Chapter 1

Introduction

Network security is becoming increasingly important in modern internetworked systems. The evolution of the Internet and its use in day to day life has increased the need for network security systems. With the development of networking and interoperation on public networks, the number and the severity of security threats has increased significantly. Many intrusion prevention techniques were developed to protect the systems. Some of these techniques are user authentication, authorization, avoiding design and programming errors, information protection (encryption), and firewall. These techniques are used to protect computer systems. But intrusion prevention alone would not be sufficient because system becomes more complex. There are always security loopholes in the network system. These weaknesses are due to various reasons including incorrect system configuration and operations, errors present in design and programming. The policies trying to strict balance between control of a system and information access may fail to provide airtight security. Many systems deploy Intrusion Detection System (IDS) as another layer of security mechanism to protect the system. Intrusion can be defined as an attempt to compromise the security services such as confidentiality, integrity and availability of resources, or to bypass the security mechanism of a network system. An intrusion may be generally described as a sequence of related actions by a malicious adversary that results in the occurrence of unauthorized breaches to a target system or network.

Intrusion Detection can be defined as "the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource." More specifically, the goal of intrusion detection is to identify entities attempting to subvert in-place security controls. These intrusions include violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. These
intrusions are the results of attackers accessing systems from the Internet or the authorized users of the systems who attempt to gain unauthorized privileges and misuse these privileges. It is a common misunderstanding that firewalls can recognize and block intruders. But a firewall is just like a fence which restricts the access to the designated points on the network. Having IDS to complement a firewall can provide an extra layer of protection to a system such as:

- Identifying attacks that firewall legitimately allow through
- Identifying various attempts such as ping sweep or port scan
- Identifying insider hacking
- Provide additional checks for holes/ports opened through firewalls, intentionally or unintentionally.

IDS can generate and report alarms to system operators when it detects intrusive or abnormal activities.

The history of intrusion detection can go back to 1980s, when James Anderson first proposed to monitor security threats through audit trails [1]. In the early years, pattern recognition referred to algorithms that allowed the computer to search for patterns. The pattern refers to repetitive behaviour that stands out. These patterns could be a change in individual feature or group of features associated with specific attack. There are many well-known pattern recognition algorithms for classification purpose. The focus of this work is on to show how pattern matching is a critical ability, and that it also be the strength of the IDS.

The changes in patterns are mainly the manifestations of attack. Pattern based IDS provides very low false alarms as compare to heuristic/anomaly based IDS. It also provides detail contextual analysis providing steps for preventive or corrective actions. Hence this work has primarily focused on pattern classification and matching techniques to identify the attacks or intrusions.

1.1 Evolution of Intrusion Detection

Intrusion detection originates from traditional audit systems. In early computing environments, large mainframes produced chronological records of system events. These records could then be examined manually for purposes such as accounting and security.
U.S. Department of Defence (DOD) outlined security goals in 1970s for such audit mechanisms. James Anderson first articulated the goal of automatic audit reduction to remove irrelevant records. From 1984 to 1986, Dorothy Denning and Peter Neumann developed a theoretical model of a real-time Intrusion Detection System, the Intrusion Detection Expert System (IDES) [2]. IDES was implemented from 1986-1992 at SRI International and served as the basis for many future research efforts. The Network System Monitor developed in 1991 at the University of California at Davis. It used network packets as its primary data source, rather than OS-level records. Hence, it was the first network Intrusion Detection System. U. C. Davis and others with the collaboration of the U.S. Air Force developed the Distributed Intrusion Detection System (DIDS) in 1992. A DIDS can be defined as: multiple Intrusion Detection Systems (IDS) spread over a large network, all of which communicate with each other, or with a central server that facilitates advanced network monitoring, incident analysis, and instant attack data [3]. DIDS, however, faces the new problem that it needs analyze massive aggregating data generated by individual IDSs placed across large-scale networks. How to analyze these huge amounts of data with high efficiency is a considerable complicated and comprehensive process.

1.2 Concepts of Intrusion Detection

This section briefly discusses the main concepts underlying network intrusion detection. Depending upon the approach taken by IDS to detect the suspicious traffic, they have been classified in two different categories. Broadly IDS are classified as: Network Intrusion Detection Systems (NIDS) and Host-based Intrusion Detection Systems (HIDS). Network Intrusion Detection Systems are placed at a strategic point(s) within the network. NIDS monitor traffic to and from all devices on the network e.g. Snort [4] and Bro [5]. Host-based Intrusion Detection Systems are run on individual hosts or devices on the network. HIDS rely on information gathered on individual hosts.

A Network-based Intrusion Detection System (NIDS) may take the form of an independent network appliance or device tapped into the network with associated processing capabilities. It monitors network activity. Input for NIDS is solely in the form of the traffic on the network. Since frequent attacks on networks or machines within the
network originate outside of the network in question, NIDSs have a wide range of possible attacks to detect from the outside (ingress). These attacks include worms, or other malware self-spreading on the network, Denial of Service (DOS) attacks, spreading viruses, port-scans, and attempts to break into or exploit vulnerabilities in computer systems by malicious individuals. However, NIDSs can also help to warn about or guard against attacks and sensitive data within the network.

HIDS will monitor all the incoming and outgoing packets on the device only and can notify the administrator or user about any suspicious activity. A commonality often seen in HIDSs is the use of an object or checksum database that catalog the last or known good states of the objects being monitored. Attackers that know of a HIDS on their target system may try to circumvent the HIDS’s detection by covering up traces of their attacks through modifying entries in this database so as not to set off alarms during the next HIDS scan. Therefore HIDS database needs to be strongly protected.

The main niche of examination in this research work is the intersection between network-based and pattern-based Intrusion Detection Systems. In this work, the focus is on network-based systems, i.e., NIDS. When a NIDS believes that it has detected an intrusion, it raises an alert. If the system has correctly identified an intrusion, the alert is a true positive. If the NIDS alarms although no intrusion has occurred, it is a false positive. On the other hand, if the NIDS fails to report an actual intrusion, the alert is a false negative. If it correctly remains quite when no intrusion has taken place, then it is a true negative. Table 1.1 summarises the activity and the type of alarms for the respective event.

Table 1.1: Types of Alarms

<table>
<thead>
<tr>
<th>Action against Events</th>
<th>Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctly identified an intrusion</td>
<td>True positive</td>
</tr>
<tr>
<td>Alarms although no intrusion has occurred</td>
<td>False positive</td>
</tr>
<tr>
<td>Fails to report an actual intrusion</td>
<td>False negative</td>
</tr>
<tr>
<td>Remains quite when no intrusion has taken place</td>
<td>True negative</td>
</tr>
</tbody>
</table>
Conceptually, NIDSs often do not make their decisions based on individual network packets but on events which are (policy-neutral) abstractions of network activity. For example, the worm signature pattern generator Poligraph is vulnerable to deliberate noise injection. An adversary can pollute the training set made up of suspicious traffic flows to force Poligraph to generate signature patterns with wrong features.

1.2.1 Intrusion Detection Techniques

Intrusion detection techniques have been traditionally classified into one of two methodologies: misuse detection or anomaly detection.

a) Misuse Detection

Misuse detection is concerned with identifying intruders who are attempting to break into a system using some known technique. If a system security administrator were aware of all the known vulnerabilities then a misuse detection system would be able to identify their occurrences and eliminate them. Misuse detection systems identify known attacks. They are equipped with a library of attack patterns; each time such a pattern is found in the network stream, an attack is assumed to be in progress. Such a library requires expert knowledge to build, and it needs to be updated regularly to accommodate new attacks. Its content can be based on experiences with attacks from the past, on dissected exploits, or on anticipated but yet unseen attacks.

There are different levels at which attack patterns can be described. The most common representations are specific byte sequences which are known to appear in the raw payload stream of attack connections. All major commercial NIDSs use this technique. It is known as byte-level patterns signatures. The nature of the patterns present in an Intrusion Detection System depends on the power of the system itself and its intended use. Consequently, misuse-based IDSs are also known as signature-based IDSs. [6]

The main advantage of misuse detection systems is that the system knows for a fact how normal behaviour should manifest itself. This leads to a simple and efficient processing of the audit data.

The main disadvantage of misuse detection is the difficulty in gathering the information about known attacks and keeping them abreast with new vulnerabilities and environments. Moreover it is a more time consuming task.
b) Anomaly Detection

In anomaly detection the goal is to define and characterize legitimate behaviors of the users. Anomalous behaviors can be detected by quantifying deviations from the legitimate behaviors. However, it is difficult to identify the distance between anomalous and legitimate behaviors.

Anomaly detection can be categorized as static or dynamic. In a static anomaly detection system is based on the assumption that there is a static portion of the system being monitored. Static portions of the system can be represented as a binary string or a set of binary strings. If there is a change in this static portion of the system, then either an error has occurred or an intruder has altered it. Tripwire is an example of static anomaly detectors [7]. Dynamic anomaly detectors are harder to build since building them requires a definition of behaviour, which is often defined as a sequence (or partially ordered sequence) of distinct events. Differentiating between normal and anomalous activity in dynamic anomaly detection systems is much difficult as compare to the problem of distinguishing changes in static elements. Dynamic anomaly detection systems usually create a base profile to characterize normal, acceptable behaviour. A profile usually consists of a set of observed measures of behaviour for a selected set of dimensions. After initializing the base profile the dynamic anomaly detection systems are similar to the static ones; they monitor the behaviour by comparing the current behaviour with that implied by the base profile. Typically, there is a wide variation of acceptable behaviours and statistical methods are employed to measure deviation from the base profile. The main challenge in dynamic anomaly detection systems is that they must build accurate base profiles and then recognize behaviours that significantly deviate from the profile.

The main advantage of dynamic anomaly detection systems is that they do not require any configuration since they automatically learn the behaviour of large number of subjects. Lacking prior knowledge of how an intrusion would manifest itself anomaly detection systems can identify novel intrusions which are the variations of known intrusions. However, building base profiles and defining measures of deviations from them is not an easy computational task. For that reason it has been an active area of research, in which several machine learning, time-series analysis and other data-analysis techniques have been employed [8, 9, 10, 11, 12, 13].
Disadvantage of this approach is the high false rate of alarm because the entire scope of the behavior of an information system may not be covered during the learning phase. Also, behavior can be changed over time, creating the need for periodic online retraining of the behavior profile, resulting either in the unavailability of the intrusion detection system or in additional false alarms.

The classic anomaly-based system is IDES [2] which uses a combination of statistical metrics and models such as thresholds, confidence intervals, and Markov models. Since then, many more approaches have been suggested. Often, they borrow ideas from the AI community, including information theory, neural networks, genetic algorithms, and artificial immune-systems.

Irrespective of how intrusions are classified, the main techniques for detecting them are the same: anomaly detection and misuse detection approach. Both these approaches make implicit assumptions about the nature of intrusions that can be detected by them.

One of the few intrusion detection methodologies which have departed from the established use of misuse and anomaly detection techniques is pattern matching. In this approach, a series of penetration scenarios are coded into the system. Pattern matching possesses a distinct advantage over anomaly and misuse detection methods. Pattern matching IDS is capable of identifying attacks which may occur over an extended period of time, or a series of user sessions, or by multiple attackers working in concert. This approach is effective in reducing the need to review a potentially large amount of audit data.

1.2.2 Types of Networking Attacks

There are four major categories of networking attacks. Attack on a network can be categorised into one of these groupings.

- Denial of Service (DOS): It is a class of attacks in which an attacker makes some computing or memory resource too full or too busy to handle legitimate requests, or denies legitimate users access to a system. Examples are Neptune, Back, Pod, Land, Mail bomb, Ping of death, Process table, Smurf, Teardrop, Udp_storm etc. [14]
• Remote to User attacks (R2L): It is a class of attacks in which an attacker sends packets to a machine over a network, and then exploits machine’s vulnerability to illegally gain local access as a user. Examples are multihop, Spy, Warezclient, Warezmaster, Ftp_write, Guest, Imap, Named, Phf, Sendmail, Xlock, Xsnoop etc. [14]

• User to Root Attacks (U2R): It is a class of attacks in which an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Examples are Eject, Ffbconfig, Fdformat, Load module, rootkit, Ps, Xterm, Buffer_overflow etc. [14]

• Probing: It is a class of attacks in which an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits. Examples are Ipsweep, Mscan, Nmap, Saint, Satan etc. [14]

The purpose of classifiers in IDSs is to identify all possible attacks especially these four attacks as accurately as possible.

1.2.3 Detection vs. Prevention

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. Intrusion prevention is the process of performing intrusion detection and attempting to stop detected possible incidents.

Historically, the main purpose of NIDSs was to detect attacks. Therefore, they were called Network Intrusion Detection Systems.

Originally, the term NIPS referred to a NIDS that is installed in-line with the network, instead of using a passive network tap. The main advantage of such an in-line system is that it can block traffic that it believes to be malicious. In this way, attack traffic can be prevented from reaching its destination. A non-in-line system is, at best, restricted to stop future packets. Unfortunately, the term NIPS is rather blurred by now. This is mainly due to a marketing hype produced by a famous Gartner report. The report declared intrusion detection to be "dead" and to be replaced by intrusion prevention. As a consequence,
commercial vendors hurried to re-brand their systems to prevention systems. Some did not care whether they were indeed working in-line, while others deliberately used terms slightly different from "prevention" for their passive systems (e.g., the non-in-line Dragon system sells as "Enterasys Dragon Intrusion Defence"). Therefore, the term "intrusion prevention" is avoided. Instead, it calls a system which follows the original intent of NIPS an in-line NIDS. Thereby, it also emphasize that the most important component is still detection. It cannot block an attack without first recognizing it. Additionally, it calls a NIDS active if, after detecting an attack, it is able to react in a way that directly impairs the attacker. All other systems are passive. By definition, all in-line systems are active.

1.3 Existing Problems of IDS
At present, two fundamental problems, quantity and quality of the outputs i.e. alarms or alerts of IDS, have not been solved well. The problem of the quantity of the alarms is that IDS usually generates too many alarms which will overwhelm its system operator. Consequently, too many alarms will be ignored by the operator, which make IDS valueless. The second problem is that the performance will degrade because the normal behaviour continuously changes and new attacks continuously emerge. The quality of alarms will degrade in two ways: miss alarms in true attacks and report too many false alarms on normal behaviour. False alarms waste operator’s time and reduce the operator’s trust in IDS. The alarms will be ignored by the operator when IDS are not trusted by the operator and makes IDS useless.
Some research works have been done to overcome these problems. The first method is to reduce the false positive alarms that IDS could generate i.e. improve the analysis methods. The second method is to add another layer above IDS, before the alarm reaches the operator. In this layer, the alarm can be filtered or some support information on analyzing the alarms can be generated to help the operator.
There are several available solutions to address the quantity problem such as alarm correlation, alarm filter and event classification.
1.4 Machine Learning

Machine learning is critical in the study of design and development of computer programs that improve their performance through experience. In a variety of application domains, machine learning algorithms have proven to be of great practical value. Machine learning algorithms are particularly useful for: poorly understood problem domains where little knowledge exists for the humans to develop effective algorithm, domains where there are large databases containing valuable implicit regularities to be discovered; or domains where programs must adapt to changing conditions. Not surprisingly, the field of intrusion detection turns out to be a fertile ground where many security, reliability, performance, and privacy tasks could be formulated as learning problems and approached in terms of learning algorithms. Various machine learning techniques which could be used in Intrusion Detection Systems are: Concept learning, Decision tree, Neural Networks, Bayesian learning, Genetic algorithms and Genetic programming, Instant based learning, Inductive logic programming, Analytical learning, Adaptive and Analytical learning, Inductive and Analytical learning.

1.5 Motivation

The design of IDS is a challenging task because, besides the requirements such as the coverage of certain types of attacks, other limiting factors are also to be taken into account. These factors can be listed as given below.

- **Feature extraction:** For detecting a given type of attack, IDS needs to be capable of making appropriate observations. Thus, IDS designer needs to identify the data required and most importantly, the data source that provides this data in an appropriate format. However it is to be noted that different detection approaches may rely on different data sets for their operation. Hence IDS designer is not able to influence the properties of the data source which makes it a non-trivial task. Moreover, commonly available data sources may require significant pre-processing of the data they provide or do not provide all the data needed.

- **Trade-off between costs and functionality:** The use of a given data source and detection approach comes with a certain cost. Their use may, for instance, cause unacceptable performance degradation, or require impractical modifications of the
surveyed system. In order to limit such side-effects, IDSs often use entirely different or simplified solutions that may result in reduced performance of IDS [15].

- **Test data:** For the testing of IDS realistic input data is required [16, 17]. This is an issue because no two environments for which IDS is to be designed are identical and because IDSs may make use of environment-dependent optimizations and ad-hoc solutions. Thus, there is no single general-purpose data set that can be used for testing.

### 1.6 Aim and Objectives

This research work mainly aims to increase the detection rate and reduce the false positive rate for the Intrusion Detection System. Recent evaluations of actual IDS implementations have clearly revealed that currently available IDSs suffer from various weaknesses [18, 19]. There are many algorithms available for Pattern Matching. To develop an optimized algorithm and to implement it can be a challenging task. From the literature review and theoretical study, it became clear that more extensive theoretical investigation was needed on a number of existing learning algorithms. Therefore the goal of this research work is to present a comparative study of various algorithms for IDS. Another goal is to enhance the performance of the algorithm for IDS using pattern based approach.

Following objectives were in mind to achieve the goal for successful implementation of Intrusion Detection System.

- To study various machine learning approaches which can be used for IDS
- To understand the models / the algorithms for Pattern Matching and its implementation for IDS (Unsupervised, supervised and semi-supervised learning approaches)
- To implement unsupervised learning approach using Self Organized Map [SOM]
- To implement supervised learning approach such as hybrid DT-SVM model and DS-AdaBoost
- To enhance the performance of AdaBoost supervised learning algorithm
• To propose and implement semi-supervised learning algorithm using SLA model for IDS
• To carry out experimentation on the implemented algorithms using benchmarked data set (namely, KDDCup99 [20], NSL [21])
• To compare the performance of the implemented unsupervised, supervised and semi-supervised learning approaches with respect to detection rate and false positive rate
• Compare results of proposed and implemented algorithms with the algorithms available in machine learning tool (WEKA [22])

1.7 Contributions
The main objectives of this research work are to examine the problem of pattern matching for IDS and its solutions fully. The work carried out presents the applicability of the solutions to pattern matching within the context of Network Intrusion Detection Systems. The approach presented in this work supports designers of IDSs by providing them IDS with high detection rate and low false alarm rate. In detail this research work makes the following contributions:

1. It has been attempted to expedite the applicability of the solutions of pattern matching within the context of Network Intrusion Detection Systems.
2. A new improved DS-AdaBoost supervised learning algorithm for Pattern Based Network Security is proposed.
4. The approach suggested based on Pattern Based Network Security renders better performance.
5. Supervised learning approach is used as black box for semi-supervised self learning.
6. Entropy of features is used for self learning in semi-supervised approach.
7. The semi-supervised self learning Pattern Based IDS adopts itself according to new pattern of attack.
8. Dynamic Threshold based, Semi-supervised learning algorithm that improved overall IDS accuracy, reduce FPR. Further this algorithm and methodology can be used in real life network environment.

1.8 Organization of the Thesis

The chapters of the thesis are organized as:

Chapter 1 contains a brief introduction defining some terms that will aid in the comprehension of material in following chapters. Chapter 2 is dedicated to the background of supervised, semi-supervised and unsupervised approaches for Intrusion Detection System. Also briefly describes various methods such as Statistics-Based Approaches, Data-Mining-Based Approaches, Neural Network (NN), Support Vector Machine (SVM), Generative Models, Self-Training, Co-Training, Multiview Learning, Transductive SVMs (S3VMs), Gaussian Processes, Information Regularization, Graph-Based Methods, K-means and Self Organizing Map. This chapter provide the detailed literature survey with respect to research topic. Chapter 3 provides unsupervised learning approach using self organizing map. Chapter 4 describes the application of machine learning to intrusion detection using hybrid model of Decision Tree and Support Vector Machine. Chapter 5 introduces the AdaBoost algorithm and discuss and implementation of proposed DS-AdaBoost algorithm using supervised approach. Chapter 6 describes the system design and implementation of semi-supervised learning with the proposed DS-AdaBoost algorithm. Chapter 7 consists of detailed experimentation results, observations, and comparative analysis on bench mark data set i.e. KDDCup99 and NSL-KDD. It focuses on optimization stage and its effect during training and testing phase. Chapter 8 consists of discussions on the overall observations and results are presented and compared with major conclusions. The thesis ends with discussion about further work. Appendix-A presents details about KDDCup 99 data set.