrusion Detection ([Yo03] page: 842)

In the first step, we convert the input graph to converted input graph according to the extracted sub-graphs. For example, from extracted sub-graph we know 4 ←2 represents "A". So, in the input graph wherever we found 4 ← 2, we replaced that with "A". We follow the same procedure for "B".

In step 2, the contracted graph is analyzed in all possible ways and is divided into every possible sub-graph called pairs. In this example 7 pairs have been extracted.

In step 3, we select the best pair according to our requirements or the pair that satisfies certain criteria. In this example B ←7 is selected and then it is extracted as 1←3←7 as B=1←3. Then this relation 1←3←7 will be added to the extracted sub-graph and will be used in the next iteration.

Starting from the empty set of extracted sub-graphs, this algorithm can extract various sub-graphs which appear frequently in the input graph. By repeating this it extracts more complex sub-graphs in a step by step manner.

The advantage of this algorithm is that in step 3 if we select pairs those appear more than MinSupport, then this algorithm works as APRIORI algorithm and if in step 3 the best pair is selected based on entropy based index then this can be viewed as an extension of conventional classification rule learning algorithm. Although in APRIORI based systems and classification rule based systems they use attribute values table representations but in GBI this thing can be implemented by viewing root nodes (4, 1, 4, 8) as attribute classes and the value of the attributes can be seen as connecting nodes (2, 5, 6, 3, 7, 2, 6, 6, 3, 9).

Their proposed work is still under research but they have showed a very easy alternative way to APRIORI system for mining rules.

3.3.3. MINDS – Minnesota Intrusion Detection System:

MINDS is one of the most popular NIDS in current days. This system [ELK+04] was developed at department of computer science in University of Minnesota in 2002. University of Minnesota is using this system into their network from 2002 and they were capable of detecting many new attacks as they were launched (examples include "slammer worm", "NetBus worm" etc).

There are two kinds of anomaly detection techniques, which are supervised and unsupervised anomaly detection. In supervised anomaly detection, given a set of normal data to train on and given a new set of test data and the goal is to determine if the test data is normal or anomalous. In unsupervised anomaly detection system the model attempts to detect anomalous behavior without using any knowledge about the training data. Unsupervised anomaly detection systems are based on statistical approaches, clustering, outlier detection schemes etc. This MINDS [ELK+04] is a kind of unsupervised anomaly detection system.

MINDS uses a suite of data mining techniques to automatically detect attacks against computer networks and systems. The long term objective of MINDS is to address all aspects of intrusion detection. In this paper they have presented details of two specific contributions: (1) an unsupervised anomaly detection technique that assigns a score to each network connection that reflects how anomalous the connection is, and (2) an association pattern analysis based module that summarizes those network connections that are ranked highly anomalous by the anomaly detection module.

The workflow of the MINDS is described below in the figure. We will describe the system with the workflow step by step.

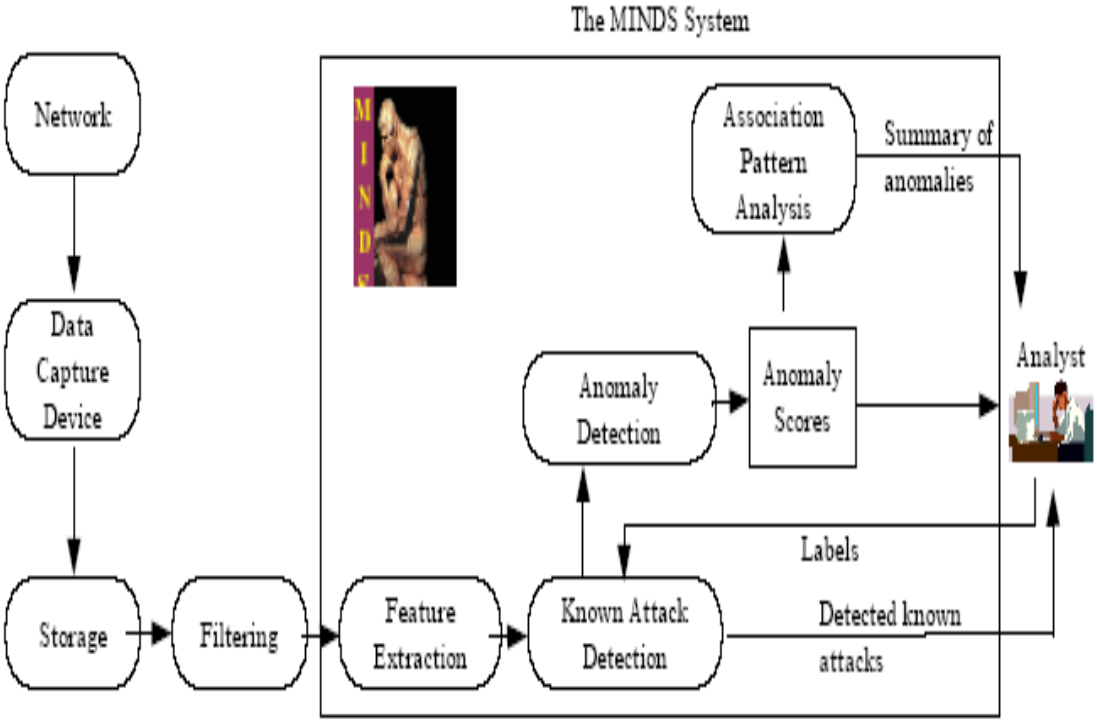

Figure 18: MINDS System ([ELK+04] page: 4)

Input to the MINDS is Netflow version 5 data collected using flow tools (for details refer to (www.splintered.net/sw/flow-tools) which are an alternative to tcpDump data. Flow-tools only capture packet header information, not the message contents. Just like tcp dump data header information contains source ip, source port, destination ip, destination port, time stump, flag values, duration of the connection etc. They have used 10 minutes time window. All data in the internet are passed as packets. All these packets have some header information and data. The system only captures the header information for all of those packets those have passed in last 10 minutes. Those data are stored and before they are fed into the main system a data filtering step is performed to remove network traffic that the analyst is not interested in analyzing. For example, filtered data may include traffic from trusted sources. For example, in University of Windsor, when an access request to port numbers between 40000 to 60000 comes from UofW campus network it is granted otherwise if the source IP is not from university network the access is denied.

Step 1: Feature Construction

Then we enter into the main system. The first step in MINDS is "feature extraction". The data came are in the binary format but we know the format (which bytes are representing what) and we extract those basic features from the audit data. These basic features include source and destination IP address, source and destination ports, protocol, flags, number of bytes and number of packets. With these basic features then derived features are computed. There are two types of derived features, (1) time window based features and (2) connection window based features. Time window based features are constructed to capture connection with similar characteristics within last T seconds. For example, how many connections were destined towards the same destination IP address in last T seconds is called count-dest. Connection window based features are constructed to capture connection with similar characteristics within last N connection. For example, within last N connection how many connection were destined towards the same destination IP address is called count-dest-conn. Sample features of both type features are presented below in the figures.

Feature name	Feature description
count-dest	Number of flows to unique destination IP addresses inside the network in the last T seconds from the same source
count-src	Number of flows from unique source IP addresses inside the network in the last T seconds to the same destination
count-serv-src	Number of flows from the source IP to the same destination port in the last T seconds
count-serv-dest	Number of flows to the destination IP address using same source port in the last T seconds

Figure 19: Time-window based features ([ELK+04] page: 4)

Feature name	Feature description
count-dest-conn	Number of flows to unique destination IP addresses inside the network in the last N flows from the same source
count-src-conn	Number of flows from unique source IP addresses inside the network in the last N flows to the same destination
count-serv-src-conn	Number of flows from the source IP to the same destination port in the last N flows
count-serv-dest-conn	Number of flows to the destination IP address using same source port in the last N flows

Figure 20: Connection-window based features ([ELK+04] page: 5)

Step 2: Known Attack Detection

After we have derived all features of connection then the next step is to compare those features with known anomalies. If we found a match then we directly sent it to the analysts. For example, suppose we got to know from time-window based features that one single source IP is trying to access to same port in many destination IPs many times within the last 3 seconds and if there is existing signature of this kind of attack then we can send it to analyst as an attack without any hesitation. Now if there is no known attack signature of this kind then we send that connection record to Anomaly Detection module, which will be the next step.

Step 3: Anomaly Detection

In this step Anomaly Detection module will use an outlier detection algorithm to assign an anomaly score to each network connection. It assigns a degree of being an outlier to each data point, which is called Local Outlier Factor (LOF) [BKN+00]. For each data example, the density of the neighborhood is first computed. The LOF of a specific data example *p* represents the average of the ratios of the density of the example *p* and the density of its neighbors. LOF requires the neighborhood of all data points be constructed. This involves calculating pairwise distances between all data points, which is $O(n^2)$ complexity. As there will be million data sets, the complexity will be huge. To reduce the complexity an approach has been taken in MINDS. They have made a sample dataset from the data and all data points are compared with the small set, which reduces the complexity to O(n*m), where m is the size of small dataset.

Cluster *C1*	Cluster *C2*
Sample dataset, *p1* in C1	Sample dataset *p2* in C2
	Sample dataset *p4* in C2
Sample dataset, *p3* in C1	Sample dataset *p5* in C2

For example, in the above table we can see that cluster C2 is denser than C1. Due to the low density in cluster C1, for most examples q inside C1, the distance between any dataset and its neighbor is greater than that of C1. For example, the distance between p1 and p3 is higher than the distance between p2 and p4. So, therefore p2 will not be considered as outlier.

Step 4: Association Pattern Analysis

After assigning each connection a score then top 10% scores are taken as anomaly class and bottom 30% scores are taken as normal class. Middle 60% scores are ignored in their system. Then these scored connections are passed into the Association Pattern Generator.

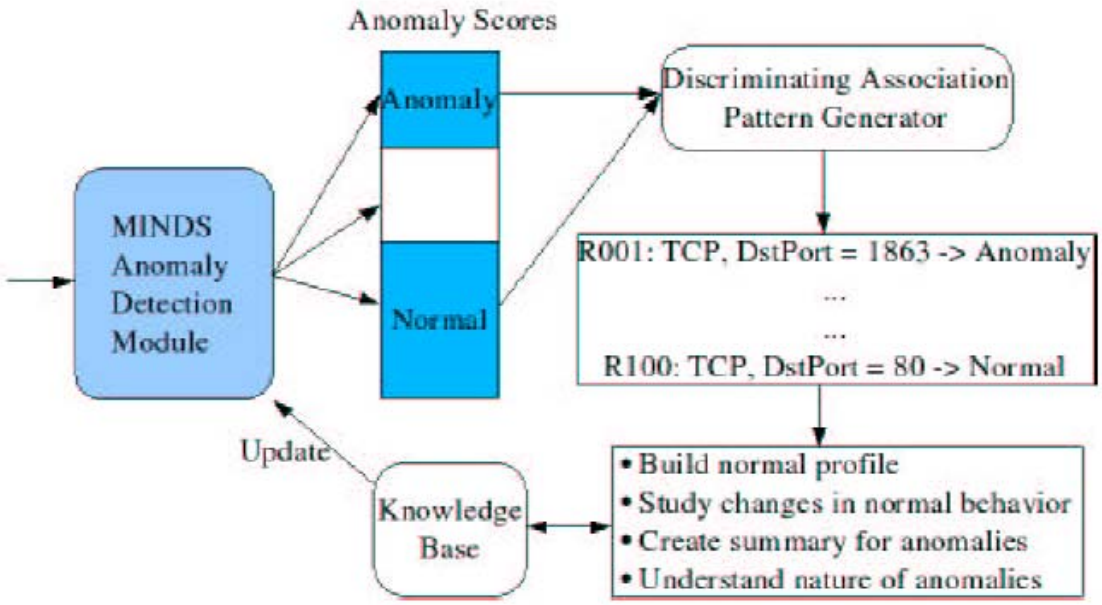

Figure 21: MINDS Association Analysis Module ([ELK+04] page: 14)

This module summarizes network connections that are ranked highly anomalous by the anomaly detection module. The goal of mining association patterns is to discover patterns that occur frequently in anomaly class or in normal class. In this step they have applied association rule [Section 2.1.3.2] to construct rulesets for anomaly class and for normal class. For example, scanning activity for a particular service can be summarized by a frequent set:

sourceIP=X, destinationPort=Y

If most of the connections in the frequent set are ranked high by previous step, then this frequent set may be a candidate signature for addition to a signature-based system. Or, if the following frequent set is scored lower and appeared many times then we can say it normal which is a web browsing activity.

Protocol=TCP, destinationPort=80, NumPackets=3…6

Their system also finds discriminating patterns. For finding discriminating patterns they have used some measures like ratio, precision, recall and harmonic mean of precision and recall. By these methods this module orders the patterns and groups similar patterns

together and presents them in front of the analyst. The measures for ordering patterns are described briefly below:

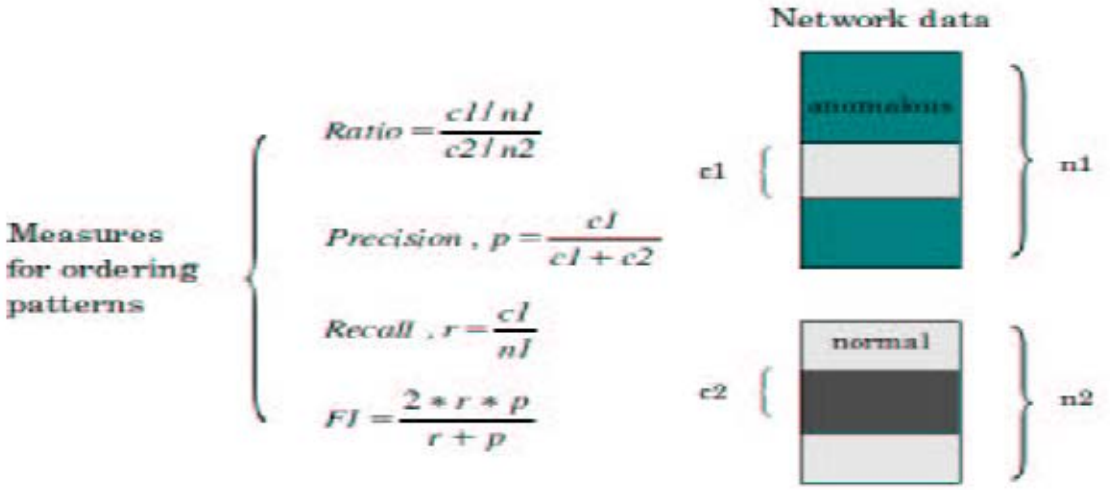

Figure 22: Measures for ordering patterns ([ELK+04] page: 13)

Consider a set of features X that occur c1 times in the anomalous class and c2 times in the normal class. Also let n1 and n2 be the number of anomalous and normal connections in the data set. Assuming that we are only interested in finding profiles of the anomalous class, the ratio c1/n1 to c2/n2 would indicate how well the pattern could distinguish anomalous connections from normal connections. Ratio or precision alone is insufficient because they often characterize only a small number of anomalous connections. In the extreme case, a rare pattern that is observed only once in the anomalous class and does not appear in the normal class will have a maximum value of ratio and precision, and yet, may not be significant. To account for the significance of a pattern, the recall measure can be used as an alternative. Unfortunately, a pattern that has high recall may not necessarily be discriminating. The F1-measure, which is the harmonic mean of precision and recall, provides a good trade-off between the two measures.

Then in the last step summary of all rules are presented in front of analyst and then analyst can update or build normal profile or can label a new attack signatures. This is how the MINDS woks as an unsupervised anomaly detection system.

4. TAXONOMY OF DATA MINING BASED NIDS:

System	Misuse/ Anomaly	Based On Alarms	Mining Technology Used	Training data Used	Strength	Limitations
ADAM	Anomaly	No	Association rule, classification rule	TCPDump	Leading research	Produce redundant rules, require careful selection of min support
MADAMID	Misuse and anomaly	No	Association rule, classification rule, frequent episodes rule	TCPDump, BSM generated data	Automate feature construction, efficiency	Off-line system, Computes only the frequent patterns of records
MINDS	Anomaly	No	Outlier, association rule	Netflow Version 5	Anomaly Scores made it easy to determine anomalies	Doesn't consider middle 60% of data which might contain important rules
Custom ID Filters	Anomaly	Yes	Association rule, frequent episodes	Sensor generated data	Can be augmented with any system	Require preparing customized filters for different

						environment
RTID Alarms	Anomaly	Yes	Association rule	Alarm context history	Reduce false alarm rate	
LERAD	Anomaly	No	Association, Classification	TCPDump	No redundancy	Careful selection of min support
Entropy based	Anomaly	No	Entropy/ Graph based	Binary audit data	Alternative way of ADAM	Require careful selection of min support

5. CONCLUSION:

Although network intrusion detection is not a new field and it has been researched since 1987 but applying data mining technique in this field is relatively very new. One of the first and ideal models in this field was ADAM [BCJ+01] which is an apriori-like algorithm. One group of researchers from Columbia University also has done a lot of research on this field and improved accuracy, efficiency and usability of data mining based NIDS [LSC+01a] [LSC+01b] [Le02] [LFM+00] [LS98] [LSF+00] [LSM98] [LSM99]. Another group of researchers concentrated on how to reduce the false alarm rate but they were few in number [MCH00] [CG00] [LCE+01].

Acknowledgement:

I would like to thank my instructor Dr. R. Frost and my supervisor Dr. C. Ezeife for their continuous guidance, direction and encouragement throughout the period of writing this report. Their detailed comments and report expectations urged me to greatly improve the quality of the report, and have given me a deeper understanding and appreciation for research.

Appendix I

Annotations of 20 important papers:

1. [AJ 01] R Agarwal, MV Joshi; PNrule**:** A New Framework for Learning Classifier Models in Data Mining**;** *on First SIAM International Conference on Data Mining, 2001;*

The authors in this paper have introduced a new framework for learning classifiers. They have introduced Positive rules (P-rules) and Negative rules (N-rules). P-rules predict the presence of a class and N-rules predict the absence of a class. Their system is learned in two phases: the first phase discovers a few P-rules that capture most of the positive cases for the target class while keeping the false positive rate at a low level. In the second step it discover a new N-rule that removes most of the false positives introduced at the first level. In order to validate their framework authors have taken part into a classifier learning competition, which was KDD-CUP 99 Classifier Contest. The authors also claimed that their system was checked against a huge dataset consisting of 5 million records each with 41 attributes and their system was best in regard of other 22 systems, which also took part into that competition. Their paper was later cited by 26 groups of researchers, some of them include D Ourston, S Matzner, W Stump, B Hopkins in 2003; M Sabhnani, G Serpen in 2003; D Musicant, V Kumar, A Ozgur in 2003 etc.

---

2. [B 00]Tim Bass; Intrusion Detection System and Multi-sensor Data Fusion; *Communication of the ACM, pp 99-105, vol. 43, no. 4, April 2000*

In this paper at the first section the author discussed about the evolution of the current ID system and their lacking. Then he introduced multi-sensor data fusion and at last he presented us a model where data mining and multi-sensor data fusion worked together to detect intrusion.

According to the author currently most of the ID system is based on examining system audit trails or network traffic. That means we have some known pattern or anomaly statistics in our own host and any incoming input is examined against those previously known patterns.

In author's opinion, this current system is the modified form of Denning's model in 1987. This model was built on host based subject profiles, system objects, audit logs, anomaly records and activity rules. The model was basically a *rule based pattern matching system*. The second approach was based on multi host network model. It was capable of monitoring the traffic analysis with Network Security Monitor.

The model, developed by the author in this paper mainly enhanced the collection of anomaly and intrusions based on previous intrusions and anomalies by data mining . So, in future those types of intrusions happen then the system can detect those intrusions.

---

3. [BBH+ 02] Jerzy Bala, Sung Balik, Ali Hadjarian, BK Gogia and Chris Manthorne; Application of a Distributed Data Mining Approach to Network Intrusion; *Proceeding of the first international joint conference on Autonomous and multiaqent Systems: Part 3, July 2002*

The problem identified by the authors in this paper is that they claimed that the current model of workflow in data mining based IDS (Wenke Lee, Salvatore J. Stolfo, 1998) is not very much realistic as that system collects all the data from different hosts and process them in a centralized monitoring machine. In this paper the authors have presented an application of distributed data mining approach in NIDS. In context to their approach the authors have referred to the works of Ingram, H. Kremerm, Steven Rowe, Neil C. in 2000, Neumann, Peter Porras, Phillip A. in 1999 and Sobirey, Michael Richter, Birk.

In this paper the authors have emphasized distributed data mining techniques to network intrusion detection. They mainly worked on TCP dump packets with a limited set of features. They conducted their experiment to build the classification models for five class types and to match the testing data against the classification models.

For their experiment they have used about 4 gigabytes of a compressed binary TCP data dump from seven weeks of network traffic as input. This was processed into about 5 million connection records. The output was labeling each connection as either normal or attack.

They have divided different features sets into 3 categories. This is like, as they have partitioned the database vertically, each sub-database is handling with different type of features set of connection records. Those three sets are: basic features of individual TCP connection, content features within a connection suggested by domain knowledge and traffic features computed using a two-second time window.

Then for building classification models for network intrusion they categorized each connection as one of the following five types: Normal: This set includes the normal, attack free data; DOS: denial-of-service. Example: SYN flag attack; R2L: unauthorized access from a remote machine. Example: guessing a password; U2R: unauthorized access to local super user (root) privileges. Example: "Buffer Overflow" attacks and Probing: surveillance and other non-harmful probing to gather data. Example: scanning port.

Later this paper was cited by 2 papers/ group of researchers: K Liu, H Kargupta, J Ryan in 2004 and ST Brugger, A Hurst, K Mehl in 2003.

---

4. **[BCJ+ 02]** Daniel Barbara, Julia Couto, Sushil Jadodia, Ningning Wu ; ADAM: A Testbed for exploring the Use of Data Mining in Intrusion Detection; *ACM SIGMOD RECORD (4): Special Selection on Data Mining for Intrusion Detection and Threat Analysis*

In this paper authors have described some existing ID systems in the first part and in the second part they have shown how data mining can be useful in this field. To do so, they have introduced **ADAM** (Audit Data Analysis and Mining). In ADAM they have introduced an improved and dynamic version of association rule.

At the first section authors have identified two kinds of ID systems currently in use. One kind is signature-based IDS; example includes P-BEST (U. Lindqvist, P.A. Porras, 1999) USTAT (K. Ilgun, 1992) and NSTAT (P.A. Porras, 1992). Another kind of IDS is statistics and data mining-based IDS (example includes IDES, NIDES (D. Anderson and T. Frivold and A. Valdes) and EMERALD (P.A. Porras and P.G. Neumann, 1997), Haystack (S. Smaha, 1990) and JAM (W. Lee and S.Stolfo and K. Mok, 1999).

The authors claimed that ADAM uses a combination of association rules and classification rule to detect any attack in a TCPdump audit trails. First, ADAM collects normal, known frequent datasets by mining into this model. Secondly, it runs an on-line algorithm to find last frequent connections and compare them with the known mined data and discards those that seem to be normal. With the suspicious ones it then uses a classifier, which is previously trained to classify the suspicious connections as known type of attack, unknown type of attack or a false alarm.

The authors have conducted their experiments in two phases. In the first phase they trained the classifier. This phase took place only once offline before using the system. In the second phase they use the trained classifier to detect intrusions.

The authors conducted their experiment on DARPA 1999 datasets. Comparing the other IDS system their system did extremely well and their system ranked third. The authors claimed that ADAM was successful at detecting DOS, PROBE, U2L attack but their system could not detect U2R attacks.

Later this paper has been cited by 34 groups of authors, some of them include R Agrawal, J Kiernan, R Srikant, Y Xu, 2002; Y Chen, A Abraham, 2005; P Ning, S Jajodia, 2003; S Peddabachigari, A Abraham, C Grosan, J Thomas, 2005 etc.

---

5. [BK 03] Yuebin Bai and Hidetsune Kobayashi.; Intrusion Detection Systems: Technology and Development.; *Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on, 27-29 March 2003*

The paper mainly presented a surveyed report on current intrusion detection systems. The main focus of this paper is the various intrusion detection systems currently in use, their advantages and disadvantages. The paper focused on both host based and network based intrusion detection systems.

Network based Intrusion Detection System obtains data by monitoring the traffic in the network to which the hosts are connected. The system gets those traffic data by sensors, which can be internal or external. As described by the author the basic properties of NIDS are as less cost, real time detection and response to the attacks, ability to detect unsuccessful attack attempts and finally operating system independence.

The author also classified the source of incoming traffic data as internal and external sensors and the paper described the advantages and disadvantages of these sources in detail. Then the paper described two current techniques used in network intrusion detection systems: Misuse and Anomaly intrusion detection. According to the author misuse detection system is based on signature patterns whereas anomaly detection system uses some kind of statistical/ data mining approach. The author discussed those techniques in details with advantages and disadvantages of both. The important remark he made about those techniques is that current systems use both techniques combined for better efficiency. Finally the author discussed some novel developments in NIDS categorized as data mining based IDS (Wenkee Lee, 1999; Wenkee Lee and Salvatore J. Stolfo, 2000) and data fusion based IDS (Waltz, E., 1998; Tim Bass, 2000).

Later this paper has been cited by four papers (Nong Ye, Qiang Chen and Borror, C.M., 2004; Krishan Kumar, R.C. Joshi and Kuldip Singh, 2005; Garon Munch; and Ya-lin Chen, 2005).

---

6. [CG 00] Chris Clifton, Gary Gengo**;** Developing Custom Intrusion Detection Filters Using Data Mining; *MILCOM 2000. 21st Century Military Communications Conference Proceedings, Volume: 1, 22-25 Oct. 2000. Pagees: 440 – 443 Vol. 1*

According to the authors in this paper, a lot of normal behaviors are triggered as attacks by false alarms. The main concern in this paper is that, if we can recognize normal patterns and identify them in alarm stream then we can easily filter out them and can decrease the rate of false alarm in a huge. The difficulty with this approach is building these filters and defining normal patterns. It also needs a considerable human effort. To reduce this effort they have tried to use data mining techniques.

They have developed filters based on sequences of alarms. The idea is that, a sequence of operations that are normal in an environment may contain some operations, which look like intrusions but really not intrusions cause false alarms. It is unlikely that a complete sequence of normal operations will be duplicated in an intrusion. So, alarms that are part of complete normal sequence can be ignored.

In their experiments they have analyzed over one million intrusion detection alarms gathered from seven machines in a network. The time period of their experiment was 2 weeks. They have loaded the logs into a relational database. The basic schema of the log is like Log(Event, FromIP, ToIP, time). Then they have used an algorithm called *"Query Flocks"*, which is basically a generalization of association rule mining algorithm to detect the frequent sequences. Query Flocks algorithm gives some more flexibility in handling complex patterns than Frequent Episodes algorithm.

---

7. **[D 87]** Denning, D. "An Intrusion Detection Model". *IEEE Transactions of software engineering SE – 13, 2. (Feb 1987), 222 – 232.*

The author of this paper was the legend in IDS and this paper was the very first paper introducing the idea of intrusion detection model in 1987. According to the author of this paper his model is based on the hypothesis that security violations can be detected by monitoring system's audit records for abnormal patterns of system usage. The author's model has six main components: subjects, objects, audit records, profiles, anomaly records and activity rules. According to the author the system can be regarded as a rule-based pattern matching system. When an audit record is generated, it is matched against the profiles. Type information in the matching profiles then determines what rules to apply to update the profiles, check for abnormal behavior, and report anomalies detected. The author also claimed that the model is independent of any particular system, application environment, system vulnerability, or type of intrusion, thereby providing a framework for a general-purpose Intrusion Detection Expert System (IDES).

---

8. **[EEL+ 03]** Levent Ertoz, Eric Eilerston, Aleksander Lazarevic, Pnag-Ning Tan, Vipin Kumar, Jaideep Srivastava, Paul Dokas. "MINDS – Minnesota Intrusion Detection System", 2003.

According to the authors, MINDS uses a suite of data mining techniques to automatically detect attacks against computer networks and systems. The long-term objective of MINDS is to address all aspects of intrusion detection. In this paper they have presented details of two specific contributions: (1) an unsupervised anomaly detection technique that assigns a score to each network connection that reflects how anomalous the connection is, and (2) an association pattern analysis based module that summarizes those network connections that are ranked highly anomalous by the anomaly detection module.

The authors described the model in four steps. In the first step they have constructed features. In the second step their system can detect all the known attacks based on its

signature profile. In third step, Anomaly Detection module will use an outlier detection algorithm to assign an anomaly score to each network connection. The last step is association pattern analysis where anomaly rules are formed from the top 10% anomalous score from the previous step. The authors have also claimed that they have reduced the complexity of step 3 from n*n to n*m where n is the size of data and m is the size of sample.

---

9. [EMZ+ 00] Eleazar Eskin, Matthew Miller, Zhi-Da Zhong, George Yi, Wei-Ang Lee, Sal Stolfo. "Adaptive Model Generation for Intrusion Detection Systems". *Workshop on Intrusion Detection and Prevention, 7th ACM Conference on Computer Security, Athens, GR: November, 2000*

In this paper the authors have presented an Adaptive Model Generation algorithm. The problem they have identified in current IDS is that building a training dataset requires a significant cost in deployment of data mining based IDS. To reduce that cost authors have introduced their model, which can build training dataset dynamically on the fly in real time. Authors have also claimed that their system is able to build anomaly detection model over noisy data.

Authors have proposed an IDS architecture that contains three components, a sensor, a detector, and an adaptive model generator. The sensor feeds formatted data to the detector for analyzing and responding to occurring intrusions, and also sends data to the adaptive model generator for learning new detection models. The adaptive model generator, upon learning a new model, feeds this model to the detector. The detection models are periodically updated by the system automatically as more data is collected.
This adaptive model generation model can be used as a black box into other systems.

They have evaluated their adaptive anomaly detection system using data from the DARPA 1999 dataset for Intrusion Detection Evaluation. They have showed that the

performance of the models improves as the system collects more training data. For measuring performances they have used ROC curves.

---

10. **[GR 02]** Giacinto, G.; Roli, F.; Intrusion detection in computer networks by multiple classifier systems; Pattern Recognition, 2002. Proceedings. 16th International Conference on Volume 2, 11-15 Aug. 2002 Page(s):390 - 393 vol.2

The authors of this paper have identified from the previous classifiers that majority of them was formulated by collecting all the available features in a single feature vector and then performing the classification task. On the other hand, the authors in this paper have identified three different sets of features. The authors have subdivided the classification problem into three smaller problems where each one was related to one feature set. According to the authors, this will accelerate the classification task with greater efficiency. Those three sets of features are: Intrinsic feature, Traffic feature and Content feature. Then the solutions of these smaller problems can be fused by Multiple Classifier System (MCS). The authors have developed this system over the system developed by J. Kittler and F. Roli in 2001.

The authors have conducted experiment of their system on a subset of the database created by DARPA in the framework of the 1998 Intrusion Detection Evaluation Program. From the experiment results, the authors have claimed that, multiple classifier approach based on distinct feature representation seems to be more suited solution for implementing an IDS.

---

11. **[ISA 93]** T. Imielinski, A. Swami, R. Agarwal. "Mining association rules between sets of items in large databases". *In proceeding of the ACM SIGMOD conference on management of data, Washington D.C. May 1993.*

This paper was not directly on Data Mining based IDS, but this paper introduced the association rule concepts and how to use it in data mining field. This concepts was later

adopted by almost all algorithms that implements association rule based IDS which include ADAM by Barbara in 2001; Matthew V. Mahoney**,** Philip K. Chan in 2003; Yoshida, K in 2003; Wenkee Lee in 2002 and many more. The author in this paper mainly discussed how association rule algorithm works and how it can be implemented in industry level applications for automation of discovering of new and hidden features based on its previous behaviors or transactions.

---

12. **[J 05]** Jensen, Kenneth G.; A combined association rule/radial-basis function neural network approach to intrusion detection; M.Sc Thesis, Utah State University, 2005.

In this paper the author have introduced a combination of association and classification rule algorithm to detect intrusion in neural network. The proposed methodology is based on and addition to the algorithm of Han, J and Yin, X in 2003. The author claimed that in his algorithm he has improved the previous work by generating a smaller set of high quality rules, removing redundant rules by evaluating candidate rules and using the best k rules instead of the best one rule to classify an example. The algorithm uses greedy approach to generate rules that cover the positive examples in comparison to the negative examples. The ruleset is continually added until all positive examples are covered. By using Predictive Rule Mining (PRM) the author has minimized the removal of too many rules. Upon completion of rule generation process an evaluation of rule accuracy process has been used and in this case the author has showed the use of lace expected error estimation. According to the author by building multiple similar rules the search becomes exhaustive. Then the author's method tries to find the balance between exhaustive and greedy approaches. To validate his method the author has used the 1999 KDD CUP competition dataset and claimed that his algorithm performed comparably well against the other algorithms that classified all attack categories.

---

13. **[L 02]** Wenkee Lee; Applying Data Mining to Intrusion Detection: the Quest for Automation, Efficiency, and Credibility; *ACM SIGKDD Exploration News-letter, vol. 4, issue 2, December 2002, pages 35-42*

This paper mainly focuses on applying Data Mining techniques to build intrusion detection models. These include extracting and constructing features from audit data, computing efficient models, and improving the usability of the detection models.

The authors have improved three different aspects of IDS: automation, efficiency and credibility. For the automation they are using customizing association rules algorithm. Because the basic algorithm can generate many rules that are irrelevant. So, they are introducing *"interestingness measure"*. They also divided the attributes of data into two categories. Some attributes are essential in describing the data while others only provide auxiliary information. The essential attributes are called *axis attributes* and others are called *reference attributes*.

To develop the efficiency of the model they have introduced the "Cascaded Detection Model". The main idea was to use sequentially more complex modules. At the lower level the module would be simpler. The full dataset will be passed through the lower part to the higher part. At the lower part the unnecessary data will be filtered out and if it is interesting then it will be passed to the next complex level for further analyze.

To improve the credibility they have shown a layered IDS architecture but yet it is under research and they haven't got any result. In this paper they just showed the opportunity of Data Mining to improve the credibility of ID system.

---

14. **[LSC+ 01]** Wenkee Lee, Salvatore J. Stolfo, Philip K. Chan, Eleazer Eskin, Wei Fan, Matthew Miller, Sholmo Hershkop and Junxin Zhang; Real Time Data Mining-based Intrusion Detection; *Proceedings of DISCEX ll, June 2001*

In this paper authors have mainly focused into three important aspect of current ID system and they provided ways to strengthen those aspects. Those three areas are *accuracy, efficiency, and usability*.

In typical applications of data mining in ID, detection models are developed in such a way that they compare or match any incoming data with previous anomaly after the incoming dataset is complete. In a huge system when there are continuous flow of data it becomes a severely slow when the model has to wait for the incoming datasets to be completed. So, they have introduced *Cost-Sensitive Modeling*. In this model they have categorized feature used in for network IDS into 4 cost levels and also estimated an arbitrary cost in proportion.

To improve the usability authors have mainly focused into two problems: First, management of both training data and historical datasets are very difficult task, especially if the system handles too many types of data. Second, after new data has been analyzed then the system or the database needs to be updated. To ease these two problems they have introduced *adaptive learning*. In adaptive learning model they are using flexible model which is algorithm independent model. In reality, when a new type of intrusion is discovered, it is very desirable to be able to quickly adjust an existing detection system to detect new attacks, even if the adjustment is temporary and may not detect all occurrences of new attacks. So, they proposed to use a plug-in model, which can be easily replaced by the old model. They efficiently generated a light-weight classifier just for the new pattern. The existing models remain the same. When the old model predicts some anomaly, this data record is sent to the new classifier for further classification. The final prediction is a function of the predictions of both the old classifier and the new classifier

15. **[LXY 03]** Quing-Hua Li, Jia-Jun Xiong, Hua-Bing Yang; An Efficient Algorithm for Frequent Pattern in Intrusion Detection; *Machine learning and cybernatics, 2003 International Conference on, Volume 1, 2-5 Nov. 2003. Pages: 138-142 vol.1*

The problem addressed by authors is to find frequent episodes based on minimal occurrences introduced in Mannila H, Toivonen H. in 1996. In this paper they have improved and extended the basic association rule algorithm to detect network intrusion detection.

To solve the addressed problem of eliminating redundant rules they have improved the basic association rule and introduced DHP (Direct Hashing and Pruning) algorithm. The basic difference between the basic and improved algorithm is that in the improved version it doesn't create the unnecessary rules those are already exist, more precisely it excludes those unnecessary subsets. For this purpose the authors have introduced another factor "Interestingness" measures on patterns that includes information on the attributes in addition to "support" and "confidence" in association rule algorithm.

The authors have claimed that their algorithm improved the association rule mining algorithm (Aggarwal, R., 1993 and 1994) with great efficiency and their rule formalism not only eliminates irrelevant patterns but also provides rich and useful information about the audit data.

---

16. **[MC 03]** Matthew V. Mahoney, Philip K. Chan.; Learning Rules for Anomaly Detection of Hostile Network Traffic; *Data Mining, 2003. ICDM 2003. Third IEEE International Conference 2003.*

The problem addressed by the authors in this paper is the high false alarm rate in implementing APRIORI (Aggarwal, R., 1993) algorithm in NIDS. In this paper the authors have presented an efficient algorithm called LERAD (Learning Rules for Anomaly Detection). They presented it as an alternative of APRIORI algorithm. The main difference between APRIORI algorithm and LERAD algorithm is that in APRIORI

produces all possible association rules and relations and as a result the rate of false alarm is also very high. On the other hand LERAD produces fewer selected rules, which are free of redundancy and the false alarm rate is also lower than APRIORI algorithm.

The algorithm they introduced in this paper has two parts. In the first part the algorithm generates all the rulesets. Initially the rule starts from one condition but as the loop goes on new antecedents are added to the rule forming a complete rule. In the second step the algorithm orders those rules in decreasing order and removes any redundant rules. To sort these rules, the authors have used a score of n/r, where n is the number of training instances satisfying the antecedent and r is the number of allowed value.

The authors have evaluated their system with an experiment with two datasets: one is 1999 DARPA dataset and another is 623 hours of traffic collected from university departmental-server over 10 weeks of time. The authors claimed that LERAD detected 64% of TCP attacks on DARPA 1999 dataset and 40% of University dataset with only 10 false alarms per day. The authors also claimed that LERAD could process 10000 packets or 3500 TCP sessions per seconds on a 750 MHz PC. They also claimed that 8.9 GB of DARPA 1999 dataset was filtered in 7 minutes and was processed in under 2 minutes with LERAD. Later this paper was referred/ cited by 18 papers.

---

17. [MCZ+ 00] Stefanos Manganaris**,** Marvin Christensen, Dan Zerkle, Keith Hermiz.; A Data Mining Analysis of RTID Alarms; *Computer Networks vol. 34, issue 4, October 2000, pages 571-577*

In this paper authors have tried to improve anomaly detection on IBM's Network Operation Center (NOC), which uses misuse detection. Generally IDS generate a lot of alarms everyday. In this paper they developed an algorithm where system would learn to detect anomalies based on its experience from the history of its alarms. Mainly they are applying data mining techniques (association rule) on the alarm context history. The authors of this paper also focused on sensor profiling.

As each sensor has its own history of alarm behavior and they also vary by alarm types, alarm rates, distribution of alarm over day-of-week, time-of-week, so, in this paper they tried to generate each sensor's own behavior table and later they applied it to customer market segmentation field. In this experiment they used 27 NetRanger sensors for one month time and collected roughly 12x10^6 alarms, corresponding to about 2GB data. Each alarm was described in its own type, priority level, time stamp, source and destination IP address and port and customer and sensor ID.

They were able to cluster those sensors based on these data. It showed that one group of sensors looking at internet traffic, one grouped together the sensors of a particular client suggesting a very abnormal network and another grouped together with those sensors those seem to need improvement by tuning. The last group showed that this strategy can be successfully applied in servicing the faulty sensors.

The algorithm the used to cluster those sensors is very simple. At each step the algorithm decides whether to assign a new vector (Sensor profile) to an existing cluster or to create a new cluster. And this process goes until there is no change in existing clusters. This algorithm is a global measure called *Condorset Criterion* which has not been discussed in this paper.

---

18. **[SLC+ 01]** Salvatore J. Stolfo, Wenke Lee**,** Philip K. Chan**,** Wei Fan and Eleazar Eskin; Data Mining-based Intrusion Detectors: An Overview of the Columbia IDS Project; *ACM SIGMOD RECORD (4): Special Selection on Data Mining for Intrusion Detection and Threat Analysis, Deniel Barbara, editor, Dec. 2001*

In this paper the IDS group emphasizes on the factor "Cost-Sensitive". They have focused onto two types of costs: *damage cost* and *operational cost*. As an example of damage cost, some attacks may be annoying but incur no damage; others may disable the whole network of computers: both have a different *damage cost*. On the other hand, some

attacks can be detected by inspecting the header only and some may be conducted over long period of time and require an expensive archive of data in memory for successful detection. In such cases different attacks have different *operational cost*. These costs vary over network to network or site to site. The Columbia IDS group built a facility that generates models of attacks based upon site-specific cost models. So, each IDS installed at a network or a site is cost-optimized for that site, automatically. The steps in their IDS project are as follows:

- Gather data in an RDBMs data warehouse
- Analyze data to identify useful features
- Define the costs associated with the observable features, systems and services being protected
- Automatically compute cost-effective models of known attacks and normal system behavior to build an effective misuse and anomaly detection system
- Deploy those models into a functioning detection system

The core technologies developed at Columbia can be applied to any environment in which audit data can be gathered using any standard commercial method like NFR (Network Flight Recorder), BSM (Basic Security Module in UNIX). But they have developed their own auditing facility which is BSM, which provide the same utility as UNIX BSM but in WINDOWS platform.

---

19. [SZ 02] Karlton Sequeira, Mohammed Zaki. "ADMIT: Anomaly based data mining for intrusions". *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, July 2002*

In this paper the authors have identified that current intrusion detection systems are not fully real time as they take much time in training their datasets. Keeping this in mind authors have presented a new methodology "ADMIT", which is better suited to real time application and which requires much shorter training time, summarizes the data and achieves model scaling simultaneously. The authors have also claimed that their system

is better in performance than the method of Du Mouchel and Lane as ADMIT uses shorter window sequences.

Instead of network level data their method concentrates on user command level data. According to the authors, ADMIT creates user profile of uses and find anomaly based on that profile. If user performs any abnormal command then it flags an anomaly. During training the commands entered by a user are stored in that user's audit data table according to the time of entry. During testing, the command data is directly used to detect anomalies via online sequential *classification*. The authors of this paper have claimed that their developed system, ADMIT, is a user profile dependent, temporal sequence clustering based, real-time intrusion detection system with host based data collection and processing. They have also claimed that their system achieved 80% detection rate and as low as 15% false positive rate.

---

20. [Y 03] Yoshida, K; Entropy based Intrusion Detection; *Communications, Computers and signal Processing, 2003. PACRIM. 2003 IEEE Pacific Rim Conference on, Volume: 2, 28-30 August 2003. Pages: 840-843 vol. 2*

The author in this paper identified the selection of minimum support parameter in APRIORI association algorithm as critical problem and therefore presented another model which can substitute the APRIORI model (Aggarwal, R., 1993).

In this paper the author briefly described two models- APRIORI based data mining and Entropy based data mining. Then he compared both systems and showed us the advantages of his Entropy based system over the APRIORI based system.

So the author in this paper has proposed the use of "Entropy based Data Mining" method for intrusion detection in alternative of APRIORI based system. Mainly entropy based system is called more precisely as "*Graph Based Induction* ***(GBI)***" method for rule finding. The algorithm is as follows:

The advantage of this algorithm is that it selects pairs those appear more than MinSupport, then this algorithm works as APRIORI algorithm and if in step 3 the best pair is selected based on entropy based index then this can be viewed as an extension of conventional classification rule learning algorithm. Although in APRIORI based systems and classification rule based systems they use attribute values table representations but in GBI this thing can be implemented by viewing root nodes as attribute classes and the value of the attributes can be seen as connecting nodes.

The author claimed this algorithm as an easy alternative to the APRIORI and ADAM (Barbara, D., 2001).

Bibliography:

1. [AJ 01] R Agarwal, MV Joshi; PNrule**:** A New Framework for Learning Classifier Models in Data Mining**;** *on First SIAM International Conference on Data Mining, 2001;*

2. [Ba00] Tim Bass. "Intrusion Detection System and Multi-sensor Data Fusion". *Communication of the ACM, pp 99-105, vol. 43, no. 4, April 2000*

3. [Ba03] Yuebin Bai. "Intrusion Detection Systems: Technology and Development". *Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on, 27-29 March 2003.*

4. [BBH+02] Jerzy Bala, Sung Balik, Ali Hadjarian, BK Gogia and Chris Manthorne. "Application of a Distributed Data Mining Approach to Network Intrusion Detection". *Proceeding of the first international joint conference on Autonomous and multiaqent Systems: Part 3, July 2002*

5. [BCH+01] Eric Bloedorn, Alan D. Chritiausen, W. Hill, C. Skorupka, Lisa M. Talbot, Jonathan Tivel. "Data Mining for Network Intrusion Detection: How to get started". *Technical report of the MITRE Corporation, VA*

6. [BCJ+01] Daniel Barbara, Julia Couto, Sushil Jadodia, Ningning Wu. "ADAM: A Testbed for exploring the Use of Data Mining in Intrusion Detection". *ACM SIGMOD RECORD (4): Special Selection on Data Mining for Intrusion Detection and Threat Analysis.*

7. [BCL02] D. Barbara, J. Couto, Y. Li. "Coolcat: An entropy based algorithm for categorical clustering". *In proceeding of the 11th ACM conference on Information and knowledge management (CIKM), McLean, VA, November 2002.*

8. [BCV01] E. Biermann, E. Cloete, L.M. Venter. "A comparison of Intrusion Detection System". *Computer and Security, Volume 20, Issue 8, pp. 676-683, December 1, 2001.*

9. [BLC03] Daniel Barbara, Yi Li, Julia Couto, Jia-Ling Lin, Sushil Jajodia. "Bootstrapping a data mining intrusion detection system". *Proceedings of the 2003 ACM symposium on Applied computing, March 2003.*

10. [BKN+00] Markus Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jorg Sander."LOF: Identifying densitybased local outliers". *In proceedings of the ACM SIGMOD Conference, Dallas, TX, 2000.*

11. [BR01] B. Balajinath, S.V. Raghavan. "Intrusion Detection through Learning Behavior Model". *Computer Communications, Volume 24, Issue 12, pp. 1202-1212, July 15, 2001.*

12. [BTS+01] Bloedorn, E.L. Talbot, C. Skorupka, A. Christiansen, W. Hill, and J. Tivel [2001]. "Data Mining applied to Intrusion Detection: MITRE experiences". *Submitted to the 2001 IEEE International Conference on Data Mining.*

13. [CG00] Chris Clifton, Gary Gengo. "Developing Custom Intrusion Detection Filters Using Data Mining". *MILCOM 2000. 21st Century Military Communications Conference Proceedings, Volume: 1, 22-25 Oct. 2000. Pagees: 440 – 443 Vol. 1*

14. [D03] Margaret H. Dunham; Data Mining Introductory and Advanced Topics; Prentice Hall 2003, ISBN 0-13-088892-3

15. [De87] Denning, D. "An Intrusion Detection Model". *IEEE Transactions of software engineering SE – 13, 2. (Feb 1987), 222 – 232.*

16. [DFP+04] Holger Dreger, Anja Feldmann, Vern Paxson, Robin Sommer."Operational experiences with high volume network intrusion detection". *Proceedings of the 11th ACM conference on Computer and communications security, October 2004.*

17. [ELK+04] L Ertoz, E. Eilertson, A. Lazarevic, P. Tan, J. Srivastava, V. Kumar and P. Dokas, The MINDS - Minnesota Intrusion Detection System; in Data Mining: Next Generation Challenges and Future Directions 2004.

18. [EMZ+00] Eleazar Eskin, Matthew Miller, Zhi-Da Zhong, George Yi, Wei-Ang Lee, Sal Stolfo. "Adaptive Model Generation for Intrusion Detection Systems". *Workshop on Intrusion Detection and Prevention, 7th ACM Conference on Computer Security, Athens, GR: November, 2000*

19. [FSL+01] Wei Fan, Salvatore J. Stolfo, Wenke Lee, Matthew Miller, Philip K. Chan. "Using Artificial Anomalies to Detect Unknown and Known Network Intrusions". *Data Mining 2001, ICDM 2001, proceedings IEEE International Conference on, 29 Nov – 2 Dec, 2001.*

20. [GRD03] Giorgio Giacinto, Fabio Roli, Luca Didaci. "Fusion of multiple classifiers for intrusion detection in computer networks". *Pattern recognition letters Volume 24, Issue 12, pp. 1795-1803, August 2003.*

21. [HC03] Sang-Jun Han, Sung-Bae Cho. "Detecting Intrusion with rule based integration of multiple models". *Computers and Security, Volume 22, Issue 7, pp. 613-623, October 2003.*

22. [HHE+02] Andrew Honig, Andrew Howard, Eleazar Eskin, and Salvatore Stolfo. "Adaptive Model Generation: An Architecture for the Deployment of Data Mining-based Intrusion Detection Systems." *Data Mining for Security Applications. Kluwer 2002*

23. [HJI00] Hashim, S.J, Jumari, K, Ismail, M. "Computer network intrusion detection software development". *TENCON 2000. Proceedings , Volume: 3 , 24-27 Sept. 2000 Pages:117 - 123 vol.3.*

24. [ISA93] T. Imielinski, A. Swami, R. Agarwal. "Mining association rules between sets of items in large databases". *In proceeding of the ACM SIGMOD conference on management of data, Washington D.C. May 1993.*

25. [J 05] Jensen, Kenneth G.; A combined association rule/radial-basis function neural network approach to intrusion detection; M.Sc Thesis, Utah State University, 2005.

26. [JA00] Mahesh Joshi, Ramesh Agarwal. "A new framework for learning classifier model in data mining (a case study in network intrusion)". *In proceeding of the 1st SIAM Conference on Data Mining, April 2000.*

27. [JHK+01] Mahesh Joshi, Euihong Han, George Karypis, Vipin Kumar; Parallel Algorithms in Data Mining; Technical report TR 01-001, Department of Computer Science and Engineering, University of Minnesota, January 26, 2001. http://www.cs.umn.edu/tech_reports_upload/tr2001/01-001.pdf

28. [KTK02] Christopher Krugel, Thomas Toth, Engin Kirda. "Service specific anomaly detection for network intrusion detection". *Proceedings of the 2002 ACM symposium on Applied computing, March 2002.*

29. [Le02] Wenkee Lee. "Applying Data Mining to Intrusion Detection: the Quest for Automation, Efficiency, and Credibility". *ACM SIGKDD Exploration Newsletter, vol. 4, issue 2, December 2002, pages 35-42.*

30. [LB97] T. Lane, C.E. Brodley. “An application of machine learning to anomaly detection”. *In proceeding of 20th National Information System Security Conference, 1997.*

31. [LFM+00] Wenke Lee, Wei Fan, Matthew Miller, Sal Stolfo, and Erez Zadok. “Toward Cost-Sensitive Modeling for Intrusion Detection and Response”. *Workshop on Intrusion Detection and Prevention, 7th ACM Conference on Computer Security, Athens, GR: November, 2000.*

32. [LJ99] E. Lundin, E. Jonsson. “Some practical and fundamental problems with anomaly detection”. *In: Proceedings of the Fourth Nordic Workshop on Secure IT Systems (NORDSEC'99), Kista, Sweden, 1-2 November 1999.*

33. [LJ00] Emilie Lundin, Erland Jonsson. “Anomaly-based Intrusion Detection: Privacy concerns and other problems”. *Computer Networks, Volume 34, Issue 4, pp. 623-640, October 2000.*

34. [LS00] Wenke Lee, Salvatore J. Stolfo. “A Framework for Constructing Features and Models for Intrusion Detection Systems”. *ACM Transaction on Information and System Security, vol. 3, no. 4, November 2000, pages 227-261.*

35. [LSC+01a] Salvatore J. Stolfo, Wenke Lee**,** Philip K. Chan**,** Wei Fan and Eleazar Eskin. “Data Mining-based Intrusion Detectors: An Overview of the Columbia IDS Project”. *ACM SIGMOD RECORD (4): Special Selection on Data Mining for Intrusion Detection and Threat Analysis, Deniel Barbara, editor, Dec. 2001*

36. [LSC+01b] Wenkee Lee, Salvatore J. Stolfo, Philip K. Chan, Eleazer Eskin, Wei Fan, Matthew Miller, Sholmo Hershkop and Junxin Zhang. “Real Time Data Mining-based Intrusion Detection”. *Proceedings of DISCEX ll, June 2001*

37. [LP99] U. Lindqvist, P. A. Porras. “Detecting Computer and Misuse Through the Production-based Expert System Toolset (P-Best)”. *In proceedings of the 1999 IEEE symposium on security and privacy. pp.146-161*

38. [LS98] Lee, W., and S. Stolfo [1998]. “Data Mining Approaches for Intrusion Detection”. *In proceedings of the 7th USENIX security symposium, San Antonio, TX*

39. [LSF+00] W. Lee, S.J. Stolfo, W. Fan, A. Prodromidis, and P. Chan. “Cost sensitive modeling for fraud and intrusion detection: results from the JAM project. *In proceedings of the 2000 DARPA information survivability conference and exposition, Jan 2000*

40. [LSM98] Wenke Lee, Sal Stolfo, and Kui Mok. “Mining Audit Data to Build Intrusion Detection Models”. *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD '98), New York, NY, August 1998*

41. [LSM99] W.Lee, S. J. Stolfo and K. Mok. “Data mining in workflow environments: Experiences in intrusion detection”. *In proceedings of the 1999 conference on knowledge discovery and data mining (KDD -99), 1999*

42. [LXY03] Quing-Hua Li, Jia-Jun Xiong, Hua-Bing Yang. “An Efficient Algorithm for Frequent Pattern in Intrusion Detection”. *Machine learning and cybernatics, 2003 International Conference on, Volume 1, 2-5 Nov. 2003. Pages: 138-142 vol.1*

43. [MC03] V. Mahoney, Philip K. Chan. “Learning Rules for Anomaly Detection of Hostile Network Traffic”. *Data Mining, 2003. ICDM 2003. Third IEEE International Conference 2003*

44. [MCZ+00] Stefanos Manganaris, Marvin Christensen, Dan Zerkle, Keith Hermiz. “A Data Mining Analysis of RTID Alarms”. *Computer Networks vol. 34, issue 4, October 2000, pages 571-577*

45. [MT96] Manila, H., Toivonen, H. “Discovering Generalized Episode using minimal occurrences”. *In proceedings of the 2nd International conference on knowledge discovery in databases and data mining, Portland, Oregon, August 1996.*

46. [MTV97] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo, “Discovery of Frequent Episodes in Event Sequences”, Data Mining and Knowledge Discovery l(3): 259-289 (1997)

47. [NC03] Caleb C. Noble, Diane J. Cook. “Graph based Anomaly Detection”. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, August 2003.*

48. [NCD+02] Peng Ning, Yun Cui, Douglas S. Reeves. “Constructing attack scenarios through correlation of intrusion alerts”. *Proceedings of the 9th ACM conference on Computer and communications security, November 2002.*

49. [PES01] Leonid Portnoy, Eleazar Eskin and Salvatore J. Stolfo. “Intrusion detection with unlabeled data using clustering”. *Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001). Philadelphia, PA: November 5-8, 2001*

50. [QW02] Yan Qiao, Xie Weixin. “A Network IDS with low false positive rate”. Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on. Volume 2, 12-17 May 2002. Pages 1121-1126.

51. [Su96] A. Sundaram. “An Introduction to Intrusion Detection”. *Crossroads: The ACM student magazine, 2(4), April, 1996.*

52. [SGF+02] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, S. Zhou. "Intrusion Detection: Specification based anomaly detection: a new approach”. *Proceedings of the 9th ACM conference on Computer and communications security, November 2002.*

53. [SGV+99] R. Sekar, Y. Guang, S. Verma, T. Shanbhag. “A high performance network intrusion detection system”. *Proceedings of the 6th ACM conference on Computer and communications security, November 1999.*

54. [SZ02] Karlton Sequeira, Mohammed Zaki. “ADMIT: Anomaly based data mining for intrusions”. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, July 2002*

55. [TUA+98] Dick Tsur, Jeffrey D. Ullman, Serge Abiteboul, Chris Cliffon, Rajeev Motwani, Svetlozar Nestorov and Arnon Rosenthal; “Query Flocks: A generalization of Association Rule Mining”; *In proceedings of the 1998 ACM SIGMOD Conference on Management of Data, Seattle, WA, June 2 – 4, 1998.*

56. [XQJ01] Yang Xiang-Rong, Song Qin-Bao, Shen Jun-Yi. “Implementation of sequence patterns mining in network intrusion detection system”. *Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences.*

57. [XSZ04] Hongxia Xia, Qi Shen, Luo Zhong, Shan Feng, Rui Hao. “Application of Data Mining Technology and generic algorithm to intrusion detection system”. *Proceedings of the 3rd international conference on Information security, November 2004.*

58. [Yo03] Kenichi Yoshida. “Entropy based Intrusion Detection”. *Communications, Computers and Signal Processing, 2003. PACRIM. 2003 IEEE Pacific Rim Conference on, Volume: 2, 28-30 August 2003.*

59. [YGF+98] Nong Ye; Giordano, J.; Feldman, J.; Qiu Zhong. “Information fusion techniques for network intrusion detection”. Information Technology Conference,1998.IEEE, 1-3 Sept 1998. Pages 117-120.

60. [Z02] Zhen Zhang; Data Mining Approach For Network Intrusion Detection; Presentation at the Department of Computer Science, California State University, Sacramento, 24th April, 2002. http://gaia.ecs.csus.edu/~wang/ppt/ids.ppt

61. [ZS04] S Zanero, Sergio M Savaresi. "Unsupervised learning techniques for an Intrusion Detection System". *On proceeding of the 2004 ACM symposium on Applied computing, March 2004.*

62. [ZW02] Zhen Zhang, Chung-E Wang; Master's Thesis, Department of Computer Science, California State University, 2002.

63. http://www.aspfree.com/c/a/MS-SQL-Server/Using-Data-Mining-for-Business-Intelligence/