"Performance Analysis for Intrusion Detection Using Data Mining Techniques"

A

Project Report

Submitted in partial fulfillment of the
requirements for the award of the degree of

MASTER OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWORK

By

Radhika Shivhare
Roll No. R102213007

Under the guidance of

Dr.Ajay Prasad
Assistant Professor, CIT, UPES Dehradun

UPES
THE NATION BUILDERS UNIVERSITY

Department of Computer Science & Engineering
Centre for Information Technology
University of Petroleum & Energy Studies
Bidholi, Via Prem Nagar, Dehradun, UK

The innovation driven
E-School

CANDIDATE'S DECLARATION

I hereby certify that the project work entitled **"Performance Analysis for Intrusion Detection System Using Data Mining Technique"** in partial fulfillment of the requirements for the award of the Degree of MASTER OF TECHNOLOGY In ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWORK **and submitted to the** Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my work carried out during a period from **January, 2015** to **April, 2015** under the supervision of **Dr. Ajay Prasad, Assistant Professor**.

The matter presented in this project has not been submitted by me for the award of any other degree of this or any other University.

Radhika.
(Radhika Shivhare)
Roll No. R102213007

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 06/05/2015

(Dr. Ajay Prasad)
Dissertation Guide

Dr. AmitAgrawal
Program Head – M.tech
Center for Information Technology
University of Petroleum & Energy Studies
Dehradun – 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

I wish to express our deep gratitude to our guide **Dr.Ajay Prasad**, for all advice, encouragement and constant support he has given us throughout my dissertation work. This work would not have been possible without his support and valuable suggestions.

I sincerely thank to our respected Program Head of the Department, **Dr.Amit Agrawal**, for his great support in doing my dissertation in **Security** at **CIT**.

I am also grateful to **Dr. Manish Prateek, Associate Dean** and **Dr. Kamal Bansal Dean CoES, UPES** for giving us the necessary facilities to carry out our project work successfully.

I am very grateful to my respected class coordinator **Mr. Vishal Kaushik**, for showing a very keen interest in my work and motivating me to expand the horizons of the work.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Radhika

Name **Radhika Shivhare**

Roll No. **R102213007**

ABSTRACT

The usage of computer over network is increasing outstandingly but more exploitation of this facility for the communication may cause the network from severe kind of intruders which affect the security and integrity of the network. They can also stop the network resources and services which help in communication. To prevent the network from intruders a system is designed which is known as intrusion detection system (IDS).

Although different classification models have been developed for network intrusion detection, each of them has its strengths and weaknesses, including the most commonly applied Support Vector Machine (SVM) method and the K Nearest Neighbor (KNN). Our new approach combines the KNN method with Ant Colony Optimization (ACO) to take the advantages of both while avoiding their weaknesses. The algorithm is implemented and evaluated using a standard benchmark KDD99 data set.

In this dissertation, the performance of Intrusion Detection with various classifiers is compared. The implementation of our method is performing on MATLAB2012a simulation and the comparative analysis is done among SVM and KNN using performance metrics such as specificity, sensitivity and accuracy. The results of our methodology give more accurate and precise results than the SVM and KNN approach.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
INTRODUCTION

1.1 History

Now a days the use of computer system increases rapidly via internet technology which lessen the security, privacy, reliability and availability of computer systems and its resources to protect the ability of references and unauthorized access to a computer system modification and use of the refuse safely to protect data and resources. For host intrusion detection system, the most powerful methods for extracting information hidden in large data sets from the data mining methods, implemented. To apply data mining techniques in intrusion detection, preprocessing of the data collected is the first step. Then, in a special format for exchanging data mining process the configuration is used for classification and clustering. Rule-based classification model: a decision tree-based, Bayesian network-based or based on the neural network.

Generally KDDCUP99 is standard dataset which is used for testing the proposed approach and there results are a show potential, especially for minimum False Alarm Rate and high detection rate to build a good mode, and it produced that outperforms the existing methods [9]. The goal is to improve efficiency and accuracy for intrusion detection system. Traditional intrusion prevention techniques like firewalls way to control or encryption, have failed to fully keep safe (out of danger) systems from increasingly not simple attacks and malware. As an outcome, IDS are used to make discovery of these attacks before they give stretched wide damage [1]. When building IDS, they need to take into account many issues such as data pre-processing, data collection, intrusion recognition, reporting, and response. Intrusion recognition is most important out of them. Audit data are making a comparison with detection models describing the patterns of intrusive or benign behavior, so that audit data can be identified both successful and unsuccessful intrusion attempts. Since Denning first made an offer a go into discovery, design to be copied in 1987, many research efforts have been gave one's mind to an idea on how to effectively and without error make discovery models. Between the late 1980s and the early 1990s, a joining together of expert systems and to do with facts as numbers comes, goes near was very pleasing to all. Discovery models were formed (from) from the lands ruled over knowledge of safety experts. From the mid-1990s to the late 1990s, getting acquaintance of normal or not normal behavior had curved from done with the hands to automatic. The artificial intelligence and machine learning techniques were used to build unearthing the close relation models from a group of training facts. Intrusion detection systems are being developed as devices to detect attacks and thus are becoming very important. IDS are useful in detecting successful intrusion, and also in monitoring suspicious activity and the attempts to break the security. Intrusion detection is the practice of observing and examining the actions going on in a system in order to identify the attacks and susceptibilities.

There are some terminologies involved in this research work and it is represented in Table 1.1 below [1].

Terms and their Definition Terms	**Definition**
Alert/Alarm	It a signal that is generated by the system to report the administrator that the system has been or is being attacked.
True Positive (TP)	Attack detected by the system and the signal is raised.
False Positive (FP)	The system generates the alarm when there is no attack detected.
False Negative (FN)	A failure of IDS to detect an actual attack.
True Negative (TN)	No attack is identified and no alarm is raised.
Noise	Unwanted data that causes the system to raise the alarm.
Alarm Filtering	Process to distinguish between false positives and actual attacks.
SVM	Support Vector Machine
IDS	Intrusion Detection System

Table 1.1: Shows Terms and their Definition Terms

1.2 Research Objectives

The aim of this dissertation is to develop systems which have broad attack detection coverage and which are not specifically in detecting only the previously known attacks.It also intended to enhance the sensitivity, specificity and accuracy generated by anomaly and hybrid intrusion detection systems in that way improving their attack detection accuracy. Issues such as scalability, accessibility of training data and toughness to noise in the training data and others are also completely addressed.

1.3 Problem Statement

We consider intrusion detection as a classification problem, that is, we wish to classify each audit record into one of a discrete set of possible categories, normal or a particular kind of intrusion. We can thus apply machine learning approaches to inductively learn classifiers as detection models. 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. However, before we can apply classification algorithms, we need to first select and construct the right set of system features that may contain evidence (indicators) of normal or intrusions. As we have read various research paper and they all achieved some best accuracy levels as well as they also got less accuracy levels for the few classes. Here they also got 100% accuracy only for one class (R2L), if they have been used one particular classifier for particular class with their advantages then they got 100% accuracy for Normal, Probe, U2R, Dos also.

1.4 Dissertation Organization

The thesis is organized into six main chapters. Chapter 2 presents the literature review & intrusion detection system describes the related techniques of Intrusion Detection along with various schemes and focuses on the taxonomy of Intrusion Detection. Chapter 3 discusses about the data mining and machine learning techniques for intrusion detection Chapter 4 describes the plans to deal with the domain of problem and proposed solution. It also portrays the methodology of our work & Algorithm used to develop the application. Chapter 5 gives the result & analysis states under various assumptions taken & and also present a view of working environment in which application will run. This chapter also gives us the type of view of working systems and also gives an analysis of results obtained in our work. The analysis is in terms of different parameters. Chapter 6 draws conclusion and future work.

Chapter 2
LITERATURE SURVEY

2.1. Related Work

A good intrusion detection system must be capable to distinguish malicious activities. However, what is normal and what is an attack is not defined, i.e., an event may be given thought to be as normal with respect to some criterion but the same may be labeled inconsistent when this criterion is changed. Consequently, the intention is to find anomalous test patterns which are like to the anomalous patterns which occurred during training. The close relation thing taken as certain is that the valuing criterion is unchanged and the system is rightly trained such that it can safely separate normal and anomalous events.

Wolfgang et al. [1] Reported Intrusion detection based upon computational intelligence is currently attracting substantial interest from the research area. Its attribute, such as adaptation, fault tolerance, high calculation speed and mistake pliability in th;e face of noisy information fit the prerequisite of constructing a good intrusion detection system. This paper presents the state-of-the-art in research progress of computational intelligence methods in intrusion detection systems. The scope of this review was on core process in CI together with artificial neural networks, fuzzy systems, evolutionary calculation methods, artificial immune systems and swarm intelligence. On the other hand, the practice of these methods reveals that each of them has advantages and disadvantages. Soft computing has the power to combine the strengths of these methods in such a way that their disadvantages will be compensated, thus offering better solutions. The assistance of research work in each method are methodically summarized and compared which allows us to obviously define existing research confronts and highlight promising new research guidelines. It is hoped that this survey can hand out as a useful guide through the confusion of the literature.

Marina Thottan et al. [2]. Reported in this paper provides a review of the area of network anomaly detection. Based on the case studies presented, it is clear that there is a significant advantage in using the wide array of signal processing methods to resolve the quandary of anomaly detection. A greater synergy between the networking and signal processing areas will help develop better and more effective tools for detecting network anomalies and performance problems. A few of the open issues in the application of statistical analysis methods to network data are discussed below.

Sandhya Peddabachigari et al. [3] investigate some new technique for incursion detection and evaluate their concert on the benchmark KDD Cup 99 Intrusion data. They first investigate a decision tree as an intrusion detection model. They furthermore conduct experiment with support vector machines (SVM) and compare the decision tree performance with this model while the decision tree was used as a binary classifier; they employed five classifiers for 5-class classification. The observed results designate that decision tree gives enhanced correctness than SVM for Probe, U2R and R2L classes while for the Normal class both give same accurate and for DOS class decision tree gives somewhat inferior accuracy than decision tree. From pragmatic results of U2R and R2L classes which have miniature training data and for which decision tree gives better performance than SVM, they said that decision tree works well with little training facts. The results also show that testing time and training time of the classifiers are to some extent better than SVM.

Giorgio Giacinto et al. [24] presented a pattern recognition approach to network intrusion detection based on the fusion of multiple classifiers is anticipated. The five decision fusion methods are assessed by experiments and their performances analyzed. The potentialities of classifier fusion for the development of this paper, proposed a multiple classifier approach based on distinct feature representation and experimented with five different fusion rules. As one of the main criticisms raised from network Security experts against previously proposed pattern recognition methods in intrusion detection was related to the high false alarm rates that such methods usually produce. This main conclusion of their work is clearly supported by the reported results, which show that fusion of multiple classifiers allows achieving a better trade-off than provided by individual classifiers between simplification capacities and false alarm generation.

Weiming Hu et al. [15] proposed an AdaBoost-based algorithm for intrusion detection. In the algorithm, decision stumps are used as scrawny classifiers. The decision rules are provided for both unconditional and continuous features. The relations between unconditional and continuous features are handled naturally without any forced conversions between these two types of features. A simple over-fitting handling is used to advance the learning results. In the explicit case of network intrusion detection, they use adjustable preliminary weights to make the tradeoff involving the detection and false-alarm rates. The experiment results show that their algorithm has a very low false-alarm rate with a high detection rate, and the run speed of their algorithm is faster in the learning stage compared with the published run speeds of the existing algorithms.

JingboYuan et al. [4] proposed a SVM classification (HTSVM) a conventional SVM classification theory is hypothesis test. Classification process, a soft edge version of the update that is compatible with the conventional SVM, but especially in the fortitude of the boundaries of the soft edge of the attribute data for hypothesis testing in preceding training. Simulation experiment results show that the intrusion detection system performance can be improved.

Mohhamadreza Ektefa et al. [6] proposed Classification tree and two leading data mining methods as support vector machine techniques to detect network intrusions. As Experimental results depict C4.5 algorithm is enhanced Detection and false alarm performance of the SVM on a data sample rate.

P Amudha et al. [7] proposed Classification and measurement in relation to the attack and hybrid attribute selection and ensemble of classifiers in order to analysis the efficiency of a series of experiments on KDD Cup'99 dataset. The experimental results, NBTree have small training data and a better detection rate and false alarm rate for R2L dataset and U2R datasets that allows for better accuracy, while the random forest, good accuracy, and detection rate, false alarm rate for DOS. They also build the model to the time taken by NBTree is further observed that compared to other classifiers. They conclude the random forest for DOS and probe dataset and NBTree for U2R and R2L dataset gives better performance.

Farah Jemili et al. [8] proposed Bayesian network intrusion detection system using an adaptive framework emphasized. Bayesian networks provide an automated search capability; they learn from the audit data and can detect both normal and abnormal connections. From audit information and can detect both normal and abnormal connections. Such Intrusions showed the high performance of their system. The system is capable to make available recommendations based on attack types which can be enhanced by integrating expert system. An additional alternative to symbolize a qualitative assessment of the risk of infiltration of possibility networks, Bayesian networks are used.

A. S. Aneethaet. Al. [9] proposed neural network for a network anomaly detection method has been and combined with the use of clustering algorithms has been proposed. Last modified self-organizing map of the SOM but for k-means that the implementation of the 1.5% increase it more than 2% higher detection rate has improved. The rate of increase in the number of output nodes, reducing the rate of learning, and learning to play a major role in the spread of the observed map found. DOS attacks, the detection rate of 98.5%, the proposed work is to identify the most effective.

H. GünesKayacıket. Al. [10] Comprehensive analysis of the relevance of a feature of the machine is used by education researchers, who are on the KDD 99 training set. As a discriminating feature to high relevance feature of the information is expressed in terms. Training for all categories of feature sets in order to measure relevance, information gain per class gain as a result of separate data for each feature, binary classification is calculated on. Recent research, decision trees, artificial neural networks, and the potential for functional classification and the remote user root and local attacks are very intricate to categorize search terms and the false alarm rate report.

Ajith Abraham et al. [11] presented two hybrid approaches for modeling IDS. Decision trees (DT) and support vector machines (SVM) are combined as a hierarchical hybrid intelligent system model (DT–SVM) and an ensemble procedure combining the base classifiers. The hybrid intrusion detection model coalesce the particular base classifiers and other hybrid machine learning paradigms to maximize detection accuracy and lessen computational complexity. The Empirical results illustrate that the proposed hybrid systems provide more accurate intrusion detection systems. In this research, they have investigated some new techniques for intrusion detection and evaluated their performance based on the benchmark KDD Cup 99 Intrusion data. Next they designed a hybrid DT–SVM model and an ensemble approach with DT, SVM and DT–SVM models as base classifiers. Pragmatic results reveal that DT gives better or equivalent accuracy for Normal, Probe, U2R and R2L classes. The Ensemble technique gave the best performance for Probe and R2L classes. The ensemble technique gave 100% accuracy for Probe class, and this suggests that if proper base classifiers are chosen 100% accuracy might be feasible for other classes too.

Heba F. Eidet al. [16] proposed intrusion detection system by using Principal Component Analysis (PCA) with Support Vector Machines (SVMs) as an approach to select the optimum feature subset. They have verified the effectiveness and the probability of the proposed IDS system by several experiments on NSL-KDD dataset. The reduction process has been used to lessen the number of features in order to lessen the complication of the system. The experimental results demonstrate of the proposed system is capable to speed up the process of intrusion detection and to minimize the memory space and CPU time cost.

S. Revathi et al. [17] proposed a new technique of combining swarm intelligence (Simplified Swarm Optimization) and data mining algorithm (Random Forest) for feature selection and reduction. SSO is used to find more fitting set of attributes for classifying network intrusions and random forest is used as a classifier. In the preprocessing step, they optimize the dimension of the dataset by the proposed SSO-RF approach and uncover an optimal set of features. SSO is an optimization method that has a strong global search capability and is used here for dimension optimization. The experimental outcome shows that the proposed approach performs improved than the other approaches for the detection of all kinds of attacks available in the dataset.

Shafigh Parsazad et al. [18] proposed a very simple and fast feature selection method to eliminate features with no helpful information on them. Result faster learning in process of redundant feature omission. They have compared their proposed method with three most successful resemblance based feature selection algorithm together with Correlation Coefficient, Least Square Regression Error and Maximal Information Compression Index. After that they used recommended features by each of these algorithms in two popular classifiers including: Bayes and KNN classifier to determine the feature of the commendation. The experimental result shows that even though the proposed method can't outperform evaluated algorithms with high differences in correctness but in computational cost it has enormous superiority over them.

L. Prema Rajeswari et al. [19] ·detection system that uses a amalgamation of tree classifiers which uses improved C4.5 which rely on labeled training data and an Enhanced Fast Heuristic Clustering Algorithm for mixed data (EFHCAM). The major advantage of this approach is that the system can be trained with unlabelled data and is accomplished of detecting formerly "unseen" attacks. Authentication tests have been carried out by using the 1999 KDD Cup data set. Since this work, it is observed that significant enhancement has been achieved from the viewpoint of both high intrusion detection rate and reasonably low false alarm rate.

Asim Das et al. [30] focused on the association rule mining in KDD intrusion dataset. Since the dataset constitutes different kinds of data similar to binary, discrete & continuous data same technique cannot be applied to determine the association patterns. Therefore, this paper used varying techniques for each type of data. The proposed method is used to generate attack rules that will detect the attacks in network audit data using anomaly detection. Rules are formed depending upon assorted attack types. For binary data; a-priori approach is used to abolish the non-frequent item set from the rules and for discrete and continuous value the proposed techniques are used.

LI Han et al. [31] used the unsupervised K-MEANS algorithm to model and detects anomaly activities. The endeavor is to advance the detection rate and lessen the false alarm rate. The K-MEANS algorithm based on information entropy (KMIE) is proposed to identify anomaly activities. KMIE can filter the outliers on the dataset to diminish the negative impact and indentify the initial cluster centers using entropy method. After that, KMIE can use these centers to iterative analyze and categorize records into different clusters. This paper used KDD CUP 1999 dataset to analysis the performance of KMIE algorithm. The results illustrate that our method has a higher detection rate and a subordinate false alarm rate which achieves expectant aim.

Devendra kailashiya et al. [32] presented a method to improve accuracy Rate of intrusion detection using decision tree algorithm. Intrusion detection systems endeavor to recognize attacks with a high detection rate and a low Error rate. In this paper they have supervised learning with preprocessing step for intrusion detection. They used the stratified weighted sampling techniques to generate the samples from inventive dataset. These sampled applied on the proposed algorithm. The accurateness of proposed model is compared with existing results in order to verify the validity and accuracy of the proposed model. The results showed that the proposed methodology gives better and vigorous representation of data. The experiments and evaluations of the proposed intrusion detection system are performed with the KDD Cup 99 dataset. The experimental results clearly show that the proposed system achieved higher Accuracy and Low Error in identifying whether the records are normal or attack one.

Yang Li et al. [33] proposed a novel supervised network intrusion detection method based on TCM-KNN (Trans-ductive Confidence Machines for K-Nearest Neighbors) machine learning algorithm and active learning based training data selection method. It can effectively detect anomalies with high detection rate, low false positives under the circumstance of using much fewer selected data as well as selected features for training in comparison with the traditional supervised intrusion detection methods. A series of experimental results on the well-known KDD Cup 1999 data set reveals that the proposed method is more vigorous and helpful than the state-of-the-art intrusion detection methods as well as can be additional optimized as discussed in this paper for real applications.

2.2 Intrusion Detection System

An intrusion detection system [1] dynamically monitors the events taking place in a system, and comes to a decision whether these events are making the sign of (disease, existence) of an attack or make up a within the law use of the system Fig. 2.1 makes picture of the organization of an IDS where solid lines, giving an idea of data/control flow, while short lined lines giving an idea of moves to intrusive things done. Efficiency is one of the major issues in intrusion detection. The effectiveness is regularly attributed to high overhead and this is caused by numerous reasons. In the middle of them are continuous detection and the use of the full feature set to look for intrusive patterns in the data set. Monitored system collected data, this data is pre-processed, IDS model recognize Intrusions and sends alarm report to security administrator. Security administrator response to monitored system and its detection process of intrusion is planned as follow in fig. 2.1 [1].

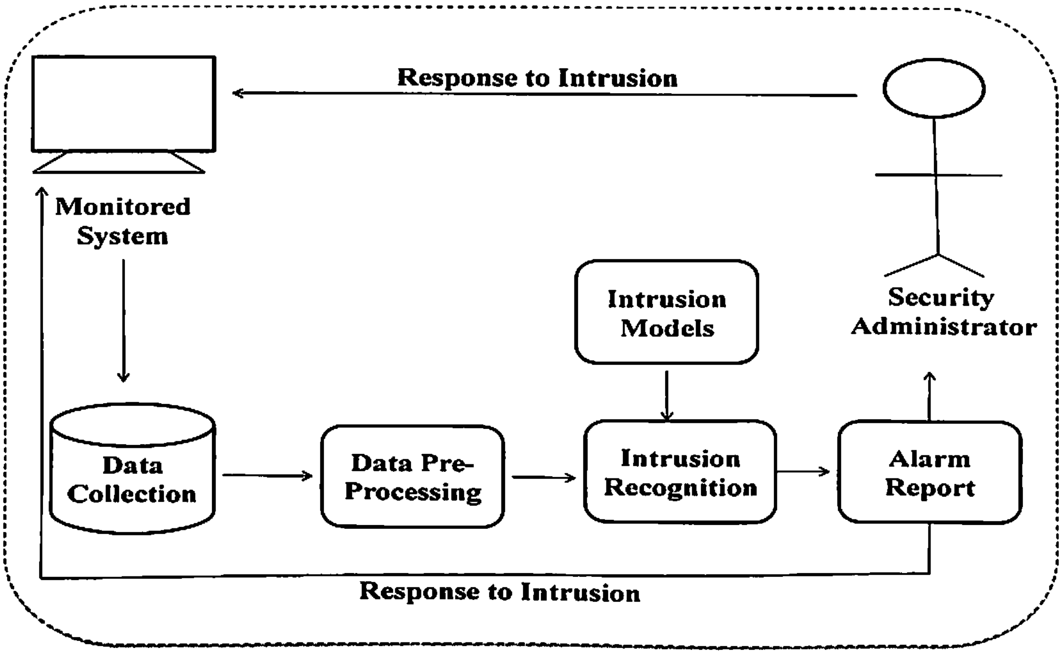


Figure 2.1: Organization of a Generalized Intrusion Detection System

The intrusion detection systems are a critical component in the security arsenal. Security is often implemented as a multilayer infrastructure and different approaches for providing security can be categorized into the following six areas [28]:

2.2.1 Attack Deterrence

Attack deterrence refers to persuade an aggressor not to commence and harass by greater than ever the seeming risk of downbeat penalty for the attacker. Having a tough legal organism may perhaps in attack anticipation. Still, it requires strong facts touching the aggressor in case an attack was launch. Research in this area focuses on methods such as those discussed in which can effectively trace the true source of attack as very often the attacks are launched with a spoofed source IP address.

2.2.2 Attack Prevention

Attack prevention aim to avoid an attack by jamming it earlier than an attack can accomplish the goal. On the other hand, it is extremely complicated to prevent all attacks. This is because, to avoid an attack; the system requires absolute knowledge of all probable attacks as well as the complete knowledge of all the allowed normal activities which is not always available. An example of an attack prevention system is a firewall

2.2.3 Attack Deflection

Attack deflection refers to tricking an attacker by manufacture the attacker believes that the attack was triumphant, while in authenticity the invader was fascinated by the system and purposely ended to disclose the attack.

2.2.4 Attack Avoidance

Attack avoidance aims to make the resource unusable by an attacker even though the attacker is able to illegitimately access that resource. An example of security mechanism for attack avoidance is the use of cryptography. Encrypting data renders the data useless to the attacker, thus, avoiding possible threat [28].

2.2.5 Attack Detection

Attack detection refers to detect an attack though the attack is still in improvement or to detect an attack which has before now occurred in the past. Detecting an attack is momentous for two reasons; first the sy^c^tem has got to recover from the damage cause by the attack and second it allows the system to take procedures to avoid equivalent attacks in future. Research in this area focuses on building intrusion detection systems.

2.2.6 Attack Reaction and Recovery

Once an attack is detected the arrangement must react to an attack and carry out the recovery mechanism as defined in the security policy.

Tools obtainable to complete attack detection followed by reaction and recovery are branded as the intrusion detection systems. Nonetheless, the difference between intrusion prevention and intrusion detection is leisurely diminishing as the present imposition detection systems gradually more focus on real-time attack detection and blocking an attack prior to it reaches the target. Such systems are better known as the Intrusion Prevention Systems.

2.3 Components of Intrusion Detection System

An intrusion detection system typically comprises of three subsystems or components [28]:

2.3.1 Data Preprocessor

Data preprocessor is responsible for collecting and providing the audit data (in a specified form) that will be used by the next component (analyzer) to make a decision. Data preprocessor is, thus concerned with collecting the data from the desired source and converting it into a format that is comprehensible by the analyzer. Data used for detecting intrusions range from user access patterns (for example, the sequence of commands issued at the terminal and the resources requested) to network packet level features (such as the source and destination IP addresses, type of packets and rate of occurrence of packets) to application and system level behavior (such as the sequence of system calls generated by a process.) They refer to this data as the audit patterns.

2.3.2 Analyzer (Intrusion Detector)

The analyzer or the intrusion detector is the core component which analyzes the audit patterns to detect attacks. This is a critical component and one of the mainly researched diverse pattern matching; machine learning, data mining and statistical techniques can be used as intrusion detectors. The capability of the analyzer to detect an attack often determines the strength of the overall system.

2.3.3 Response Engine

The responsive engine controls the reaction mechanism and determines how to respond when the analyzer detects an attack. The system may decide either to raise an alert without taking any action against the source or may make a decision to block the source for a previously period of time. Such an action depends upon the predefined security policy of the network.

2.4 Challenges and requirements for Intrusion Detection System

The principle of an intrusion detection system is to detect attacks. Conversely, it is equally essential to detect attacks at an early stage in order to minimize their impact. The major challenges and requirements for building intrusion detection systems are [28]:

1. The system must be able to detect attacks reliably without giving false alarms. It is very important that the false alarm rate is low, as in a live network with large amount of

traffic, the number of false alarms may exceed the total number of attacks detected correctly thereby decreasing the confidence in the attack detection capability of the system. Ideally, the system must detect all intrusions with no false alarms. The challenge is to build a system which has broad attack detection coverage, i.e. it can detect a wide assortment of attacks and at the same time which results in very a small number of false alarms.

2. The system must be able to handle large amount of data without affecting performance and without dropping data, i.e. the rate at which the audit patterns are processed and the decision is made must be greater than or equal to the rate of arrival of new audit patterns. Hence the speed of operation is critical for systems deployed in high speed networks. In addition, the system must be capable of operating in real-time by initiating a response mechanism once an attack is detected. The challenge is to prevent an attack rather than simply detecting it.

3. A system which can link an alert generated by the intrusion detector to the actual security incident is desirable. Such a system would help in quick analysis of the attack and may also provide effective response to intrusion as opposed to a system which offers no after attack analysis. Therefore, it is not only indispensable to detect an attack but it is also significant to recognize the type of attack.

4. It is desirable to develop a system which is resistant to attacks since, a system that can be exploited during an attack may not be able to detect attacks reliably.

5. Every network and application is different. The challenge is to build a system which is scalable and which can be easily customized as per the specific requirements of the environment where it is deployed [28].

2.5 Type of Intrusion Detection

Intrusion detection can be can be given thought to as system of ordering. The process of looking at the events taking place in a system or Network and receiving at details them for sign of intrusions is known as Intrusion detection. Intrusion detection is put in order into two makes common with a group: misuse intrusion detection and anomaly intrusion detection [1].

2.5.1 Misuse Intrusion Detection

It is pleasing designs of the attack that utilize feeblenesses in system and relevance software are used to classify the intrusions. These can be encoded in advance and used to match it being against to the user conduct to make discovery of intrusion [3]. Misuse-based IDS are also named as signature-based or pattern based IDS, which are used to make the discovery of certain things being forced into based on the attacks that stored in the knowledge-base with very low false things greater than zero It acting (play) need orders to the signatures of certain attacks and/or pattern matching of incoming packets. The discovery rate of misuse-based IDS is relatively low, because the attacker always tries to modify the basic attack sign-mark in such a way that will not match the attack sign-mark, which is already putting in the knowledge-base.

Misuse detection first consist of recording and representing the specific patterns of intrusions that exploit known system vulnerabilities or violate system security policies, then monitoring current activities for such patterns, and reporting the matches. There are several approaches in misuse detection. They differ in the representation as well as the matching algorithms employed to detect the intrusion patterns. Some systems, for example NIDES, use a rule-based expert system component for misuse detection. These systems encode known system vulnerabilities and attack scenarios, as well as intuitions about suspicious behavior into rules. For example, one such rule is: more than three consecutive unsuccessful logins within five minutes is a penetration attempt. Audit data is matched against the rule conditions to determine whether the activities constitute intrusions.

Another system, STAT, uses state transition analysis for misuse detection. It represents and detects known penetration scenarios using state 14 transition diagrams. The intuition behind this approach is that any penetration is essentially a sequence of actions that leads the target system from an initial normal state to a compromised state. Here a state in the state transition diagram is a list of assertions in terms of system attributes and user privileges. A transition is labeled by a user action (i.e., the signature action) for example, the acquisition of previously un-held privileges. Intrusions are detected in STAT when a final compromised state in the state transition diagram is reached. IDIOT uses a more formal pattern classification and matching approach for misuse detection. First, independent of the underlying computational framework of matching, the characteristics of intrusion patterns are partitioned into orthogonal categories [3]:

- Linearity: which means that the specified sequence of events must occur
- Unification: which instantiates variables to earlier events and matches these events to later occurring events, for example, the variable file2 from different audit records are bound to the same value (a file name) after the unification
- Occurrence, which specifies the relative placement in time of an event with respect to the previous events, for example, event e2 occurs within 5 seconds after event el
- Beginning: which specifies the absolute time of the beginning of a pattern
- Duration: the time duration for which an event must be active. Colored Petri Nets were used as the pattern matching model. Each intrusion signature is represented as a Petri net: linearity is represented as the sequence of 15 transitions; unification is introduced through the use of global variables; and occurrence, beginning and duration are introduced through the use of guard expressions in the Petri nets. A sequence of transitions from the start state(s) to the final state(s) constitutes a match of the intrusion signature [3].

The key advantage of these misuse detection approaches is that they can accurately and efficiently detect known attacks, those that have been coded as rules or patterns. By their nature, they are not very effective in detecting unknown attacks, those that have no matched rules or patterns, unless the new attacks employ the same system level events manifested by previously encoded exploits. Given the fact that new attack techniques are invented often, misuse intrusion systems may need to be updated frequently across many platforms. However, constructing and main-training a misuse detection system is very labor-intensive since attack scenarios and system vulnerabilities need to be analyzed and categorized, and the corresponding rules and patterns need to be carefully hand-coded and verified.

2.5.2 Anomaly Intrusion Detection

To make out the go into uses the normal user behavior to design the normal use designs are made by the arithmetical actions of the system attributes.

They have two selections to secure the system completely, either prevents the threats and vulnerabilities which come from damaging marks in the operating system as well as in the application programs are made discovery of them and take some action to prevent them in future and also get in good condition again the damage [2].

2.5.2.1 Types of Anomaly Intrusion Detection

The nature of the desired anomaly is an important point of view of an anomaly detection technique. The anomalies can be put in order into supporters groups namely: point anomaly and contextual anomaly.

A. Point Anomalies

If an individual data instance, can be given thought to as anomalous with respect to the rest of the data, then the instance is termed as a point anomaly. As a material living example, take into account credit card fraud discovery let the facts put be like to an individual's credit card transaction. For the purpose of the condition of being simple, let us assume that the knowledge of computers is formed using only one point amount made payments. A transaction for which the amount made payments is very high made an evaluation to the normal range of expenditure for that person will be a point abnormality.

B. Contextual Anomalies

If a data instance is anomalous in a specific con-text (but not if not), then it is termed as a contextual anomaly also has a relative to as conditional anomaly. The small useful things of a makes sense clearer is gotten by the structure in the data set and has given details of as a part of the hard question ordered statement of how to. Every data instance is defined using following two sets of attributes [2]:

(1) Contextual attributes

The contextual attributes are used to come to a decision about the context (or neighborhood) for that instance. For example, in spatial data groups, the longitude and latitude of a marked off are the context attributes. In time- series data, time is a contextual property which comes to a decision about the position of an instance in the entire sequence.

(2) Behavioral attributes

The behavioral attributes define the non-contextual characteristics of an instance. For example, in a spatial data set recitation the average rainfall of the whole world, the quantity of rainfall at any location is a behavioral attribute.

In table 2.1 [2] describes the advantages and disadvantages of anomaly and misuse detection.

Detection System	Advantages	Disadvantages
Anomaly Based	1. It doesn't depend on the knowledge of all kinds of possible threats. 2. It can detect also new kind of attacks, not known at the moment the IDS was configured, as anything deviating from the normal behavior triggers an alarm.	1. It is very complex to be implemented and can be heavy to be managed too, in order to define and update the normal behavior. 2. They have no detailed information about the nature, the kind of an attack (there is not a matching known attack pattern but a non-matching "normal" pattern).
Misuse Based	1. It is simpler tc be defined and updated, as you have a list of known attacks that can eventually be given by the IDS producer itself. Also incremental updates can be done in a centralized way from the producer	1. It has all the limitations of a default permit approach and can't protect the system from new threats. 2. Trying a match with all known kinds of attacks can be computationally expensive.

Table 2.1: Advantage and Disadvantage of Anomaly & Misuse Detection

2.5.2.2 Anomaly Detection Method

In this part, most commonly used networks anomaly detection methods. The methods described are rule-based approaches, finite state machine models, pattern matching, and statistical analysis [2].

Rule-Based Approaches

Early work in the region of fault or anomaly detection was based on expert systems. In expert systems, absolute knowledge-base having in it the rules of behavior of the imperfect system is used to conclude if a fault occurred. Rule-based systems are too slow for real-time applications and are dependent on preceding knowledge about the fault conditions on the network. The identification of faults in this way in depends on the symptoms that are special to a particular demonstration of a fault. Examples of these symptoms are more than enough use of bandwidth, number of open TCP connections, whole throughput exceeded etc. These rule-based systems have belief in heavily on the proficiency of the network manager and do not adapt well to the developing network environment. In this way, it is promising that completely new faults may escape detection. In, the authors portray an expert system model using fuzzy cognitive maps (FCMs) to conquer this limiting condition. FCM can be used to acquire an intelligent modeling of the propagation and interaction of network faults. FCMs are made with the nodes of the FCM symbolizing managed objects such as network nodes and the arcs denoting the fault propagation model.

Finite State Machines

The anomaly or fault detection using finite state machine model, alarm sequences that occur during and preceding to fault events. A probabilistic finite state machine model is made for a certain network fault using historical data. State machines are designed with the purpose of not just detecting an anomaly, but also possibly identifying and diagnosing the problem. The succession of alarms obtained from the diverse points in the network is modeled as the states of a finite state machine. The alarms are assumed to have within knowledge, such as the device name as well as the symptom and time of occurrence. The transitions between the states are measured using prior events. A given cluster of alarms may have a number of explanations and the goal is to find the best enlightenment among them. The best explanation is obtained by identifying a near-optimal set of nodes with a minimum cardinality such that all entities in the set give details all the alarms and at least one of the nodes in the set is the majority likely one to be in fault. In this approach, there is an underlying assumption that the alarms obtained are true. No attempt is made to produce the individual alarms themselves.

Pattern Matching

This method attempts to covenant with the unpredictability in the network environment. In this approach, online learning is used to build a traffic profile for a given network. Traffic profiles are built using symptom-specific feature vectors such as link utilization, packet loss, and the number of collisions. These profiles are then categorized by time of day, day of week, and special days, such as weekends and holidays. When newly acquired data fail to fit within some confidence interval of the developmental profiles, then an anomaly is declared. The efficiency of this pattern matching approach depends on the accuracy of the traffic profile generated. Given a new network, it may be indispensable to spend a substantial amount of time building traffic profiles. In the face of evolving network topologies and traffic conditions, this method may not scale elegantly [2].

Statistical Analysis

As the network evolves, each of the methods described above require significant recalibration or retraining. On the other hand, using online learning and statistical approaches, it is achievable to continuously track the behavior of the network. Statistical analysis has been used to discover both anomalies corresponding to network failures, as well as network intrusions. Interestingly, both of these cases make use of the standard sequential change point detection approach. The flooding detection System, which was proposed by the authors, uses measured network data that describe TCP operations to detect SYN1 flooding attacks. SYN flooding attacks capitalize on the limitation that TCP servers maintain all half-open connections. Once the queue limit is reached, future TCP connection requests are denied. The sequential change point detection employed here makes use of the nonparametric cumulative sum (CUSUM) method. Using this approach on trace-driven simulations, it has been shown that SYN flooding attacks can be detected with high precision and convincingly short detection times.

This chapter also describes the background for intrusion detection systems (IDSs). Also content along with the state of the art in pattern matching algorithms used within IDSs are important to fully grasp in order to understand the contributions of this thesis. Intrusion detection covers a broad range of digital security because IDSs have a wide range of uses. In general, these systems automate the process of extracting intelligence about past or present measures that endeavor to conciliation the confidentiality, integrity, or availability of a source. The definition of an intrusion in this context is not fixed, but rather is a concept that changes depending on the administration or objective of the system. More specifically, the intelligence and information provided by IDS is contingent upon how the system is being used, and is as important as the chosen IDS itself. Indeed, there are many ways to use IDSs. If and when IDS discovers an intrusion, regardless of how it has been defined, it is common for a system to make a record or report of the intrusion, typically by way of logging or generating an alert that is sent off [2].

2.6 Passive versus Reactive IDS

The passive IDS simply detect and alert when suspicious or malicious traffic is detected an alert is generated and sent to the administrator or user and it is up to them to take action to block the activity or respond in some way.

Reactive IDS will not only detect suspicious or malicious traffic and alert the administrator, but will take pre-defined proactive actions to answer to the threat. Normally this means blocking a in a passive system the IDS detects an intrusion and then alerts the user in some way. There is several different ways IDS can do this.

Examples:

- It can in some way show the alert in the user's GUI, for example as a message in a console.
- It can log the event in detail
- It can in some external way notify the user; email, sms, pager etc.

In a reactive system the IDS does something more concrete when an intrusion is detected.

Examples:

- It can block the intruder access to the system, for example, with reconfiguration of routers/firewall's ACL lists
- Reset the TCP connection
- In host based IDS disable the user account of the intruder or just terminate the user's session
- Trace the origin of the intruder

Some also define a third version which is called proactive IDS. In proactive IDS the system doesn't wait for the intrusion to happen and then reacts. It stops the intrusion happening altogether before it has succeeded in doing its evil work. Proactive IDS is also sometimes used as another name for reactive IDS.

2.7 Host-Based versus Network-Based IDS

The difference between host- and network-based intrusion detection is marking the location of the system and most importantly its input. Systems that can handle both types of input are called hybrids or distributed intrusion detection systems (DIDSs) [5]. A host-based intrusion detection system (HIDS) consists of an application, generally software, on a machine that is designed to inspect input actions that are internal to the machine like system calls, application and audit logs, file-system amendment and other host activities and states. A commonality often seen in HIDSs is the use of an object or a checksum database that catalogs 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 to not set off alarms during the r :xt HIDS scan. For this reason a HIDS database needs to be strong, often cryptographically, protected.

Host-based versus Network-based IDSs 10 pendent network appliance or device tapped into the network with associated processing capabilities. It monitors network activity, and therefore, its input is solely in the form of the traffic on the network. Since frequently attacks on networks or machines within them originate outside of the network in question, NIDSs have a wide range of possible attacks to detect from the outside (ingress).

These typically include but are not limited to denial of service (DoS) attacks, port-scans, spreading viruses, and attempts to break into or exploit vulnerabilities in computer systems by malicious individuals, worms, or other malware self-spreading on the network. However, NIDSs can also help to warn about or guard against sensitive data and attacks within the network or leaving the relevant network [5].

2.7.1 Network Based Monitoring

Advantages:

- Reduces the processing overload from the mobile phone.
- Detects external intrusions.

Disadvantages:

- No access to monitoring data on the mobile phone that is useful for detection
- Communication environment is very fragmented as a mobile device can be connected to multiple sources on multiple interfaces at the same time
- Cannot detect intrusions on the device itself like malware etc.
- Collecting all relevant networks based monitoring data for all networks and communication interfaces that the mobile device interacts with may be very difficult.

2.7.2 Host Based IDS

Advantages:

- Have access to private information on the mobile device that is useful to detect intrusions as the information collected from the mobile device will reflect the device behavior accurately.
- Intrusion detection models with host-based data collection provide more accurate and reliable results than other approaches [3].

Disadvantages:

- Difficult to implement because of the processing limitations of mobile phones.
- Difficult to provide security for the data that is directly collected from the mobile device.

CHAPTER - 3

DATA MINING & MACHINE LEARNING TECHNIQUE

3.1 Data Mining

Data mining, a process of analyzing data to identify patterns or relationships for autonomously extracting useful information or knowledge, has been increasingly developed to provide solutions for uncovering useful and/or unexpected information from large volumes of data in various research areas, such as multimedia data analysis, visualization, biomedicine, market analysis, homeland defense, threat assessment systems, intrusion detection, credit card fraud detection, and other applications. Data mining can also be described as the process of extracting hidden patterns from data. As more data is gathered, with the amount of data doubling every three years, data mining has become an increasingly imperative tool to renovate this data into information. It is generally used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery.

Data mining can be applied to data sets of any size. However, while it can be used to uncover hidden patterns in data that have been collected, obviously it can neither uncover patterns which are not already present in the data, nor can it uncover patterns in data that have not been collected. Among various data mining techniques, classification is important as it is one of the key techniques in most data mining applications. It is the act of distributing things into classes or categories of the same type or characteristics. In other words, it is a basic cognitive process of arranging things or objects into classes or categories. A classification process is obtained from given training sets and then tested on test dataset. For multivariate data, a classification procedure predicts different output patterns [3].

Classification techniques have been applied to numerous research areas including market analysis, homeland defense, threat assessment systems, intrusion detection systems, credit card scam detection, face identification, etc. Generally speaking, there are two broad types of classification procedures: supervised and unsupervised classification. Supervised classification can be defined when the output patterns and their range of values are given.

As its counterpart method, unsupervised classification can be defined when none or only part of the output pattern information is known before the classification process Supervised classification takes into use a class quantity that is high enough to distinguish a class from other class types. In supervised classification, classification accuracy or detection rate is most important performance evaluation measure of a classifier. On the other hand, operation merits, another important performance evaluation measure, refer to the usage benefits of the classifier in terms of how fast it executes, how much memory it consumes, how large the programming code is, being lightweight, and so on. For instance, a classifier may have a high detection rate or high accuracy, but is slow to execute and/or requires a considerable amount of storage space to save, for instance, generated decision rules; whereas, another classifier may execute faster and use some form of heuristic that does not require so many rules to be stored. All these merits taken together give the overall performance merit for a classifier. When relatively little information is known about the data before classification, unsupervised classification is usually employed. In addition, no human effort is required to provide the foreknowledge of the class labels of the data set. This is the reason behind, clustering algorithms being used to aggregate and classify data instances into classes while perforr:ing unsupervised classification. Unsupervised classification is highly efficient and useful when real-world applications are considered because usually there is very less class-related information available for them.

The figure 3.1 depicts the data mining process for intrusion detection [7].

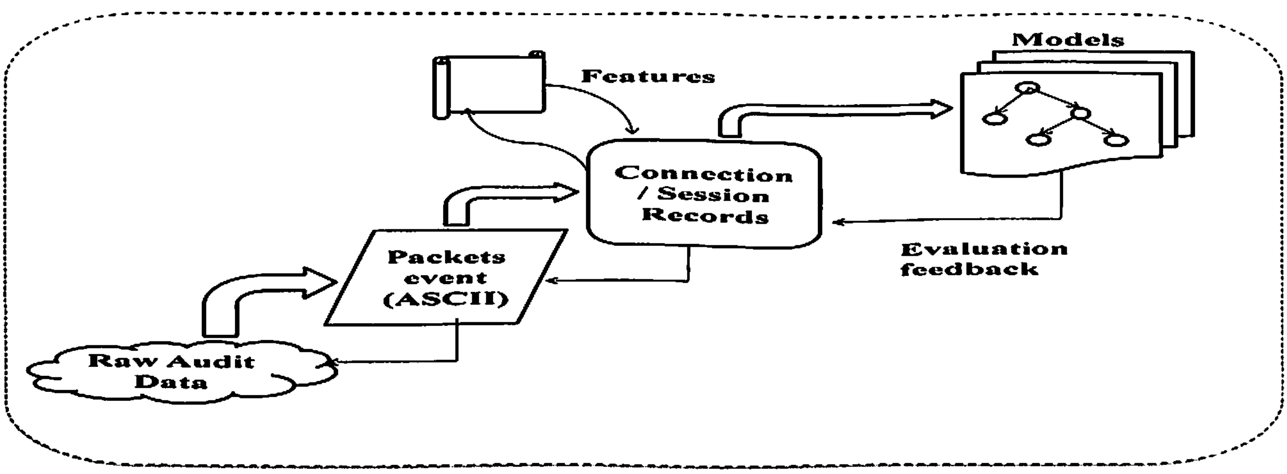


Figure 3.1: Data Mining Process of Intrusion Detection

Data mining approaches for intrusion detection was first implemented in Mining Audit Data for Automated Models for Intrusion Detection. The data mining process of constructing intrusion detection models is depicted in the figure: 3.1. Data mining algorithms used in this approach are RIPPER (rule based classification algorithm), meta-classifier, frequent episode algorithm and association rules. RIPPER classification algorithm is then used to learn the detection model. Meta classifier is used to learn the correlation of intrusion substantiation from multiple detection models and generate combined detection model [7].

In the training segment database of persistent item sets is created for the attack-free items from using only attack-free data set. This serves as a profile against which frequent item sets bring into being soon after willed be compared. Unfamiliar attacks are the ones which are not expert to establish either as false alarms or as famous attacks. This system attempts to determine only anomaly attacks [6].

3.2 Machine Learning

Data mining and machine learning methods focus on analyzing the properties of the audit patterns rather than identifying the process which generated them. These methods include approaches for mining association rules, classification and cluster analysis. Classification methods are one of the most researched and contain methods like the decision trees, Bayesian classifiers, artificial neural networks, k-nearest neighbor classification, support vector machines and others. There is a tremendous explosion in the amount of data that organizations generate, collect and store. Managers are beginning to recognize the value of this asset, and are increasingly relying on intelligent systems to access, analyze, summarize, and interpret information from large and multiple data sources. These systems help them make critical business decisions faster or with a greater degree of confidence [6]. Data mining is a promising new technology that helps bring business intelligence into these systems. While there is a plethora of data mining techniques and tools available, they present inherent problems for end-users, including complexity, required technical expertise, lack of flexibility and interoperability, etc. Data mining, a process of analyzing data to identify patterns or relationships for autonomously extracting useful information or knowledge, has been increasingly developed to provide solutions for uncovering useful or unexpected information from large volumes of data in various research areas such as multimedia da.a analysis, visualization, bio-medicine, market analysis, homeland defense, threat assessment systems, intrusion detection, credit card fraud detection and other applications. Data mining can also be described as the process of extracting concealed patterns from data. Since more data is gathered with the amount of data doubling every three years, data mining has become an increasingly significant tool to convert this data into information. It is generally used in a wide range of profiling practices, such as marketing, supervision, fraud detection, and scientific discovery. Data mining can be functional to data sets of any size. Nevertheless, while it can be used to expose hidden patterns in data that have been collected apparently, it can neither expose patterns which are not by now present in the data nor can it expose patterns in data that have not been collected. It is methodical process designed to investigate data (usually large amounts of data - normally business or market related) in search of regular patterns and/or methodical relationships between variables and then to authenticate the findings by applying the detected patterns to novel subsets of data. The decisive target of data mining is prediction. Predictive data mining is the most ordinary type of data mining and one that has the most direct business applications. The progression of data mining includes of three phase: (1) the preliminary exploration, (2) model building or pattern recognition with validation/verification and (3) exploitation (i.e., the application of the model to new data in order to generate predictions). These steps are explained beneath [27].

Phase 1: Exploration

This stage typically starts with data preparation, which may entail cleaning data, data transformations, selecting subsets of records and - in case of data sets with large numbers of variables ("fields") - performing some prelude feature selection operations to bring the number of variables to a expedient range (depending on the statistical methods which are being considered). After that, depending on the nature of the analytic problem, this first stage of the process of data mining may engage everywhere between a simple choice of straightforward predictors for a regression model, to sophisticate exploratory analyses using a extensive assortment of graphical and statistical methods in order to recognize the most relevant variables and conclude the complexity and/or the all-purpose nature of models that can be taken into consideration in the next stage.

Phase 2: Model building and validation

This phase involves considering different models and selecting the best one based on their predictive performance (i.e., explaining the changeability in question and producing stable results across samples). This may sound like a simple operation but in fact, it occasionally involves a very convoluted process. There are a variety of techniques developed to accomplish that objective, many of which are based on so-called "competitive appraisal of models," that is, applying different models of the matching data set and then comparing their performance to prefer the best. These techniques - which are again and again considered the core of predictive data mining comprise: Bagging (Voting, Averaging), Boosting, Stacking (Stacked Generalizations) and Meta-Learning.

Phase 3: Deployment

That concluding phase involves using the model selected as best in the preceding phase and applying it to novel data in order to generate predictions or estimates of the expected outcome [27].

3.2.1 Classification

Surrounded by different data mining techniques, classification is an imperative as it is one of the key techniques in most data mining applications. It performs distributing things into unit or categories of the similar type or characteristics. In other words, it is a basic cognitive process of arranging things or objects into classes or categories. We have a Training set surrounding data that have been formerly categorized. Based on this training set, the algorithm finds the group that the new data points belong to. Because a Training set exists, we illustrate this technique as supervised learning. Predicts categorical class labels – Classifies data (constructs a model) based on a training set and the values (class labels) in a class label attribute – Uses the model in classifying novel data.

Example: We use a training dataset which categorized customers that have agitated. Now based on this training set, they categorize whether a customer will agitate or not.

A classification procedure is erudite from giving training dataset then tested on test dataset. For multivariate data, a classification procedure predicts different output patterns. Classification techniques have been applied to numerous research areas including market analysis, homeland defense, threat assessment systems, intrusion detection systems, credit card scam detection and face identification, etc. Generally speaking there are two broad types of classification procedures: supervised and unsupervised classification.

3.2.1.1 Supervised & Unsupervised Classification

Supervised classification can be defined when the output patterns and their range of values are given. As its counterpart method, unsupervised classification can be defined when none or only part of the output pattern information is known before the classification process Supervised classification takes into use a class quantity that is high enough to distinguish a class from other class types [27]. In supervised classification, classification accuracy or detection rate is most important performance evaluation measure of a classifier. On the other hand, operation merits, another important performance evaluation measure, refer to the usage benefits of the classifier in terms of how fast it executes, how much memory it consumes, how large the programming code is, being lightweight, and so on. For instance, a classifier may have a high detection rate or high accuracy, but is slow to execute and/or requires a considerable amount of storage space to save, for instance, generated decision rules; whereas, another classifier may execute faster and use some form of heuristic that does not require so many rules to be stored. All these merits taken together give the overall performance merit for a classifier.

When relatively little information is known about the data before classification, unsupervised classification is usually employed. In addition, no human endeavor is required to make available the foreknowledge of the class labels of the data set. This is the reason behind clustering algorithms being used to aggregate and classify data instances into classes while performing unsupervised classification. Unsupervised classification is highly efficient and useful when real-world applications are considered because usually there is very less class-related information available for them.

3.2.2 Classifiers

An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. Classifiers are used to classify the data based on a learning model. There are many classifiers in data mining such as naive Bayes classifier, decision tree, support vector machine, neural network, K nearest neighbor [6].

3.2.3 Clustering

Clustering of data has been applied extensively for intrusion detection using a number of methods such as k-means, fuzzy c-means and others. Clustering methods are based upon calculating the numeric distance of a test point from different cluster centers and then adding the point to the closest cluster. One of the main drawbacks of clustering technique is that since a numeric distance measure is used, the observations must be numeric. Observations with symbolic features cannot be readily used for clustering, which results in inaccuracy. In addition, clustering methods consider the features independently and are unable to capture the relationship between different features of a single record which results in lower accuracy. Another issue when applying any clustering method is to select the distance measure as a different distance measures result in clusters with different shapes and sizes. Frequently used distance measures are the Euclidian distance and the MAHALANOBIS distance. Clustering can, however, be performed in case only the normal audit patterns are available. In such cases, density based clustering methods can be used which are based on the assumption that intrusions are rare and dissimilar to the normal events. This is similar to identifying the outlier points which can be considered as intrusions.

3.2.4 Bayesian Classifiers

Naive Bayes classifiers are also proposed for intrusion detection. However, they make strict independence assumptions between the features in an observation resulting in lower attack detection accuracy when the features are correlated, which is often the case. Bayesian network can also be used for intrusion detection. However, they tend to be attack specific and build a decision network based on the special characteristics of individual attacks. As a result, the size of a Bayesian network increases rapidly as the number of features and the type of attacks modeled by the network increases.

3.2.5 Decision Trees

Decision trees have also been used for intrusion detection. Decision trees select the best features for each decision node during tree construction based on some well defined criteria. One such criterion is the gain ratio which is used in C4.5. Decision trees generally have very high speed of operation and high attack detection accuracy and have been successfully used to build effective intrusion detection systems [5].

3.2.6 Artificial Neural Networks

Neural networks contain extensively to build network intrusion detection systems. Though, the neural networks can effort with noisy data, like other methods, they entail huge data for training and it is often hard to select the most first-rate potential structural design for the neural network. Neural networks included in anomaly intrusion detection as well as in misuse intrusion detection [9].

3.2.7 Support Vector Machines

Support vector machines map real valued input feature vector to higher dimensional feature space through nonlinear mapping and have been used for detecting intrusions. They can provide real-time attack detection capability, deal with large dimensionality of data and perform multi class classification. For data mining and machine learning based approaches, the accuracy of the trained system also depends upon the amount of audit patterns available during training. Generally, training with more audit patterns result in a better model. The above discussed methods often deal with the summarized representation of the audit patterns and may analyze multiple features which are considered independently. The prime reason for working with summary patterns is that the system tends to be simple, efficient and give fairly good attack detection accuracy. Similar to the pattern matching and statistical methods, these methods assume independence among consecutive events and hence do not consider the order of occurrence of events for attack detection.

3.2.8 Markov Models

Markov chains and hidden Markov models can be used when dealing with sequential representation of audit patterns. Hidden Markov models have been shown to be effective in modeling sequences of system calls of a privileged process, which can be used to detect anomalous traces. However, modeling system calls alone may not always provide accurate classification as various connection level features are ignored. Further, hidden Markov models cannot model long range dependencies between the observations. However, in order to gain computational efficiency the multivariate data analysis problem is broken into multiple univariate data analysis problems and the individual results are combined using a voting mechanism. This, however, results in inaccuracy as the correlation between the features is lost. The authors showed that modeling the ordering property of events, in addition to the duration and frequency, results in higher attack detection accuracy. The drawback with modeling the ordering of events is that the complexity of the system increases, which affects the performance of the system. Hence, there is a tradeoff between the detection accuracy and the time required for attack detection [9].

3.2.9 Others

Other approaches for detecting intrusion include the use of genetic algorithm and autonomous and probabilistic agents. These methods are generally aimed at developing a distributed intrusion detection system.

3.2.9.1 K- Nearest Neighbors

The first machine-learning algorithm we'll look at is k-Nearest Neighbors (KNN). K-NN is a type of instance based learning. K-NN algorithm is amongst the simplest of all Machine learning algorithms. Object is classified by a Majority vote of its neighbors. There is main two parts Training Sets and Test Sets [12].

KNN procedure

1. Store all input data in the training set.
2. for each pattern in the Test set.
3. Search for the K nearest patterns to the input pattern using a Euclidean Distance Measure.

4. For classification computes the confidence for each class as Ci/K.

3.3 Data Mining Learning Techniques

3.3.1 Decision Tree

Decision Trees as predicament where each relationship or user is taken to be moreover as one of the attack types or normal based on having existing knowledge four computers. Decision trees can get answers to this classification problem of intrusion detection as they be trained the model from the data set and can classify the new data item into one of the classes specified in the data set. Decision trees learn a model based on the training data and can be used as misuse intrusion detection and can predict the future data as one of the attack types or normal which is based on the learned model.

In data mining, decision tree induction is one of the classification algorithms. The Classification algorithm constructs a model from the pre classified data set which is learned inductively. Each data item is defined by the values of the attributes. Classification may be viewed as a mapping from a set of attributes to a particular class. The Decision tree classifies the given data item using the values of its attributes. The decision tree is at first started making from a group of pre-classified data. The main way in is to select the properties, which best makes a division the data items into their classes. According to the values of these attributes the data items are division into parts. This process is recursively applied to each partitioned subset of the data items. The process puts an end to when all the data items in the current subset belong to the same class. A node of a decision tree gives details of an attribute by which the data are to be divisions into parts. Each node has a number of edges, which are labeled according to a possible value of the attribute in the parent node. An edge makes connection either two nodes or a node and a leaf. Leaves are labeled with a decision value for grouping of the data [5]. Technology decision tree is a common method of classification, intuitionist and quickly. The Construction process is top-down allocation and based on dividing and rule mechanism. It is important, greedy algorithms. Start from the root node for each node is non leaf node, first select an attribute to try to set an example; Second, a model train set into several sub-sample set, according to the experimental results, a set of samples creates a new leaf node. Third, repeat the above divisional process, until having reached specific end conditions. In practice, because the size of the sample set of training is usually large tree branches and layers of more. In addition, abnormal noise occurs in the sample set of training will also cause some branches are unusual, so we need to prune the decision tree. One of the greatest advantages of the decision of the algorithm, the classification tree is that: it does not require the user to know that a lot of background knowledge in the learning process. J48 algorithm is the most representative and widely used. It was proposed by Quinlan in 1993. J48 is a one-class classifier so it gives best result for one-class. They used the C4.5 algorithm which is updated algorithm of j48 [3]. The advantages and disadvantages of decision tress is described in table 3.1 [3].

Advantage	Disadvantage
1. Decision tree works well with large data sets.	1. Decision tree algorithm are unstable.
2. High performance of decision tree makes them useful in real time intrusion detection.	2. The tree created from numeric dataset can be complex.
3. Generalization accuracy-ability to detect new intrusions.	3. Limited to one o/p attributes.

Table 3.1: Advantages and Disadvantage of Decision Tree

3.3.2 C4.5 Decision Tree Algorithm

It is a classification algorithm in data mining. Based on the assumptions deduced, it tries to classify the new data set. It is also called as Classification tree algorithm. In this algorithm, there is a root node, and then comes internal nodes on which the tests are performed. On getting the result we reached to the leaf node which describes the ultimate result. On the base of attributes, given data items are classified by the decision tree algorithm. Initially a decision tree is constructed with the help of pre-classified data set. Each and every data item has some set of attributes, which has some value on which they are defined. Selection of attribute is the key issue, as we have to select the best attribute that divides the data item into corresponding classes. The partitioning of a data item is made on the basis of the values of the attributes of the data element. This process is applied to every part of data items. When all the data items are categorized together that is of the same class the process gets terminated. At the end the name of leaf node is the result of classification. C4.5 algorithm can deal with continuous attributes, missing attribute's value, and gives computational efficiency. Nodes, leaves and edges make a decision tree. Node describes that attributes on the basis on which the partitioning of the data takes place. Every node comprises of several edges. According to the values of edges, values of attributes in parent node, the labeling is done. Two nodes or node and leaf are joined together with an edge.

Figure 3.2 shows the basic and common example of decision of playing depending upon the conditions of weather [5].

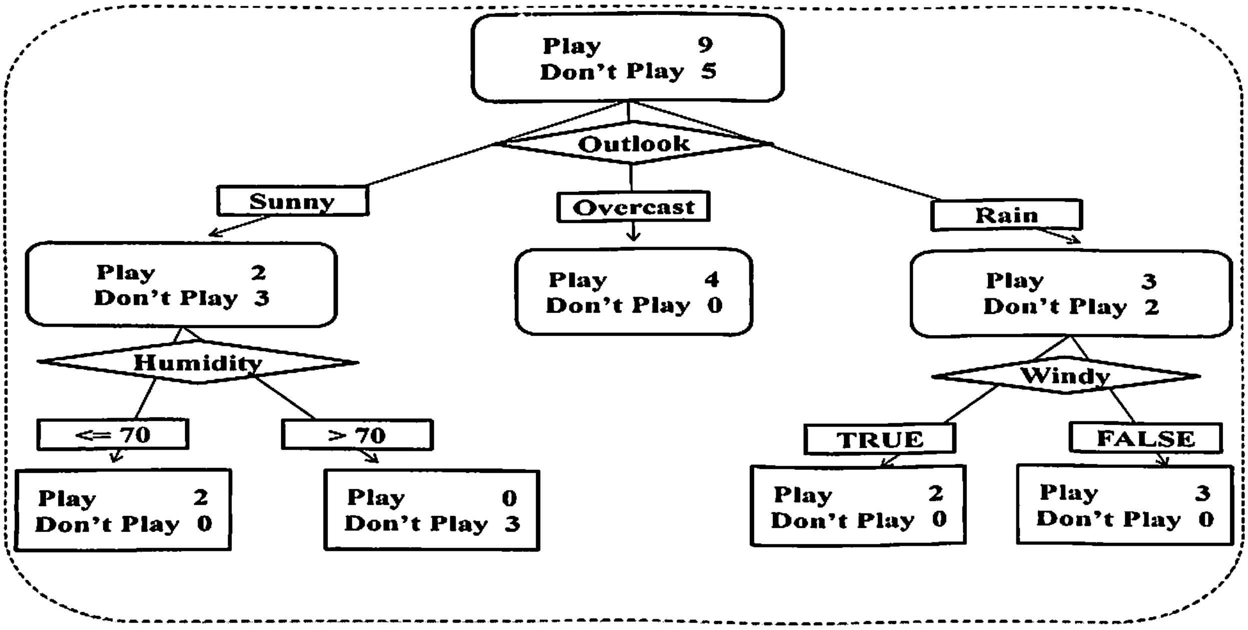


Figure 3.2: Decision Tree of Weather Data

3.3.3 Support Vector Machine as Intrusion Detection Model

Support Vector Machines have been projected as a novel technique for intrusion detection. A Support Vector Machine (SVM) maps input material valued point gives directions to be taken into a higher to do with measures point space through some nonlinear mapping [5]. SVM tools are powerful for providing solutions to classification, regression and density estimation problems. These are undergone growth on the principle of structural risk minimization. Structural risk minimization seeks to find a speculation for which one can get lowest probability of error. The structural risk minimization can be achieved by finding the hyper plane with greatest possible separable amount in addition to the data. Binary classification problems can be handled using SVM. It uses a feature called, kernel function for mapping. Kernel functions like polynomial, radial basis function are used to separate the feature space by making a hyper plane. The kernel functions can be used at the time of training of the classifiers which selects support vectors along the surface of this function. SVM classifies data by using these support vectors that outline the hyper plane in the feature space. In this figure 3.3 support Vector Machines is applied as novel intrusion detection [5].

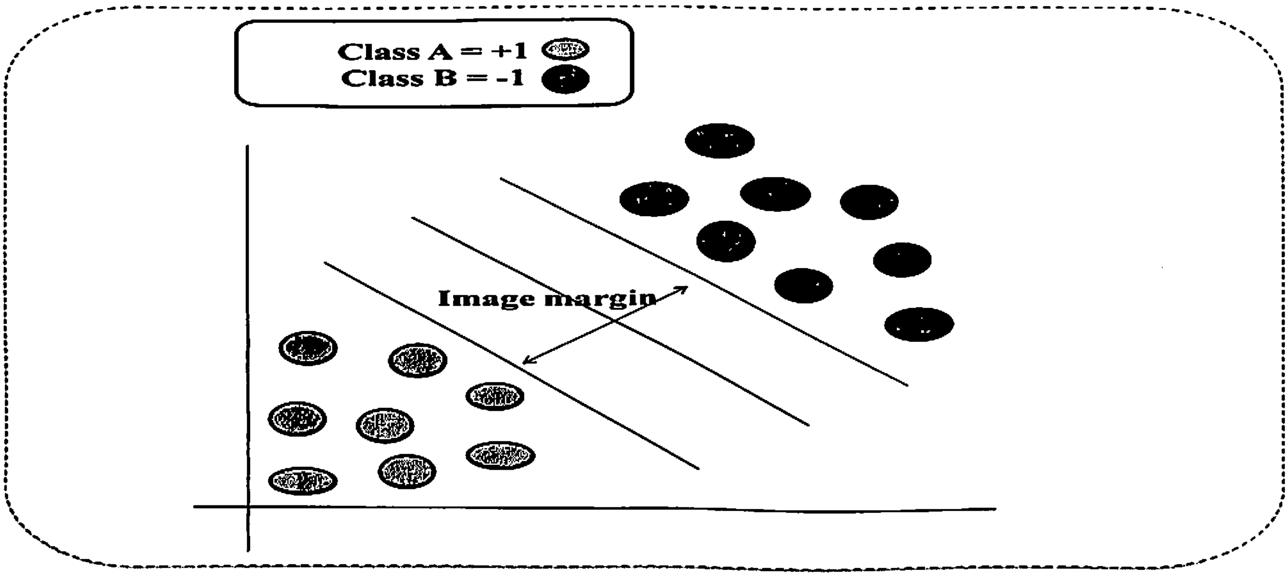


Figure 3.3: Classifications through SVM

The Support Vector Machine (SVM) maps to evaluate the subject gives instructions input point to be made in higher space to the point measures through some nonlinear mapping. SVM is a powerful tool for finding the solutions to the problem of evaluating classification, regression and density. The framework builds on the principle of reducing risk. They speculated that a low probability of error approach in order to try to reduce the risk [5].

With the help of multi class SVM, it can classify attacks of different class. SVM uses non-linear mapping that maps the real values into higher dimensional feature space. The linear separating hyper plane is used by SVM for the creation of the classifier. Through the use of hyper-plane SVM separates the data into different classes. There is an attribute that is called as the kernel that SVM uses for solving the problem. The User has to provide the kernel function at the training phase of the algorithm. With the help of support vectors, SVM does the classification. There are many kernel functions like linear, radial basis functions, polynomial, sigmoid.

They solved the multiclass classification problem in SVM by using one-against-one method. In this method they constructed n(n-1)/2 classifiers where each classifier is trained on data from two classes."One against one" strategy, which is also known as "Pair wise Coupling", "all pairs" or "round-robin", each pair consists of the construction of an SVM Classes. Usually, the classification of an unknown sample is done by the maximum voting, where each SVM classifier Votes for one class. The advantages and disadvantages of SVM are described in table 3.2 [5].

3.3.4 Ensemble Technique

The ensemble approach [29] is a relatively new trend in artificial intelligence in which several machine learning algorithms are combined. The main idea is to exploit the strengths of each algorithm of the ensemble to obtain a robust classifier. Ensembles are particularly useful when a problem can be divided into sub problems. In this case, each module of the ensemble, which can include one or more algorithms, is assigned to one particular sub problem. Classification of the ensemble methods are shown in figure 3.4 [29].

ADVANTAGE	DISADVANTAGE
1. Speed of SVM, as the capability of detecting intrusion in real time.	1. SVM can only handle binary-class classification, whereas intrusion detection requires multi-class classification.

2. SVM can learn a larger set of pattern.	2. If the number of features is much greater than the number of samples, the method is likely to give poor performances.
3. SVM also has ability to update the training pattern dynamically.	3. The biggest limitation of the support vector approach lies in the choice of the kernel.

Table 3.2: Advantages and Disadvantages of SVM

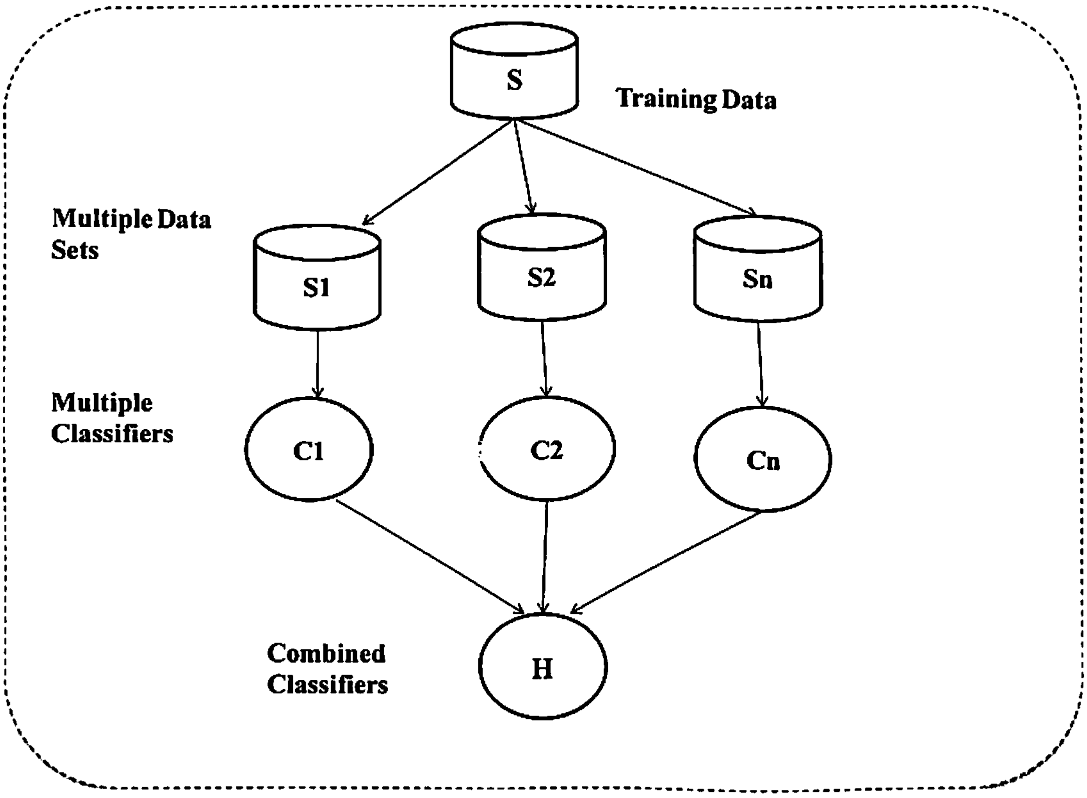


Figure 3.4: Classifications with Ensemble Method

In this figure 3.4 ensemble method multiple learners are trained to resolve the identified problems, where is the machine learning paradigm. The generalization capability of an ensemble is regularly stronger than the base learners. It can make very accurate predictions of the strong learners slightly better than a random guess is; the weak learners are able to increase. Therefore, the basis of learners and "weak learners" are called. In general, Base learners, the decision tree, neural network or other types of machine learning algorithms may be based on learning algorithm that is generated from the training data. There are many ensembles learning technique which is Boosting, Bagging, Stacking. We are using BOOSTING ensemble learning technique [29].

A supervised machine learning task involves constructing a mapping from input data (normally described by several features) to the appropriate outputs. In a classification learning task, each output is one or more classes to which the input belongs. The goal of classification learning is to develop a model that separates the data into the different classes, with the aim of classifying new examples in the future. For example, a credit card company may develop a model that separates people who defaulted on their credit cards from those who did not based on other known information such as annual income. The goal would be to predict whether a new credit card applicant is likely to default on his credit card and thereby decide whether to approve or deny this applicant a new card. In a regression learning task, each output is a continuous value to be predicted (e.g., the average balance that a credit card holder carries over to the next month). Many traditional machine learning algorithms generate a single model (e.g., a decision tree or neural network). Ensemble learning methods instead generate multiple models. Given a new example, the ensemble passes it to each of its multiple basemodels, obtains their predictions, and then combines them in some appropriate manner (e.g., averaging or voting). As mentioned earlier, it is important to have base models that are competent but also complementary [11].

3.3.4.1 IDS Using Ensemble Methods

The main idea of ensemble method is to build multiple classifiers and then integrate the outputs of all classifiers to make decisions for the final outcome. The core purpose of the ensemble is to increase classification accuracy and decrease error rate. Because each type of classifier can produce different results, ensemble method takes advantage of the strong points of each individual classifie- to induce a better final outcome. There are many types of ensemble proposed in the machine learning literature. With respect to architecture, individual classifiers can in general be structured in forms of parallel (e.g., bagging), sequential (e.g., boosting), or hybrid. For making a decision, the composer of classifiers can apply various mechanisms such as majority voting, Bayesian combination, distribution summation, entropy weighting, and so on. Many studies have applied the diversity of ensemble methods to the intrusion detection problem [22], [23], [24], [25], [26], [29]. It is worth noting that most of the studies report that ensemble method considerably enhances the efficiency of 'rareclass' detection and anomaly detection. Giacinto et al. present an ensemble system, including three groups of classifiers that correspond to three subsets of features (i.e., intrinsic features, traffic features, and content features) [27]. Each group of classifiers is trained from one out of the three above feature subsets. Then, three simple fusion functions (i.e., majority vote, average, and belief) are employed for aggregation. A subsequent work of the same authors describes ensemble architecture, including multiple one-class k-means classifiers. Each classifier is trained from a training subset containing a specific attack type belonging to a specific attack class (e.g., Neptune is one of twenty one attack types and belongs to a DoS attack class in the KDD99 dataset).

The process of ensemble is based on the Decision Template method. The proposed architecture aims at labeling a given instance to belong to a normal or known attack class, and thus is called misuse detection. In another later study, Giacinto et al. propose an ensemble model, based on a modular multiple classifier system, to handle one-class classification (unlabeled training data) for anomaly detection [26]. The proposed model consists of various modules. Each module is specifically designed for each different service group. For example, the mail module contains all service related to mail, i.e., SMTP, POP2, POP3, NNTP, and IMAP4. More specifically, each module corresponds to a disjointed training subset that is used to train one individual classifier or more different classifiers. One-class classifiers employed in this study are Parzen, k-means, and v-SVC. Finally, the outputs of all modules are combined using some fusion functions such as min, max, mean, and majority vote. More adaptively, Abadeh et al. Employ Fuzzy Logic Theory to develop an ensemble method.

This study introduces a parallel genetic local search algorithm to generate fuzzy rule sets for each class label in the training set. Each of these rule sets is utilized to build a fuzzy classifier. Then, a decision fusion procedure is in charge of determining a class label for a given instance. Comparably, Zainal et al. describe an ensemble model that utilizes three different learning algorithms (classifiers), i.e., linear genetic programming, neural fuzzy inference system, and random forest [24]. Each classifier is trained by the same training set and assigned to a weight calculated given the strength of the classifier. For decision making, a composer of classifiers determines a class label for a given instance, according to the weights of classifiers. Xiang et al. build a multi-level hybrid model by combining two techniques, i.e., supervised decision tree and unsupervised Bayesian classification [23]. The classifier model is hierarchically structured in forms of class labels in the training set.

By experimenting on the KDD99 data set, the authors motivated that the model is especially efficient in improving false negative rate compared to other methods. Likewise, Peddabachigari et al. develop a hybrid intrusion detection model by combining DT and SVM. In this model, the training set is passed through the DT classifier to generate leaf-node information. Then, the SVM classifier is trained using the training set together with leaf-node information (as an additional attribute) to produce the final output [3]. Apart from other methods that build classifiers from network header data, they introduce a multiple classifier system for anomaly detection given network payload data [26]. This ensemble system comprises several one-class SVM classifiers. In this study, different compact representations of payload in different feature spaces are obtained by applying a dimensionality reduction algorithm. Then, each one-class SVM classifier is trained by using these different representations of payload. Given the outputs of classifiers, a final decision is made by applying some fusion functions (e.g., average, product, majority vote). The experiment is conducted on three datasets, i.e., Attack-Free DARPA Dataset, Attack-Free Gatech (a dataset of Georgia Institute of Technology), and HTTP-Attack Dataset. Based on ROC Curve Graph, the detection rate of the proposed method fluctuates from 80% to 99%. Zhang et al. apply a Random Forest Algorithm, an ensemble method, to intrusion detection. Random Forest produces a forest of classification trees in which each tree is built from a different bootstrap sample. Instead of using the class label attribute of training datasets for classification analysis, the proposed method only uses the attribute service type (e.g., HTTP, FTP) as the purpose of classification. In misuse intrusion detection, a given instance pass through the trees and then a 'majority vote' mechanism are applied to label this instance. For outlier detection, the general idea is that if an instance is classified as the one that is different from its own service type, then this instance is regarded as an outlier. For example, if an HTTP connection record is classified as FTP service type, this connection record is determined as an outlier. More diversely, Makkamala et al. build an ensemble model using five classifiers, i.e., resilient back propagation NN, scaled conjugate gradient NN, one-step-secant NN, SVM, and multivariate adaptive regression spline. All these five classifiers are operated independently and concurrently. In this model, a final classification decision is made by majority voting. Likewise, Zainal et al. present an ensemble model of three one-class classifiers, i.e., neuro-fuzy inference, linear genetic programming, and random forest classifiers. In this model, weighted voting is used to make a final decision [25].

CHAPTER - 4
METHODOLOGY

In this section we present our methodology for the detection of intrusion. The feature reduction of intrusion is applied on KDD'99 dataset using information gain and then classification done via KNN-ACO and compared with the SVM, KNN. The overview of the methods uses is described below:

The overview of the SVM (support vector machine) is already discussed in chapter 2; SVM is powerful machine learning approach for providing good solutions to classification, regression and mass assessment problems.

4.1 K- Nearest Neighbor (KNN)

A more sophisticated approach, k-nearest neighbor (KNN) classification is to find a group of k Patterns in the training set that are adjoining to the test pattern and bases the obligation of a label on the preponderance of a meticulous class in this neighborhood. This addresses that in many data sets, it is unlikely that one pattern will exactly match another, as well as the truth that differing informªtion about the class of a pattern may be provided by the patterns closest to it. There are many key elements of this model [12]:

(1) The set of labeled patterns to be used for evaluating a test pattern's class,

(2) A remoteness or correspondence metric that can be used to compute the closeness of patterns

(3) The value of k, the number of nearest neighbors and

(4) The method used to determine the class of the target pattern based on the classes and distances of the k nearest neighbors.

In its simplest form, KNN can involve assigning a pattern of the class of its nearest neighbor or of the majority of its nearest neighbors Generally, KNN is a special case of instance-based learning and is also an example of a lazy learning technique which is a procedure that waits until the query arrives to generalize beyond the training data. Although KNN classification is a classification technique that is easy to understand and implement, it performs well in many situations. Also, because of its simplicity, KNN is easy to transform for more convoluted classification problems. For instance KNN is particularly well-suited for multimodal classes as well as applications in which an object can have many class labels [12].

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Where x_i is the i^{th} feature of the instance and is the total number of features in the data set. When all the attributes are of nominal, the distance can be measured as:

$$d(x,y) = \sum_{i=1}^{n} \delta(x_i, y_i)$$

Where $\delta(x_i, y_i)$=0 if $x_i = y_i$ and $\delta(x_i, y_i)$=1, if $x_i \neq y_i$.

DARPA dataset that contains only network data is termed as KDDCup'99 dataset. It contains seven weeks of training data and two weeks of test data. KDD dataset is widely used as a benchmark dataset for offline network traffic, which helps the researchers to test and implement their algorithms.

4.2 Information Gain

Information gain is part of the decision tree algorithm, where ID3 algorithm is a distinctive decision tree algorithm. Algorithm ID3 uses information gain (or entropy) to decide which attribute belongs into a decision node. The following are the steps involved in ID3 algorithm [13]:

1. Calculate the entropy (or information gain) of every attribute in the data set.
2. Partition the set into subsets using the attribute for which entropy is minimum (or, homogeneously, information gain is maximum).
3. Make a decision tree node surrounding that attribute
4. Recourse on partitioned subsets using the remaining attributes

Traditional ID3 algorithm selected attributes and often tends to choose the attributes that get more values because the weighted sum technique makes the classification of examples set tend to the metadata group, discarding small data cluster but the attribute has more properties which is not always most favorable one. The final decision tree classification results are not certainly consistent with the actual situation according to the traditional ID3 classification because there are many types of attributes based on Entropy [13].

Suppose that a training example set is X, the purpose is to divide the training examples into n classes recorded as C=(X_1, X_2... X_n). On the postulation that the number of i^{th} training examples is | X_i | = C_i, the opportunity that an example belongs to this training examples is P (X_i).

If we choose the attribute A to analysis with a set of properties a_1, a_2, a_3,...a_i, the numeral of examples that belonged to the i^{th} category when A = a_j is C_{ij}

$$P(X_i\!:A = a_j) = \frac{C_{ij}}{|X|} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

The value of P (X_i: A=a_j) is the probability that the test attribute A belongs to the i^{th} category. Y_j is the examples set when A= a_j, then the degree of uncertainty to the decision tree classification is the entropy of the training examples set to attributes A:

$$H(Y_j = -\_P(X_i|A = a_j)log2P(X_i|A = a_j \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

We increase the user interest when calculating the taxonomic information entropy of each leaf node = a_j extended from attribute A and then strengthen the label of significant attribute, and reduce the label of non-important attribute. The formula as follows:

$$H(X|A = \_[P(A = a_j) + \_]H(X_j)(3)$$

The information presented by attributes A for classification (the information gain of attribute A) is:

$$I(X\!:A) = H(X) - H(X|A) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

By doing all the steps which was conversed a new decision tree will be created with the possibility to identify known and unknov.:n incoming packets.

4.3 Ant Colony Optimization

ACO is a class of algorithms, whose first member, called Ant System, was initially proposed by Colorni, Dorigo and Maniezzo. The main underlying idea, loosely inspired by the behavior of real ants, is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of previously obtained result. The collective behavior emerging from the interaction of the dissimilar search threads has proved effectual in solving combinatorial optimization (CO) problems. The original suggestion comes from scrutinizing the exploitation of food resources among ants, in which ant's independently limited cognitive abilities have collectively been able to find the shortest path between a food source and the nest.

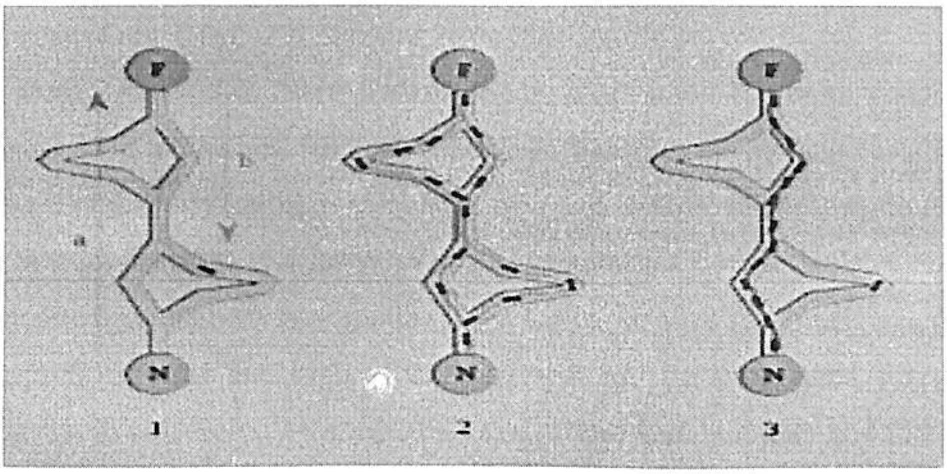


Figure 4.1 Ant follows path between nest and food

1. The first ant wanders randomly until it finds the food source (F), then it returns to the nest (N), laying a pheromone trail.
2. Other ants follow one of the paths at random, also laying pheromone trails. Since the ants on the shortest path lay pheromone trails faster, this path gets reinforced with more pheromone, making it more appealing to future ants.
3. The ants become increasingly likely to follow the shortest path since it is constantly reinforced with a larger amount of pheromones. The pheromone trails of the longer paths evaporate.

In a succession of experiments on a colony of ants with a choice among two imbalanced length paths leading to a source of food & biologists have observed that ants tended to employ the shortest route.

It has the following objectives:

- To design technical systems for optimization, and
- NOT to design an accurate model of nature.

4.4 Proposed Work

In the classification of big data domains, sometimes concealed data possibility has been occur while the classification process. Therefore generated features contain the false correlations which are not upto the mark of finding the process of intrusion detection. The weakness of extra features is that it restrains large time for the process of computing and it impacts the precisions of IDS. Here feature selection advances the more classification precision by searching for the best features, which best classifies the training data. So in the proposed system probability has been calculated of the each independently attributes, then entropy has been deliberated and lastly information gain has been calculated for every attributes disjointedly. And here they applied some logical implies that if calculated gain is very less (gain<0.16) then that type of attribute will not be contributed for the data preprocessing. So, in conclusion 14 attributes found whose gain is higher and that process is done in feature extraction and feature reduction.

4.4.1 Preprocessing and feature selection

Source_bytes	Dst_bytes	Logged_in	Root_shell	count	Dst_host_count	Dst_host	Type
200	3000	1	0	50	400	0	Normal
400	6000	1	0	50	400	0	Normal
200	6000	1	0	50	400	0	Normal
200	3000	1	0	0	400	0	Normal
200	6000	0	1	1	50	0	Normal
400	3000	0	1	1	50	0	Normal
200	3000	1	0	50	400	0	Normal
2000	0	0	0	600	400	0	Dos
2000	0	0	0	600	400	0	Dos
2000	3000	1	1	50	50	0	Dos
2000	6000	0	0	600	400	0	Dos

2000	0	0	0	600	400	0	Dos
50	0	0	0	50	50	0	Probe
50	0	0	0	50	50	0	Probe
50	3000	0	0	600	50	0	Probe
50	6000	1	1	1	400	0	Probe
50	0	0	0	50	50	0	Probe
3000	6000	0	1	1	50	0	R21
3000	6000	0	1	1	50	0	R21
3000	0	1	0	0	400	0	R21

Table 4.1: Result obtained from the solution of example

The computation process is done accordingly:

Now for example we have to find out attack type of tuple given below;

X = (Source_bytes= 200, dest_bytes= 3000, logged_in= 1, root_shell= 0, count= 50, dst_host_count= 400, dst_host_rerror_rate= 0).

P (C_i), the prior probability of each class, can be computed based on the training tuples:

P (type = normal) = 7/20 = 0.35

P (type = DOS) = 5/20 = 0.25

P (type = probe) = 5/20 = 0.25

P (type = R2L) = 3/20 = 0.15

To compute *P (X* | C_i), for *i* = 1, 2, 3, 4 we compute the following conditional probabilities:

P (source_bytes = 200 | type = normal) = 5/7 = 0.7143

P (source_bytes = 200 | type =DOS) = 0

P (source_bytes = 200 | type = probe) = 0

P (source_bytes = 200 | type = r21) = 0

P (dst_bytes = 3000 | type = normal) = 4/7 = 0.57143

P (dst_bytes = 3000 | type = DOS) = 1/5 = 0.2

P (dst_bytes = 3000 | type = probe) = 1/5 = 0.2

P (dst_bytes = 3000 | type = r21) = 0

P (logged_in = 1 | type = normal) = 5/7 = 0.7143

P (logged_in = 1 | type = DOS) = 1/5 = 0.2

P (logged_in = 1 | type = probe) = 1/5 = 0.2

P (logged_in = 1 | type = r2l) = 1/3 = 0.33

P (root_shell = 0 | type = normal) = 5/7 = 0.7143

P (root_shell = 0 | type = DOS) = 4/5 = 0.8

P (root_shell = 0 | type = probe) = 4/5 = 0.8

P (root_shell = 0 | type = r2l) = 1/3 = 0.33

P (count = 50 | type = normal) = 4/7 = 0.57143

P (count = 50 | type = DOS) = 1/5 = 0.2

P (count = 50 | type = probe) = 3/5 =0.6

P (count = 50 | type = r2l) = 0

P (dst_host_count = 400 | type = normal) = 5/7 = 0.7143

P (dst_host_count = 400 | type = DOS) = 4/5 = 0.8

P (dst_host_count = 400 | type = probe) = 1/5 = 0.2

P (dst_host_count = 400 | type = r2l) = 1/3 = 0.33

Using the above probabilities, we obtain

P(X | type= normal) = P (Source_bytes = 200 | type = normal) × P (dst_bytes = 3000 | type = normal) × P (logged_in = 1 | type = normal) × P (root_shell = 0 | type = normal) × P (count = 50 | type = normal) × P (dst_host_count = 400 | type = normal)

$$= 0.7143 \times 0.57143 \times 0.7143 \times 0.7143 \times 0.57143 \times 0.7143$$

$$= 0.085$$

To find the class, C_i, that maximizes $P(X \mid C_i)P(C_i)$, we compute

P(X | type = normal) P (type = normal) = 0.085 × 0.35 = 0.02975.

Therefore, the knnaco predict attack type = *Normal* for tuple *X*. and entropy has been calculated as follows:

Entropy H(X) $= -p_1\log_2 p_1 - p_2\log_2 p_2 - p_3\log_2 p_3 - \ldots\ldots\ldots\ldots -p_n\log_2 p_n$ ***...eq(1)***

$$= -\sum_{i=1}^{m} pi \; log2 \; pi$$

In the equation 1, the class-wise probability has been settled then entropy has been calculated of each individual attributes.

Then gain was calculated as follows:

Gain = Entropy(X) - Entropy (X|Y) ....eq (2)

So as per the above process, feature reduction has been done, where gain was higher than that attribute has been qualified for the process and less gain was reduced from dataset.

Major Steps followed as:

Step1: Initialize population

Step2: initialize ant's generation pheromone

Step3: Set the intensity of pheromone trial if related with any specified feature

Step4: Define the maximum of allowed iterations (approx)

Step5: Apply KNN as classifier for testing of all five data which is classified or misclassified.

Step6: Any ant randomly is assigned to one feature and it should visit all five features and merge the subclasses into their parent class.

Step7: Evaluation of the selected sub classes then
arrange identified sub classes according to classifier performance and distance. After that select the best optimized categories.

Step8: Check the criterion status,
If the number of iterations is more than the maximum allowed iteration,

    Terminate here,

Else

    Continue

Step9: Updating pheromone.

Finally, allow the best ant to deposit additional pheromone (classes) on parent class.

Step10: Generation of new ants //here previous ants are removed and new ants will be generated

Step11: increase counter of identified classes

Repeat from step2 to 11 until iteration is not finished

Step12: Classification: after KNN classification process the accuracy of whole process is calculated for an individual class.

Block Diagram of Proposed Approach

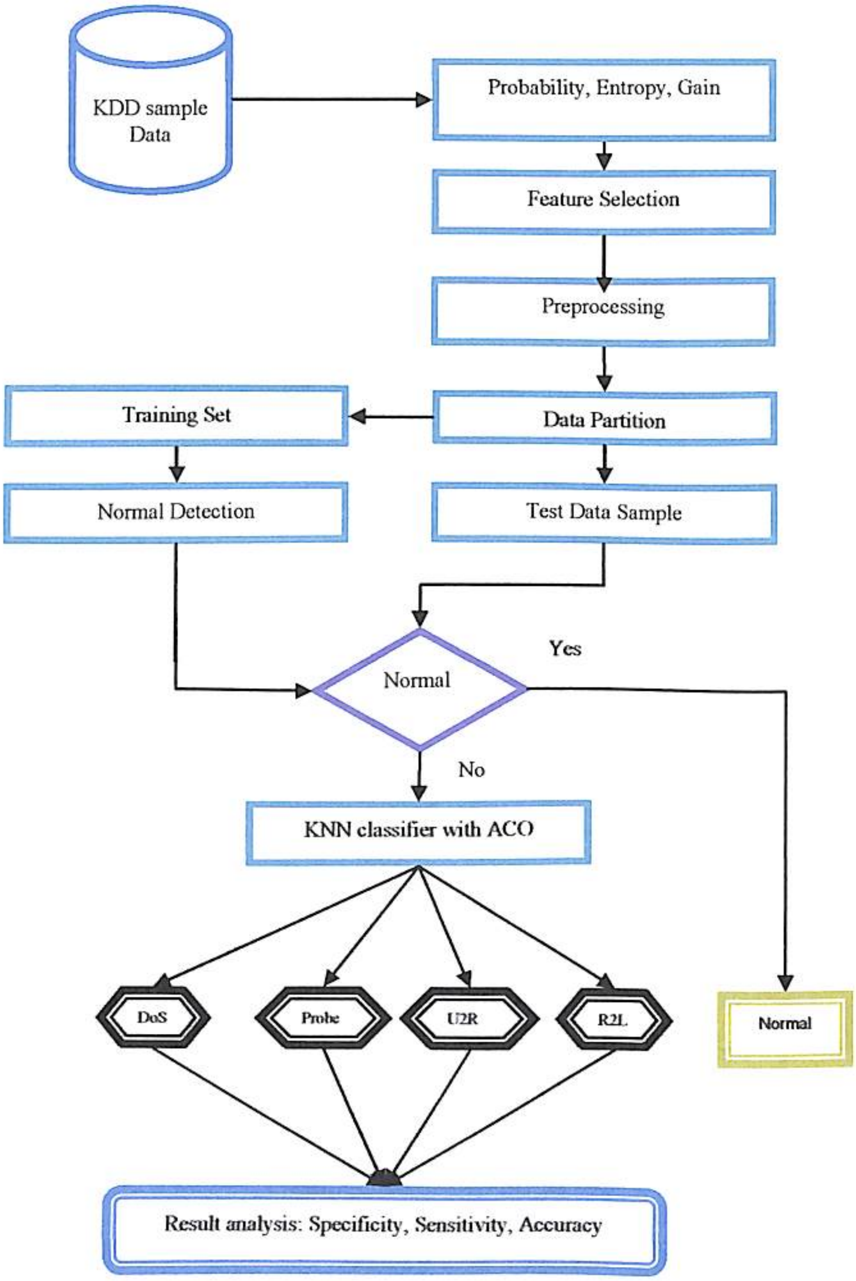


Fig. 4.2 Block diagram proposed methodology

Here block diagram shows that the working of proposed approach, where at initial state KDD99 dataset is selected for the processing, then into next stage entire dataset is logically separate for the moment due to it is containing string fields as well as numeric fields, so in the designing approach they developed separate approach for string and numeric data.

Pre-processing: It converts the data which is more reliable for unsupervised learning by removing the labels from the dataset.

Data fraction: Preprocessed data are used to partition into training & testing sets samples.

Detection of Normal: in this step normal data is separated from the training data sample, here training process is done by SVMtrain() built in function of the MATLAB

And if the normal class has been easily detected then its goes to the separately normal class otherwise if not detected then it will go to the KNNclassify() classifier and in this process each class has been accurately predicted with their own identity with the help of ant colony optimization, after successful prediction the result analysis approach follows for the detected intrusions.

CHAPTER - 5

EXPERIMENTAL RESULTS

5.1 Introduction

This chapter describes the experimental setup and result analysis of the all compared method with proposed method, also describes the dataset used and Matlab brief description.

5.2 Dataset KDD99

The KDD99cup data set used for the purpose of experimental research analysis, as they know that KDD 99 dataset has been widely used for the evaluation of signature based intrusion detection. In the novel approach they have used standard KDDCup'99 dataset, where we have chosen approx 26167 records with.50:50 testing and training ratio.

Attack types are four categories:

1. Denial of Service (DoS)
2. Remote to Local (R2l)
3. User to Root (U2R)
4. Probe

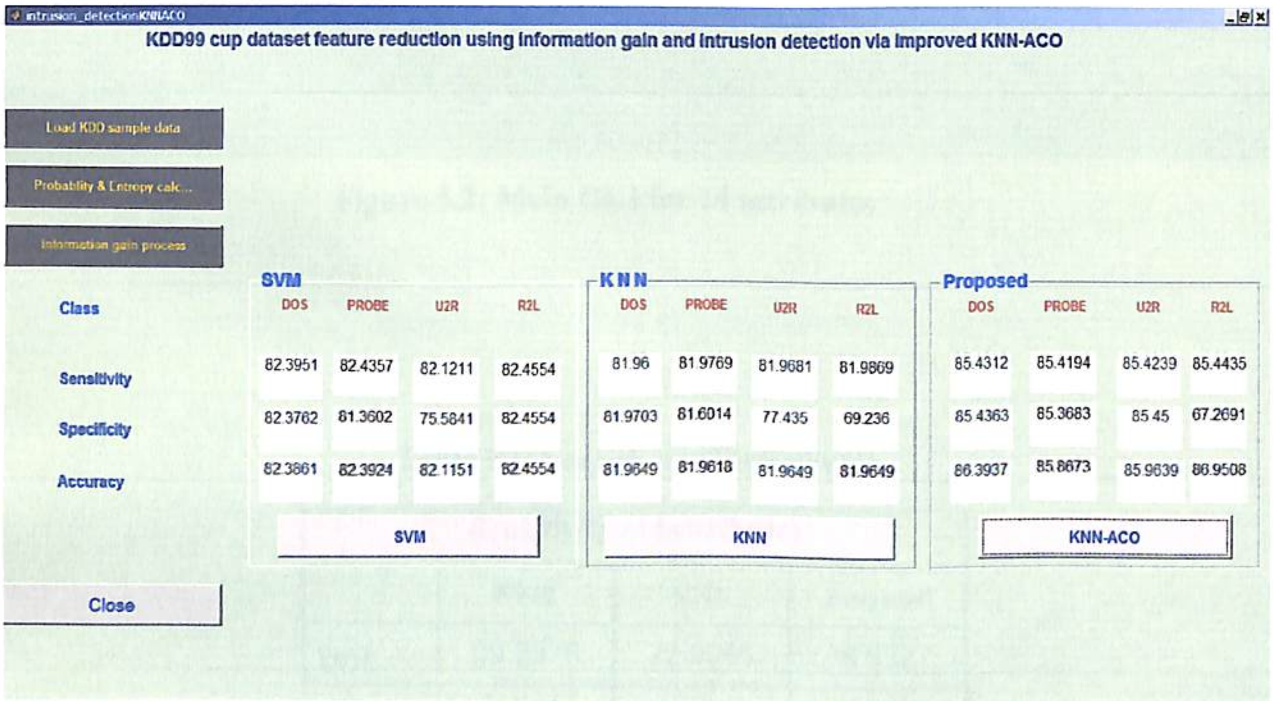
The proposed IDS has been implemented in MATLAB2012A [14] tool and the machine configuration is Intel I3 core 2.20 GHz processor, with 4GB RAM, windows 7 home basis. The proposed methodology have first used the partially ID3 algorithm for the feature reduction from the KDD, the SVM train function is use for training purpose of the trained sample, then KNN is use for the clustering and classification process for the classify or misclassify of the data, where ACO is ensemble with KNN to enhance the best classification rate and optimized the result in very efficient manner. Here classification has five classes (normal, dos, u2r, r2l, and probe). The following are the lists of features used to detect the viruses in KDD Cup 1999 dataset. The 41 features are listed in the website [36].

KNNACO classified data whic: were misclassified by alone SVM and KNN then applying KNNACO on multiple classifiers. This approach is focused on misclassified classifiers. Where ACO is putted extra efforts to optimize best classified of the category until they are not accurately classified.

Then method has been tested on full (41 attributes) dataset as well as in reduced dataset (14 attributes), and used measurement parameters are:

Sensitivity, specificity and accuracy, and method is compared with SVM, KNNACO and found that proposed method produced most accurate result into maximum cases.

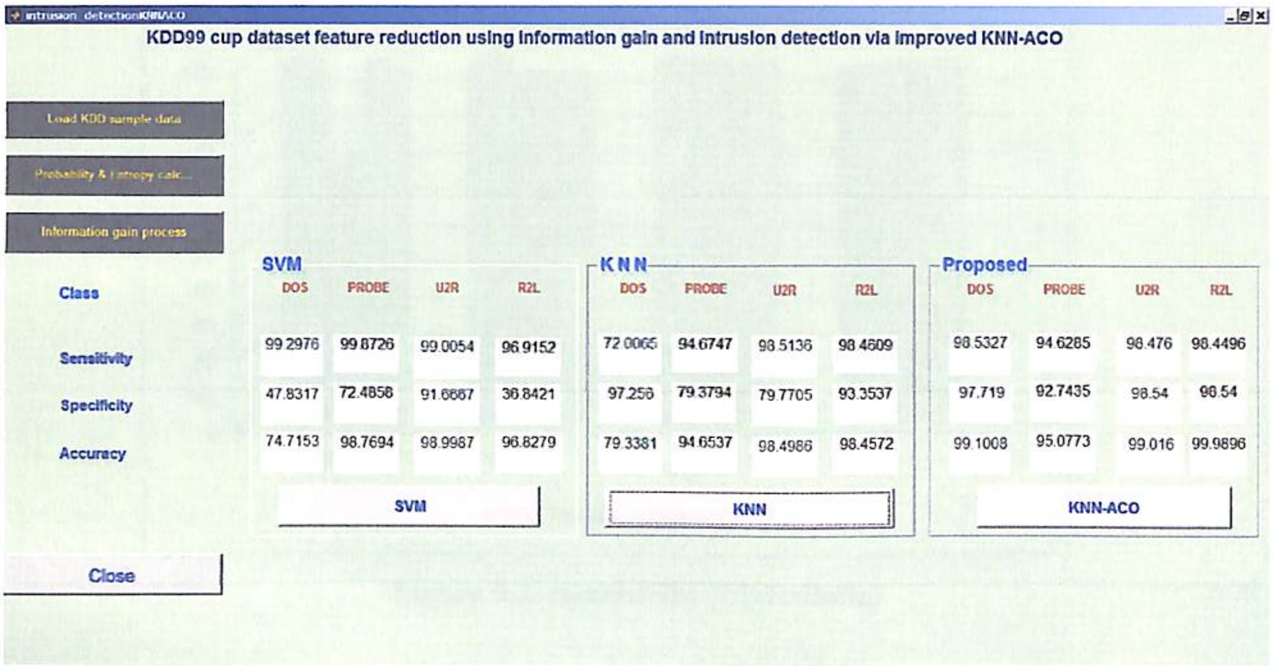
Here figure 5.1 & figure 5.2 shows that the main GUI environment of the implemented all methods along with proposed approach having 41 all dataset and 14 reduced dataset, here we clearly observe that the proposed approach yield more accurate output compared to the other previous developed methods.


intrusion_detectionKNNACO
KDD99 cup dataset feature reduction using information gain and intrusion detection via improved KNN-ACO
Load KDD sample data
Probablity & Entropy calc...
Information gain process
SVM
KNN
KNN-ACO
Close

	SVM				K N N				Proposed			
Class	DOS	PROBE	U2R	R2L	DOS	PROBE	U2R	R2L	DOS	PROBE	U2R	R2L
Sensitivity	82.3951	82.4357	82.1211	82.4554	81.96	81.9769	81.9681	81.9869	85.4312	85.4194	85.4239	85.4435
Specificity	82.3762	81.3602	75.5841	82.4554	81.9703	81.6014	77.435	69.236	85.4363	85.3683	85.45	67.2691
Accuracy	82.3861	82.3924	82.1151	82.4554	81.9649	81.9618	81.9649	81.9649	86.3937	85.8673	85.9639	86.9508

Figure 5.1: Main GUI and result for 41 attributes

Table 5.1, table 5.2, table 5.3 illustrated that the sensitivity, specificity and accuracy comparison table of SVM, KNN and proposed ID3, KNNACO approaches for reduced 18 attributes, we have also examine the same scenario for the 41 full attributes, and there we found that the all methods gave the less accuracy level and taking much time as compare to reduced attribute.



KDD99 cup dataset feature reduction using information gain and intrusion detection via improved KNN-ACO

	SVM				K N N				Proposed			
Class	DOS	PROBE	U2R	R2L	DOS	PROBE	U2R	R2L	DOS	PROBE	U2R	R2L
Sensitivity	99.2976	99.8726	99.0054	96.9152	72.0065	94.6747	98.5136	98.4609	98.5327	94.6285	98.476	98.4496
Specificity	47.8317	72.4858	91.6667	36.8421	97.256	79.3794	79.7705	93.3537	97.719	92.7435	98.54	98.54
Accuracy	74.7153	98.7694	98.9987	96.8279	79.3381	94.6537	98.4986	98.4572	99.1008	95.0773	99.016	99.9896

Figure 5.2: Main GUI for 14 attributes

Table 5.1: Sensitivity (14attribute)

Sensitivity (14attribute)			
	SVM	KNN	Proposed
DOS	99.2976	72.0065	98.5327
PROB	99.8726	94.6747	94.6285
U2R	99.0054	98.5136	98.476
R2L	96.9152	98.4609	98.4496

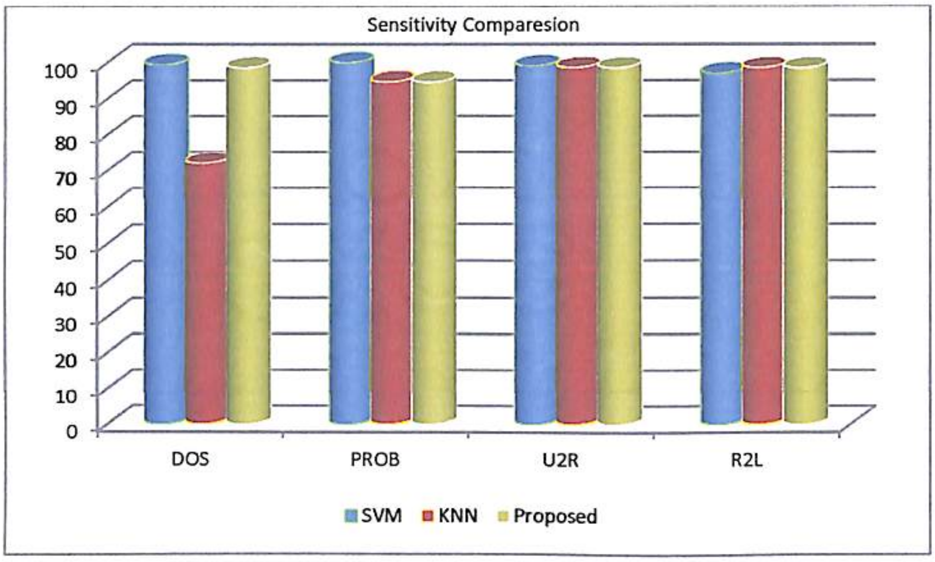


Figure 5.3: Sensitivity (14attribute)

Table 5.2: Specificity (14attribute)

Specificity(14attribute)			
	SVM	KNN	Proposed
DOS	47.8317	97.256	97.719
PROB	72.4858	79.3794	92.7535
U2R	91.6667	79.7705	98.54
R2L	36.8421	93.3537	98.54

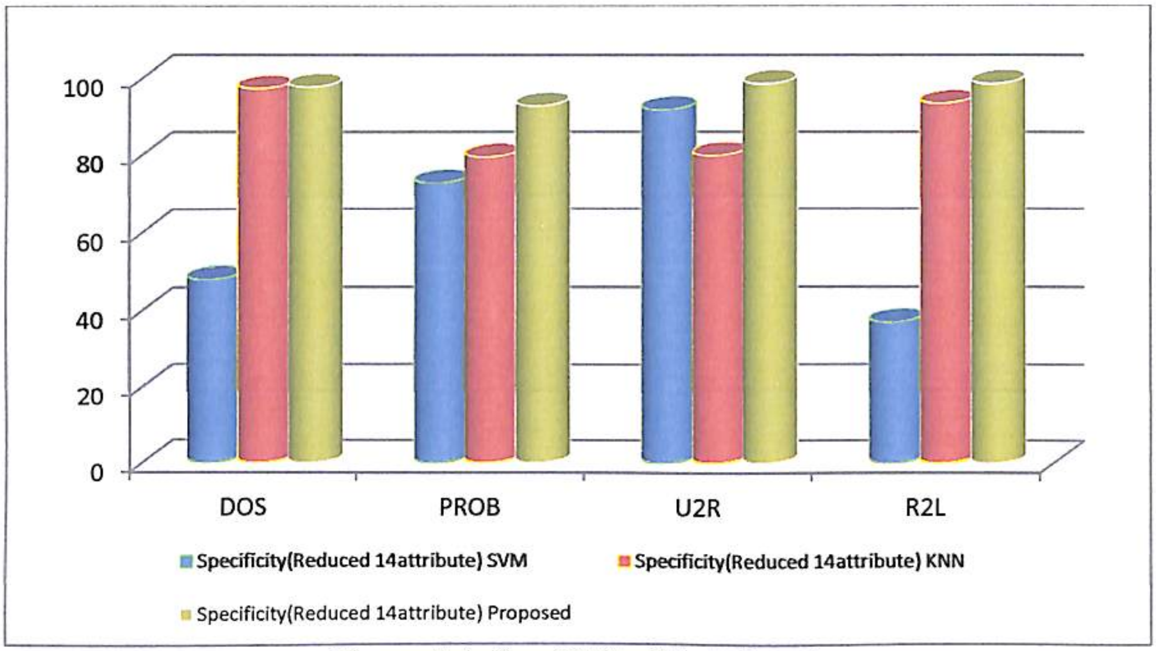


Figure 5.4: Specificity (14attribute)

Table 5.3: Accuracy (14attributes)

Accuracy(14attribute)			
Attacks	SVM	KNN	Proposed
DOS	74.7153	79.3381	99.1008
PROB	98.7694	94.6537	95.0773
U2R	98.9987	98.4986	99.016
R2L	96.8279	98.4572	99.9896

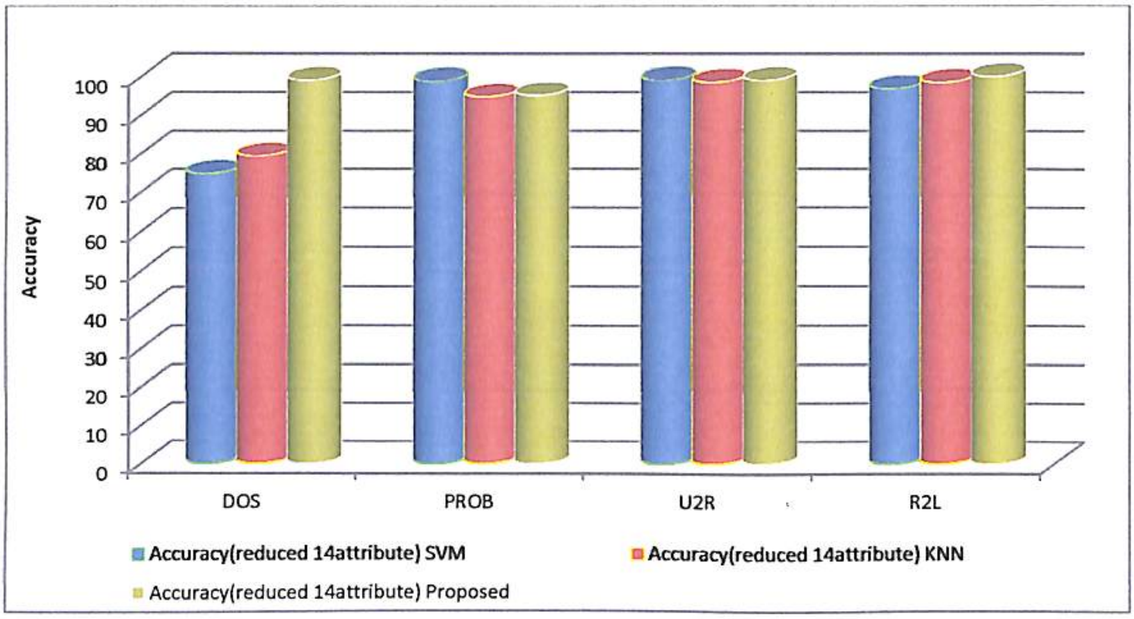


Figure 5.5: Accuracy (14attribute)

Here table 5.4 , table 5.5 and table 5.6 shows the sensitivity, specificity and accuracy obtained after simulating of the 41 attributes of KDD dataset and figure 5.6, figure 5.7 and 5.8 illustrate the result through the graph.

Table 5.4: Sensitivity (41attributes)

Sensitivity(41attribute)			
Method/ Attacks	SVM	KNN	Proposed
DOS	82.3951	81.96	85.4312
PROB	82.4357	81.9769	85.4194
U2R	82.1211	81.9681	85.4239
R2L	82.4554	81.9869	85.4435

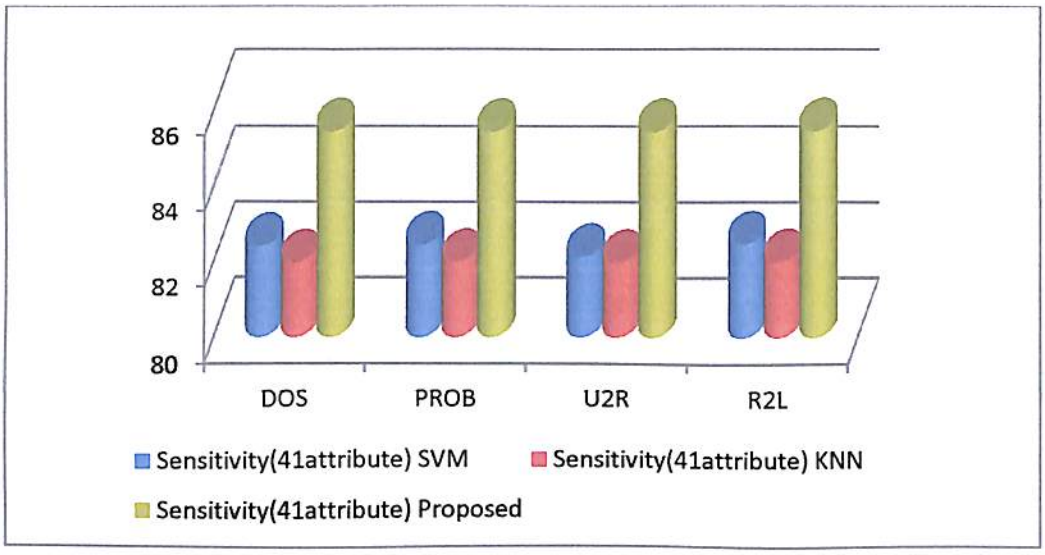


Figure 5.6: Sensitivity (41attribute)

Table 5.5: Specificity (41attributes)

Specificity(41attribute)			
Method/ Attacks	SVM	KNN	Proposed
DOS	82.3762	81.9703	85.4363
PROB	81.3602	81.6014	85.3683
U2R	75.5841	77.435	85.45
R2L	82.4554	69.236	67.2691

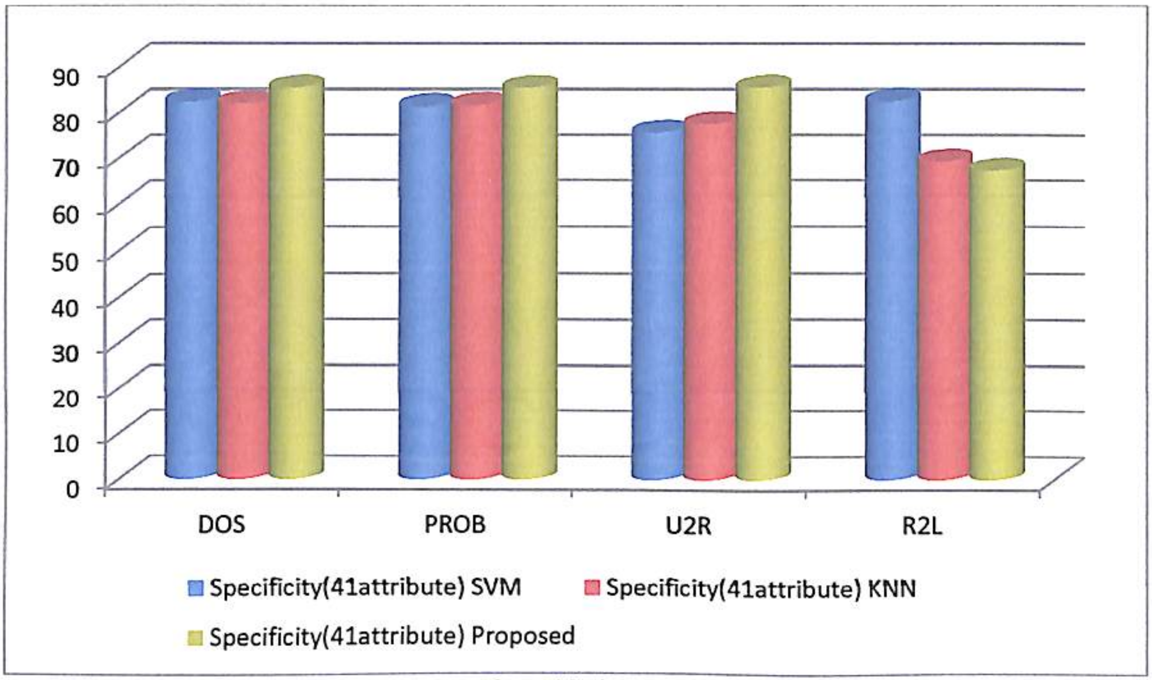


Figure 5.7: Specificity (41attribute)

Table 5.6: Accuracy (41attributes)

Accuracy(41attribute)			
Method/ Attacks	SVM	KNN	Proposed
DOS	82.3861	81.9649	86.3937
PROB	82.3924	81.9618	85.8673
U2R	82.1151	81.9649	85.9639
R2L	82.4554	81.9649	86.9508

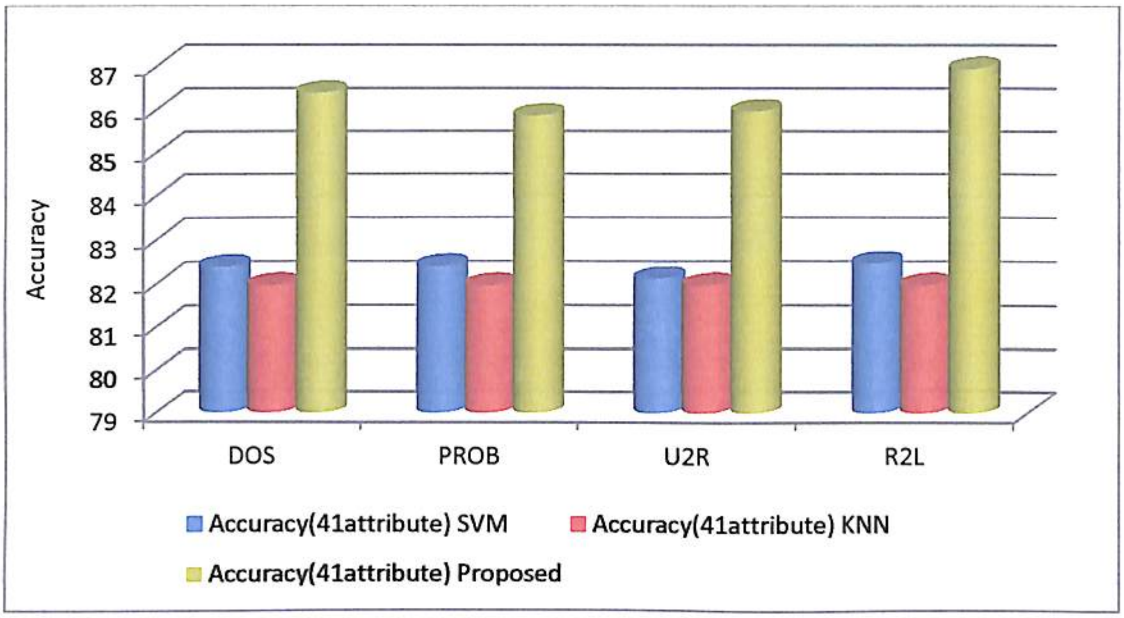


Figure 5.8: Accuracy (41attribute)

Chapter - 6
CONCLUSION

6.1 Conclusion

To develop the system for exposure and detection of severe types of intrusion this may corrupt or destroy the resources used for the access. Several author have been work in the field of intrusion detection and develop the system which can reduce the true and false alarm rate but in this dissertation we develop a novel method by applying multiple classification and feature reduction techniques. In this we use ID3, SVM and KNNACO approach for detection of suspicious activities in KDD99 and tested on standard dataset KDD99 cup. For investigating it we have apply partially decision tree for feature selection then experimental process is done by using SVM and compared with KNN based ant colony optimizer for the detection of intrusions. The analysis of the methodology is done in well known simulator MATLAB2012a using the performance metrics sensitivity, specificity and accuracy in which our method results outperform than the existing methods. The accuracy level of the proposed work is approx 99% (on selected data's) which is more than the existing method.

6.2 Future possible improvements

In future work, other than SVM or KNN classifiers we can develop a model using ensemble multiple classifier which can better expose the intrusion and greatly enhance the performance of the intrusion system.

References

[1] Wu Shelly Xiaonan and Banzhaf Wolfgang "The use of computational intelligence in intrusion detection systems: A review," ELSEVIER, 2010.

[2] Thottan Marina and Ji Chuanyi "Anomaly Detection in IP Networks," *IEEE Transaction on Signal Processing, Vol. 51, NO. 8, Aug 2003.*

[3] Peddabachigari Sandhya, Abraham Ajith and Thomas Johnson "Intrusion Detection Systems Using Decision Trees and Support Vector Machines," Department of Computer Science, Oklaho State University, USA.

[4] Yuan Jingbo and Li Haixiao "Intrusion detection Model based on Improved support Vector Machine," *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium_IEEE, pp. 465-469, 2010.*

[5] Koshal Jashan and Bag Monark "Cascading of C4.5 Decision Tree and Support Vector Machine for Rule Based Intrusion Detec.ion System," *I. J. Computer Network and Information Security, Vol. 8,* IEEE pp. 8-20, 2012.

[6] Ektefa Mohammadreza "Intrusion Detection Using Data Mining Techniques" *Information Retrieval & Knowledge Management, (CAMP), 2010 International Conference* IEEE, pp. 200-203, 2010.

[7] Amudha P and Rauf H Abdul "Performance Analysis of Data Mining Approaches in Intrusion Detection," *Process Automation, Control and Computing (PACC), 2011 International Conference,* IEEE, 2011.

[8] Jemili Farah "A Framework for an Adaptive Intrusion Detection System using Bayesian Network," *Intelligence and Security Informatics, 2007* IEEE, pp. 66-70, 2007.

[9] Aneetha A.S. and Bose S. "The Combined approach for Anomaly Detection using Neural Networks and Clustering Techniques,"*Computer Science & Engineering International Journal CSEIJ*, Vol.2, No.4, pp. 37-46, 2012.

[10] Kayacık H. Gunes "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets".

[11] Abraham Ajith "Modeling Intrusion Detection System using Hybrid Intelligent Systems," *Journal of Computer and Network Applications, Elsevier*, pp. 114-132.

[12] Swathi K., Lakshmi D. Sree, "Network Intrusion Detection Using Fast k-Nearest Neighbor Classifier", *UGC Sponsored National Seminar On "Cyber Security With Special*

Focus On Cyber Crimes & Cyber Laws *(NSCS- 2014)* November 2014-IJDCST Special Issue, Paper –ID: IJDCST-16 ISSN-2320-7884 (Online).

[13] Jaiganesh V., Rutravigneshwaran P. and Sumathi P. "An Efficient Algorithm for Network Intrusion Detection System", *International Journal of Computer Applications (0975 – 8887)* Volume-90, No. 12, March 2014.

[14] http://homes.ieu.edu.tr/hozcan/EEE281/Intro_R2012b.pdf

[15] Hu Weiming "AdaBoost-Based Algorithm for Network Intrusion Detection", *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* IEEE, Vol. 38, pp. 577-583, 2008

[16] Eid Heba F., Darwish Ashraf, Hassanien Aboul Ella and Abraham Ajith "Principle Components Analysis and Support Vector Machine based Intrusion Detection System", in proceeding of IEEE.

[17] Revathi S. and Malathi A. "Network Intrusion Detection Using Hybrid Simplified Swarm Optimization and Random Forest Algorithm on NSL-KDD Dataset", *International Journal Of Engineering And Computer Science ISSN: 2319-7242* Volume 3 Issue 2 February-2014 Page No. 3873-3876.

[18] Parsazad Shafigh, Saboori Ehsan and Allahyar Amin "Fast Feature Reduction in Intrusion Detection Datasets", MIPRO 2012, May 21-25, 2012, Opatija, Croatia.

[19] Rajeswari L. Prema, Kannan A. "An Intrusion Detection System Based on Multiple Level Hybrid Classifier using Enhanced C4.5" *IEEE-International Conference on Signal processing, Communications and Networking Madras Institute of Technology, Anna University Chennai India*, Jan 4-6, 2008. PP 75-79.

[20] Breiman, "Bagging predictors, Machine Learning" *Springer August 1996, Volume 24, Issue nno.2, pp 123-140pp. 123-140, 1996.*

[21] Freund Y., Schapire and R.E., "A Decision-Theoritic Generalization of on-line Learning and an Application to Boosting," *Journal of Computer and System Sciences, Volume 55, Issue1August 1995, pp.119-139.*

[22] Zhou and H. Z. "Ensemble Learning "Encyclopedia of Biometrics" *Springer, Vol. 1, PP. 270-273, Berlin, ISBN: 978-0-387-73002-8, 2009.*

[23] Xiang C., Yong P.C. and Meng L.S. "Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. in Pattern Recognition Letters" *ACM transaction on Journal on pattern Recognition, Vol. 29, Issue 7,May,2008.*

[24] Giacinto G. and Roli F. "Fusion of multiple classifiers for intrusion detection in computer network," *Elsevier on Pattern Recognition Letters,Volume 24,August 2003 pp. 1795-1803.*

[25] Zainal A., Maarof M. A., Shamsuddin S.M. and Abraham A. "Ensemble of one-class classifiers for network intrusion detection system," *In Information Assurance and Security, 2008.*

[26] Perdisci R., Ariu D., Fogla P., Giacinto G. and Lee W. "McPAD: A multiple classifier system for accurate payload based anomaly detection" *In Computer Networks, Vol. 53, No. 6, PP. 864-881, 2009.*

[27] Giacinto G., Perdisci R., Rio M.D. and Roli F. "Intrusion detection in computer networks by a modular ensemble of one class classifiers", *In Information Fusion, 9, 69-82, 2008.*

[28] Dongre S S. and Wankhade K. "Intrusion Detection System Using New Ensemble Boosting Approach", *International Journal of Modeling and Optimization, Vol. 2, No. 4, Aug 2012.*

[29] Kruegel Christopher, Valeur Fredrik, and Vigna Giovanni "Intrusion Detection and Correlation: challenges and solutions", *Springer- 2005.*

[30] Das Asim and Sathya S. Siva "Association Rule Mining for KDD Intrusion Detection Data Set", *International Journal of Computer Science and Informatics ISSN (PRINT): 2231 – 5292, Volume-2, Issue-3, 2012.*

[31] Han LI "Research of K-MEANS Algorithm based on Information Entropy in Anomaly Detection", *2012 Fourth International Conference on Multimedia Information Networking and Security.*

[32] kailashiya Devendra and Jain R.C. "Improve Intrusion Detection Using Decision Tree with Sampling" *International Journal of Computer Technology & Applications, Vol. 3 (3), 1209-1216, ISSN:2229-6093.*

[33] Li Yang and Guo Li "An active learning based TCM-KNN algorithm for supervised network intrusion detection", *Computers & Security 2 6 (2007), PP- 459 – 467 in proceeding of Elsevier.*

[34] Crosbie Mark and Spafford Gene "Applying genetic programming to intrusion detection", *In Working Notes for the AAAI Symposium on Genetic Programming, pages 1–8. MIT, Cambridge, MA, USA: AAAI, 1995.*

[35] Farid Dewan Md, Rahman Mohammad Zahidur, and Rahman Chowdhury Mofizur "Adaptive intrusion detection based on boosting and naıve bayesian classifier", *International Journal of Computer Applications, 24(3):12–19, 2011.*

[36] KDD Cup 1999 Intrusion Detection Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html,2010.