Model Based Intrusion Detection System

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Computer Science and Engineering

Submitted By
Jyotsna Goyal
(801632018)

Under the supervision of
Dr. Ajay Kumar
Associate Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA 147004, PUNJAB, INDIA

June 2018

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **Model Based Intrusion Detection System** , in partial fulfillment of the requirements for the award of degree of **Master of Engineering** submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Ajay Kumar** and refers other researchers work which are duly listed in the reference section. The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature: Jyotsna Goyal

(Jyotsna Goyal)

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

Signature: A Kumar

Dr. Ajay Kumar

Associate Professor

Acknowledgement

I would like to use this opportunity to express my gratitude towards those who provided me with the possibility to complete this research, without whose kind support and help it would have not been possible. A special thanks to my supervisor **Dr. Ajay Kumar** for providing necessary information regarding the project and also for his help and support in completing the project. Without his unfailing support and belief in me, this thesis would have not been possible.

Furthermore, I would also like to appreciate the crucial role of the staff of Thapar University and the organisation Oracle India Pvt. Ltd. where I worked as an Intern alongside, who gave me the permission to use all required equipment and the necessary material to complete the research on my research topic **Model Based Intrusion Detection System**. I would also like to express my gratitude to our Head of Department **Dr. Maninder Singh** and Dean of Academic Affairs **Dr. SS Bhatia** for their constant motivation and encouragement.

Finally, I would like to express my sincere and deep gratitude to my family members for their love, encouragement, care and support.

Submitted By

Jyotsna Goyal

Jyotsna Goyal

Abstract

The technological advances have to lead to a more digitized world where data is handled through machines rather than paper. Every day a huge amount of data and information is generated and this needs to be stored for further references and analysis. With this growth in production and storage of information, the issue of security vulnerabilities also rise. The attack on this critical information and data with the intention of misusing it is called intrusion. These intrusions pose a great threat to the data stored, like tampering with the stored information or loss of information which makes the database and the repositories insecure. Therefore, detection of these activities is the need of the hour as it is very important to secure the data especially the user data from any unwanted criminal activity, as misuse of data can lead to serious issues and breaches in the system. The detection of these unwanted activities is called intrusion detection.

An intrusion detection model is built using the data mining techniques and the intrusion detection dataset. The NSL-KDD dataset is used for detection which is an intrusion detection database. The dataset is divided into two parts, the training set and testing set. The training set is at the time of model creation and testing set is used to test the model. Various classification and clustering techniques are used. Clustering techniques like K-means clustering and classification techniques like C4.5, naive Bayes, random forest, Ripper K-nearest neighbours are used for building the model. Further two types of model are built which are classification models and hybrid models. The classification model is built using a classification algorithm and the hybrid model is built by using both classification and clustering algorithms.

The model detects three types of intrusions which are misuse-based, anomaly-based and hybrid intrusions. Misuse based intrusions are those which the system had already encountered and so are already present in the database. For these attacks models generally, give high true positive rates. Anomaly-based attacks are new attacks or unknown attacks which the system has not seen earlier and so are not present in the database and therefore difficult to detect. The third one is the hybrid attacks which can lead to both types of attacks in the system.

A comparison is done between the prediction results of the models built using the above techniques. Ripper algorithm gave the highest accuracy and a good true positive rate for the classification algorithm. C4.5 tree algorithm with K-means gave the best accuracy with a good true positive rate among hybrid models. The Results show that hybrid models which are used to detect both types of attacks outperform other models and classification models as well.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

With the advances in technology, the data prevention and storage techniques have also emerged widely. From physical paper records to digital records and database repositories, the data storage practices have changed. But this digitization of records have also posed threats to data and made systems vulnerable to attacks and have lead to serious security breaches. Computer viruses and unwanted attacks have always been a threat for important confidential data. The threats include serious crimes like hacking and forgery, theft of confidential data by criminals, vulnerabilities in the system. The main purpose of these hackers or attackers is to steal important information, to make money related transactions or frauds, to change or destroy such important information.

As the number of users and technological advances is increasing so is the data and so does the chances of attacks and the threats posed to the system. The customer and users data is very confidential and the lost of its loss or access from the illegitimate user is very costly and can not be tolerated. The user data contains user personal information and account information which is very much vulnerable, leaking of this information can lead to loss of customer assets. The attacks and threats on the confidential data will go on if not monitored and so need to be slowed down. Computer security is much required for securing the systems from these dangerous attacks. These threats have made the researchers move towards intrusion detection and prevention as it has become an important issue.

Intrusion detection is a security mechanism which detects malicious activities occurring at vulnerable places. The detection of these activities allows timely action towards an issue, for example, stop an ongoing attack. Every day millions and tons of information are sent and received by the connected nodes, which could be accessed by an illegitimate user and so must be protected from unwanted attacks. The intrusion detection systems(IDS) are the systems which can definitely help in the detection of these threats. Intrusion detection includes collecting information which could be conferred as an intrusion by monitoring the system and saving logs in the system and then analyzing the output. The data is then characterized as either intrusion or normal behaviour.

1.1 Types of Intrusions

There are three main types of attacks which need to be detected and can be external attacks (attacks which happen due to access from an element outside the organization)

and internal attacks (attacks from one among the users of the organization). These attacks are further divided as misuse-based, anomaly-based and hybrid attacks. The Figure 1.1 shows the different types of attacks.

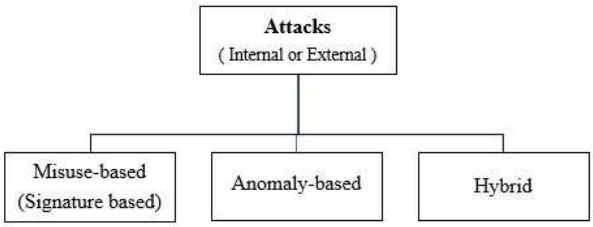

Figure 1.1: Types of Attacks

Each attacks have there further detection techniques and are explained ahead.

1.1.1 Misuse Based Attacks

Misuse based or signature-based intrusions are those which the system had already encountered that is they are known attacks. These are already present in the database. For these attacks models generally, give high true positive rates. Misuse detection technique is used to detect these type of attacks. Misuse detection is a technique in which known attacks (attacks already recorded in the database) can be detected as they are already present in the database. This technique works only for those attacks present in the database so are unable to detect any new attacks i.e. zero-day attacks. Although, they do not generate high false alarm rates yet they require the database to be updated frequently along with the revision of all of the rules and signatures.

1.1.2 Anomaly Based Attacks

Anomaly-based attacks are new attacks or unknown attacks which the system has not seen earlier and so are not present in the database. These are therefore difficult to detect. The anomaly-based detection technique is used to detect these type of attacks. Anomaly-based techniques build a model using the normal behaviour and detect attacks by identifying the deviation from the modelled normal behaviour. This approach performs better as it can detect any new malicious attack which is not already present in the database. Moreover, the data generated by detecting new attacks can be used as information for misuse based attacks and can be stored in the database to define signatures for misuse-based techniques. But anomaly based techniques result in high false positive rates(FPR) because unknown but normal system behaviours may be categorized as attacks.

1.1.3 Hybrid Attacks

The third one is the hybrid attacks which can lead to both types of attacks in the system. These attacks, therefore, need the hybrid type of techniques. Hybrid techniques combine misuse and anomaly detection methods as anomaly detection technique can work as a learning element for misuse detection as the recorded behaviours can be used to enhance the signatures so can use advantages of both the systems. Hybrid methods decrease the rate of false positives and detect unknown attacks better.

1.2 Key Research Area

Intrusion detection models are mainly built using data mining and machine learning methods. Data mining is extracting useful data and patterns from the previously unknown data by applying some specific algorithms. Machine learning, on the other hand, is making the model to learn and then lets to predict the new domains based on known characteristics of the data on which it is built. There is generally an overlap between the two terms, in the broader sense data mining is extracting useful information from unlabeled and unprocessed data and machine learning is using the extracted information to build a learned model for making new predictions. Data mining is divided into two categories which are as follows.

1.2.1 Supervised learning

Supervised learning is a class of data mining technique which has origins from machine learning. These methods are mainly used where the dataset has a predefined class on which the data is to classified and the model is constructed using this pre-classified dataset. The dataset has some input variable or vector of features and an output variable generally known as the class variable on which the dataset is to be classified. The class attribute divides each data point into different sets or classes and new data points are classifieds into different sets to which they should belong on the basis of some similarity measure.

1.2.2 Unsupervised learning

Unsupervised learning is methods which are applied to the datasets which are not pre-classified, so the classification is done on the basis of clustering. Clustering is distributing the data among different clusters where each cluster has similar types of data points. The clusters differ from each other depending on some parameters. The data points are classified into different clusters on the basis of similarity between the data points and the mean points for each cluster. The new data points are then classified into different

clusters to which they most appropriately belong. Clustering techniques are used for datasets which do not have a class attribute.

Sometimes both supervised and unsupervised learning need to be combined for detecting both known and unknown attacks which are achieved by combing classification and clustering known as classification via clustering. The research mainly focuses on building a model using the data mining techniques so that known and unknown intrusions can be detected and the system can then be used securely. For detecting unknown intrusions clustering methods are used and for detecting known intrusions classification methods are used. These methods are combined to create a single model called hybrid models so that the system can be detected over both types of intrusions. Models are created using different techniques and a comparison is done using the machine learning evaluation parameters to find the best models among them. The models which gave better true positive rate, lower false positive rate and thus better accuracy are termed the best models of all. Models are also evaluated on other parameters which are explained in further sections. The hybrid models perform better than classification models as they detect unknown attacks also along with known attacks, classification models can only detect known attacks and they ignore unknown instances making the system more vulnerable to intrusions. The objective is to select those models which reduce false alarm rate so that the resources are not wasted in calling those activities attacks which were otherwise normal.

Chapter 2
LiteratureReview

This chapter explains the work is done and the ongoing work in the field of intrusion detection. The whole work in the field of intrusion detection is not explained here as it is not possible but some of the remarkable works in this field are described.

2.1 Research Advances

The intrusion detection is said to be started in the early 1980's by Anderson's work[1] or Denning work[2]. But the work came in force in the late 1990's with the bombarding of a huge number of research papers and relevant research. A huge time started to be spent in this field for detection.

The first model of IDS was developed in 1984 which monitored the user activities and stored them in a database for further analysis[3]. After this the work in this field came across many hurdles and at a point in time was also discontinued as at that time there were not much intrusion activities. The attacks were very less when compared with data storage systems. The cost was also much higher from ignoring these intrusions than to install intrusion detection systems. This also made a decrease in the research in this area. But the need for intrusion detection has been increasing with time which has diverted the attention of researchers to this area.

MIT Lincolon Laboratories created a dataset DARPA using the US AIR BASE STATION data. This data contained the air traffic over different frequencies and the intrusions of an outsider over these frequencies to decode the messages of the US Navy. The change in frequencies or any unknown suspicious frequencies were taken as a sign of intrusion. They created a 9-week dataset using TCP dump data which was a simulation of the US Navy air traffic. The IDS created was tested using the two weeks traffic which not included the normal US navy traffic but the traffic created by automatic machine as well. The traffic fed also contained human created traffic as it was not feasible to write so many large scripts for automatic generation[4, 5]. They used Receiver Operator Metric(ROC) for the system evaluation, ROC is a good evaluation metric. Their results show that there were positive false rates while prediction. These results surely advised not to use their research criteria as a while but their work gave much guidance in the field of intrusion detection.

Canadian Institute of Security (CIS) have created IDS2012 and IDS2017 dataset which contain all the possible cyber attacks and are working on cybersecurity from these

attacks. They created a complete topology of networks which included routers, switches and other network devices for simulating the real world systems[6]. The dataset is an addition to the earlier dataset including the 1998 dataset. The dataset contains the most common and up to date attacks. The system also triggers the firewall penetrating attacks. The work is still going on and aims at tackling more difficult and new attacks with efficiency.

2.2 Distributed IDS

These type of systems deploy the intrusion detection system over the distributed network. The distributed system is a network in which the computers from different networks are connected with each other and they communicate with each other by passing messages. But this heterogeneous network is much vulnerable to attacks because these systems need to monitor a large number of systems at a time. This makes an attacker more liable for intrusion into the system and access to more information. These systems also make it difficult to manage the resources by a single intrusion detector as it also needs to monitor which resource is given to which computer at a particular interval. The continuous monitoring and switching of the analyzing systems make them a commodity of higher risk. The data collected is also huge and redundant in these types of networks so IDS with redundancy removal capacity need to be deployed.

For these type of intrusion detection system, the data is collected from different computers distributed over the network. This collected data is then reduced and stored in a centralized database. The reduction is done using LAN Monitors and individual hosts which remove the redundancy in data[7]. The system where all the gathering and analysis of data is done is called the expert system. They have used C2 or higher rated computers which means that computers connected will make uniform audit logs. Uniform here means that while connected to different types of topologies and working on the different operating system will have C2 rated audit logs. The expert system uses rule-based learners for prediction purposes. A NID (network ID)which is a unique ID is generated every time a user login the system. This new generated ID and the user ID of the user are matched at regular and constant intervals. If at any point in time there is any discrepancy then it is detected as an intrusion.

EMERALD, is said to be the most advanced distributed Intrusion detection system and is developed by [8] Sri International Laboratories. This id uses both signature-based expert system as mathematical constructs for detecting attacks. This DIDS has a well-formed architecture with adept application programming interface[8] lest the architecture and hardware constructs are left undefined. The system works by creating a subscription ticket at each node login and maintains and monitors it throughout.

IFIP international surveyed CARDS, a distributed intrusion detection system[9]

which helped in tackling with the problem of attacks occurring due to no coordination in the network. These systems also tackled the issues regarding the selection of a portion of the data to be analyzed and then how to relate this information collected from different sources. They used signature-based techniques for detection of the data centralized at the central server and then the detection is decentralized.

2.3 Host Based IDS

Host-based intrusion detection system is capable of detecting intrusion in a system as well as over the network in a way similar to the network intrusion detection systems. The intrusions are detected by monitoring of files, Log files, analyzing connections and intrusion detection at the kernel level. In file monitoring, the new files are compared with the already saved files. The files are compared on the basis of size, editing etc. Analysis can also be done on the basis of recorded log files, the network packets activities, type of connection used also forms a base for monitoring. In kernel-based, the kernel is liable for detection and prevention of unwanted activities.

Don Bosco University created a HIDS which can detect misuse as well as anomaly based intrusions[10]. They have used decision tree algorithm and Support vector machine algorithm for implementation of the intrusion detection system. Ying yan yang proposed a model which monitors a model using the log files and then applied BP neural networks which are good at detecting anomaly-based detection. Their system detects both types of intrusions.

Ng et al. [11] presents a tool which is used for detecting both anomaly detection and a signature-based detection using a log file to detect patterns that may be considered an unauthorized activity. If the pattern is considered as an attack it is stored in the database for future use. The tools keep on learning and get more powerful, the new attacks are detected and classified using clustering which includes grouping similar activities together based on popular trends. The concept of reoccurring matches with the DoS attack which enables detection of possible password guessing.

Zhu et al. [12] intrusions are detected by first recording host logs and then intrusions are detected using methods like ARIMA time series, Apriori association algorithm. Then, intrusion detection methods which are misuse-based and anomaly-based methods are integrated and applied to the host system and through this, the system security is enhanced. The system is unable or has some issues for detecting unknown or zero-day attacks but the detecting of data from logs is done with accuracy. Their methods also coped with the rapid increase in the size of log files thus improving the system stability.

2.4 Network Based IDS

Network intrusion detectors detect the intrusions over the network mainly over the internet. These systems control the unwanted access of the network packets sent and also control the tempering of the crucial information sent over the net. They need to control attacks like denial of service where the attacker bombards the internet making the resource inaccessible by the user. They also need to detect and control man in the middle attacks.

Taiwan university proposed a network intrusion detection model WRS[13] which uses active network technology with programming ability. This system can stop attacks at the time of arrival only by detecting and preventing the intrusions at the time of occurrence so that the damage to the data can be reduced. They have used Support vector technology as an additional functionality for enhanced results.

Sultana et al. [14] contributed to the detection of network intrusion using Average one-dependence estimators (AODE) which according to the research is an improvement over the nave Bayes algorithm. This model first creates some estimators and then averages the results produced. This algorithm helps in detecting different types of attacks and gave efficient results. Zhao et al. [15] presented work which analyzes the audit data by extracting the properties deeply and then these intrusion characteristics of the network are analyzed and then combined intrusion detection techniques along with human intervention to establish a rule base using the C.45 algorithm. This paper used optimal pattern matching algorithm so that the detection rate can be improved.

2.5 Automata Based IDS

In automata based Intrusion detection system the method of parsing is used. The pattern matching methods are used as they can overcome the problems faced by Definite finite automata(DFA) as well as Non-deterministic finite automata(NDFA)[16]. DFA is inefficient in terms of memory and NDFA are very slow while parsing a large string. These parsers take payload or packet data as input and compare them with regular expressions already constructed. If the pattern matches then an alert is fired. To detect the type of attack, a finite automaton is created for the corresponding regular expression. The packet data is then given as input, and the attack types are found out. If the input data is a normal behaviour then it is passed on, otherwise termed as an attack and stored for future use.

Becchi et al.[17] tried to reduce the memory related problem in DFA as they require more memory for storing states as compared to NDFA. The dissimilar states of the DFA are marked and then are merged with similar states using data structure implemented by them. Jiekun Zhang et al.[18] have also proposed a system for the similar problem but

have tried to reduce state transitions rather than states. They used the data state merging algorithm for merging the state transitions.

Gerald Tripp[19] devised a high-speed pattern matching algorithm. A Xilinx Field Programmable gate array(FGPA) with automata implementation and compresses some of the input patterns that are posed similar by the automata. The FPGA contains the hardware design of this system and is tested using the simulators. The regular expression rules are first created using a software and then loaded into the circuit. The expressions can be modified without posing any harm to the device.

Xu et al.[20] proposed a new automaton for regular expressions called Tunable Finite automata(TFA). TFA eliminates the problem of DFA and NDFA stated earlier. TFA allows more than one state to be active at a time, therefore, allowing the need of lesser number states. TFA allows bounding of a number of states till a proposed factor. TFA is used for pattern matching and intrusion detection.

2.6 Model Based IDS

Model-based intrusion detectors create a model for detecting intrusions. These model are created using different techniques. The techniques related to the domain of machine learning, data mining etc can be used. Hybrid models can be created as well which can detect signature based as well as anomaly based attacks.

Sri International Laboratories built a host detection model called IDES[21] used pattern recognition method to determine the behaviour of an activity. The system also includes an expert system that detects the unknown behaviours and classifies them as an attack or normal. If classified as an attack, then these systems store these activities in the database. They also built a model which could protect SCADA networks[22] using the similar type of procedures but used network monitoring protocols instead for detection purposes. Puthran et al. [23] has used different data mining methods to detect and analyze different attacks using methods like classification and clustering. They have mainly focused on the attacks prevailing in the network. They concluded that attacks are detected accurately when using both types of mining techniques. Shahadat et al. [24] used decision Table to detect intrusions. The decision table is a rule-based mining technique. This algorithm gave better performance and accuracy over the existing models. To select important features they used a new technique called Dropout (DP). which does a sequential search and drops out all non-relevant features keeping only the important features. Kathleen et al. [25] used a model formed by using Nave Bayes, decision trees and SVM as base classifiers This model showed high accuracy while reducing false positive rates. Firstly, they used SVM to divide the dataset into normal and attack, then attack points are fed to a decision tree and Naive Bayes which is used to classify these attacks. Gupta et al. [26] used techniques like K-Means Clustering,

Linear Regression generates rules automatically and detect intrusions based on these rules. Singh et al. [27] used different data mining algorithm for finding correct patterns of an attack.

There are a lot more fields where intrusion detection systems are being implemented and more research is carried out.

Chapter 3
Problem Statement

Over the years there is the rapid increase in the usage of internet technology along with the emergence of, new applications areas for computer networks. With the tremendous increase in network usage and technology advancement, the user data has become vulnerable to a great extent. There is a continuous threat to the data either over the network i.e. while data communication or at the host system by some insidious malicious users or masqueraders. So there is an urgent need for intrusion detection. The existing work includes detecting intrusion using audit or host logs. The log files contain and preserve the intrusions occurred till date. Many other modes of creating logs like key press inputs, commands, application logs are there but mainly network related interactions and activities or system based host logs are used. These log files are mainly stored somewhere either indefinitely or during the processing time only. This leads to huge volumes of data collection which exceeds overtime thus forcing many researchers to develop some log data reduction methods or find different ways to intrusion detection. Currently, research is going on not only for detecting intrusions but also giving a response at the same time so that intrusions can be handled along with that. Log-based detection not only faces problem-related to a large and increasing volume of log files but also from intruders who wish to access the log files so that they can know what type of attacks remain undetected. Model-based intrusion detection systems are less vulnerable to detecting the normal behaviour as an intrusion as they are trained machine learning models which improve overtime on the other hand log based detection works by finding patterns from the log files once they are stored. Therefore model based intrusion detectors are more in trend.

3.1 Objectives

The aim of our work is to detect any unwanted activity which can be categorized as illegitimate, in the system. The detection of these unwanted activities called intrusions is important as they pose a great threat to the information stored which is very much confidential. The main objectives are as follows:

i. To detect any unwanted and abnormal activity in the system by distinguishing it from the normal.

ii. To make an efficient model which can not only detect attacks already present in

the database but also the unknown attacks which are new to the detection system.

iii. To protect the confidential data from illegal access, as this data is leaked can result in a very dangerous situation for the organisation responsible for keeping the data safe as well as for the person to whom the data is related to. In these situations, whole responsibility comes on the organisation keeping the information.

iv. To decrease the chances of false alarms or false positive rate that is the detection system should not categorize a normal activity as an attack.

v. To make the detection system such effective that it can detect true attacks with accuracy.

3.2 Research Approach

The research work described in coming sections involves the generation of a model based intrusion detector using different data mining techniques. The NSL-KDD dataset has both training and test sets which are used for building and validating the model. **K-fold cross validation** method is used for first creating 10 folds of the dataset and then using these folds as input to build models so that the variance in the data can be reduced and this method also enhances the performance of the classifier. The technique is used for dividing the dataset into 10 folds and then the model is trained over the nine models and tested over the remaining fold. Then the model is fed with the test dataset for evaluating the model over different and known attacks. The model which provides the best performance evaluated using different evaluation parameters is the model which outperforms and provides the best intrusion detection capabilities. The model can detect known attacks accurately and learns overtime for detecting unknown attacks by storing those attacks for future reference. The work is done using the tool WEKA[28].

3.3 Dataset Description

The NSL-KDD dataset is used for preparing intrusion detection model which is a refined version of KDDcupp99 dataset. The work includes different tools and techniques used to develop an effective intrusion detection model on the NSL-KDD dataset. This dataset consists of training and testing dataset. The dataset includes 41 attributes and a label class which suggests the attack type is either normal or one of the attacks. The first nine features are basic features of network connection vector, next thirteen features are content features of network connection vector, and next nine are time based network

traffic features and rest i.e. the last ten are host based network traffic features. The dataset is as described in the table 3.1.

Table 3.1: Dataset Description

No.	Feature Name	Description	Type
1.	Duration	Time for which connection was on	Numeric
2.	Protocol_type	Protocol used for establishing a connection	Nominal
3.	Service	Service to establish connection at destination	Nominal
4.	Flag	Connection Status	Nominal
5.	Src_bytes	Data Bytes number in a connection from source to destination	Numeric
6.	Dst_bytes	Data Bytes number in a connection from destination to source	Numeric
7.	Land	If IP addresses and port numbers are equal, then 1 else 0	Numeric
8.	Wrong_fragment	If packets fragmented wrongly	Numeric
9.	Urgent	Packets with urgent flag bit on	Numeric
10.	Hot	Number of indicators in the system	Numeric
11.	Num_failed_logins	Number of login in the system failed	Numeric
12.	Logged_in	Status bit 1 if user is logged in	Numeric
13.	Num_compromised	Conditions in which connection was compromised	Numeric
14.	Root_shell	If shell was attacked then 1 else 0	Numeric
15.	Su_attempted	If *su root* command was used to get shell access	Numeric
16.	Num_root	Number of times *root* accessed	Numeric
17.	Num_file_creations	Number of the file creations	Numeric
18.	Num_shells	Number of shell prompts	Numeric
19.	Num_access_files	Operations on access control files	Numeric
20.	Num_outbound_cmds	Outbound commands in an ftp connection	Numeric
21.	Is_hot_login	1 if login from hot (admin or root) else 0	Numeric
22.	Is_guest_login	1 if *guest* login else 0	Numeric
23.	Count	Connections to the same destination	Numeric
24.	Srv_count	Connections to the same service(port number)	Numeric
25.	Serror_rate	Connections that have activated the flags s0,s1,s2 or s3	Numeric
26.	Srv_serror_rate	Percentage of connections that contribute to Srv_serror_rate	Numeric
27.	Rerror_rate	Number of connections that activated the flag REJ	Numeric

to be cont'd on next page

Table 3.1: Dataset Description (Cont.)

No.	Feature Name	Description	Type
28.	Srv_rerror_rate	Percentage of connections that contributed to Rerror_rate	Numeric
29.	Same_srv_rate	Connections to same service from the connections in count(23)	Numeric
30.	Diff_srv_rate	The number of connections that were to different services, among the connections in count(23)	Numeric
31.	Srv_diff_host_rate	Percentage of connections to different destination machines	Numeric
32.	Dst_host_count	Connections with same destination host IP address	Numeric
33.	Dst_host_srv_count	Connections with same port number	Numeric
34.	Dst_host_same_srv_rate	Percentage of connections to same service	Numeric
35.	Dst_host_diff_srv_rate	Percentage of connections to different servers	Numeric
36.	Dst_host_same_src_port_rate	Percentage of connections that were to same source port	Numeric
37.	Dst_host_srv_diff_host_rate	Percentage of connections to different destination machines	Numeric
38.	Dst_host_seerror_rate	Connections that activated the flag	Numeric
39.	Dst_host_srv_serror_rate	Connections that activated the flags	Numeric
40.	Dst_host_rerror_rate	Percentage of connections that activated the flag REJ	Numeric
41.	Dst_host_srv_rerror_rate	Percentage of connections that activated the other flags	Numeric

The attack class present in the NSL-KDD training dataset contribute to four type of attacks[29]:

1. **DOS:** Denial of service is a type of attack in which an attacker tries to make a network or any resource unavailable to the users which are its legitimate users. This is done by disrupting the services of a host or a node which is connected to the Internet temporarily so that the legitimate requests cannot get fulfilled e.g. flooding the targeted machine or resource.

2. **Probe:** Probe is a type of attack in which the attacker aims at accessing the information about the remote system or tries to know the network state -e.g. A user trying to send an empty e-mail just to check if the person is online or to know the

target person’s usage patterns.

3. **U2R:** U2R is any unwanted access to exploit the vulnerabilities in the victim’s system by accessing its local account to get the root/admin privileges of the system, mainly in an organisation e.g. kernel level attacks

4. **R2L:** R2L is any unwanted access from a remote machine to a local machine(remote to local). In this type of attack, the attacker tries to get into a remote machine to gain local access to the victim machine. E.g. remotely logging into a system by knowing its username and password.
   The normal class label tells that the activity does not account for any of the attacks. The test set contains many other attacks which are not present in the training dataset.

Chapter 4

Proposed Solution

The NSL-KDD dataset is an intrusion detection dataset which is an improvement over the KDDcup99 dataset. The NSL-KDD dataset contains 41 attributes and the last attribute is a class attribute which classifies the data into normal activity or an attack. The NSL-KDD dataset is further divided into two datasets:

1. **Training Dataset**: The training dataset is used to build the model. The dataset contains 125973 instances which are classified as either normal, u2r, r2l, probe, dos.

2. **Testing Dataset**: The testing dataset is used to test the model that how accurately it can perform predictions on the new set. The test dataset contains 22543 instances with 3750 instances having attack type not present in the training dataset.

The figure 4.1 shows the steps followed to create an intrusion detection model and is as below.

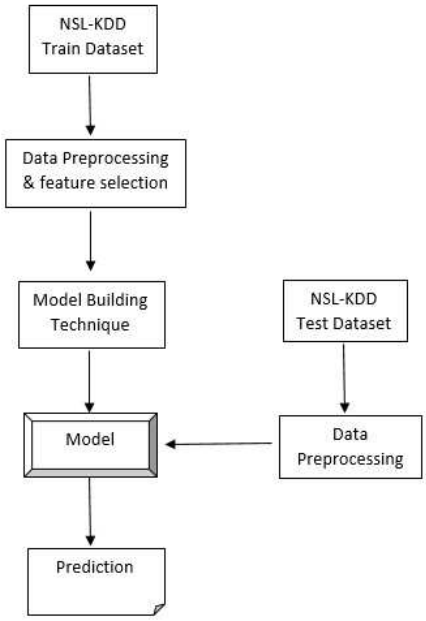

Figure 4.1: Basic Work Approach

4.1 Preprocessing

The raw data is comprised of data with varying scales. Normalizing the dataset will give a boost to the performance of the classifiers. The dataset is pre-processed by normalizing to a range 0-1, that means the highest value of the attribute which is normalized is 1 and the lowest value is 0. The normalization of the dataset can be done using feature scaling, z-score and many other techniques. The dataset here is normalized using Feature scaling which gave better results than z-score and is used more by researchers as compared to z-score

4.1.1 Feature Scaling

Feature scaling method is used for getting the range of values of data on a common scale. The reason for using this method over standardization is that it gives smaller standard deviations, which can nullify the outliers.

$$v = \sqrt{\frac{\sum_{i=1}^{n}(s_i - \overline{s})^2}{n-1}} \tag{4.1}$$

where,
v is the standard deviation a sample,
s_1,s_2.....s_n are the observed values of the sample items,
$\overline{s}$is the mean value of these observations,
n is the number of observations in the sample

The reason the most classifiers do not perform well on data which is not normalized is that the learners calculate the distance between two feature sets by Euclidean distance. If one of the features has a larger value, then the value of the resultant distance will mostly be around the feature with a higher value. To make each feature approximately proportionately to the final distance, the range of all features need to be on the same scale. There are different methods for preprocessing which are,

4.1.1.1 Rescaling

Rescaling or min-max scaling is the most simple method of all and here the values of the features are converted to a scale of [0, 1] or [1, 1]. The nature of the data is used as a decision parameter for choosing one of the two scales. The rescaling technique can be applied as:

$$f' = \frac{f - min(f)}{max(f) - min(f)} \tag{4.2}$$

where,

f is an original value,
f' is the normalized value

4.1.1.2 Mean normalization

It is similar to rescaling with the difference that raw score is subtracted with mean of the population instead of the minimum raw score.

$$f' = \frac{f - \bar{f}}{max(f) - min(f)} \tag{4.3}$$

where,
f is an original value,
f' is the normalized value

4.1.1.3 Standardization

In this method, each feature value is converted in a way such that the mean of each feature is zero and the variance is one. Standardization firstly finds the mean and variance of each feature. Next, the calculated mean is subtracted from each feature. Then this value is divided by the standard deviation.

$$f' = \frac{f - \bar{f}}{\sigma} \tag{4.4}$$

where,
f is the original feature vector,
$\bar{f}$ is the mean of that feature vector,
σ is its standard deviation.

4.1.1.4 Scaling to unit length

This method first scales each term of the feature vector so that the length of the final vector is one. For this each term is divided by the Euclidean length of the vector:

$$f' = \frac{f}{\|f\|} \tag{4.5}$$

4.1.2 Z-Score

z-score is calculated as the difference between the sample value and the mean value by the variance of that point. z-score of the values in the sample more than mean are positive, while values with lower mean have negative z-score and is a dimensionless

quantity. The Z-score can be calculated as

$$z - score = \frac{f - \mu}{\sigma} \tag{4.6}$$

where,
μ is the mean of the population,
σ is the standard deviation of the population

This formula requires the population mean and the population standard deviation which is many a time unrealistic to know. In that case, z-score can be estimated by using the sample mean and variance as:

$$z - score = \frac{f - \bar{f}}{S} \tag{4.7}$$

where,
$\bar{f}$ is the mean of the sample.
S is the standard deviation of the sample.

Here, Feature scaling method is used over z-score method to normalize the dataset as the standard score requires the whole population mean and variance rather than a set of sample elements, therefore, making it complex to get the normalized values for such a huge dataset. z-score are mainly used for data which follow a normal distribution or are nearly bell-shaped and The NSL-KDD dataset is not normally distributed over the mean.

4.2 Feature Selection

Then, feature selection is done as it is required for dimensionality reduction and removal of unimportant features which in turn also improves the accuracy and other parameters related to the outcome. The feature selection techniques like CFS subset evaluation with greedy search, Pearson correlation, Info gain attribute evaluation and Info gain with filter method techniques were used and are described below.

4.2.1 Correlation Feature selection(CFS) Subset Evaluation

CFS first creates a subset of features and then evaluate each group rather than adding one by one to the set. CFS uses both Wrapper and Filter methods. In Wrapper methods, a search algorithm is used to select appropriate features for each feature subset and then each subset is made to execute on a model. Wrapper methods are generally very expensive and can cause overfitting. These methods can be an overhead and very time

consuming if the dataset is huge. Filters methods on the other hand also do searching, but instead, the subset is evaluated on a more simple filter. In filter methods, each feature has a defined score and on the basis of that score the features are added or removed. This method further uses heuristic methods like hill climbing, which first create a subset of features and add the new subset of features created if they improve the model performance over the older subset. Sometimes large datasets can lead to exhaust the memory, so in such cases, a stopping point is generally applied. The criteria are stopped when the subset gives the highest score over a model and adding new subset does not contribute for any more good. Alternative search-based techniques can be Best first, Greedy forward selection, Greedy backward elimination and are as the name describes.

This method uses a hypothesis to evaluate a feature which is: "Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other". To get a more meaningful estimate of the worth of the output generated by an attribute selection scheme measures like merit are calculated. The following equation gives the merit of a feature subset C consisting of n features:

$$M_{C_n} = \frac{n\overline{x_{cf}}}{\sqrt{n + n(n-1)\overline{x_{ff}}}} \tag{4.8}$$

Here,
$\overline{x}_{cf}$ is the average value of correlation and feature values calculated from the features,
$\overline{x}_{ff}$ is the average value of correlations between features

The Subset evaluation criterion can be calculated as:

$$CFS = \max_{C_n} \left[\frac{x_{cf_1} + x_{cf_2} + \cdots + x_{cf_n}}{\sqrt{n + 2(x_{f_1f_2} + \cdots + x_{f_if_j} + \cdots + x_{f_kf_1})}} \right] \tag{4.9}$$

where,
x_{cfi} and $x_{f_if_j}$ variables are correlations

4.2.2 Person Correlation

The Pearson correlation coefficient is a measure of the linear correlation between two variables A and B having a value between +1 and -1, where 1 is the total positive linear correlation, 0 is no linear correlation, and 1 is the total negative linear correlation. The coefficient for population is represented by the Greek letter called rho and is calculated as:

$$\rho_{A,B} = \frac{\operatorname{cov}(A, B)}{\sigma_A \sigma_B} \tag{4.10}$$

where,

σ_A is the variance of A,
σ_B is the variance of B

*cov(A,B)*is calculated as,

$$\operatorname{cov}(A, B) = \mathrm{E}[(A - \mu_A)(B - \mu_B)] \tag{4.11}$$

where,
μ_A is the mean of A,
μ_B is the mean of B,
E is the expectation

Pearson's correlation coefficient for a sample is mostly represented by the letter r and can be derived by using the values of one dataset *x1,...,xn* containing n values and another dataset *y1,...,yn* containing n values:

$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{n}(b_i - \bar{b})^2}} \tag{4.12}$$

where,
n is size of the sample,
a_i,b_i are the individual sample values,
sample mean is,

$$\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i \tag{4.13}$$

The above formula can also be written as,

- More is the size of sample than the population distributed normally then the sample is the best estimate of the population and therefore more accurate prediction can not be made.
- If population is not normally distributed and sample size is huge then also the coefficient is not biased but is not an appropriate prediction.

4.2.3 Info Gain Evaluation Feature Selection

The InfoGain is an implementation of a feature selection method by using information gain theory. It measures the information gained through a specific attribute for prediction of a class variable after the value for a feature is observed. This technique is mainly used to reduce the bias towards multi-valued attribute while evaluating an attribute. In-

formation gain for a sample S using attribute R can be calculated as:

$$Gain(S,R) = I - \sum_{v \in values(R)} \left(\frac{t_i}{s} I_i\right) \tag{4.14}$$

where,
I and I_i is the information before and after the split over the attribute *i*,
t_i is the number of instances in *i*,
s is the total number of instances in *R*

The intrinsic value for I^{th} feature is:

$$IV = I_i = -\sum_{v \in values(R)} \frac{|\{x \in S | value(x,r)\}|}{|S|} \cdot \log_2 \left(\frac{|\{x \in S | value(x,r)\}|}{|S|} \right) \tag{4.15}$$

The ratio between information gain and the intrinsic value is called Gain Ratio and is,

$$GR(S,R) = \frac{Gain(S,R)}{IV(S,R)} \tag{4.16}$$

After applying these techniques for feature selection, the feature set is passed through a decision table algorithm to check the accuracy of the evaluated feature set and thus selecting the best feature set among them. A decision table is a technique in which a table is formed which is called a decision table containing the conditions and corresponding rules which are to be followed to reach the actions to be performed. For feature selection four techniques are used namely Co-related feature subset evaluation with greedy approach, Pearson Correlation, Info gain Evaluation. CFS method creates subsets and then find correlated features using greedy approach. Pearson correlation is finding the interdependence between features and thus finding the features with lowest standard deviation. Info Gain evaluation is selecting the features having the highest information gain. The techniques and results are as shown in the table 4.1.

Table 4.1: Feature Selection Techniques

Selection Technique	**No. of Features**	**Features Selected**	**Accuracy with Decision Table**
CFS+greedy search	11	3,4,5,6,12,14,26,29,30, 37,38,42	98.82%
Pearson Correlation	12	3,4,5,6,12,14,26,29,30, 37,38,14,42	98.81%

to be cont'd on next page

Table 4.1: Feature Selection Techniques (Cont.)

Selection Technique	**No. of Features**	**Features Selected**	**Accuracy with Decision Table**
Info gain evaluation	30	1,2,3,4,5,6,8,10,12,22, 23,24,25,26,27,28,29,30, 31,32,33,34,35,36,37,38, 39,40,41,42	98.26%
Our approach	23	2,3,4,5,6,12,23,24,25, 26,29,30,31,32,33,34,35, 36,37,38,39,40,42	98.27%

The last approach includes applying a filtering technique to the info gain attribute evaluation technique until the accuracy improves thus reducing the feature set tremendously. Here, the first feature selection technique gives the best accuracy so the features of the first approach will be used for further evaluation. The selected features are as shown in the table 4.2.

Table 4.2: New Feature Set

Feature No.	**Feature Name**
1.	Service
2.	Flag
3.	Src_bytes
4.	Dst_bytes
5.	Logged_in
6.	Root_shell
7.	Srv_serror_rate
8.	Same_srv_rate
9.	Diff_srv_rate
10.	Dst_host_srv_diff_host_rate
11.	Dst_host_serror_rate
12.	Class(normal or attack)

4.3 Model Building Techniques

After the training dataset is refined and is ready for usage, it is used as a basis to build the model for making the predictions and evaluating the model using various evaluation parameters.

4.3.1 Classification Algorithm

The model built using classification techniques are more appropriate in detecting signature-based attacks i.e. attacks which are already present in the database, they are not as efficient in detecting novel attacks. So, models built using these algorithms are helpful in environments where the intrusion to be detected are mainly from known sources or the database used as a base is very powerful and highly well developed. In these kinds of situations, these model give low false positive rates. The figure 4.2 shows the steps used to build classification model.

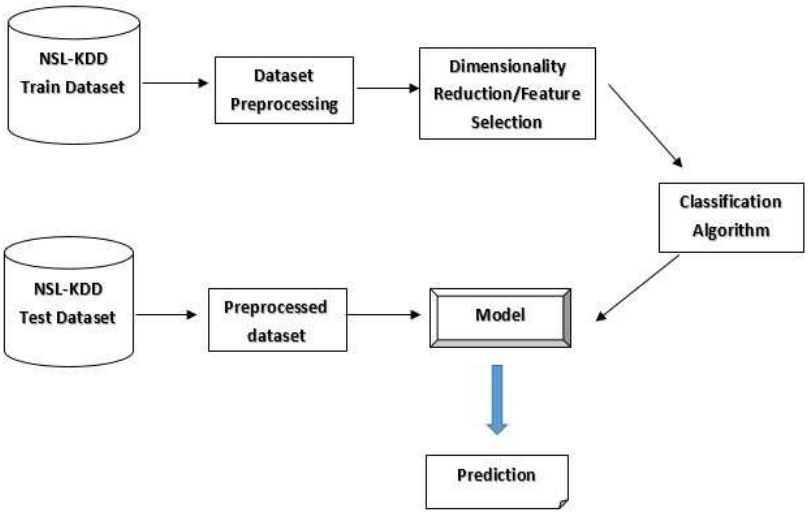

Figure 4.2: Work Mechanism for classification model

4.3.2 Hybrid Algorithm

The model is built using both supervised and unsupervised learning i.e. classification as well as clustering methods. Many new types of attacks remain undetected using classification techniques so the new type of attacks are handled using clustered data and the signature attacks are detected and classified using the classification data. Hybrid algorithms combine both misuse and anomaly-based detection so that they can preserve the quality of high detection rate using misuse detection and efficient classification of a new type of attack using anomaly-based detection methods. The figure 4.3 shows the steps used to build hybrid model.

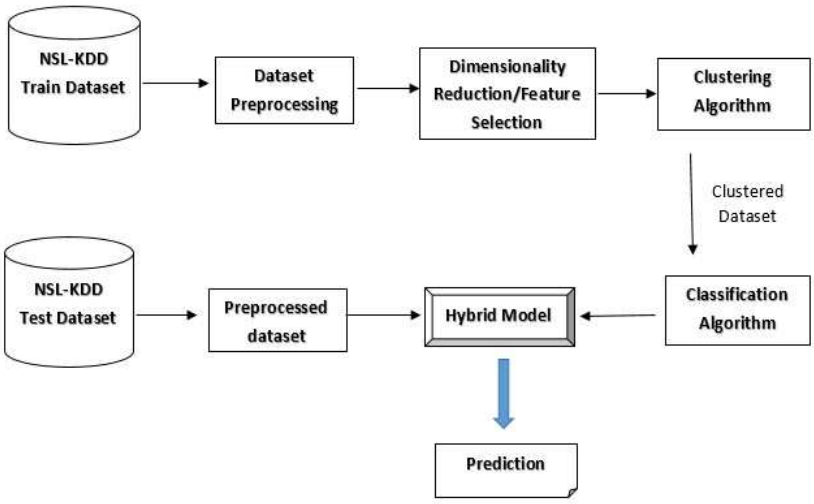

Figure 4.3: Work Mechanism for hybrid model

4.4 Performance Evaluation

True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, Mathews Correlation Coefficient (MCC) and accuracy are used as evaluation parameters which are derived using confusion matrix. The matrix is as shown in the table 4.3.

Table 4.3: Confusion Matrix

	Normal	**Intrusion**
Normal	TP	FP
Intrusion	FN	TN

Where,

TN - Instances correctly predicted as not being intrusions.

FN - Instances wrongly predicted as not being intrusions.

FP - Instances wrongly predicted as intrusions.

TP - Instances correctly predicted as intrusions.

$$Accuracy = \frac{Numberof Samplescorrectlyclassified}{Totalnumberofsamplesintestdata} \tag{4.17}$$

$$TruePositiveRate(TPR) = \frac{TP}{TP + FN} \tag{4.18}$$

$$FalsePositiveRate(FPR) = \frac{FP}{FP + TN} \tag{4.19}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.20}$$

$$MCC = \frac{TP\ X\ TN - FP\ X\ FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.21}$$

Chapter 5
Implementation

The model is implemented using various data mining algorithms. The models can detect misuse based attacks (known attacks) as well as anomaly attacks (unknown attacks). The model is implemented using classification and hybrid algorithms. The model implemented using the classification model detects misuse based attacks efficiently but is not as much efficient in detecting unknown or anomaly attacks. This problem is addressed by models built using hybrid algorithms that is they can detect both types of attacks aptly. The hybrid algorithms are constructed using the combination of classification and clustering algorithms.

5.1 Classification Algorithms

These algorithms are best suitable for detecting misuse based intrusions i.e. intrusions which are already present in the database. These algorithms are applied using the WEKA tool [30].

5.1.1 C4.5

C4.5 is a supervised learning algorithm and is used to generate a decision tree which is an improvement over the earlier decision tree algorithm ID3. The C4.5 algorithm is a statistical classifier which classifies the nodes on the concept of information gain as in section 4.2.3. C4.5 generates a tree and select each attribute by first calculating the information gain of an attribute over the parent attribute and the node with the maximum information gain is selected as the next parent node. 4.14. In these tree structures, leaves nodes are the class labels which if reached gives the class label to the test sample and thus a prediction occurs, and branches are features that help in finding and traversing to the leaf or class label for the test point. Information gain or Entropy is as:

$$H(E) = I_E(p_1, p_2, ..., p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{5.1}$$

where, $p_1, p_2, ...$ are fractions that sum up equal to one

The information gained through a particular attribute for prediction can be calculated by,

$$IG(E, r) = H(E) - H(T|r) \tag{5.2}$$

where,
IG(E,r) is information gained for a particular attribute r on the class attribute E,
H(E) is the entropy of the class attribute and is calculated as in equation 5.1,
n is the no. of attributes further dividing that particular attribute,
H(E|r) is calculated as,

$$H(E|r) = \sum_{r} p(r) \sum_{i=1}^{n} -Pr(i|r) \log_2 Pr(i|r) \tag{5.3}$$

The attribute which has the lowest entropy and thus the highest information gain is selected as the next split attribute. To make the decision that the next node should be selected or not is done by estimating the error rates. The information gain of a particular attribute needs to be calculated as after that only the decision of a particular node to be selected or rejected will be established.

C4.5 Algorithm works by taking the set of instances to reach each node and then for each of this possibility calculate certain errors in reaching to that node. This gives certain errors E, out of total number of instances N.[31].The formula can be derived using a given confidence level and estimating the confidence limits as,

$$P\left[\frac{r-t}{\sqrt{t(1-t)/N}} > z\right] = c \tag{5.4}$$

where,
N is the number of samples,
r = E/N is the observed error rate,
t is the true error rate
The formula leads to an upper confidence limit for t. Estimate for the error rate e can be calculated as,

$$e = \frac{r + \frac{z^2}{2N} + z\sqrt{\frac{r}{N} - \frac{r^2}{N} + \frac{r^2}{4N^2}}}{1 + \frac{z^2}{N}} \tag{5.5}$$

The node with the least error rate is chosen until the leaf node is reached. The confidence level is generally taken as c=25%.

5.1.2 K-nearest neighbors(KNN)

KNN is a supervised learning algorithm which is also called a lazy learner. When a lot of training samples are fed into the learner then it stores the whole dataset in memory or rather plots the dataset over n-dimensional data space. When the test sample is fed to this model then it plots whole test sample into n-dimensional space and then finds

the k-nearest neighbours using any of the distance measures (like Manhattan distance, Euclidean distance) from the training plot or sample. The learner does this whenever it needs to predict for a test sample and it goes through the full training dataset every time for this step, so it is called a lazy learner.

The distance measures can be calculated as,

- Manhattan Distance:

$$d_M(p,q) = \sum_{i=1}^{N} |p_i - q_i| \tag{5.6}$$

- Euclidean Distance:

$$d_E(p,q) = \sum_{i=1}^{N} \sqrt{p_i^2 - q_i^2} \tag{5.7}$$

  where,

  p,q are instances composed of *N* features, such that *p*=$p_1, ..., p_N$,*q*=$q_1, ..., q_N$

Given a test point s for which the attack class is to be found, and a data set *T* = $t_1, c_1 ... t_n$, where t_i is the i^{th} element and l_i is the label, the algorithm will find a subset m= $m_1, l_1 ... m_q$ such that $m \in T$ and $\sum_1^q d(s, m_q)$ is minimal. Y thus is set of k points which are neighbours and the closest to the test point. Where l is the class label and is $l = f(l_1 ... l_k)$.

The main obstacle while using the k-nearest neighbours is the choice of the value of k. If the value of k chosen is too large then the neighbourhood will contain many irrelevant points leading to difficulty in predicting with precision and if taken too small will lead to the wrong number of points.

5.1.3 Ripper

Ripper or Repeated Incremental Pruning to Produce Error Reduction is a rule-based learner which creates a set of rules for classifying the problem and reduces the error. An error is the number of misclassified samples occurred while using these rules to evaluate a model. The learner does not make any a priori assumptions to reach the final concept rather it works on the assumption that the data on which it is trained is similar in a way as the unseen data. Ripper creates association rules with reduced pruning. Firstly, an initial rule set is created and new rules are added using some heuristic method, which is called the growing phase. Then pruning is applied to this large dataset and is applied to that rule which is least effective. The pruning is stopped when further application leads to growth in error rate. The algorithm is applied as:

1. The rules are added greedily to the rule set R= till the rule formed is optimum. The value of each feature is evaluated and the condition which gives the highest information conferring to information gain is selected.

2. Then the rule set is pruned inclementally pruning each rule. The pruning metric is,

$$m = \frac{p}{(p+n)} \tag{5.8}$$

where,
p and n are positives and negatives of the predicted rules

3. After the generation of initial rule set R_i, two more variants of each rule are generated, one from the empty rule set and another by appending next rules to the rule set R. The pruning is done by using the formula of accuracy as in Equation 4.17 Then the smallest DL(description length) for each rule is calculated. The condition with the smallest DL is selected. If there are still more residual positives then the whole procedure is followed again.

5.1.4 Naive Bayes

Naive Bayes algorithm comes under the class of probabilistic algorithms. The nave Bayes algorithm use probability theory and the Bayes theorem for prediction. Here, the probability of each category is calculated for a given sample and then the sample is classified based on the highest one. In the Bayes theorem, the probability of each feature is calculated by using the previous knowledge or the given conditions that might be related to that feature.

For a test instance *t=(*$t_1, ..., t_n$*)* having n features, instance probabilities p($Z_m|t_1, ..., t_n$) are assigned for each M outcomes or classes C_m. But this theorem performs problematically for large datasets as biasing such models using probability theory is infeasible. Therefore Bayes' Theorem make it more in tactable and the probability is estimated as,

$$p(Z_m \mid t) = \frac{p(Z_m)p(t \mid Z_m)}{p(t)} \tag{5.9}$$

However, the denominator is always constant as the values of feature t_i do not depend on Z and the t_i is always provided. Therefore the above equation can be rewritten as,

$$p(Z_m, t_1, \ldots, t_n) = p(Z_m) \prod_{i=1}^{n} p(t_i \mid Z_m) \tag{5.10}$$

The independence of attributes from each other is assumed while using this classifier. These methods vary of errors and can generally handle missing values

5.1.5 Random Forest

Random forest algorithm creates a forest of a number of trees for classifying the problem. If the trees are more in number, then better is the accuracy. In random forest instead of choosing the root node by using the Gini index or information gain, the root node and the further nodes will be chosen randomly. This will let in the creation of different decision trees and then these trees will contribute to the random forest. These are ensemble learning techniques for classification or regression models. ID3 and decision tree classifiers are used as base learners in this random forest algorithm. Decision trees are mostly used when talking about tree algorithms. In general, trees that are very large or go to the depth learn much about the problem and are very efficient and but sometimes overfit that is very high variance between the points. So they average the outcomes of these huge trees which are trained on different parts of the dataset thus reducing variance. This comes a little bit expensive but generally boosts the model. They apply the bootstrap aggregating, or bagging, to subtrees or learners.The test set *S*=($s_1, ..., s_n$) produce tree results *T*=($t_1, ..., t_n$) after applying bagging. Then a sample is selected randomly with replacement and each tree is fitted to the selected sample i.e training a classification tree f_b on S_b, T_b where b is the times for which bagging is performed. The test samples or unseen queries can be calculated as:

$$\hat{r} = \frac{1}{N} \sum_{n=1}^{N} f_n(x') \tag{5.11}$$

where,

N is the number of times bagging performed,

x' is the sample in the data set

and for classification trees use the majority vote principle. Bagging results in good outcomes and high performance because the variance lessens using this algorithm. Creating many trees from the same training set give trees having a high correlation.

5.1.6 Random Subspace

Random subspace or feature bagging method uses ensemble techniques for generating a model. The technique used is bagging. In Bagging technique, additional data is produced from the original dataset which decreases the deviation thus resulting in an effective prediction. Rather, in random subspace, the features are randomly selected with replacement and then fed to a learner which helps in refraining from the selection of feature with higher importance every time. The individual learners are combined by using majority voting or posterior probabilities[32].

For the selection of base learner, any good learner can be used from among the

explained earlier. AdaBoost is used as a base learner here as the AdaBoost algorithm has the ability to combine many weak classifiers into one strong classifier thus enhancing the prediction results. Moreover, an equal voting weight scheme does not work as most classifiers may contain noisy features so, this algorithm uses a variable weight scheme for each classifier and thus performs well as compared.

Let the dataset to be classified be $t_1, ...t_m$ where t_i be the samples and weak classifiers$c_1, ...c_l$ that classify these samples which produce outputs as $c_j(t_i)$ having value between 1 and -1. Then the classifier is as:

$$R_{(n)}(t_i) = k_1 c_1(t_i) + \cdots + k_n c_n(t_i) \tag{5.12}$$

where,
k_i is constant value produced for a given classifier,
n is the number of weak classifiers

The difference between random forest and random subspace is that random subspace does the splitting of the node in top-down fashion randomly whereas random forest use information theory for choosing the way of a splitting of an attribute.

5.2 Hybrid Algorithms

These algorithms apply both clustering and classification approaches for intrusion detection and can detect misuse as well as anomaly based intrusions. The techniques are implemented using WEKA tool. K-means algorithm is used as the clustering algorithm in all the combinations used ahead to create hybrid model as it is the most widely used algorithm and has good prediction characteristics. K-means is a clustering technique which lets then instances to be grouped in different clusters on the basis of similarity. Here k denotes the number of clusters which are areas distinct from each other as much as possible for better clustering of data. The optimal value of k need to be chosen an thus is calculated from the given data. This algorithm helps in minimizing the variance between each cluster formed. The k-means algorithm is as below:

1. In each step k data points are chosen as the centre.

2. The distance of other points from these k centroids is calculated using one of the distance measures (Euclidean distance here) and then the points are clustered to the centroid with least distance.

3. At each step value of the data points in each cluster is averaged and then it is chosen as the new centroid.

4. The process continues till the value of cluster centroids keep changing. Then finally the data is clustered into k clusters.

Let, $x_1, ..., x_n$ be the number of instances in a dataset, from them k number of points are chosen as centroid for each cluster. Then each instance x_i is subtracted from every centroid c_i to know to which center it is the closest. Then these instances are put in the respective clusters. This algorithm continues as explained in above algorithm. The K0means clustering uses the following objective function and is as:

$$O = \sum_{j=1}^{k} \sum_{i=1}^{n} \|t_i^{(j)} c_j\|^2 \tag{5.13}$$

where, *o* is the objective function,
n and *k* are the number of instances and clusters respectively,
$t_i^{(j)}$ the i^{th} instance, $\|t_i^{(j)} c_j\|^2$ is the distance function

In this method value of k need to be chosen efficiently for the data beget clustered well. Anyhow there no practical or theoretical formulation for choosing k, one way is to use hit and trial method that is based on some criteria choose some values of k and for each value form the clusters to choose the best one.

5.2.1 Kmeans + C4.5

K-means is an unsupervised clustering algorithm and C4.5 is a supervised algorithm that is it has a set of predefined classes. First K-means is applied to the dataset and then, the C4.5 algorithm is applied to the clustered dataset as explained in section **??**.

5.2.2 K-means + Naive Bayes

Firstly, the K-means algorithm is applied on the dataset which is applied in the same way as above and then the naive Bayes algorithm as explained in section **??** is applied on the clustered dataset.

5.2.3 K-means + Ripper

The dataset is firstly passed through the K-means clustering algorithm and then the Ripper classification algorithm as explained in section **??**.

5.2.4 K-means + Random Forest

After applying K-means algorithm to the dataset, the random forest classifies the instances into respective class. Random Forest is implemented in the same way on the clustered dataset as explained in section 5.1.5.

Chapter 6

Results and Analysis

6.1 Classification Algorithms

Table 6.1: Performance Evaluation of Classification Algorithms

Classification Algorithm / Performance Measure	TPR	FPR	Precision	MCC	Accuracy
C4.5	0.855	0.113	0.859	0.750	85.54%
KNN	0.843	0.117	0.858	0.735	84.27%
Ripper	0.857	0.136	0.881	0.732	85.67%
Naive Bayes	0.787	0.183	0.803	0.646	78.72%
Random Forest	0.839	0.137	0.843	0.711	83.89%
Random Subspace	0.837	0.141	0.905	0.5864	83.70%

The performance evaluation of various parameters as in the table 6.1 shows that Ripper and C4.5 outperform among all the techniques giving a good accuracy score with the former giving highest true positive rate and latter giving the lowest false positive rate.

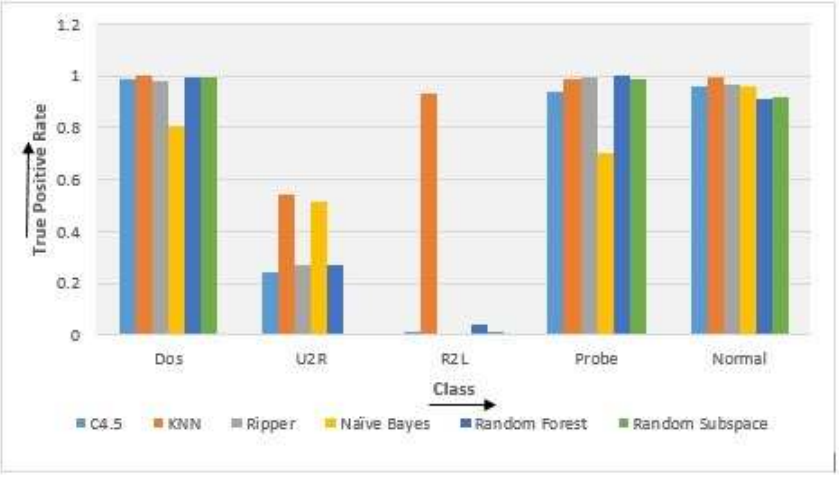

Figure 6.1: Analysis of Classification Algorithms for different class labels over True Positive Rate

From the figure 6.1 it can be visualized that KNN is giving better results for the attacks U2R and R2L as compared but these attacks constitute very lesser part of the

complete data set which nullifies the effect of giving the best performance for these attacks thus Ripper and C4.5 outperforms among all and can be seen from the table as well as figure above.

6.2 Hybrid Algorithms

Table 6.2: Performance Evaluation of Hybrid Algorithms

Hybrid Algorithm/ Performance Measure	TPR	FPR	Precision	MCC	Accuracy
K-means+C4.5	0.859	0.130	0.854	0.743	85.11%
K-means+Naive Bayes	0.845	0.087	0.890	0.779	84.51%
K-means+Ripper	0.848	0.146	0.887	0.778	84.81%
K-means+Random Forest	0.827	0.143	0.831	0.682	82.73%

The performance evaluation of various parameters as shown in the table 6.2 shows that all the techniques give fairly comparative result and give good accuracy on unknown data also. K-means with C4.5 gives the highest true positive rate and K-means with Nave Bayes gives the lowest false positive rate.

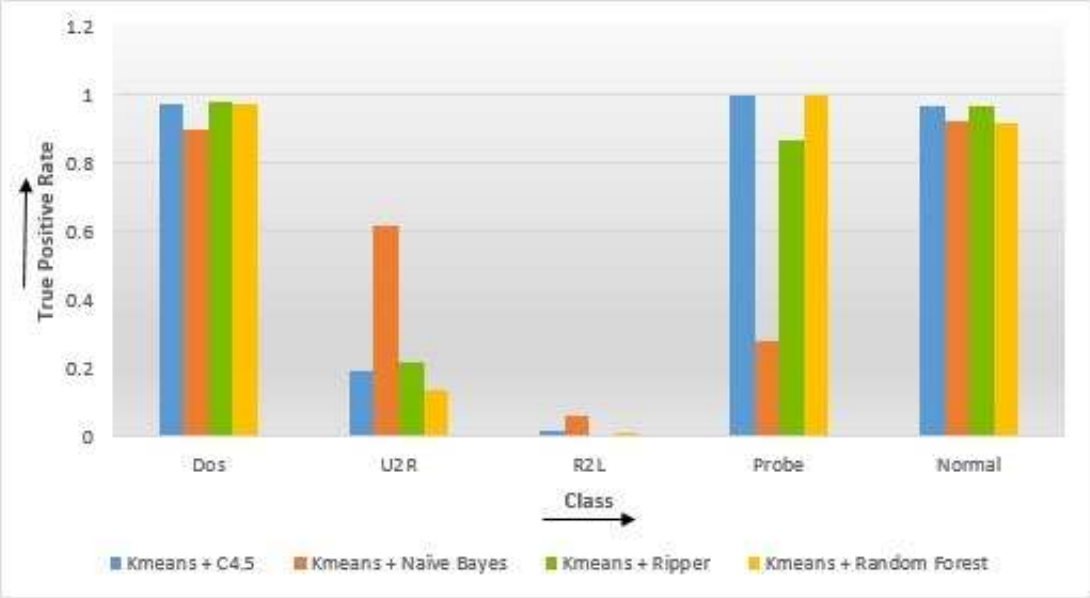

Figure 6.2: Analysis of Hybrid Algorithms for different class labels over True Positive Rate

The figure 6.2 depicts that K-means with Naive Bayes improves the true positive rate of every attack and also gives a good accuracy. However other three techniques

still give better results as U2R and R2L contribute to very lesser part of the dataset and these techniques have a higher true positive rate as compared to K-means with Naive Bayes for other more likely attacks.

Chapter 7
Conclusion

7.1 Conclusions

i. From the results of classification models and hybrid models, it can be concluded that hybrid models perform much better as they detect new types of attacks(unknown) whereas, the classification algorithm ignore the new attacks

ii. Hybrid models give better true positive rate and lower false positive rate compared to classification models.

iii. Hybrid models use the advantage of classification algorithm by maintaining lower false positive rate for known attacks as they are implemented using classification algorithms as well.

iv. On comparing the results of classification models it can be concluded that among all the classification models C4.5 model has given the best performance.

v. On comparing the results of hybrid models it can be concluded that among all the hybrid models, K-means + C4.5 model has given the best performance.

vi. C4.5 classification model has given an accuracy of 85.54% and K-means + C4.5 hybrid model has given an accuracy of 85.11% as shown in the tables 6.1 and 6.2.

vii. Rest of the models have also shown a good performance as can be seen from Table 4 and Table 5 and, figure 5 and figure 6.

7.2 Future Scope

i. The dimensionality reduction of a dataset using principal component analysis can be achieved for better results and better data visualization.

ii. More advanced algorithms like neural networks can be used to model the system for better detection rate.

iii. The system can be made to respond to intrusions along with the detection with lesser false alarms.

References

[1] J. P. Anderson, "Computer security threat monitoring and surveillance," *Technical Report, James P. Anderson Company*, 1980.

[2] D. E. Robling Denning, *Cryptography and data security*. Addison-Wesley Longman Publishing Co., Inc., 1982.

[3] D. Denning and P. G. Neumann, *Requirements and model for IDES-a real-time intrusion-detection expert system*. SRI International, 1985.

[4] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 darpa offline intrusion detection evaluation," *Computer networks*, vol. 34, no. 4, pp. 579–595, 2000.

[5] J. W. Haines, R. P. Lippmann, D. J. Fried, M. Zissman, and E. Tran, "1999 darpa intrusion detection evaluation: Design and procedures," tech. rep., Massachusetts Inst Of Tech Lexington Lincoln Lab, 2001.

[6] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset," in *Information Science and Security (ICISS), 2016 International Conference on*, pp. 1–6, IEEE, 2016.

[7] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, L. T. Heberlein, C.-L. Ho, K. N. Levitt, B. Mukherjee, S. E. Smaha, T. Grance, *et al.*, "Dids (distributed intrusion detection system)-motivation, architecture, and an early prototype," in *Proceedings of the 14th national computer security conference*, vol. 1, pp. 167–176, Washington, DC, 1991.

[8] P. G. Neumann and P. A. Porras, "Experience with emerald to date.," in *Workshop on Intrusion Detection and Network Monitoring*, pp. 73–80, 1999.

[9] J. Yang, P. Ning, X. S. Wang, and S. Jajodia, "Cards: A distributed system for detecting coordinated attacks," in *IFIP International Information Security Conference*, pp. 171–180, Springer, 2000.

[10] Y. Lin, Y. Zhang, and Y.-j. Ou, "The design and implementation of host-based intrusion detection system," in *Intelligent information technology and security informatics (iitsi), 2010 third international symposium on*, pp. 595–598, IEEE, 2010.

[11] J. Ng, D. Joshi, and S. M. Banik, "Applying data mining techniques to intrusion detection," in *Information Technology-New Generations (ITNG), 2015 12th International Conference on*, pp. 800–801, IEEE, 2015.

[12] M. Zhu and Z. Huang, "Intrusion detection system based on data mining for host log,"

[13] H.-P. Huang and C.-M. Chang, "An active network-based intrusion detection and

response systems," in *Networking, Sensing and Control, 2004 IEEE International Conference on*, vol. 2, pp. 1317–1322, IEEE, 2004.

[14] A. Sultana and M. Jabbar, "Intelligent network intrusion detection system using data mining techniques," in *Applied and Theoretical Computing and Communication Technology (iCATccT), 2016 2nd International Conference on*, pp. 329–333, IEEE, 2016.

[15] Y. Zhao, "Network intrusion detection system model based on data mining," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2016 17th IEEE/ACIS International Conference on*, pp. 155–160, IEEE, 2016.

[16] P. Vij and A. Kumar, "Effect of rumor propagation on stock market dynamics using cellular automata," in *Inventive Computation Technologies (ICICT), International Conference on*, vol. 3, pp. 1–8, IEEE, 2016.

[17] M. Becchi and S. Cadambi, "Memory-efficient regular expression search using state merging," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 1064–1072, IEEE, 2007.

[18] J. Zhang, D. Zhang, and K. Huang, "A regular expression matching algorithm using transition merging," in *Dependable Computing, 2009. PRDC'09. 15th IEEE Pacific Rim International Symposium on*, pp. 242–246, IEEE, 2009.

[19] G. Tripp, "Regular expression matching with input compression: a hardware design for use within network intrusion detection systems," *Journal in Computer Virology*, vol. 3, no. 2, pp. 125–134, 2007.

[20] Y. Xu, J. Jiang, R. Wei, Y. Song, and H. J. Chao, "Tfa: A tunable finite automaton for pattern matching in network intrusion detection systems.," *IEEE journal on selected areas in communications*, vol. 32, no. 10, pp. 1810–1821, 2014.

[21] T. D. Garvey and T. F. Lunt, "Model-based intrusion detection," in *Proceedings of the 14th national computer security conference*, vol. 17, 1991.

[22] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for scada networks," in *Proceedings of the SCADA security scientific symposium*, vol. 46, pp. 1–12, Citeseer, 2007.

[23] V. Singh and S. Puthran, "Intrusion detection system using data mining a review," in *Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), 2016 International Conference on*, pp. 587–592, IEEE, 2016.

[24] N. Shahadat, I. Hossain, A. Rohman, and N. Matin, "Experimental analysis of data mining application for intrusion detection with feature reduction," in *Electrical, Computer and Communication Engineering (ECCE), International Conference on*, pp. 209–216, IEEE, 2017.

[25] K. Goeschel, "Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive

bayes for off-line analysis," in *SoutheastCon, 2016*, pp. 1–6, IEEE, 2016.

[26] D. Gupta, S. Singhal, S. Malik, and A. Singh, "Network intrusion detection system using various data mining techniques," in *Research Advances in Integrated Navigation Systems (RAINS), International Conference on*, pp. 1–6, IEEE, 2016.

[27] V. Singh, S. Puthran, and A. Tiwari, "Intrusion detection using data mining with correlation," in *Convergence in Technology (I2CT), 2017 2nd International Conference for*, pp. 620–625, IEEE, 2017.

[28] N. Z. University of Waikato, Hamilton, "Weka, data mining software in java." `https://www.cs.waikato.ac.nz/ml/weka/`, Last Access, 11 June 2018.

[29] S. Paliwal and R. Gupta, "Denial-of-service, probing & remote to user (r2l) attack detection using genetic algorithm," *International Journal of Computer Applications*, vol. 60, no. 19, pp. 57–62, 2012.

[30] T. C. Sharma and M. Jain, "Weka approach for comparative study of classification algorithm," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 4, pp. 1925–1931, 2013.

[31] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[32] X. Li and H. Zhao, "Weighted random subspace method for high dimensional data classification," *Statistics and its Interface*, vol. 2, no. 2, p. 153, 2009.