ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A data mining approach for Intrusion Detection System Using Wrapper Based Feature Selection Method

MARTHA TEFFERA

JUNE, 2014

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A Data Mining Approach for Intrusion Detection System Using Wrapper Based Feature Selection Method

A Thesis submitted to the school of graduate studies of Addis Ababa University in partial fulfilment of the requirements for the degree of Master of Science in Information Science

By

MARTHA TEFFERA

JUNE, 2014

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A Data Mining Approach for Intrusion Detection System Using Wrapper Based Feature Selection Method

By

Martha Teffera

Name and Signature of Members of the Examining Board

Name	Signature	Date
_____ Chair Person	_____	_____
_____ Advisor	_____	_____
_____ Examiner	_____	_____

Declaration

I declare that the thesis is my original work and has not been presented for a degree in any other university

_____

Date

The thesis has been submitted for examination with my approval as university advisor.

_____

Advisor

Dedication

I would like to dedicate this thesis to my father late Teffera G/hiwot who gave everything he had to see his children succeed in their education.

Acknowledgement

I would like to give my heartfelt gratitude to the source of knowledge and wisdom, the Almighty God, who has always been with me and guided me in all paths of my life and in the works of this thesis.

I would like to thank my advisor, Ato Workshet Lamenew for being very supportive and encouraging and for the invaluable comments.

I would also like to thank my friends and class mates who were with me in the processes of the preparation of this thesis. Thank you for being there and for encouraging me to pass through the difficult times.

Abstract

Data mining techniques can be used for network intrusion detection systems. Network traffic data are usually large in number of instances and in number of features. Different techniques are available in reducing large data set size. Applying data mining techniques on a large data set may result in wrong output and may also imply computational and time cost. Feature reduction by reducing redundant and irrelevant feature may be a good approach in finding optimal data for applying classification algorithms and finding out accurate result. The main challenge in building intrusion detection system is building a system which can identify newly introduced attack types which were not included in training set. Most intrusion detection systems are developed meant to identify already trained intrusion or attack types. The objective of this research is to explore the possibility of developing a predictive model for intrusion detection using efficient wrapper based feature selection technique. The data used for the research is NSL KDD data set which is a network traffic data with manually injected network intrusion attempts. In this research a wrapper based feature selection approach is used to identify an optimal subset of features from NSL-KDD data set.

After applying wrapper based feature selection and using the induction algorithm to analyze the KDD data set a promising result has been obtained in classifying the different attack types in the NSL- KDD data set. The test set which has new attack types that were not included in the training data set seem to be effectively classified using the classification models built. Using the predictive model built, the attack types that were correctly identified were 95.16%. Which was a better result compared to the same algorithm being applied on data set on which filter based feature selection is used.

Since wrapper based feature selection uses classification algorithms in evaluating the relevance and optimality of a feature the time complexity is significant. Finding techniques for reducing this problem is important by using techniques like reducing stopping criterions or by experimenting different combination of searching and evaluating algorithms.

Table of Content

List of Figures

List of Tables

List of Acronyms

ARFF	Attribute Relation File Format
CFS	Correlation based feature selection
CSV	Comma Separated Values
DARPA'98	Defense Advanced Research Projects Agency
DoS	Denial of Service
FN	False Negative
FP	False Positive
GA	Genetic algorithm
GA	Genetic Algorithm
IDS	Intrusion Detection System
IP	Internet Protocol
KDD	Knowledge Discovery in Database
MS Excel	Microsoft Excel
NIDS	Network Intrusion Detection System
NSL	Network Simulation Language
NSL-KDD	Network Simulation Language- Knowledge Discovery in Database
R2L	Remote to User attacks
ROC Area	Receiver Operator Characteristic Curve
TCP	Transmission control Protocol

TN	True Negative
TP	True Positive
U2R	User to Root Attacks
WEKA	Waikato Environment for Knowledge Analysis

Chapter One

Background

1.1 Intrusion Detection

Intrusion detection method is one of the new techniques currently used to detect malicious intrusion attempts and identify them to protect a network system. There are different techniques used to develop an intrusion detection system. Some intrusion detection systems use signature based techniques to identify normal traffic from malicious one. Others use data mining techniques to filter out and classify the normal traffic from the malicious one (Tigabu, 2012). Data mining techniques applied on data captured from network traffic may help to classify good traffic from bad ones.

Signature based techniques is one of the techniques that was widely used to protect a network system from attacks. In this technique only attacks that are known before and which has a corresponding signature in the database will be detected and any new attack type has to be added manually for the attack type to be detected in the future. This practice has limitations in the area of network intrusion detection as new attack types need to be introduced and hence updating the database with the attack types is essential for the IDS system to be effective. This limitation has led to the need of a means that can detect intrusions automatically without the need to enter known attack types into the system. And this has brought a wide interest in using data mining techniques in intrusion detection system (O and S, 2008).

Many researches were carried out with success on the development of Network intrusion detection system using data mining techniques. And few others are done with the integration of knowledge based system with machine learning techniques to develop a system that successfully identify intrusion attacks. However network traffic data are naturally bulky in size and dimension. This has been a challenge for the efficiency and effectiveness of data mining algorithms in successfully classifying the data set. For this purpose many research have been done to get optimal feature subset from large data set. The purpose of this research has been to attempt identifying relevant and non-redundant

feature set from a network data set using wrapper based feature subset selection. The wrapper based feature selection method has been experimented with different induction algorithms to compare their performance.

1.2 Statement of the problem

Network security is vital in protecting the usability, reliability, integrity and safety of one's network and data. There are different attacks that put the integrity and safety of a network at jeopardy. These threats can include denial of service, data interception and theft, identity theft, hackers attack and others similar network threats.

Considering the intensity of threats that exist currently, network security systems are applied in different layers using different technologies in both hardware and software formats. Some of these network security components are antivirus and antispyware, firewalls that can block unauthorized access to one's network, and virtual private systems that help in identifying malicious attacks.

Network intrusion detection systems have become vital components in network security implementation. The problem is that current network intrusion detection systems are capable of detecting known network attacks. And other attempts to develop network intrusion detection systems using other different means have resulted in too much false positive output which made the efforts fruitless. Network intrusion detection is an application that assists in detecting unauthorized intrusions to network systems. These intrusions might origin from different sources like malicious applications or from different entities with various intentions. Some of the function a Network intrusion detection system perform are recording information related to network access events, notifying security administrators regarding potential threats and producing reports (Terry ,2004).

Different researchers did various works regarding network intrusion detection systems and locally three researchers did valuable researches that have become the ground for this research. The first researcher (Zewdie, 2011) attempted to develop a machine learning intrusion detection system that investigates the application of cost sensitive learning using data mining approach to network intrusion detection that considered cost

sensitive learning techniques by testing decision tree algorithm on labeled records. Another research by (Adamu, 2010) attempted to develop a predictive model for network intrusion detection using information gain value for feature selection that used indirect cost sensitive feature selection approach using decision tree as classification techniques. And (Tigabu, 2012) attempted to model intrusion detection system using semi supervised approach by considering unlabeled and labeled records and found out that it is promising to identify those network intrusions either normal or attacks and put forward tangible mechanisms for detecting and presenting them using the appropriate data mining approaches.

There are two kinds of intrusion detection systems these are misuse detection and anomalous detection. In misuse detection a set of rules will be identified and entered in the IDS system based on known intrusion types. In anomalous IDS systems the system itself will attempt to identify attack types by first training the system using training data set based on which a model will be built.

Data mining techniques play a great role in developing anomalous intrusion detection systems for a better prediction of intrusion types. Network traffic data from which knowledge extraction and classification models will be developed are usually very large in dimension and in quantity. It won't be practical to use the total amount of data and feed it in a knowledge extraction data mining algorithm and try to develop a prediction model out of it as it will consume too much system resource. To tackle this problem there are various techniques that are being applied to reduce the data size and dimension. A data may be said large if it contains a large number of instances of data or a large number of attributes or both (Liu and Steiono, 1998).

It is almost always true to say that using all the content of a large data set both the instances and the features may not be important in extracting a useful knowledge from the data. It may even affect the performance of a good and effective algorithm because a large data set may contain irrelevant and redundant data and features.

Different researchers have used different techniques to reduce the instances of a large data set. And in the same way there are various other techniques to reduce the

dimensionality of a data set by selecting a subset of features using feature importance evaluation techniques. These feature subset selection approaches fall in one of the following three approaches; these are filter, wrapper and hybrid approaches.

Filter approaches use some statistical techniques to evaluate the importance of the feature subset. Some of the techniques can be correlation measurement, information gain ratio techniques and other similar statistical techniques can be used to evaluate the feature's relevance. In wrapper based approach an induction algorithm is used in the process of selecting the feature subset which will be used in developing a predictive model. The relevance of a feature will be evaluated on how successfully the feature can classify the data using the induction algorithm that is used. Wrapper based approach are said to be resource consuming but is more effective than any filter method in feature subset selection. Finding out effective feature subset has a paramount importance in deriving a successful predictive model. In order to reduce the cost of a wrapper based feature selection algorithm finding out less resource consuming and effective feature subset evaluator, search method and induction algorithm which will be used as evaluator of the classification power of a feature is critical.

Therefore in this research by considering the relevance of features in identifying targeted results and also the necessity of suitable features for certain algorithms in finding accurate results, wrapper based feature selection method has been used. By using wrapper based feature selection, the problems in misuse intrusion detection systems and the issues that arise in using filter based feature selection methods have been tried to be addressed.

In this research an effective wrapper based feature selection methods which will help in identifying relevant features data set was identified and used to select optimal features from the NSL_KDD data set. The wrapper based feature approach was evaluated using different induction algorithms which were used for classification of the data set with the selected features using each induction algorithm. The classification accuracy of each induction algorithm and the prediction model were evaluated in different performance evaluation metrics. By using wrapper based feature selection method, it is believed to build a better prediction model which will result in a better accuracy in intrusion detection. And

reduce the limitations of filter based feature selection method which doesn't consider the relevance of features for prediction algorithms that will be applied on data sets and which in turn causes less prediction accuracy.

1.3 Scope

The Scope of this research is limited to applying wrapper based feature selection method on NSL-KDD data set and build predictive model. The data used for the research is a publicly available network traffic data which is used for experiment purposes. The experiments done on this research are not applied on real time network traffic data. The algorithms used to build prediction models and evaluate the optimality of the features selected are only two since other algorithms are found to be non efficient in the feature selection process.

1.4 Objective(s)

The general objective of this research is to explore the possibility of developing a predictive model for intrusion detection using wrapper based feature selection technique.

1.4.1 Specific objectives

In line with the general objective of this research, the following will be specific objectives

i) Reviewing literatures written in the area of intrusion detection system using data mining techniques and the approaches used in the reduction of large data set and features. This will help in understanding the problem domain and to come up with an effective solution in optimal feature selection.

ii) To select a network traffic data set and understand the content of the data and apply preprocessing technique to make it ready for the experiments to be done.

iii) To compare different algorithms and feature selection methods using wrapper based feature selection method and select the effective ones in selecting optimal subset of features from the data set using different induction algorithms.

iv) Apply the induction algorithms used in the feature selection process and apply classification on the data with the selected features.

v) On the result obtained evaluate the accuracy by applying it on a test data set and analyze the result.

vi) Compare the performance of the models built with other recent similar works and compare the performance of each induction algorithms with each other.

vii) Report the methods used and the result achieved and draw conclusions and recommendations.

1.5 Significance of the Research

The goal of intrusion detection is to identify, preferably in real time, unauthorized use, misuse and abuse of computer systems by both insiders and external penetrators. The intrusion detection problem is becoming a challenging task due to the production of heterogeneous computer networks since the increased connectivity of computer systems gives greater access to outsiders and makes it easier for intruders to avoid identification (Mukherjee, haberlein & Levitt, 1994).

So far wrapper based feature selection method has not been used to select optimal features from the NSL-KDD data set from which intrusion instances would be identified. Thus in this research wrapper based feature selection methods were used to identify optimal features from a large data set of network traffic. And hence by selecting a relevant and non redundant feature type a reliable predictive model will be built in order to reduce the high rate of false positive result obtained by many other researches made using different techniques of feature selection and classifying algorithms.

1.6 Methodology

In order to conduct a relevant research and provide a significant and useful output the researcher did extensive literature review to get a thorough understanding of the subject matter and to identify the right methods to achieve the best result.

In order to experiment the feature selection methods proposed in this research, the NSL-KDD data set is selected. The data set is a network traffic data set with normal and attack traffic data and it is divided in training and test sets. The training and testing set have 125,973 and 22,544 instances respectively with 41 features. The data set is labeled with normal class and 38 different attack types. The attack types are categorized into four different attack classes which are Denial of Service, User to root attack, Remote to local and probing (Chuang, Yang and C, 2009). The testing data set has new attack types which are not included in the training set and this was taken as a good opportunity to evaluate the performance of the predictive model to be developed in classifying new attack types.

To extract a meaningful information or knowledge from a network traffic data, data mining technique was applied and for this purpose WEKA tool was used. Data mining is the process of extracting knowledge and patterns in large data sets. The patterns that will be discovered help in making predictions on new data. This makes data mining an important tool in discovering patterns from network traffic data and help predicting new attack types from new data sets. There are various applications which help in mining patterns from data and one of these tools is WEKA (Frank, 2005). WEKA is open source data mining software which contains a collection of machine learning algorithms for data mining tasks. The algorithms and tools that are included in WEKA help in discovering patterns and knowledge from a data set. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules and visualization.

One of the processes that help in extracting accurate and relevant patterns from a data set is feature reduction and feature selection. The NSL-KDD data set which has 41 features in total need a feature selection process to be applied on it in order to remove redundant and irrelevant features. WEKA has various methods of attribute selection which helps in segmenting subset feature. WEKA's supervised attribute selection methods, enable a combination of evaluation and search methods to be specified, where the objective is to determine a useful subset of attributes as input to learning scheme (Remco, 2008).

WEKA also helps to apply algorithms on a data set to perform all processes of experimental data mining including preparing the input data, evaluating learning schemes

statistically and visualizing the input data and the result of the learning process. The diverse algorithms can be accessed through a common interface which helps the user to compare different methods and identify the most appropriate ones easily (Frank, 2005).

As mentioned earlier, in order to get a useful and accurate prediction model which help in predicting patterns from new dataset, having optimal and relevant features is mandatory. And hence effective wrapper based feature selection algorithms and methods were used in order to select an optimal subset of features from the NSL-KDD dataset. For this process the algorithms used were genetic algorithm, Bayesnet, Naïve Bayes, K-means and Decision tree. These algorithms were used in the feature selection through the wrapper method and in the prediction process. After applying those algorithms the performance of the techniques was compared with the accuracy of those algorithms on features selected using filter feature selection methods.

The evaluation method used was confusion matrix which shows the accuracy of classification algorithms. And from the confusion matrix the accuracy, precision, recall, true and false positive rates are derived. In order to compare the performance of the selected classification algorithms their accuracy is compared with other algorithms both used on wrapper feature selection methods and filter feature selection methods.

1.7 Organization of the Thesis

This thesis contains five chapters. The first chapter discusses background of the study area, statement of the problem, objective and scope of the study.

The second chapter is dedicated for literature review and narrates about network security, how network systems can be intruded by different attack types and intrusion prevention methods. It also briefly highlights the various data mining techniques. Moreover, it explains about feature selection approaches and algorithms and finally discusses the data set used in the experiment.

The third chapter deals about data preparation, feature selection methods and tools used for the experiment. It also discusses about the evaluation metrics used to analyze results.

Chapter four covers experiment setup, the different algorithms and the steps used to undergo the experiment and also discusses the result obtained. And finally, chapter five deals about conclusion and recommendation of the study..

Chapter Two

2. Literature Review

2.1 Network Security

Keeping a network system to be secured involves guarding the confidentiality, integrity and availability of the system. Confidentiality is making information to be available only to authorised users and entities. Information may be forcefully disclosed intentionally by deciphering a security protection or by incompetency of users who handle the information carelessly.

Integrity of a network system involves three areas these are preventing modification of information by unauthorized users, preventing modification of information by authorized users but unintentionally or preventing authorized users from unauthorized modification and preservation the consistency of internal and external information. Availability is making information and a network system available and accessible to all authorized users.

Network security attacks can be categorized into different groups depending on the characteristics of the attacks.

- Passive: Passive attacks include those attacks which analyse network traffics, monitors unprotected communications and decrypts traffics which are weakly encrypted in the aim to capture passwords and other authentication information.
- Active: Active attacks work to break and intrude to a network protection system and spread malicious code or steal or modify information.
- Close-in: These attacks usually involve individuals who have the privilege of being close to the network system that is the target of the attack.
- Insider: These kind of attacks can be intentional or non intentional. Intentional insider attacks may steal or damage information and use information in a fraudulent manner or deny access to other authorized users.

- Distribution: Distribution attacks focus on the modification of hardware or software during distribution of the item. These attacks can introduce malicious code into a product so that to gain unauthorized access to information or a system function in future. (Eric and R, 2005)

2.2 Types of Intrusion

The different kinds of attacks which act for different purposes including economic gain, maliciousness and fraud can be said they are all against the confidentiality, integrity and availability of a network system. Some of specific types of attacks are listed below.

- Normal connection (Normal): are those network traffics which act to be a normal traffic with common behaviours and doing legible actions.

- Denial of Service (DoS) : This kind of attacks make a network resource too busy by sending too much traffic that exceeds the limit if the ability for the network resource to attend properly.

- Spoofing- A spoofing attacker captures an IP address from a network system and sends packets giving a known IP address to the network system so that the target host assumes the source of the packet is an internal client (Eric and R. , 2005).

- Port Scanning- Using scanning software, an intruder aiming to attack a system can scan a network system and determine which hosts are active and which are down (Eric and R. , 2005).

- User to Root (U2R) - is a kind of attack in which an intruder gains the privilege of an administrator or root user while the user actually has a normal limited user privileges.

- Probing (probe) - is a kind of attack that scans a network system in hopes that to find vulnerabilities and holes that it can use the weak part of the network to send attacks.

2.3 Network Intrusion Preventions Methods

To protect a network system from such groups of attacks and intrusions, the following and more other different techniques are being used. Defence in multiple places which is installingg information protection mechanisms on various locations of the network system. Layered defences is Implementing multiple information protection and detection means to prevent an attacker from having easy access to a network. Security robustness is measured in terms of assurance and strength of the information assurance component. Deploy intrusion detection system is implementing intrusion detection technique to detect intrusions and prevent system.

2.4 Data mining

The large amount of data collected from various sources needed to be stored in large databases and data warehouses. Using the collection of data for decision making and devising a marketing strategy were not possible due to lack of powerful data analysis tools which help in extracting useful knowledge from the collection of data. In addition to that the vast collection of data has become beyond the capacity of the data repositories available. Data warehouse technology which includes data cleaning, data integration analysis techniques with functionalities such as summarization, consolidation and aggregation as well as the ability to view information from different angles lead to the concept of data mining. Data warehouse is a non volatile, time-variant, subject oriented relational database management system responsible for the collection and storage of data to support management decision making and problem solving.

Data mining is a means of 'mining' knowledge from large amounts of data and supports associations, construction of analytical models, performing classification and prediction and presenting the mining results using visualization tools. Data mining techniques can be applied on different kinds of data formats including relational databases, data warehouses, transactional databases flat files, data streams and others (Kamber and J, 2011).

Knowledge discovery from database is the process of finding useful information and knowledge from a dataset. Data mining techniques and algorithms are used in Knowledge

discovery in databases (KDD) which helps in extracting useful information from data sets (Julish,2006). KDD process include the processes of selection of data for the appropriate knowledge domain, pre-processing of the data set, transformation of the dataset by applying data reduction and data categorization to ease data mining , Mining the data and interpretation or evaluation which presents results to user in meaningful manner.

2.4.1 Pre-processing

Pre-processing data set before applying data mining techniques to discover knowledge is vital. Large data collections almost always contain incomplete, noisy and inconsistent data. The existence of these types of data in a dataset makes the knowledge to be discovered unreliable and sometimes may give wrong information. The major tasks in data pre-processing are data cleaning, data integration, data transformation, data reduction and data discretion (Kamber and J, 2011).

Data cleaning is the process of filling missed values in a data set, removing outliers, removing redundant data and smoothing noisy data in a data set to keep the integrity and consistency of the data. The use of data summarization techniques and the measures of central tendency in a data, help in identifying outliers and noisy data from a data set. Measures of central tendency like mean, median and mode help in identifying the values of normal data in the dataset and greatly help in identifying the outliers and noisy data values. (Kamber and J, 2011).

Data integration involves the process of integrating data from various sources like databases, data warehouses or files whenever analyzing data from different sources is necessary. Data reduction is applied when there is a very large amount of data at hand and when applying data mining techniques. Techniques that can be used to apply data reduction are, attribute subset selection, numerosity or size reduction and data compression. Data can also be reduced by generalization with the use of concept hierarchies. A concept hierarchy uses the technique of information abstraction which hides detailed information by replacing the information with a general idea (Kamber and J, 2011).

In data mining there are also different techniques that help in identifying patterns. Some of these techniques are association rules, classification and clustering. Association rules is a process in which common behaviours between data features and data sets will be recognized. In this technique regular patterns and relations occurring between sets of items in a data set are discovered.

2.4.2 Feature Selection

Data acquisition, feature selection, applying classification and clustering algorithm are steps of knowledge mining process. Selecting the relevant features and minimizing the complexity of a data set and maximizing the performance and accuracy of an algorithm used for knowledge discovery is some of the purposes of feature selection from a data set (Jinsong and C, 2010).

Feature selection process includes identifying relevant attributes from a dataset which has a large dimensionality. In feature selection, selected attributes may be important enough to be used for experimental purposes like classification and clustering. one of the reason for applying feature selection process on a large data set is some classification algorithm's performance degrade with the presence of irrelevant and redundant features in the data set to be classified. In the process of a feature selection there is no feature extraction or construction done it is only the most important subset of features will be selected for an optimal accuracy of a classification algorithm (Ron and G, 1996). Trying to discover knowledge from a data set which has a large number of attributes is resource taking and sometimes even impossible (Jian and S, 2004). Usually the large number of features a dataset contains may be redundant and irrelevant and hence selecting those features which are highly discriminating is vital. According to (Jian and S, 2004) removing those features which are redundant features is as important as removing irrelevant ones because they both reduce the accuracy of knowledge discovery and take high resource for learning algorithms. Redundant features are features which are highly correlated to each other (Hall, 1999).

Those features which highly classify the data set and give an optimal subset of the data set are called strongly relevant features. Weakly relevant features are those features which don't have strong discriminating power and which are not always necessary for optimal subset. And the irrelevant features are those features which don't have any discriminating power and which are not necessary at all. According to some scholars an optimal data set has to contain all strongly relevant features and few weakly relevant features. Some of redundant features may be categorised in the weakly relevant features group and hence finding out which features are redundant and which are not is a necessary task in feature selection (Hall, 1999).

Predictive models which are based on highly relevant features have a high chance of having classification accuracy but when those features which have high predictive power is eliminated from the data set the accuracy of the model will be degraded (Hall, 1999).

In a large data set feature selection can be handled in one of the two approaches. These approaches are individual evaluation and subset evaluation. Individual feature selection inspects each feature and assigns them weights based on their relevance. And this approach takes a linear time complexity and seems to be the best approach however since redundant features have equal relevance weigh they may not be identified as redundant. And this approach is not good for high dimensional data with a large number of redundant features. Subset evaluation approach handles feature selection by selecting a subset of features and compares the relevance with previously selected features and chooses the best one. If a new subset appears to be better, it replaces the previous best subset. The process of subset generation and evaluation is repeated until a given stopping criterion is satisfied and this approach takes into account the existence and effect of redundant features. Many methods which are used for subset evaluation proved to be effective in removing irrelevant as well as redundant features (Liu, 2004).The searching methods used for this approach may take a quadratic time complexity and this prevents them from scaling well to data sets containing tens of thousands of features.

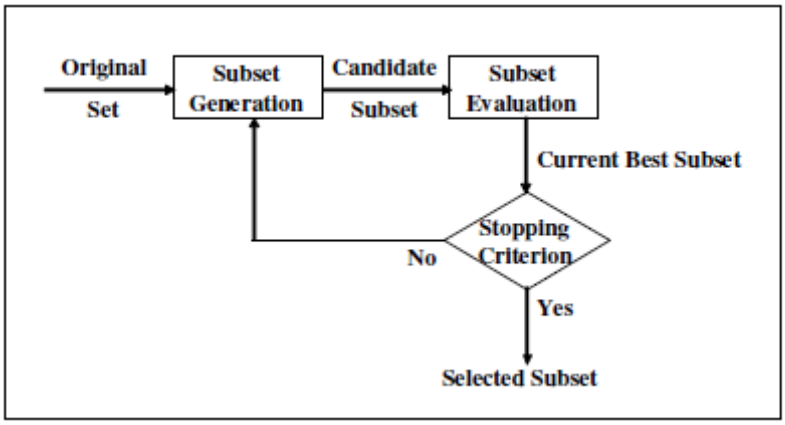


Figure 1 Traditional feature selection approach (Liu, 2004)

To find an optimal subset of features, a better approach can be used by using two steps which are first by using relevance analysis and in which the relevant features selected. And second redundancy analysis determines and removes redundant feature from relevant ones and produce the final subset. The advantage over the traditional approach is that it produces by individually handling the relevance and redundancy analysis, the approach allows both efficient and effective way in finding a subset that approximates an optimal subset.

Because of these problems and others a subset of features selected will be said optimal feature subset for a particular induction algorithm therefore the process of finding out a subset feature can be said it is the process of finding the optimal feature for a specific algorithm. For an inducer I and a dataset D with features of X1,X2...Xn over a labelled data set an optimal feature subset Xopt is a subset of the features such that the accuracy of the induced classifier C= I(D) is maximal (Ron and G., 1996).

In some cases a classifying algorithm may give highest accuracy of classification with a subset feature which may or may not contain all relevant features. Relevance of a feature does not imply that it should be in the optimal feature subset and in the same way if a feature is irrelevant it does not mean it shouldn't be in the optimal feature subset. Rather some classifying algorithms may perform better with the absence of the most relevant features (JInsong and C, 2010).

Based on the method used for computing importance of features in a data set, feature selection algorithms are classified into filter approach, wrapper approach and hybrid approach (P and C, 2002).

2.4.2.1 Filter feature selection

The filter approach computes the feature evaluation weight but without performing classification of data, eventually finding the good subset of features. Mostly the measuring criteria's for optimality of a feature are statistics based in nature. The principle of filter approaches is to select the subset of features which have high dependency on the class and while have less correlation among them. One group of filter methods is to measure the importance by maximizing the clustering performance, Other approaches are to find redundant or irrelevant feature to be removed that carries little or additional information using statistics measures (King, 1997). Sequential forward search, sequential floating forward search, stepwise clustering, feature selection through clustering and others have been used for selecting best features from a data set.

The evaluation measures that will be used for filtering approach of feature selection are:-

1. Distance Measure

   In this evaluation measuring technique a feature is said to be better than another feature when the first feature can induce greater difference between two classes based on conditional probabilities than the second one. If the difference the two features make is zero then it will be impossible to differentiate. Euclidean distance is an example of this evaluation measuring approach.

2. Information Measure

   A feature is evaluated based on the information that will be gained from the feature. A feature is selected based on the comparative result of the amount of information that would be gained from features.

3. Dependence Measure

   Dependence Measure of correlation measure attempts to measure the correlation value between features and feature and class. If the correlation value of a feature is higher another feature with a class then the first feature will be selected as a good feature for showing dependency of the feature and the class. Using the correlation measure it is possible to identify redundant features from a dataset.

4. Consistency Measure

   This type of evaluation measure is characteristically different from other measures because of their heavy reliance on the training dataset and use of Min-Features bias in selecting a subset of features. Min-Features bias prefers consistent hypotheses definable over as few features as possible. These measures find out the minimal size subset that satisfies the acceptable inconsistency rate that is usually set by the user.

2.4.2.2 Wrapper feature selection

In wrapper approach feature subset selection is done using an induction algorithm which undertakes a search for a good subset and the algorithm itself will be part of the evaluation function. In this approach a search is undertaken in the collection of attributes and a search requires a state, an initial state, a termination condition, heuristic or evaluation and a search algorithm. The goal of the search is to find the state with the highest evaluation, using a heuristic function to guide it (Ron & R, 1996).

A wrapper featuer selection method can be seen as the combination of a search technique for proposing new feature subsets along with an evaluation measure which scores the different featuer subsets and an induction algorithm for evaluating the optimality of the feature subset. According to (Sklansky., 2000) the wrapper based approaches can perform better than filter based approaches. A combination of some filters and inductive classification algorithm approaches has also been proposed to resolve the short comings of filter and wrapper methods (P and B, 2010). The wrapper approaches of feature selection aim to find the minimum discriminative features to reach the high

classification accuracy, while the filter approaches are to compute the best subset of features in terms of some criteria (JInsong and C., 2010).

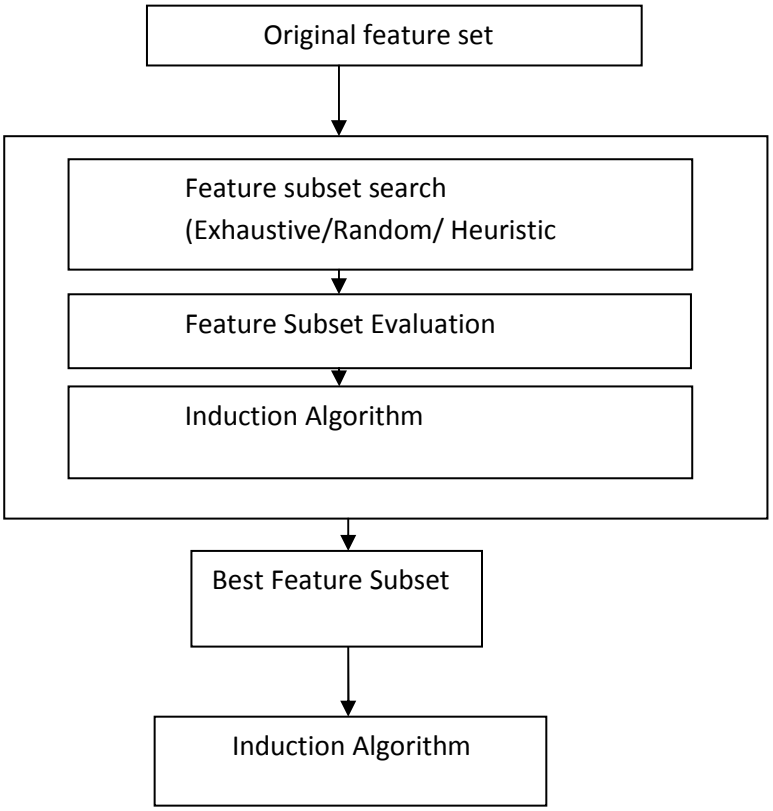


Figure 2 Wrapper based feature selection steps (Asha, Jayaram, & Manjunath, 2010)

2.4.2.3 Hybrid Feature selection

Hybrid feature selection method uses a combination of the techniques of wrapper and filter approaches to take advantages of both approaches.

2.4.3 Search Technique

Most methods for attribute selection involve searching the space of attributes for the subset that is most likely to predict the class best. The goal of the search is to find the state with the highest evaluation, using a heuristic function to guide it. There are various techniques of searching methods that are used in wrapper method some of the methods are Best first search, Exhaustive serach, Genetic Search method, Greedy stepwise search and others.

- Best first search is a method in which an attribute is selected if it gurantees a best classification accuracy.
- In Greedy search a search is taken place in one of two directions either top to bottom or bottom to top. At each stage, a local change will be made to the current attribute subset by either adding or deleting a single attribute.
- Genetic Search selects the best feature subset based on the random sampling strategy. Genetic algorithm generates the sample population with certain number of features the optimality of the features will be evaluated with a certain classification algorithm.

  The basic processes GAs undergo to search for the best feature subset are:-
  - Population: it generates n number of subsets of features randomly.
  - Fitness: It evaluates the related fitness of each feature.
  - Iteration: It repeats the processes until a predefined number N of feature subsets found
    - Selection: Chooses two features for crossover. This method selects features which will be reproducing. The method used to evaluate the fitness of the feature to identify the fitter feature which will be selected to reproduce (Liu and Steiono, 1998).
    - Crossover: Form a new feature subset based on the crossover strategy. The crossover operator performs recombination, creating two new offspring by randomly selecting a locus and exchanging subsequences to the left and right of that locus between two chromosomes chosen during selection (Liu and Steiono, 1998).
    - Mutation: Mute the new features by the mutation probability. The mutation operator randomly changes the bits or digits at a particular locus in a chromosome: usually, however, with very small probability (Liu and Steiono, 1998).
    - Fitness: Compute the fitness of the muted feature.
    - Update: Replace the muted feature in the population.

- Evaluation: If fitness is satisfied keeps the feature selected and generate a new subset and compute the fitness of each feature.

o Return: Find N feature subset.

2.4.4 Association Rules

Association rule or pattern discovery tries to identify hidden relationships between data items. Naive algorithm, apriori algorithm and frequent pattern growth approach are some of the algorithms that are used to identify association rules from a data set (Kamber and J, 2011).

2.4.5 Classification

Is another technique of data mining, assigns labelled dataset records to predefined classes and used in supervised learning. The challenge in this technique is identifying the classes as they may need to be derived from attributes of the dataset (Julish and K, 2011).

Some of the pre-processing techniques are vital before applying classification technique on a data set in order to obtain a relevant result. Classification can be done using different techniques some of these techniques are classification by decision tree, Bayesian classification, rule based classification and classification by association rule analysis (Kamber and J, 2011)

- K- Nearest Neighbors

KNN is a supervised learning algorithm the purpose of this algorithm is to classify a new object based on attributes and training samples. KNN algorithm uses neighborhood classification as the prediction value of the new query instance. It works based on minimum distance from the new instance to the training samples to determine the K-nearest neighbors. To calculate the distance between the new instance and all the training samples Euclidean distance can be used

$$Dist(X,Y) = \sqrt{\sum_{i=1}^{D}(Xi - Yi)2}$$

Using the training samples using the Euclidean distance the k^{th} nearest neighbours can be determined and so does the class of the nearest neighbours.

- Decision tree classifier

Decision tree is a widely used learning method. It performs classification by constructing a tree based on training instances with leaves having class labels. The tree is traversed for each test instance to find a leaf and the class of the leaf is the predicted class. This is a directed knowledge discovery in the sense that there is a specific field whose value will be predicted. The steps in creating a decision tree classification model is select optimal attributes on the basis of a heuristic or statistical measure like information gain then partition training samples based on selected attributes. The Conditions for stopping partitioning are when all samples for a given node belong to the same class and there are no samples left, when there are no remaining attributes for further partitioning majority voting is employed for classifying the leaf.

- Bayesian Network

Bayesian networks can readily handle incomplete data sets in situations where two of the explanatory or input variables are strongly anti-correlated. Such cases may not be an issue for standard supervised learning techniques, provided all inputs are measured in every case. When one of the inputs is not observed, however, many models will produce an inaccurate prediction, because they do not encode the correlation between input variables. Bayesian networks offer a natural way to encode such dependencies. Bayesian networks also allow one to learn about causal relationships between data instances or features. Learning about causal relationships is important for at least two reasons. In situations where cause and effect are needed to be studied Bayesian network becomes important.

A BN is a graphical structure that allows us to represent and reason about an unknown domain. In many practical settings the BN is unknown and one needs to learn it from the data (Faltin, Kenett, and Ruggeri, 2007). The nodes in a Bayesian network represent a set of random variables, X= X1,…Xi,…Xn from the domain. In Bayesian network

the process of conditioning or probability propagation is performed using flow of information through the network.

- Naive Bayes

A naive bayes classifier assumes that the presence or absence of a particular feature of a class is unrelated to the presence or absence of any other feature. Depending on the precise nature of the probability model, naive bayes classifiers can be trained very efficiently in a supervised learning setting. The naive Bayes classifier requires a small amount of training data to estimate the parameters like means and variances of the variables necessary for classification because independent variables are assumed only the variances of the variables for each class need to be determined and not the entire covariance matrix. The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible.

For simplifying the naive Bayes model it can be written as follows.

$$Posterior = \frac{prior\ x\ liklihood}{evidence}$$

And the above can be re written using bayes' theorem as follows

$$P(C|F1, \dots, Fn) = \frac{p(C)p(F1, \dots . Fn|C)}{p(F1, \dots, Fn)}$$

The above naive Bayes classifier combines Bayes probability model with a decision rule and a classifier can be made. A common decision rule that can be made from the combination is by taking into consideration the maximum a posterior or MAP decision rule and the following classifier will be defined as follows.

$$classify(f1, \dots, fn) = argmax\ p(C = c) \prod_{i=1}^{n} p(Fi = fi|C = c)$$

Even if the feature independence assumptions are usually not correct, Naive bayes classifier are usually advantageous in using it in large datasets.

2.4.6 Clustering

The third approach in mining data is clustering in this technique data records with similar attributes and values will be categorized in one cluster based on their similarity (Julish and K,2010). Groups of data in a cluster are similar to each other but they are different from the data in different cluster. Since there is no training data based on which clustering of the data records is done, the process is called unsupervised learning. Evaluation of clustering is done by measuring the similarity of data in a cluster and the dissimilarity of the data in different clusters. The various methods of clustering are partitioning methods, hierarchical methods, density- based methods, grid based methods and model-based clustering methods (Julish and K,2010).

Each clustering problem is based on some technique of distance or nearness measurement between data points. Distance measurement help to measure similarity or disimilarity between data instances.

Similarity measurement techniques that usually are used to measure similarity and disimilarity are Minkowski distance if $X=(X_1,X_2,\ldots X_n)$ and $Y=(Y_1,Y_2,\ldots Y_n)$ are two n dimensional data objects and n is size of vector attributes of the data object q=1,2,3…then

$$dis(X,Y) = \sqrt[q]{\sum_{i=1}^{n} (|xi - yi|)^q}$$

If q = 1 dis becomes Manhattan distance

$$dis(X,Y) = \sum_{i=1}^{n} (|xi - yi|)$$

Some of clustering algorithms are K-Means algorithm, hierarchical clustering,

- K Means Clustering

In K means clustering K number of cluster points will be created as initial centroid which are selected randomly. After partitioning objects into K non empty subsets recomputing the centroids or mean points of each K clusters of the partiton will be done then each object in the data set will be assigned to each cluster based on the nearest seed point.

- Hierarchical Clustering

In hierarchical clustering the data are not partitioned into a particular cluster in a single step instead a series of partitions takes place which may run from a single cluster containing all objects to n clusters each containting a single object. In network intrusion data mining techniques, performing the task of removing those activities with no suspicious pattern from the data help in identifying real attacks easily. They also help in identifying those alarms which generate false positive results, finding out in extracting mischievous activities that uncover real attack and also helps in identifying new data sets with different source but with same activity which another source was applying (Eric and A,2005).The quality of data that will be analyzed by data mining techniques greatly affects the quality of the result to be obtained. Clearing redundant and irrelevant data from a data set helps in extracting relevant result from applying data mining technique on the data. If the techniques applied on data set without performing the cleaning the result that will be obtained will be unreliable and it will be waste of resources (Anita and P,2005).

2.5 Data Mining Based Network intrusion Systems

The importance of network intrusion detection system has become more and more significant to keep the security of a network system. One can define intrusion as any actions that may threaten the well being, security and availability of a network resource (Liu, 2004). As the development of network technology increases so does the various ways of attempts to intrude to a network for different purposes. To tackle the problem of being affected by intrusions different techniques of protection are being used. One of the techniques is network intrusion detection systems, which helps to monitor and analyze

traffics to and from a network system to detect security problems, which are being developed based on different technologies.

The systems that are being developed using data mining techniques usually have their own challenges to be effective. These challenges are regarding the accuracy of the system, efficiency and usability of the system and the fact that the systems require large amounts of training data and their degree of complexity is very high when compared with other same purpose systems. The fact that their being complex may make them to be inefficient as they require more computational resources to provide reliable results. And as a result data mining based network intrusion detection systems are offline systems as they require high computational resources to perform their task online (Eric & R, 2005). As work around for the expensive computational need of data mining based network intrusion detection systems, detection rules are forced to base on low cost features which are identified from the total data set in belief that they are more powerful in identifying the data at hand.

In order to evaluate the performance of a data mining based network intrusion detection system, the points that need to be seen are the amount of attack the system detects and the amount of data that are wrongly identified as attack (Eric & R, 2005).

The two general categories of intrusions are anomaly attacks and misuse attacks. The current IDS systems normally target misuse detection and the intrusion detection systems for targeting this attack usually produce a high true positive rate. Academically anomaly intrusions are favoured to do several researches as there are not much intrusion detection systems which successfully detect anomaly–based attacks (Mahobod & E, 2009). Misuse detection identifies intrusions by selecting those activities which are different from the common ones. But for the data to be identified as normal there need to be a labelled training data based on which new data can be classified in one of the categories. The drawback of this method is that if a new data type appears other than the pre-defined ones the system will not be able to identify the data correctly. Supervised systems are those systems which use labelled training data set. In supervised learning labelled training data

set are used and new data is classified according to the training set. While in unsupervised learning, the class labels of the training data is not known rather the existing classification in the data set will be used to categorize the data set at hand.

The approaches used in misuse detection are expert systems, signature analysis, state transition analysis and data mining (Steven and D, 2002). The disadvantages of Misuse detection is that it tends to provide false positive results and a packet may not be examined in context with its precedent or following packets in addition to that it relies on signature based technique.

Whereas in anomaly detection the IDS system defines class to which the normal data will be categorized into. If a data appears to be different than this class will be classified as an attack. This may lead normal data to be categorized as attacks if they somehow are different from the ordinary network packet and hence may result high false positive. However this method helps to identify new attacks which are different from the normal or known attacks. In anomaly detection approach profiles of normal behaviour are automatically discovered (Steven and D, 2002).

It is also possible to classify network intrusion detection systems based on source of data used to apply the IDS system at hand. There are two possible sources of data for the purpose, these are host based and network based data source. In host based detection system, the host data files and operating system processes which will possibly be targets, will be monitored. And therefore identifying the resource which may be target of a particular attack will be possible. For network based intrusion detection, the data source is traffic across the network. And such systems help in detection of the intrusion (Steven and D, 2002).

The third way of classifying IDS activity is a method called degree of attack guilt. This method uses the correctness of the result of the system to identify attack and normal data to distinguish the activity of the system (Steven and D, 2002).

The challenge with the current intrusion detection systems is that the rate of producing false positive is very high however IDS systems have also become the hope in securing one's network system (Anita and P, 2013). Most Intrusion detection systems do not provide online intrusion detection results as the systems are resource consuming and the data to be analyzed are enormous in quantity, as a result the output from the analysis of the data will not be real time (Eric and A, 2005).

Several researches have been done on identifying the right methods and techniques to reduce the false positive results. According to (Choi and H, 2008) there is no one single algorithm that is able to produce low false alarm rate for the different kinds of network attacks that are available today. In order to be able to detect the various kinds of threats that are attacking network integrity and security using the combination of several algorithms is vital. Experiments showed that for certain kinds of attacks a specific algorithm is effective in detection of the attacks.

2.6 The KDD data set

The KDD data set is a data set that contains network traffic data with manually injected network intrusion attempts as well. It is a data set, which was provided by DARPA it was prepared and managed by MIT Lincoln Labs in 1998.The purpose of preparing the data set was for research competitions on the area of intrusion detection systems. And it is the main standardised data set used for any research done on the area of network intrusion detection approaches. The data set is a collection of raw TCP dump data of a LAN simulating a typical U.S Air Force LAN. The raw training data contained around 4GB of compressed binary TCP dump data from seven weeks of network traffic. In the data set the connection is labelled as normal or as an attack, with exactly one specific attack type. A connection is taken as any traffic of a TCP packet starting and ending in certain amount of time limit, and the traffic of data moves from one IP source to the destination IP under a defined protocol. The size of each connection is about 100 bytes. The training data set contains a different attack types in total 24 attack types and the testing dataset conations 34 attack types. The attacks in the data set are in one of the following attack types.

- DOS- Denial of Service- is a kind of attack in which legitimate users of a network system are denied of access to the system.
- R2L- Occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.
- U2R-Is an attack type which tries to have access of resources by getting the passwords and permission to root user. This may be achieved by using sniffing passwords, a dictionary attack or other methods.
- Probing- It is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls (Yong and W, 2004).

According to (Mahobod and E, 2009) KDD'99 features can be categorised into three groups.

1) Basic Features- This feature contain those attributes that can be derived from a TCP/IP connection. And most of these features lead to a delay in detection.
2) Traffic features- This category of features include those features that are obtained from calculating with respect to a window interval and falls into the either of the following two sub groups.
    a. "Same host" features- inspects connections occurred in the previous 2 seconds and with the same destination host as the current connection, and computes figures concerned about protocol behaviour, service, etc.
    b. "Same service" features- inspect the connection in the previous 2 seconds with the same service as the current connection.
3) Content features- Since R2L and U2R attacks are normally in the data portions of packets and involve a single connection, they don't have any intrusion frequent sequential patterns.

feature name	Description	Type
Duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	Discrete
Service	network service on the destination, e.g., http, telnet, etc.	Discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
Flag	normal or error status of the connection	Discrete
Land	1 if connection is from/to the same host/port; 0 otherwise	Discrete
wrong_fragment	number of "wrong" fragments	continuous
Urgent	number of urgent packets	continuous

Table 1 Basic features of individual TCP connections.

feature name	Description	Type
Hot	number of "hot" indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	Discrete
num_compromised	number of "compromised" conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	Discrete
su_attempted	1 if "su root" command attempted; 0 otherwise	Discrete
num_root	number of "root" accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise	Discrete
is_guest_login	1 if the login is a "guest"login; 0 otherwise	Discrete

Table 2 Content features within a connection suggested by domain knowledge

feature name	description	Type
Count	number of connections to the same host as the current connection in the past two seconds	Continuous

Note: The following features refer to these same-host connections.		
serror_rate	% of connections that have "SYN" errors	Continuous
rerror_rate	% of connections that have "REJ" errors	Continuous
same_srv_rate	% of connections to the same service	Continuous
diff_srv_rate	% of connections to different services	Continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	Continuous
Note: The following features refer to these same-service connections.		
srv_serror_rate	% of connections that have "SYN" errors	Continuous
srv_rerror_rate	% of connections that have "REJ" errors	Continuous
srv_diff_host_rate	% of connections to different hosts	Continuous

Table 3 Traffic features computed using a two second time window Critics on the KDD CUP 99 data set

Some Criticizes the KDD Cup 99 data set as being an inefficient data set to be used for any research. The first reason that they outline is that the KDD data set contains a large amount of redundant data and hence the redundancy of the data may affect results that will be obtained from data set. According to (Mahobod and E, 2009) 78% of the records in the training data and 75% of the records in the test data set are redundant. It is feared that having this much of redundant records may affect the result obtained from data mining algorithms. As some algorithms may favour the most frequent records and hence ignore non frequent traffics and make them skip harmful records like U2R which are usually infrequent records. It was also proven that similar results are obtained in various experiments by using different portions of the KDD'99 data set in various experiments. And this made it difficult to evaluate the outputs of the researches that used different techniques and algorithms as most of the results range from 85 % to 100% of success rate in correctly classifying the records.

In the hope to prove the reliability of the KDD data set to provide accurate results by using the data set, researchers made several experiments (L and E, 2001) made an experiment which revealed that the distribution of the attacks in the KDD'99 data set were unevenly distributed. This was proved by dividing the data set into ten groups which contain equal 10% of the total dataset records. Many of the subgroups contain only a single attack type.

To avoid the above mentioned problems with the KDD99 data set an alternative data set was prepared by (Mahobod and E, 2009).The data set was prepared to avoid the problem which existed in KDD99 data. The new data set contains a manageable number of records which avoids excessive consumption of resources to run the data set. And it also avoids the need to select a portion of the dataset to save resources which otherwise will be demanded if the whole set of the KDD 99 data set is used for an experiment.

	Original Records	Distinct Records	Reduction Rate
Attacks	3,925,650	262	93.32%
Normal	972,781	812,814	16.44%
Total	4,898,431	1,074,992	78.05%

Table 4 Redundant records in the KDD training set (Mahobod and E, 2009)

	Original Records	Distinct Records	Reduction Rate
Attacks	250,436	29,378	88.26%
Normal	60,591	47,911	20.92%
Total	311,027	77,289	75.15%

Table 5 Redundant records in the KDD test set (Mahobod & E, 2009)

To avoid the problem of redundancy the new data set named NSL-KDD was made by removing the redundant records from the train and test sets. In addition to this, there are

no duplicate records in the new data set. And hence the performance of the algorithms used will not be biased by those methods which have better detection rates on the frequent records. As mentioned above the number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD data set. And hence the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques. The number of records in the train and test sets is fair which makes using the whole dataset for an experiment not much of a waste of resource and there will not be the need to use a subset of the dataset. And as a result evaluation outcomes of different experiments made using the data by applying different machine learning algorithms be appropriate and worth doing (Ali and W,2010).

The following table shows previous woks by local researchers and their findings

Author	Objective	Method	Key findings
Adamu Teshome	Application of cost sensitive learning using data mining approach to intrusion detection system	IDS models were developed using cost sensitive classification tree CS-CRT and cost sensitive decision tree CS-MC4 algorithms on full training NSL-KDD dataset using a powerful machine learning and data mining Tanagra tool.	CS_MC4 gave better result than CS-CRT in terms of accuracy, false positives rate and average misclassification costs.
Zewdie Mossie	Advancing classification performance of algorithms using cost sensitive learning and feature selection using data mining for IDS.	Train cost sensitive CS-CM4 and metacost sensitive C4.5 algorithms to generate models on training set and build predicated instance to classify test data set for each of the selected attributes on each algorithms using WEKA	Information gain ratio and CFS algorithms are found to be the best technique in detection accuracy, average misclassification cost, false alarm rate and training time.
Tigabu Dagne	Constructing a predictive model using	For analysing data and classification of network	The model created using 10-fold cross validation

	a semi supervised approach for intrusion detection system that will enhance network security system	attacks from a network environment J48 decision tree classifier, Naive Bayes classifier and simple K-means clustering are used using WEKA	using the J48 decision tree algorithm with the default parameter values showed better prediction accuracy than Naive Bayes simple algorithm.

Table 6 Researches done on the area and their findings

Chapter Three

3. Dataset Preparation

3.1 Data collection and preparation

The process of Knowledge discovery and analysis involves data acquisition, data preprocessing, feature selection and data mining stages. The first step in the KDD process is data acquisition.

The data set that was used for this research is NSL-KDD which is a collection of selected instances of data from KDD CUP 99 data set. The NSL-KDD data set is a data set built by correcting some of the drawbacks of the KDD CUP 99 data. The data set is freely available for network intrusion detection experiments and it has been used in many such researches.

The NSL-KDD data set has a training and test set with 125,973 and 22,544 instances respectively with 41 features. These instances of data are extracted from the KDD data set by removing the redundant and irrelevant data which proved to lead to wrong results as many algorithms tend to be biased by frequent records. The data set is labeled with normal class and 38 different attack types which will be later categorized into four different attack classes. The attack classes are Denial of service (Dos), User to Root attack (U2R), Remote to Local Attack (R2L) and Probing attack. DoS is an attack in which the attacker makes some computing or memory resources too busy or too full to handle legitimate requests ,or denies legitimate users access to a machine. U2R is a class of exploit in which the attacker starts accessing to a normal user account on the system by either sniffing passwords, a dictionary attacker and by other similar means and is able to exploit some vulnerability to gain root access to the system. R2L occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine (Yang and Chuang, 2009).

The testing data is found as a different collection and in addition to the attack types in the training set, it contains additional attack types which are not included in the training data. The fact that new attacks are included in the testing data, makes a data mining model developed using the training data set to be realistic because in reality new attack types are always introduced and any successful intrusion detection models should be evaluated by how effectively they can identify new attack types in a testing data.

The Improvements that were made in NSL–KDD data set is the redundant records in the KDDCUP'99 data set in both the training and test sets are removed which would have bias classifiers towards the more frequent records. As a result the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques. The number of records in the train and test sets is realistic which makes it reasonable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable (Ali and W, 2010).

The attack types in the training and testing sets of the original KDD 99 data set are in total 38. Seventeen of the attack types are only included in the test data set. These different attack types are categorized into the four different classes of the attack types mentioned above. According to (Adetunmbi and A, 2010) the following are the corresponding classes of the attack types.

	Attack Names	Classes
1.	Back, land, Neptune, pod, smurf, teardrop	Dos
2.	Satan, ipsweep, nmap, portsweep	PROBE
3.	Guess_passwd, ftp_write, imap, phf, multihop, warezmaster, warezclient, spy	R2L
4.	Buffer_overflow,loadmodule,perl,rootkit	U2R

Table 7 List of attack names and their classes in NSL-KDD data set (Adetunmbi & A, 2010)

3.2 Pre-processing

Data sets from which any knowledge and pattern to be extracted using data mining algorithms, need to be preprocessed in order to remove redundant records, outliers and irrelevant data and fill out missing data as well. This will help to get a reliable result from the dataset as otherwise even if the right algorithm is used to extract a desired knowledge from a bad data set the result may not be reliable.

A preprocessing task needs to be done on a data set to make the data accurate, complete and consistent. The redundant instances of the KDD CUP 99 data set were removed and the data was made to be more accurate and consistent in the NSL-KDD data set. And hence preprocessing tasks were not required to do on the NSL KDD data set except the following few tasks.

The preprocessing tasks that have been done on the data are:-

1. The NSL-KDD training and test data set is found in text format and hence in order to make it readable by WEKA 3.6.10, the file format has been changed to csv format. As WEKA 3.6.10 reads files in CSV and arff formats.
2. The different attack types of the training data set were replaced in the training and test sets by the respective classes of attacks for making classification easier.
3. Applying feature selection algorithms in order to remove redundant and irrelevant features has been done.
4. As mentioned above the original KDD data set contains many redundant and irrelevant data. In addition to that the distribution of the attack types in the training data set is unevenly distributed. These problems are corrected in the NSL-KDD data set and hence there is no major preprocessing task that has been done on the dataset except removing records which doesn't have attributes.

3.3 Feature Selection

Feature selection is the process of finding out important subsets of features which can clearly define the original data set and which can classify or cluster the data set more correctly than the whole set of original features would have been taken (JInsong & C, 2010). The purpose of feature selection is to reduce the number of redundant and

irrelevant features from large dataset and use the more important features to classify the data set at hand. Most researches have focused on elimination of irrelevant features than redundant data. However as stated in chapter two redundant features have equal negative impact on learning algorithms as well as on discovering reliable knowledge from a dataset.

Of the three kinds of feature selection approaches, wrapper based method is used in this research for feature selection process. Wrapper based approach proved to be the most effective one as it employs an evaluation criteria of identifying good features based on their power of classifying the data set accurately using a specific classifying algorithm (JInsong & C, 2010)

3.4 Performance Evaluation Metrics

A performance evaluation method is a method which helps in measuring the efficiency and accuracy of a data mining model. There is different kind of evaluation metrics depending on the kind of data mining technique we have applied on the data set at hand. That is for supervised and unsupervised approaches of mining knowledge from a dataset the kind of evaluation metrics that should be applied on the derived model may differ.

The output of the prediction models was evaluated using expert opinions on the relevance of features selected and on the accuracy of the models.

In addition to expert opinions to evaluate the performance of the classification models, the other evaluation metrics that are used on this thesis are

3.4.1 Confusion matrix

A binary classification model classifies each instance into one of two classes. This kind of classifications makes the accuracy of the classification to fall in one of the following four possibilities: True positive, true negative, false positive or false negative.

For classification algorithm that identifies an attack correctly it would fall on the true positive, if it labels non attack instance as a non threat it will be True negative, for identifying a normal instance as an attack will make it a false negative and if it classifies an attack as normal it would be a false positive.

The result of the accuracy of a classification algorithm would be put on the confusion matrix and for a perfect model the values of true positive and true negative fields would be filled out and the other fields' values would be zero.

		Predicted Class	
		TP	TN
Actual Class	TP	TP	FN
	TN	FP	TN

Table 8 confusion matrix

In addition to speed, robustness, scalability and interpretability of a model to run a classification algorithm and classify a data set there are other performance metrics that can be derived from the confusion matrix these are accuracy, error rate, recall and precision (Andre and R, 2007).

Accuracy shows the number of predictions which are correct and it is determined by the following formula. Accuracy is usually estimated by using an independent test set that was not used at any time during the learning process.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Error rate is the rate of incorrect predictions made by a model over a data set.

The error rate is computed using a testing dataset obtained through one of applied re-sampling techniques. An error rate is the sum of instances miss-classified over the total number of instances in the test data set. (Soman, Diwakar, and Jay, 2006)

$$Error = \frac{FP+FN}{TP+TN+FP+FN}$$

Recall is the proportion of positive instances that were correctly identified and can be calculated as follows.

$$Recall = \frac{TP}{TP + FN}$$

Precision measures the number of instances that are correctly classified by the data mining model developed. (Okach,2014)

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

3.5 Tools used for the experiments

3.5.1 MS EXCEL 2007

Since Excel is a handy tool to manipulate, visualize and do editing on datasets, MS Excel 2007 has been used for reading and converting the originally text file format of the NSL-KDD data set to CSV file format. In addition to that to understand and visualize the content of the dataset and to label the different attack types in the training and test set to probe,R2L, U2R and Dos classes of attacks was done with Excel as filtering editing using excel is much easier.

3.5.2 WEKA

Since WEKA is open source data mining software which contains a collection of machine learning algorithms for data mining tasks and all the selected algorithms are supported in WEKA 3.6.10 version, WEKA is used for data analysis process in this research. In addition to this it is a widely used data mining tool for data analysis and feature selection processes and also it has graphical user interface as well as command based interface which makes it attractive to be used in this research.

The algorithms and tools that are included in WEKA help in discovering patterns and knowledge from a data set. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules and visualization (Frank and I., 2005).

WEKA has various methods of attribute selection which helps in segmenting subset feature. WEKA's supervised attribute selection methods, enable a combination of evaluation and search methods to be specified, where the objective is to determine a useful subset of attributes as input to learning scheme (Remco, 2008).

WEKA also helps to apply algorithms on a data set to perform all processes of experimental data mining including preparing the input data, evaluating learning schemes statistically and visualizing the input data and the result of the learning process. The diverse algorithms can be accessed through a common interface which helps the user to compare different methods and identify the most appropriate ones easily (Frank and I., 2005).

Chapter Four

4. Experimentation

4.1 Experimentation and Discussion

In this chapter the experiment that has been conducted on NSL-KDD data set and the feature selection approach and the algorithms used to identify attacks from normal traffic will be explained in detail. As discussed in the previous chapters, wrapper feature selection approach has been used to select important features and remove the redundant and irrelevant ones from the NSL-KDD Data set. For all the processes of the knowledge discovery feature selection and analysis of NSL- KDD WEKA 3.6.10 and Ms Excel 2007 has been used.

4.2 Experimentation Setup

The default values for memory setting of WEKA 3.6.10 were taken as they are since they were found to be suitable for loading and processing the 125,973 instances of the NSL-KDD data set and for applying the different classifying and feature selection algorithms. In addition to that rather than affecting the performance of the algorithms by changing parameters it was believed performance of algorithms can better be compared if identical parameter settings are taken and the accuracy of algorithms are evaluated depending on their output when used for predicting attack types.

In all the experiments the whole set of training and test set were used and cross validation is not used. The reason behind this is that since cross validation is necessary when classes are not well represented in test and training sets and when this makes random sampling unable to give the whole picture of the data. However in NSL-KDD data set attack types are fairly distributed in the training and test sets (Mahobod and E., 2009).

4.3 Experimentation on feature selection

A feature selection algorithm can be seen as a combination of a search technique for proposing new feature subsets along with an evaluation measure which scores the

different feature subsets. A feature selection algorithm needs to be applied on a data set when a dataset is large. A dataset is said to be large when it consists of either a large number of instances or a large number of potentially predictive features or both. From a computational perspective, the time complexity of feature selection algorithms makes it infeasible to use all of the data in a large dataset (Liu and Steiono, 1998). In theory, more features should provide more discriminating power, that is, all available features of a large data set should be fed to a knowledge discovery algorithm, to get a better result of pattern extraction and hidden knowledge discovery. However in practice, with a limited amount of training data, excessive features will not only significantly slow down the learning process, but also cause the classifier to over-fit the training data because irrelevant or redundant features may confuse the learning algorithm (Yu and H, 2004), (JInsong and C, 2010).

Feature selection methods can broadly be classified as supervised and unsupervised by taking into consideration the usage of class information of the data in the feature selection process. It also can be further classified as wrapper and filter selection based on the kind of evaluation methods to be used in the feature selection process.

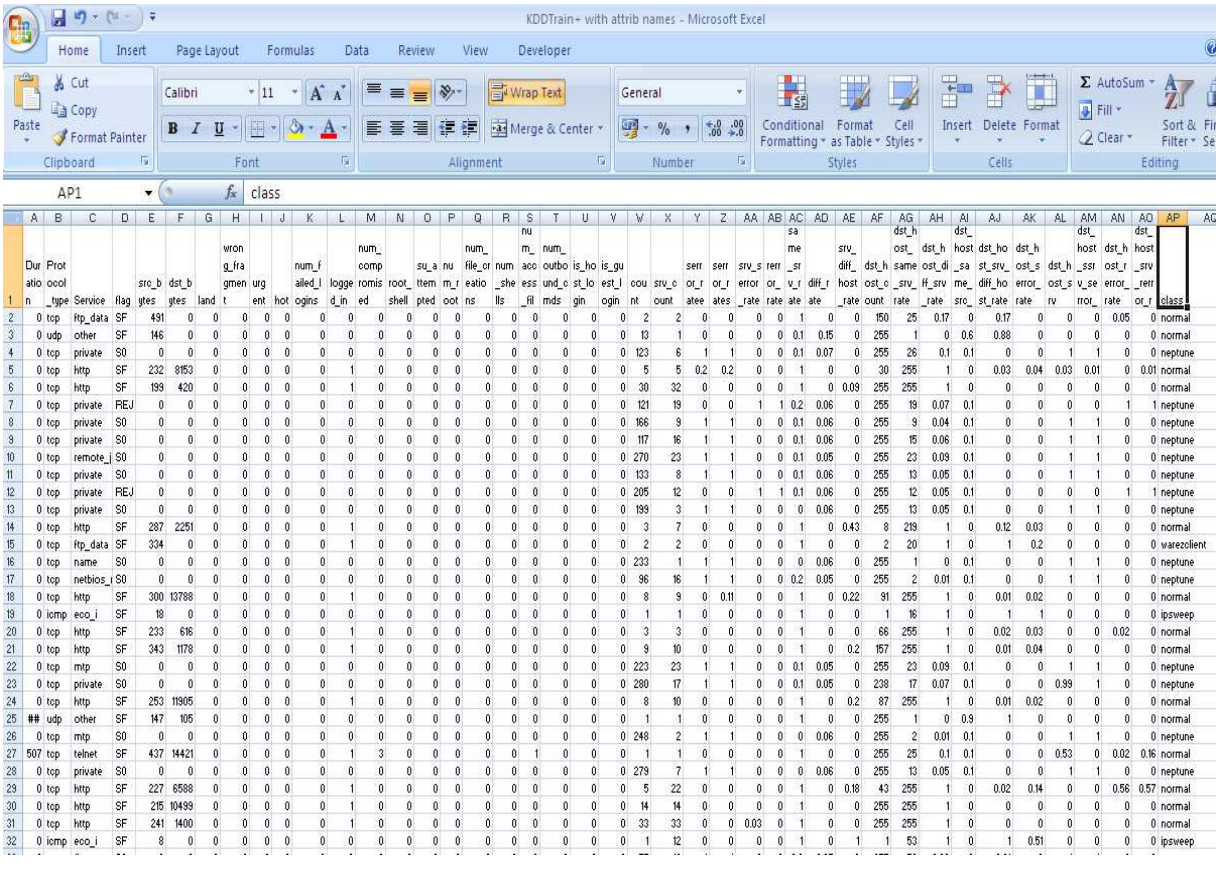
Even if filter methods are prefered because of their efficiency for feature subset selection and various filter methods has been applied in extracting relevant features from the NSL- KDD data set (Lutu and P, 2010), same kind of classification accuracy would not be achieved if different classification algorithms had been used on the selected feature subset (Ron and G., 1996) and hence optimal feature subset is not guaranteed. This shows the importance of using the input of an induction algorithm for the evaluation of optimality of a feature for the algorithm. In wrapper method if a feature subset of a large data set is found out to be optimal for any particular algorithm, the same algorithm has to be used in the classification of the dataset for high accuracy rate.

The NSL-KDD data set is one of the data set which can be called a large dataset. It has 41 features and 125,973 instances in the training dataset and 22,544 instances in the test set. Even if the number of instances is affordable to be used for knowledge extraction using a learning algorithm, the number of features needs to be filtered to remove irrelevant

and redundant features (Asyiqin and Mohd., 2012). In this research wrapper method is used to get the optimal features from the data and to increase accuracy of the prediction model.

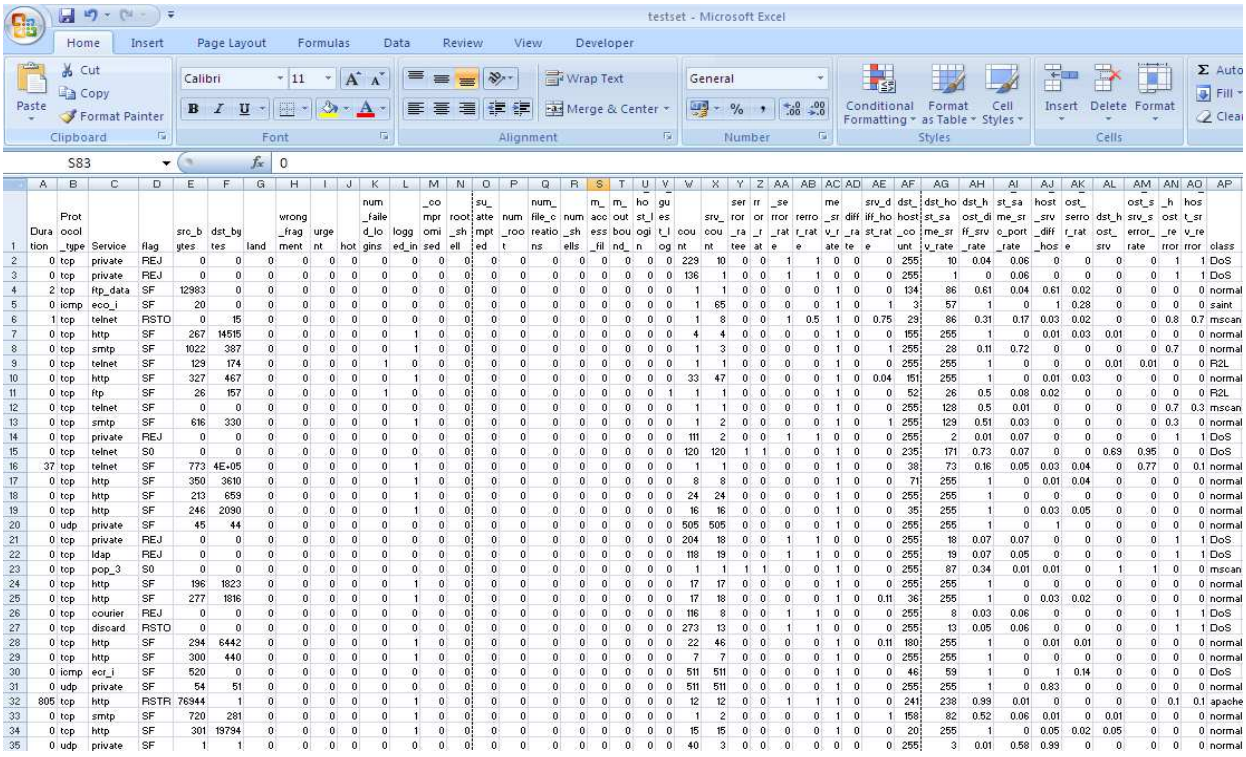
The main purpose of feature selection is to reduce irrelevant and redundant features and to make a learning algorithm be effective in classifying data sets and to increase the accuracy rate of the prediction model. In network intrusion detection systems reducing the false positive rate of the systems is the main challenge in the area. In wrapper feature selection methods finding out the most optimal features for a particular learning algorithm helps the algorithm to provide a high accuracy in the classification process.

The following figures show sample data from the NSL-KDD data set which has training and testing set with 41 features.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ
1	Duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_shell	su_attempted	num_root	num_file_creations	num_shells	num_access_files	num_outbound_cmds	is_host_login	is_guest_login	count	srv_count	serror_rate	srv_serror_rate	rerror_rate	srv_rerror_rate	same_srv_rate	diff_srv_rate	srv_diff_host_rate	dst_host_count	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	class	
2	0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	150	25	0.17	0	0.17	0	0	0.05	0	normal	
3	0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	1	0	0	0	0	0.1	0.15	0	255	1	0	0.6	0.88	0	0	0	0	normal	
4	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	6	1	1	0	0	0.1	0.07	0	255	26	0.1	0.1	0	0	1	1	0	neptune	
5	0	tcp	http	SF	232	8153	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	5	5	0.2	0.2	0	0	1	0	0	30	255	1	0	0.03	0.04	0.03	0.01	0	0.01	normal	
6	0	tcp	http	SF	199	420	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	30	32	0	0	0	0	1	0	0.09	255	255	1	0	0	0	0	0	0	normal	
7	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	121	19	0	0	1	1	0.2	0.06	0	255	19	0.07	0.1	0	0	0	0	1	neptune	
8	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	9	1	1	0	0	0.1	0.06	0	255	9	0.04	0.1	0	0	1	1	0	neptune	
9	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	16	1	1	0	0	0.1	0.06	0	255	15	0.06	0.1	0	0	1	1	0	neptune	
10	0	tcp	remote_j	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	270	23	1	1	0	0	0.1	0.05	0	255	23	0.09	0.1	0	0	1	1	0	neptune	
11	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133	8	1	1	0	0	0.1	0.06	0	255	13	0.05	0.1	0	0	1	1	0	neptune	
12	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	205	12	0	0	1	1	0.1	0.06	0	255	12	0.05	0.1	0	0	0	0	1	neptune	
13	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	3	1	1	0	0	0	0.06	0	255	13	0.05	0.1	0	0	1	1	0	neptune	
14	0	tcp	http	SF	287	2251	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	7	0	0	0	0	1	0	0.43	8	219	1	0	0.12	0.03	0	0	0	normal	
15	0	tcp	ftp_data	SF	334	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	2	20	1	0	1	0.2	0	0	0	warezclient	
16	0	tcp	name	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	233	1	1	1	0	0	0	0.06	0	255	1	0	0.1	0	0	1	1	0	neptune	
17	0	tcp	netbios_	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96	16	1	1	0	0	0.2	0.05	0	255	2	0.01	0.1	0	0	1	1	0	neptune	
18	0	tcp	http	SF	300	13788	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	8	9	0	0.11	0	0	1	0	0.22	91	255	1	0	0.01	0.02	0	0	0	normal	
19	0	icmp	eco_i	SF	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	16	1	0	1	1	0	0	0	ipsweep	
20	0	tcp	http	SF	233	616	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	1	0	0	66	255	1	0	0.02	0.03	0	0	0.02	normal	
21	0	tcp	http	SF	343	1178	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	9	10	0	0	0	0	1	0	0.2	157	255	1	0	0.01	0.04	0	0	0	normal	
22	0	tcp	mtp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	223	23	1	1	0	0	0.1	0.05	0	255	23	0.09	0.1	0	0	1	1	0	neptune	
23	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	280	17	1	1	0	0	0.1	0.05	0	238	17	0.07	0.1	0	0	0.99	1	0	neptune	
24	0	tcp	http	SF	253	11905	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	8	10	0	0	0	0	1	0	0.2	87	255	1	0	0.01	0.02	0	0	0	normal	
25	##	udp	other	SF	147	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	1	0	0.9	1	0	0	0	0	normal	
26	0	tcp	mtp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	248	2	1	1	0	0	0	0.06	0	255	2	0.01	0.1	0	0	1	1	0	neptune	
27	507	tcp	telnet	SF	437	14421	0	0	0	0	1	3	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	255	25	0.1	0.1	0	0	0.53	0	0.02	0.16	normal	
28	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	279	7	1	1	0	0	0	0.06	0	255	13	0.05	0.1	0	0	1	1	0	neptune	
29	0	tcp	http	SF	227	6588	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	5	22	0	0	0	0	1	0	0.18	43	255	1	0	0.02	0.14	0	0	0.56	0.57	normal	
30	0	tcp	http	SF	215	10499	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	14	14	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	normal	
31	0	tcp	http	SF	241	1400	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	33	33	0	0	0.03	0	1	0	0	255	255	1	0	0	0	0	0	0	normal	
32	0	icmp	eco_i	SF	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	12	0	0	0	0	1	0	1	1	53	1	0	1	0.51	0	0	0	ipsweep	

Figure 3 Sample data set of the NSL-KDD training data set



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP
1	Dura tion	Prot ocol _type	Service	flag	src_b ytes	dst_by tes	land	wrong _frag ment	urge nt	hot	num _faile d_lo gins	logg ed_in	_co mpr omi sed	root _sh ell	su_ atte mpt ed	num _roo t	num_ file_c reatio ns	num _sh ells	m_ acc ess _fil	m_ out bou nd_	ho st_l ogi n	gu es t_l og	cou nt	srv_ cou nt	ser ror _ra te	rr or _r at	_se rror _rat e	rerro r_rat e	me _sr v_r ate	diff _ra te	srv_d iff_ho st_rat e	dst_ host _co unt	dst_ho st_sa me_sr v_rate	dst_h ost_di ff_srv _rate	st_sa me_sr o_port _rate	host _diff _hos t	ost_ serro r_rat e	dst_h ost_ srv	ost_s srv_s error_ rate	_h ost _re rror	hos t_sr v_re rror	class
2	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	229	10	0	0	1	1	0	0	0	255	10	0.04	0.06	0	0	0	0	1	1	DoS
3	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	136	1	0	0	1	1	0	0	0	255	1	0	0.06	0	0	0	0	1	1	DoS
4	2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	134	86	0.61	0.04	0.61	0.02	0	0	0	0	normal
5	0	icmp	eco_i	SF	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	65	0	0	0	0	1	0	1	3	57	1	0	1	0.28	0	0	0	0	saint
6	1	tcp	telnet	RSTO	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	8	0	0	1	0.5	1	0	0.75	23	86	0.31	0.17	0.03	0.02	0	0	0.8	0.7	mscan
7	0	tcp	http	SF	267	14515	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	1	0	0	155	255	1	0	0.01	0.03	0.01	0	0	0	normal
8	0	tcp	smtp	SF	1022	387	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	1	0	1	255	28	0.11	0.72	0	0	0	0	0.7	0	normal
9	0	tcp	telnet	SF	129	174	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	255	1	0	0	0	0.01	0.01	0	0	R2L
10	0	tcp	http	SF	327	467	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	33	47	0	0	0	0	1	0	0.04	151	255	1	0	0.01	0.03	0	0	0	0	normal
11	0	tcp	ftp	SF	26	157	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	52	26	0.5	0.08	0.02	0	0	0	0	0	R2L
12	0	tcp	telnet	SF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	128	0.5	0.01	0	0	0	0	0.7	0.3	mscan
13	0	tcp	smtp	SF	616	330	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	1	0	1	255	129	0.51	0.03	0	0	0	0	0.3	0	normal
14	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	111	2	0	0	1	1	0	0	0	255	2	0.01	0.07	0	0	0	0	1	1	DoS
15	0	tcp	telnet	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	120	1	1	0	0	1	0	0	235	171	0.73	0.07	0	0	0.69	0.95	0	0	DoS
16	37	tcp	telnet	SF	773	4E+05	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	38	73	0.16	0.05	0.03	0.04	0	0.77	0	0.1	normal
17	0	tcp	http	SF	350	3610	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	1	0	0	71	255	1	0	0.01	0.04	0	0	0	0	normal
18	0	tcp	http	SF	213	659	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	24	24	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	0	normal
19	0	tcp	http	SF	246	2090	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	16	16	0	0	0	0	1	0	0	35	255	1	0	0.03	0.05	0	0	0	0	normal
20	0	udp	private	SF	45	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	505	505	0	0	0	0	1	0	0	255	255	1	0	1	0	0	0	0	0	normal
21	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	204	18	0	0	1	1	0	0	0	255	18	0.07	0.07	0	0	0	0	1	1	DoS
22	0	tcp	ldap	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	118	19	0	0	1	1	0	0	0	255	19	0.07	0.05	0	0	0	0	1	1	DoS
23	0	tcp	pop_3	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	255	87	0.34	0.01	0.01	0	1	1	0	0	mscan
24	0	tcp	http	SF	196	1823	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	17	17	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	0	normal
25	0	tcp	http	SF	277	1816	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	17	18	0	0	0	0	1	0	0.11	36	255	1	0	0.03	0.02	0	0	0	0	normal
26	0	tcp	courier	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	116	8	0	0	1	1	0	0	0	255	8	0.03	0.06	0	0	0	0	1	1	DoS
27	0	tcp	discard	RSTO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	273	13	0	0	1	1	0	0	0	255	13	0.05	0.06	0	0	0	0	1	1	DoS
28	0	tcp	http	SF	294	6442	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	22	46	0	0	0	0	1	0	0.11	180	255	1	0	0.01	0.01	0	0	0	0	normal
29	0	tcp	http	SF	300	440	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	7	7	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	0	normal
30	0	icmp	ecr_i	SF	520	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	511	511	0	0	0	0	1	0	0	46	59	1	0	1	0.14	0	0	0	0	DoS
31	0	udp	private	SF	54	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	511	511	0	0	0	0	1	0	0	255	255	1	0	0.83	0	0	0	0	0	normal
32	805	tcp	http	RSTR	76944	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	12	12	0	0	1	1	1	0	0	241	238	0.99	0.01	0	0	0	0	0.1	0.1	apache
33	0	tcp	smtp	SF	720	281	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	1	0	1	158	82	0.52	0.06	0.01	0	0.01	0	0	0	normal
34	0	tcp	http	SF	301	19794	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	15	15	0	0	0	0	1	0	0	20	255	1	0	0.05	0.02	0.05	0	0	0	normal
35	0	udp	private	SF	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	3	0	0	0	0	0	0	0	255	3	0.01	0.58	0.99	0	0	0	0	0	normal

Figure 4 Sample data of the NSL- KDD test data set

The following figure shows the steps followed in this research to select optimal subset and to classify the attack types from the NSL-KDD data set.

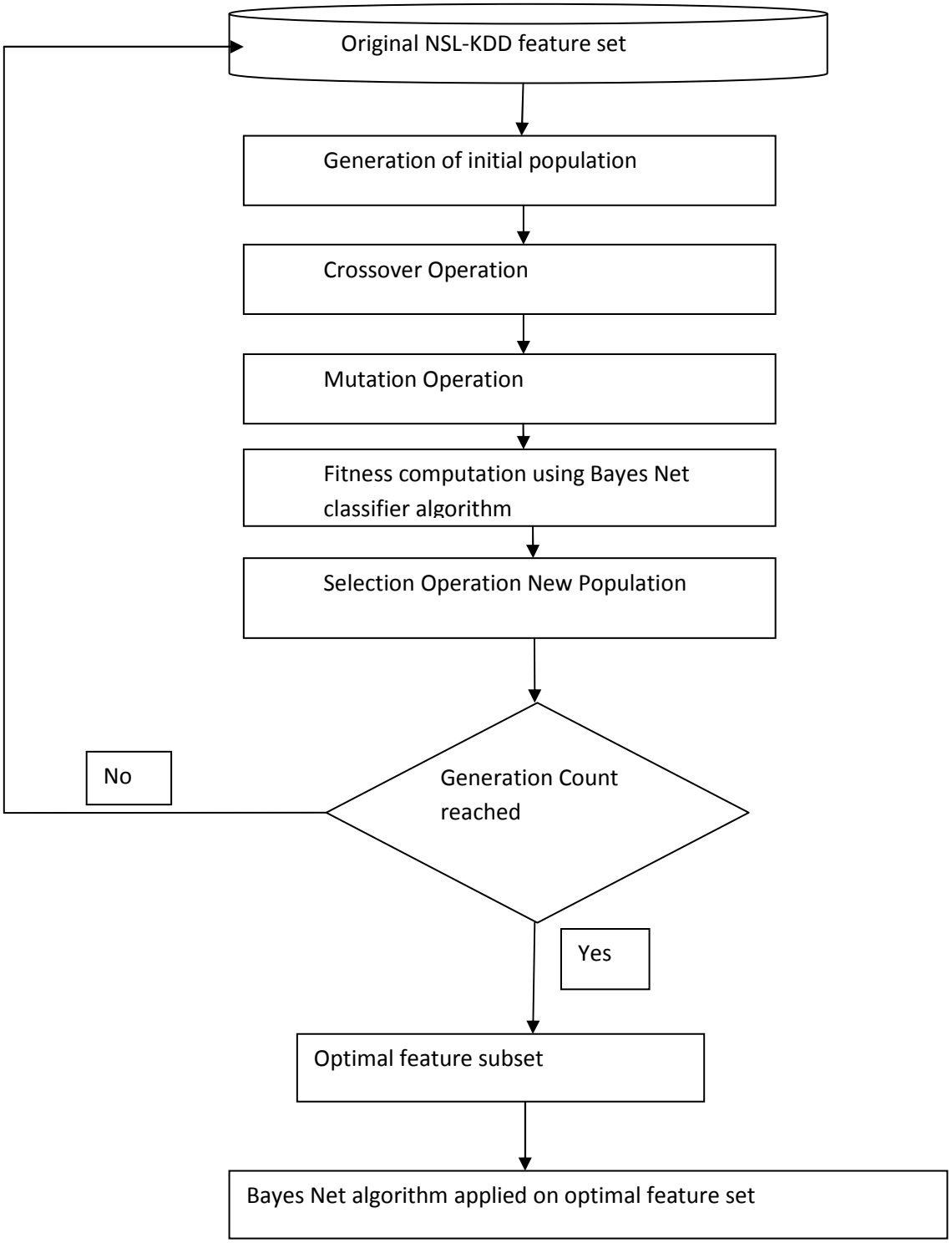


Figure 5 Proposed feature selection approached in this research

In this research in order to experiment the wrapper feature subset evaluation approach on NSL-KDD data set four induction algorithms were selected and applied on the data. In all the feature selection process the searching algorithm used is genetic algorithm. The algorithms used are Genetic Algorithm which is used as searching algorithm for searching through the attributes; the subset evaluator is wrppersubseteval which evaluates attribute sets by using a learning algorithm. The learning algorithms used for this purpose are BayesNet and Naïve Bayes algorithm, Kmeans algorithm and J48.

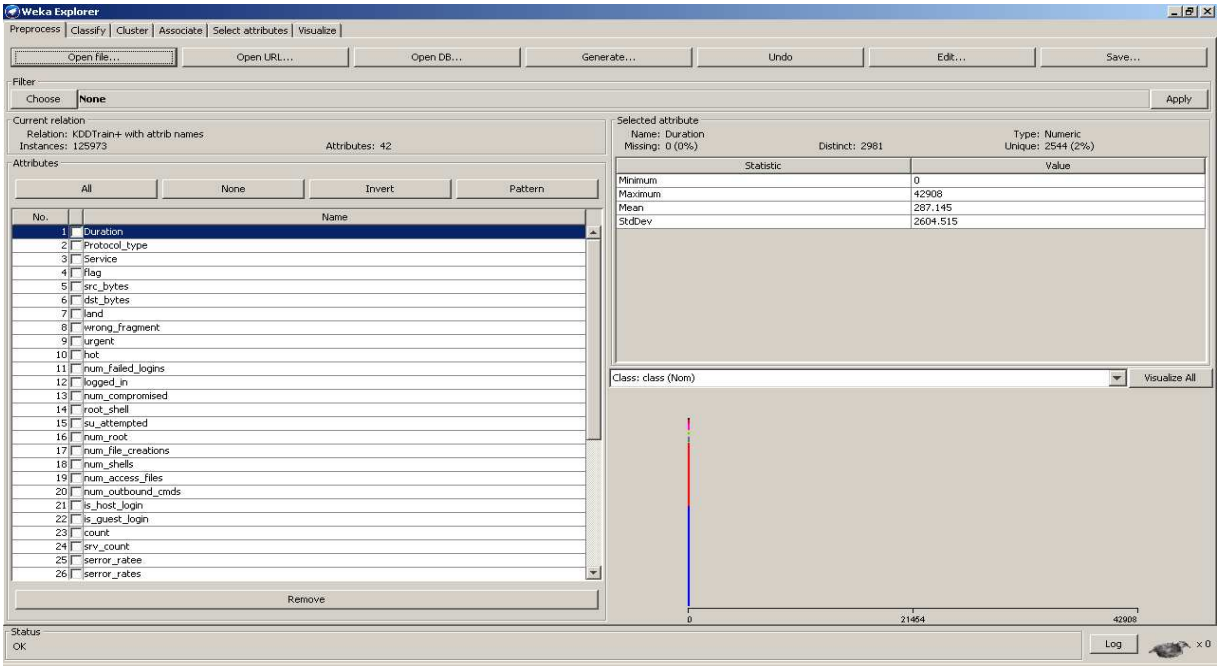


Figure 6 NSL-KDD training data set opened in WEKA for feature selection

4.4 Description of algorithms used

4.4.1 Genetic Algorithm

The theory of natural evolution was the base for genetic algorithms which is a member of evolutionary algorithm. A genetic algorithm searches for the best approximation of a solution for a problem by creating consecutive populations of individuals which will be considered as feasible solutions for the problem. To find out how good features are for the solutions a fitness function is used. And to create new populations

genetic operators based on selection and reproduction is used. (Gabriella, Massimiliano and Rocco., 2014)

The parameter setting for the major features of Genetic algorithm on WEKA was as follows, the population size which is the number of attribute sets in the population is 20, the Number of generation which is the stopping criterion is 20, crossover rate which is the probability that two population members will exchange genetic material is set to 0.6 and the mutation probability which is the probability of mutation occurring is 0.033.These parameters are used for all classifying algorithms used.

4.4.2 BayesNets

The main induction algorithm used for the evaluation of the feature subsets in the optimal feature subset search was BayesNets. Other induction algorithms are also used to compare their performances with the Bayesian network.

Bayes Networks are graphical representation for probabilistic relationships among a set of random variables. This algorithm is derived from Bayes theorem which describes how the conditional probability of a set of possible causes for a given observed event can be computed from knowledge of the probability of each cause and the conditional probability of the outcome of each cause. The basic idea of the algorithm is to test the dependency of attributes in predicting the value of the class and the attribute. The dependency of two attributes is measured by the conditional probabilities of the class attribute. If two attributes are identified to be dependent, either of them can be removed to achieve attribute reduction. The reason why this algorithm selected is that because the process of attribute selection is simple and that makes it efficient in providing output. Wrapper based feature selection approach is criticized by its inefficiency in feature selection process and the performance of the Bayesian network believed to counter attack this drawback of the approach.

Some of the parameters or options of the BayesNet algorithm for which the default values of WEKA are taken in the experiment are:-

- Estimator- select estimator algorithm for finding the conditional probability tables of the Bayes Network.

- SearchAlogrithm- selects method used for searching network structures.

- useADTree- when ADTree (the data structure for increasing speed on counts, not to be confused with the classifier under the same name) is used learning time goes down typically. However, because ADTrees are memory intensive, memory problems may occur. Switching this option off makes the structure learning algorithms slower, and with less memory (Faitin, Kennet and Ruggeri, 2007).

4.4.3 Naïve Bayes

In Naive Bayes a presence and absence of a particular feature of a class is unrelated to the presence or absence of any other feature. Depending on nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting. The Naive Bayes classifier requires a small amount of training data to estimate the parameters like means and variances of the variables necessary for classification because independent variables are assumed only the variances of the variables for each class need to be determined and not the entire covariance matrix.

Some of the parameters or options of the algorithm Naïve Bayes for which the default values of WEKA are taken in the experiment are:-

- Debug- If set to true, classifier may output additional information to the console.

- dispalyModelInOldFOrmat- Use old format for model output. The old format is better when there are many class values. The new format is better when there are fewer classes and many attributes.

- useKernelEstimator- Uses a kernel estimator for numeric attributes rather than a normal distribution.

- useSupervisedDiscretization-use supervised discretization to convert numeric attributes to nominal ones.(Faltin, Kenett and Ruggeri, 2007).

4.4.4 K Means

Weka has a feature selection option which enables to use a clusterer for classification. And WEKA uses various clusterers which can be used for the purpose in this research K-means clustering was used.

Some of the options or parameters that simple K-means requires to be defined are:

- Dispalystdevs which is required for standard deviation of numeric attributes and counts of nominal attributes to be either to be displayed or not.
- Distance function- is the distance function which will be used for comparison of instances. For this Euclidean distance is used.
- Dontreplacemissingvalues - Which gives the option whether missing values globally need to be replaced with mean/mode or not.
- MaxIterations- it helps to set maximum number of iterations.
- Numclusters- helps to set number of clusters
- Preserveinstancesorder- Helps to decide whether to preserve order of instances.
- Seed- sets the random number of seed to be used. (R, 2008).

K means algorithm is a classic clustering technique in which the number of clusters is defined in advance which will be taken as a parameter k. Then k points are chosen at random as cluster centres. Then all instances are assigned to their closest cluster centre based on Euclidean distance metric. Next the mean of the instances in each cluster is calculated and taken as cancroids which will be taken as new centre values for their respective clusters. And the whole process is repeated with the new cluster centres and iteration will continue until the same points are assigned to each cluster in consecutive rounds at which stage the cluster centres have stabilized and will remain the same forever.

K means clustering is simple and effective choosing the cluster centre to be the centroid minimizes the total squared distance from each of the cluster's points to its centre. Once

the iteration has stabilized each point is assigned to its nearest cluster centre, so the overall effect is to minimize the total squared distance from all points to their cluster centres.

The default values of WEKA for all the parameter of the Kmeans algorithm for this experiment were used.

4.4.5Decision Tree

Decision tree is a widely used learning method. It performs classification by constructing a tree based on training instances with leaves having class labels. The tree is traversed for each test instance to find a leaf and the class of the leaf is the predicted class. This is a directed knowledge discovery in the sense that there is a specific field whose value will be predicted. The steps in creating a decision tree classification model is select optimal attributes on the basis of a heuristic or statistical measure like information gain then partition training samples based on selected attributes. The Conditions for stopping partitioning are when all samples for a given node belong to the same class and there are no samples left, when there are no remaining attributes for further partitioning majority voting is employed for classifying the leaf. At each node, the best attribute is selected for splitting the training samples using goodness function. The best attribute is the one that separates the classes of the training samples quickly so that it results in the smallest tree. The common evaluation measures for goodness of the best attributes are Information gain and Entropy.

Decision trees are the most powerful approaches in knowledge discovery and data mining. Decision trees are highly effective tools in many areas such as data and text mining, information extraction and machine learning. Some of the benefits that decision trees offer to data mining are handling a variety of input data like nominal, numeric and textual and it also handles dataset with error. (Bhargava et al., 2013)

J48 is an algorithm that is used to create decision trees and in WEKA J48 is used for this purpose. The J48 decision tree classifier requires attribute values of available training data inorder to create a decision tree based on which it classifies new item. When training data is fed to J48 decision tree the algorithm identifies the attribute with highest infromation

gain. And creates decision tree by deciding what combination of attributs gives a particular target value until the end of attribute is reached. By checking attributes and their values with those in the decision tree model the target value of new instances will be predicted . (Bhargava et al., 2013)

Some of the options or parameters of J48 decision tree algorithim for which the default values of WEKA were:-

- Debug-if set true classifier may output additional info to the console.
- BinarySplits-Whether to use binary splits on nominal attributes when building the trees.
- minNumObj- The minimum number of instances per leaf.
- numFolds-Determines the amount of data used for reduced error pruning. Onefold is used for pruning the rest for growing the tree.
- SubtreeRaising-Whether to consider the subtree raising operation when pruning

Of the algorithm used for feature selection using wrapper approach in this research, J48 decision tree algorithm and K means algorithms were found to be very inefficient in the feature selection process and hence were not selected in this research but has been used for experiment. Bayesnet and NaiveBayes algorithms were the algorithms which were found to be efficient and were used to compare their performances and effectiveness in selecting relevant and non redundant features which have high input in classifying attack types correctly. (Bhargava et al., 2013)

The optimal features selected after applying the above wrapper based feature selection algorithm are shown in the table below. The accuracy estimation of the feature selection was calculated based on the classification error of the features.

Feature selection Evaluator	Search method	Classifying induction algorithm	No of features selected	Subset of Features selected
Wrappersubseteval	Genetic Algorithm	BayesNet	17	3,5,6,7,8,9,11,14,18,19,22,24,29,30,38,40,42
Wrappersubseteval	Genetic Algorithm	NaiveBayes	13	2,3,4,8,12,15,17,21,26,30,32,42

Table 9 Feature subset selection results using the different induction algorithms

	Attribute	Attribute description
3	Service	Network service on the destination. Eg. http,telnet
5	Src-bytes	Number of data types from source to destination
6	Dst_bytes	Number of data types from destination to source
7	Land	1 if connection is from/to the same host/port, 0 otherwise
8	Wrong_fragment	Number of 'wrong' fragment
9	Urgent	Number of urgent packets
11	Num-failed login	Number of failed login attempts
14	Root shell	1 if root shell is obtained, 0 otherwise
18	Num shells	Number of shell prompts
19	Num access files	Number of operation on access control files
22	Is guest login	1 if the login is a 'guest' login, 0 otherwise
24	Srv count	Number of connections to the same service as the current connection in the past two seconds
29	Same srv rate	% of connections to the same service
30	Diff rate	
38	Dst host srv	
40	Dst host r error	
42	Class	

Table 9 Selected features using Bayes net algorithm

	Attribute	Attribute description
2	Protocol_Type	Type of the protocol,eg tcp, udp, etc
3	Service	Network service on the destination. Eg. http,telnet
4	Flag	Normal or error status of the connection
8	Wrong_fragment	Number of 'wrong' fragment
12	Logged_in	1 if successfully logged in; 0 otherwise
15	Su_attempted	1 if 'su root' command attempted; 0 otherwise
17	Num_file_creations	Number of file creation operations
21	Is_hot_login	1 if the login belongs to the 'hot' list; 0 otherwise
26	Serror_rates	% of connections that have 'SYN' errors
30	Diff rate	
32	Dst_host_count	
42	Class	

4.5 Training phase of the experiment

In the training phase of the experiment, using the selected subset features of the NSL KDD data set BayesNet algorithm was used to classify the dataset in order to build a classifying model. As it has been already mentioned, the NSL KDD data set is divided in training and test data set. The different types of attacks in the training data set are categorized in classes of PROBE, Dos, R2L and U2R and the rest of the data are labeled as normal. However in the test set of the data in addition to the attack types in the training set there are more attack types which are not included in the training data. This has been taken as an opportunity to evaluate the classifying power of the model that is built. Anomalous method of intrusion detection systems should be built by taking into consideration the existence of new kinds of intrusion and attack types are on production with the growth of network systems.

Duration	Protocol_t	Service	flag	src_bytes	dst_bytes	land	wrong_fra	urgent	hot	num_faile	logged_in	num_com	root_shell	su_attem	num_root	num_file_	num_shel	num_
0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	0	0	0	0	
0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	0	0	0	0	
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	remote_jc	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	http	SF	287	2251	0	0	0	0	0	1	0	0	0	0	0	0	
0	tcp	ftp_data	SF	334	0	0	0	0	0	0	1	0	0	0	0	0	0	
0	tcp	name	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	netbios_n	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	http	SF	300	13788	0	0	0	0	0	1	0	0	0	0	0	0	
0	icmp	eco_i	SF	18	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	http	SF	233	616	0	0	0	0	0	1	0	0	0	0	0	0	
0	tcp	http	SF	343	1178	0	0	0	0	0	1	0	0	0	0	0	0	
0	tcp	mtp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	tcp	http	SF	253	11905	0	0	0	0	0	1	0	0	0	0	0	0	
5607	udp	other	SF	147	105	0	0	0	0	0	0	0	0	0	0	0	0	

KDDTrain+ with attrib names

Figure 7 Sample training data

The effectiveness of the feature subset selected was tested using the induction algorithms used for subset evaluation. The classifying algorithms used as an induction algorithm are BayesNet and Naïve Bayes which are also used as a classifier algorithm in building the classifying model.

The classifying performance of the model was evaluated using different performance evaluator methods. Based on the training data set the correctly classified instances are 124,714 out of the 125,973 instances of the NSL-KDD data set. This is 99.0006% of the instances were correctly classified as Normal, PROBE, R2L, and U2R classes. The rest 1259 of the data set were incorrectly classified and this makes a 0.9994% of the total dataset. The other accuracy evaluation method used is TP Rate, FP Rate, Precision, Recall, F-Measure, ROC Area.

		Predicted Class				
		Normal	DoS	R2L	PROBE	U2R
Actual Class	Normal	66466	59	198	499	121
	DoS	122	45776	0	29	0
	R2L	38	5	945	3	4
	PROBE	92	57	5	11499	3
	U2R	23	0	0	1	28

Table 10 confusion matrix of the training model

4.6 Testing phase for the experiments

As has been mentioned above 17 new attack types are included in the test set. This makes building an effective intrusion detection model for the NSL-KDD data set difficult. The different types of the attacks in the test set which could be categorized in PROBE, R2L, U2R and DoS classes are classified. The rest of the attack types were left as they are to evaluate the classifying model performances. It is known that anomaly based IDS helps in identifying new kinds of attack types being introduced in a network system and intrusion detection methods are categorized in signature based and anomaly based IDS. Signature based intrusion detection system normally realize whether a traffic is an attack or normal traffic based on prior definition of attack patterns. The major problem with signature based systems is that since the system learn attack types from the rules that are entered into the system manually, new types of attacks which are not included in the rules will be left undetected (T & K, 2008). The Major benefit of anomaly detection network intrusion systems is that it takes into consideration the continuously introduction of new attack types to a network system. In this experiment wrapper based feature selection method and the classification algorithms used with the search method of the feature selection process applied provided a promising result in anomaly intrusion detection approach (Farooq and D., 2008).

As has been said above, the NSL-KDD data set contains various number of attack types in the training and test set. There are thirty eight attack types in total in both training and test set. Of the total attack types around seventeen are found only in the test set (Adetunmbi and A., 2010). The model that has been developed using the training set is shown to be very effective on identifying and classifying the new attack types in the test correctly.

In this research to select a subset of optimal features a wrapper based feature selection approach is used. Using genetic search algorithm with Bayes net and Naïve Bayes induction algorithms a subset of features are selected and the same algorithms are used on the respect subsets for effective classification. Their performances are compared with each other as well as the algorithms performance on filter based subset features selection approach.

4.7 Result analysis and discussions

A promising result has been obtained by using the feature subsets obtained from wrapper approach and after using the induction algorithms for classifying the different attack types in the NSL-KDD data set. The test set which has new attack types that were not included in the training data set seem to be effectively classified using the classification models built. The attack types that are correctly identified on the testing phase are 95.16% and only 4.84% are classified wrongly which means the method used is promising in identifying new attack types in network traffic.

To compare the performance of the various wrapper and filter feature selection algorithms used in this research, the widely used performance measures for supervised classification which are True positive rate, recall, precision, false positive rate and F-measure are used.

True positive (Accuracy) is defined as the ratio between the number of components correctly predicted which are expressed in TP and TN and the total number of components which is the sum of TP,TN,FP and FN.

Precision is defined as the ratio between the number of components classified as TP and the number of components classified as TP or FP.

Recall is defined as the ratio between the number of components classified as TP and the number of components classified as TP or FN. This shows that precision concerns the correctness of the responses provided by the methods while the completeness of the responses is measured by employing Recall. A measure that provides an indication of a balance between correctness and completeness is the harmonic mean of precision and recall or F-measure.

Feature selection method and classification algorithm	Precision	Recall	F-measure	TP Rate	FP Rate
Wrapper-BayesNet	.958	.952	.958	.953	.006
Filter-BayesNet	0.945	0.93	0.93	0.93	.01
Wrapper Naïve Bayes	.896	.863	.873	.867	.018
Filter Naïve bayes	.843	.597	.654	.597	.037

Table 11 Performance comparison of wrapper and filter feature subset selection

The table shows performance comparison of two classification algorithms which were applied on NSL-KDD data set after a feature subset was selected using wrapper and filter feature selection methods. In the case of wrapper method the induction algorithms used to evaluate the optimality of the feature subset selected however in the filter approach other evaluation methods are used to select the features.

The above performance evaluation measures shows the values of the selected accuracy measures which are Precision, Recall, F-measure, TP rate and FP-rate. In all the above filter and wrapper approaches the wrapper approach performed better than the filter methods. The reason behind this fact is that the classifying algorithms are involved in feature evaluation in the feature selection process and hence the selected features are optimal for the respective algorithm and made the accuracy higher. Of the entire

classification algorithm which involves both wrapper and filter feature selection methods, the bays net classification algorithm applied on feature subset selected by wrapper method provided the highest performance rate. Even though wrapper based feature selection approach have high time and computational costs, the effectiveness of the approach makes it unavoidable for critical systems like IDS.

Some of the challenges observed in this research were even if the wrapper based feature subset selection promises a high classification accuracy rate, the time complexity of the approach is one of the challenges. This challenge has become an obstacle to only compare the output of two algorithms which have a lower computational cost in feature selection using the wrapper method. The other challenge this fact has caused is in order to find efficient algorithms to use for the subset selection processes, an intensive experiments needed to be done to select proper feature subset selection algorithms like subset evaluator and searching methods.

Chapter Five

5. Conclusion and Recommendation

5.1 Conclusion

Feature selection process involves identifying relevant and useful features from a data set. And using relevant and optimal features is crucial in extracting a useful knowledge and pattern from a large data set. It also has a significant impact on the performance of

classifying and clustering algorithms as even successful algorithms may come up with wrong results if applied on redundant and irrelevant data set.

In this research a wrapper based approach is used to identify the optimal subset of features from the NSL-KDD data set. The attribute evaluation methods used to select an optimal feature subset from NSL-KDD dataset is the wrapperSubsetEval and the induction algorithm for evaluation of the features is Bayes Net using genticSearch which is a search method for attribute selection. Subset evaluators take a subset of attributes and return a numeric measure that guides the search.

The most relevant list of feature subsets which are effective for classification of attack types in the NSL-KDD is not available to compare the result of the experiments in this research. And hence the result of the experiments in this research was evaluated using the different evaluation metrics mentioned in chapter three. According to the results found in the wrapper based feature selection algorithm is found to be effective in selecting relevant and optimal features from a data set for a particular classifying algorithm. And hence in this research a promising result was conducted in using Bayes Net as an induction algorithm in the feature selection process and classifying algorithm in building a prediction model.

Even if the time complexity of the wrapper feature selection approach seem to be impractical to use the approach for high dimensional data, the effectiveness of the method for building a model for crucial systems like maintaining a network system secured is inevitable. And this fact has been proved in this research which shows that the features selected using wrapper based approach provided the highest accuracy for attacks classification using Bayes net algorithm. The computational cost the approach has incurred may need to be approached with techniques which may reduce the high computational costs.

5.2 Recommendation

- Experiments need to be done by applying different classification algorithms other than the one used in this research and use them as induction algorithm for

evaluating the classification power of the induction algorithms in selecting an optimal feature subset for classification of attack types.

- Research needs to be done on how to integrate the result found from an effective data mining algorithms on real network security systems and find out their effectiveness on a real time network system.
- The feature subset selection research done in this paper is for supervised learning algorithm the problem of a large number of features also affects the performance of unsupervised learning algorithms (Talavera, 2005) and hence research needs to be done for unsupervised learning algorithms using wrapper based feature selection.
- The time complexity of feature selection process using wrapper method is significant. Finding a technique for reducing this problem is mandatory by using techniques like reducing stopping criterions or by experimenting other combination of searching and evaluating algorithms which may reduce the time complexity of the method is needed. Using algorithms which may lower the computational costs of the wrapper feature selection algorithms may help for identifying a better classification algorithm for building a model for IDS.

References

1. Adetunmbi A.Olusola., A. S. (2010). Analysis of KDD ’99 Intrusion Detection Dataset for Selection of Relevance Features .
2. Ali Ghorbani, W. L.(2010.). Network intrusion detection and prevention: concepts and techniques.
3. Andre Gregio, R. S. (2007). Evaluation of data mining techniques for suspicious.
4. Anita R. Zope, P. D. (2013). Data Mining approach in network security.
5. Asha, G. K., Jayaram, M., & Manjunath, A. S. (2010). Feature Subset Selection Problem Using Wrapper Appoach in Supervised Learning. International Journal of Computer Applications .
6. Asyiqin, D., & Mohd Hanapi, Z. (2012). HYBRID OF FUZZY CLUSTERING NEURAL NETWORK OVER NSL DATASETFOR INTRUSION DETECTION SYSTEM. Journal of Computer Science .
7. Bhargava, N., Sharma, G., Bhargava, R., & Mathuria, M. (2013). Decision Tree Analysis on J48 algorithm for data mining. *International journal of advanced research in computer science and software engineering* , 1114-1119.
8. Chae, H.-S., Jo, B.-o., Choi, S.-H., & Park, T.-k. (2013). Feature Selection for Intrusion Detection using NSL-KDD.
9. Choi, H. A. (2008). Application of data mining to network intrusion detection:classifier selection model.
10. Dagne, T. (2012.). Constructing predictive model for network intrusion detection ..
11. Eric Bloedorn, A. D. (2005). Data mining for network intrusion detection: how to get started.
12. Eric Cole, R. L. (2005). Network Security Bible.
13. Faltin, F., Kenett, R., & Ruggeri, F. (2007). Bayesian Networks.
14. Farooq Anjum, D. S. (2008). Signature based intrusion detection for wireless Ad-Hoc Networks: A comparative study of various routing protocols.
15. Fayyad, U. M.-S. (1996). The KDD process for extracting useful knowledge from volumes of data .
16. Frank, I. H. (2005). Data mining: practical machine learning tools and techniques 2nd edition .
17. Gabrielle, Massimiliano & Rocco (2014), Search based software maintenance: Methods &tools.

18. Garrney, J. E., & Jacob, U. W. (2001). Evaluation of Intrusion Detectors: A Decision Theory approach. IEEE .

19. Hall, M. A. (1999). Correlation-based feature selection for Machine learinig .

20. Jian Pei, S. J. (2003.). Data Mining for intrusion detection techniques, applications and systems.

21. JInsong Leng, C. V. (2010). A wrapper based feature selection for Analysis of large data sets .

22. Julish, K. (2005). Data mining for Intrusion detection .

23. Kamber, J. H. (2012). Data mining concepts and techniques.

24. King, B. (1997). Step-Wise Clustering Procedures. . Journal of the American Statistical Association .

25. L. Portnoy, E. E. (November, 2001). Intrusion detection with unlabeled data using clustering,. Proceedings of ACM CSS Workshop on Data Mining Applied to Security,

26. Lee, W. &. (1998). Data mining approaches for intrusion detection.

27. Lee, W. &. (2000). A framework for constructing features and models for intrusion detection systems.

28. Liu, b. L. (2004). Efficient feature selection via analysis of relevance and redundancy

29. Liu, H., & Steiono, R. (1998). Scalable feature selection for large sized databases.

30. Lutu, P. E. (2010, September). Data set selection for aggergate model implementation in predictive data mining. Pretoria, South Africa.

31. M, D. K., & Solomon, M. G. (2012). *Fundamentals of Informarion Security.*

32. Mahobod Tavallaee, E. B. (2009). A detailed analysis of the KDD CUP 99 Data set. *IEEE* .

33. Mossie, Z. (2011). Optimal Feature selection for network intrusion detection: a data mining approach .

34. Mukherjee, B., Haberlein, L., & Levitt, K. (1994). Network intrusion Detection.

35. O, A., & S, B. (2008). *Intrusion detection in computer networks based on machinelearning algorithms.*

36. okach, L. (2014). Data mining with decision trees: Theory and application .

37. P. Yang, B. Z. (2010). A Multi-filter Enhanced Genetic Ensemble System for Gene Selection and Sample Classification of Microarray Data. BMC Bioinformatics.

38. P.Mitra, C. a. (2002). Unsupervised Feature Selection Using Feature Similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence .

39. R. R. (2008). WEKA Manual.

40. Ron Kohavi, G. H. (1996). Wrappers for feature subset selection, . Elsevier .

41. Sajja, P. &. (2010). Knowledge based system for development .

42. Sklansky., M. K. (2000). Comparison of Algorithms that Select Features for Pattern Classifiers. Pattern Recognition,.

43. Soman, K. P., Diwakar, S., & Jay, V. (2006). Data mining : Theory and practice.

44. Steven Noel, D. W. (2002). Modern intrusion detection , data mining and degrees of attack guilt .

45. T.s Chou, K. Y. (2008). Network intrusion detection design using feature selection of soft computing paradigms . International Journal of information and mathematical sciences .

46. Talavera, L. (2005). An evaluation of filter and wrapper methods for feature selection in categorical clustering.

47. Y. Saeys, I. I. (2007). A Review of Feature Selection Techniques in Bioinformatics. .

48. Yang, C. H., Chuang, L. Y., & Yang, C. H. (2009). IG-GA: A Hybrid fitler/wrapper method for featuer selection of microarray data. *Journal of Medical and Biogogical Engineering* , 23-28.

49. Yong Shi, w. x. (2004). Data mining and Knowledge management.

50. Yu, H. L. (2004). efficient feature selection via analysis of relevance and redundancy.

51. Yu, L., & Liu, H. (2003). Feature Selection for High-Dimensional Data:A Fast Correlation-Based Filter Solution.

APPENDICES

Appendix A Features of NSL-KDD Data set

feature name	description	type
duration	length (number of seconds) of the connection	continuous
protocol_type	type of the protocol, e.g. tcp, udp, etc.	discrete
service	network service on the destination, e.g., http, telnet, etc.	discrete
src_bytes	number of data bytes from source to destination	continuous
dst_bytes	number of data bytes from destination to source	continuous
flag	normal or error status of the connection	discrete
land	1 if connection is from/to the same host/port; 0 otherwise	discrete
wrong_fragment	number of "wrong" fragments	continuous
urgent	number of urgent packets	continuous

feature name	description	type
hot	number of "hot" indicators	continuous
num_failed_logins	number of failed login attempts	continuous
logged_in	1 if successfully logged in; 0 otherwise	discrete
num_compromised	number of "compromised" conditions	continuous
root_shell	1 if root shell is obtained; 0 otherwise	discrete
su_attempted	1 if "su root" command attempted; 0 otherwise	discrete
num_root	number of "root" accesses	continuous
num_file_creations	number of file creation operations	continuous
num_shells	number of shell prompts	continuous
num_access_files	number of operations on access control files	continuous
num_outbound_cmds	number of outbound commands in an ftp session	continuous
is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise	discrete
is_guest_login	1 if the login is a "guest"login; 0 otherwise	discrete

feature name	**description>**	**type**
count	number of connections to the same host as the current connection in the past two seconds	continuous
Note: The following features refer to these same-host connections.		
serror_rate	% of connections that have "SYN" errors	continuous
rerror_rate	% of connections that have "REJ" errors	continuous
same_srv_rate	% of connections to the same service	continuous
diff_srv_rate	% of connections to different services	continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	continuous
Note: The following features refer to these same-service connections.		
srv_serror_rate	% of connections that have "SYN" errors	continuous
srv_rerror_rate	% of connections that have "REJ" errors	continuous
srv_diff_host_rate	% of connections to different hosts	continuous

Appendix B

Bayesian network classification output on wrapper based feature selection

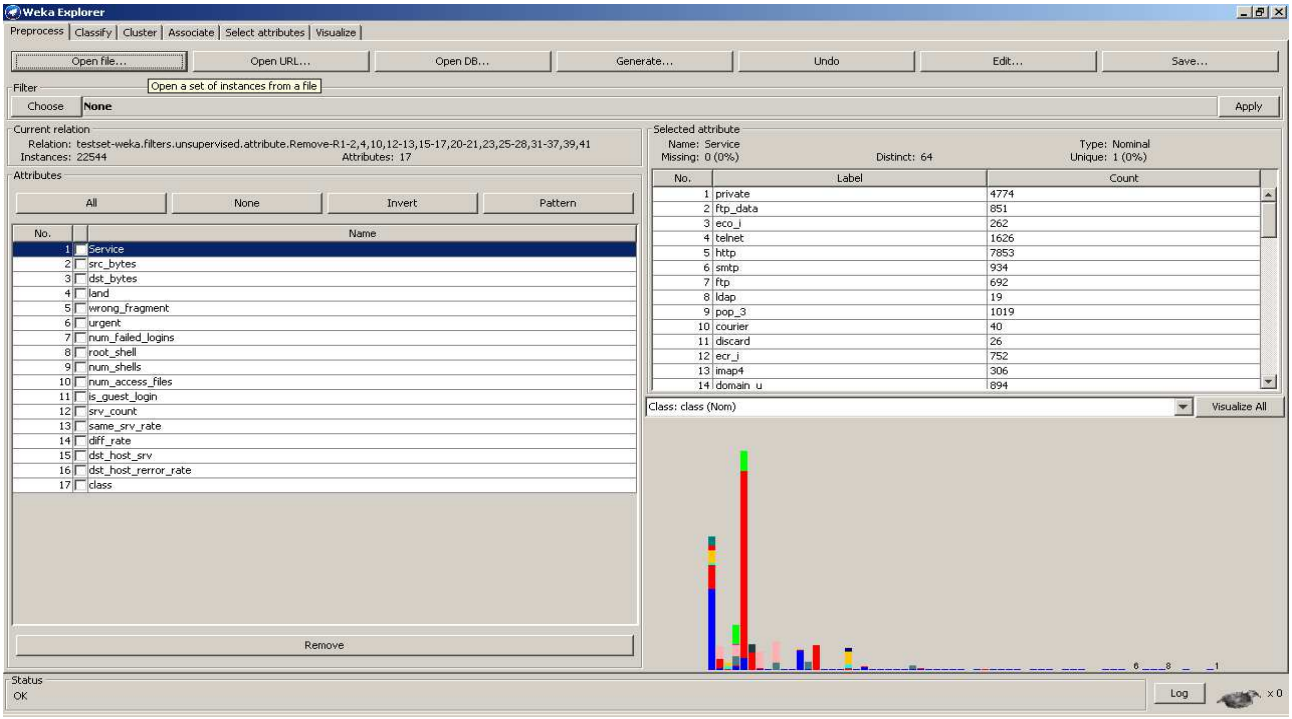
```
Time taken to build model: 0.39 seconds

=== Evaluation on test set ===
=== Summary ===

Correctly Classified Instances       21452               95.1561 %
Incorrectly Classified Instances      1092                4.8439 %
Kappa statistic                          0.9348
Mean absolute error                      0.0048
Root mean squared error                  0.0581
Relative absolute error                  7.2622 %
Root relative squared error             31.8489 %
Total Number of Instances            22544

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.989     0.007      0.98      0.989     0.984      1         DoS
                 0.933     0.008      0.989     0.933     0.96       0.995     normal
                 0.799     0.006      0.673     0.799     0.731      0.995     saint
                 0.993     0.001      0.984     0.993     0.989      1         mscan
                 0.988     0.005      0.954     0.988     0.971      1         R2L
                 0.991     0.001      0.959     0.991     0.975      1         apache2
                 0.822     0.011      0.791     0.822     0.806      0.995     PROBE
                 0.649     0.001      0.421     0.649     0.511      0.995     U2R
                 1         0.008      0.486     1         0.654      0.997     snmpgetattack
                 0.991     0          0.99      0.991     0.991      1         processtable
                 0.94      0.002      0.735     0.94      0.825      0.998     httptunnel
                 0.2       0          0.333     0.2       0.25       0.997     ps
                 0.991     0.003      0.832     0.991     0.905      1         snmpguess
                 1         0          0.997     1         0.998      1         mailbomb
                 0.471     0          0.571     0.471     0.516      0.995     named
                 0.857     0          0.6       0.857     0.706      1         sendmail
                 0.308     0          0.8       0.308     0.444      0.996     xterm
                 0         0          0         0         0          0.999     worm
                 0.556     0          0.625     0.556     0.588      1         xlock
                 0         0          0         0         0          0.997     xsnoop
                 0         0          0         0         0          1         sqlattack
                 0         0          0         0         0          0.9       udpstorm
Weighted Avg.    0.952     0.006      0.958     0.952     0.953      0.998
```

Appendix C

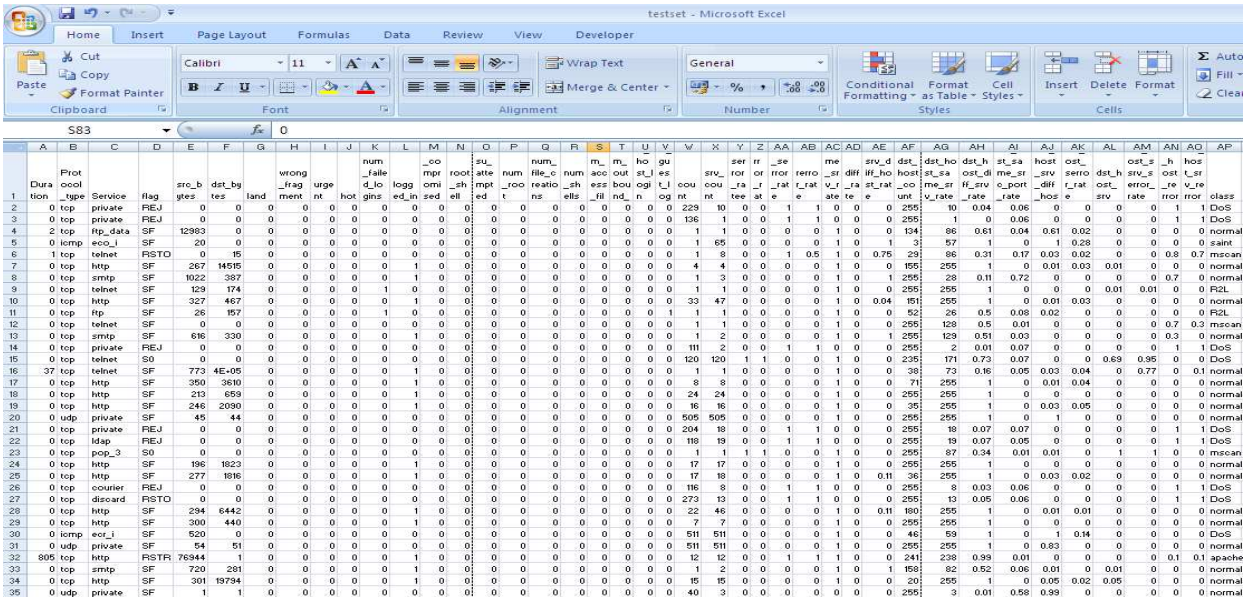
Bayesian network classification result on filter based feature selection

```
Relative absolute error                 8.6311 %
Root relative squared error            35.1206 %
Total Number of Instances           22544

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.967     0.004      0.989     0.967     0.978      0.999    DoS
                 0.922     0.013      0.981     0.922     0.951      0.992    normal
                 0.809     0.006      0.674     0.809     0.735      0.994    saint
                 0.984     0.001      0.97      0.984     0.977      1        mscan
                 0.987     0.008      0.929     0.987     0.957      1        R2L
                 0.996     0.002      0.957     0.996     0.976      1        apache2
                 0.859     0.009      0.825     0.859     0.841      0.996    PROBE
                 0.676     0.003      0.266     0.676     0.382      0.991    U2R
                 1         0.008      0.493     1         0.66       0.997    snmpgetattack
                 0.996     0          0.991     0.996     0.993      1        processtable
                 0.962     0.002      0.762     0.962     0.85       0.999    httptunnel
                 0.4       0          0.375     0.4       0.387      0.996    ps
                 0.997     0.005      0.752     0.997     0.857      1        snmpguess
                 1         0          0.987     1         0.993      1        mailbomb
                 0.529     0.001      0.29      0.529     0.375      0.998    named
                 0.714     0.001      0.294     0.714     0.417      0.999    sendmail
                 0.462     0          1         0.462     0.632      0.999    xterm
                 0         0          0         0         0          1        worm
                 0.556     0          0.556     0.556     0.556      0.999    xlock
                 0.25      0          0.333     0.25      0.286      0.985    xsnoop
                 0         0          0         0         0          0.999    sqlattack
                 0         0          0         0         0          0.969    udpstorm
Weighted Avg.    0.944     0.008      0.953     0.944     0.947      0.996
```

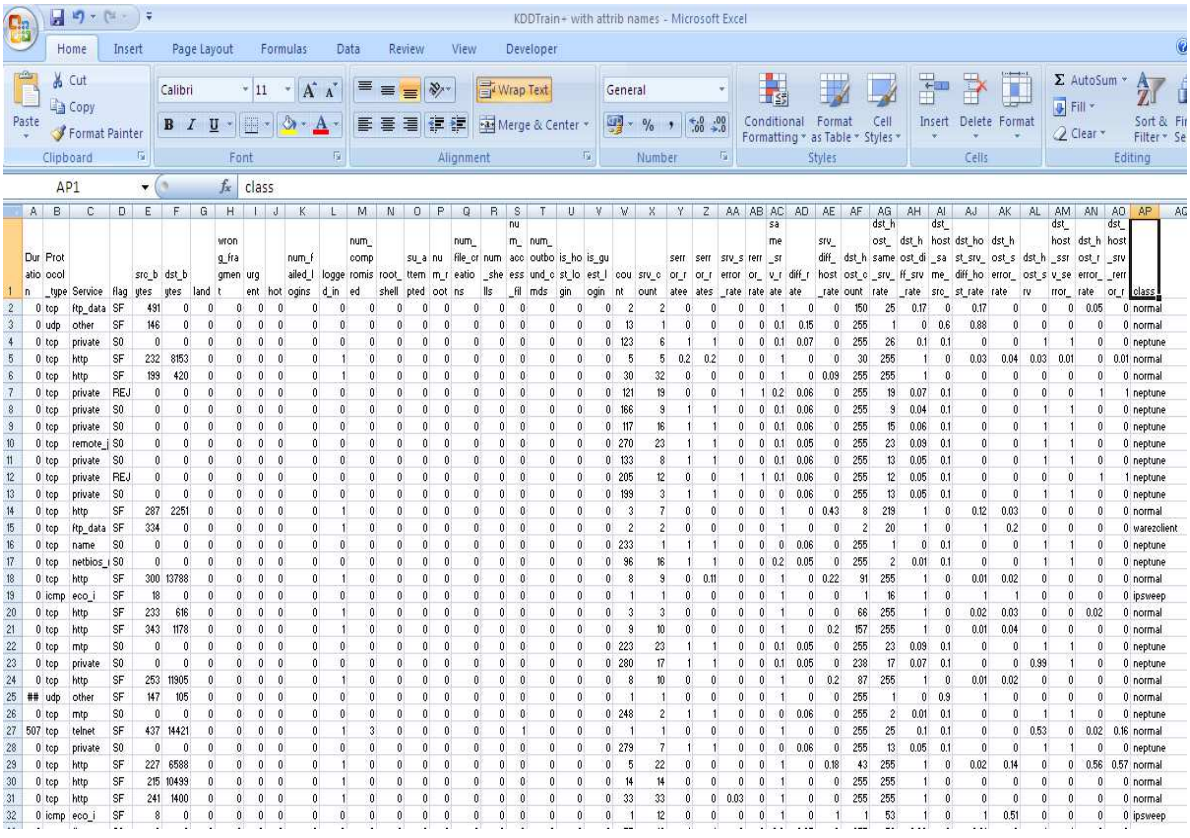
Appendix D Testing data with selected features using wrapper based feature selection with Bayes-net



Appendix E Sample data set of the NSL-KDD training data set



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP
1	Duration	Protocol_type	Service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_shell	su_attempted	num_root	num_file_creations	num_shells	num_access_files	num_outbound_cmds	is_host_login	is_guest_login	count	srv_count	serror_rate	srv_serror_rate	rerror_rate	srv_rerror_rate	same_srv_rate	diff_srv_rate	srv_diff_host_rate	dst_host_count	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	class
2	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	229	10	0	0	1	1	0	0	0	255	10	0.04	0.06	0	0	0	0	1	1	DoS
3	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	136	1	0	0	1	1	0	0	0	255	1	0	0.06	0	0	0	0	1	1	DoS
4	2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	134	86	0.61	0.04	0.61	0.02	0	0	0	0	normal
5	0	icmp	eco_i	SF	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	65	0	0	0	0	1	0	0	1	3	57	1	0	1	0.28	0	0	0	0	saint
6	1	tcp	telnet	RSTO	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	8	0	0	1	0.5	1	0	0.75	29	86	0.31	0.17	0.03	0.02	0	0	0.8	0.7	mscan
7	0	tcp	http	SF	267	14515	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	1	0	0	155	255	1	0	0.01	0.03	0.01	0	0	0	normal
8	0	tcp	smtp	SF	1022	387	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	1	0	1	255	28	0.11	0.72	0	0	0	0	0.7	0	normal
9	0	tcp	telnet	SF	129	174	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	255	1	0	0	0	0.01	0.01	0	0	R2L
10	0	tcp	http	SF	327	467	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	33	47	0	0	0	0	1	0	0.04	151	255	1	0	0.01	0.03	0	0	0	0	normal
11	0	tcp	ftp	SF	26	157	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	52	26	0.5	0.08	0.02	0	0	0	0	0	R2L
12	0	tcp	telnet	SF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	128	0.5	0.01	0	0	0	0	0.7	0.3	mscan
13	0	tcp	smtp	SF	616	330	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	1	0	1	255	129	0.51	0.03	0	0	0	0	0.3	0	normal
14	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	111	2	0	0	1	1	0	0	0	255	2	0.01	0.07	0	0	0	0	1	1	DoS
15	0	tcp	telnet	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	120	1	1	0	0	1	0	0	235	171	0.73	0.07	0	0	0.69	0.95	0	0	DoS
16	37	tcp	telnet	SF	773	4E+05	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	38	73	0.16	0.05	0.03	0.04	0	0.77	0	0.1	normal
17	0	tcp	http	SF	350	3610	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	8	8	0	0	0	0	1	0	0	71	255	1	0	0.01	0.04	0	0	0	0	normal
18	0	tcp	http	SF	213	659	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	24	24	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	0	normal
19	0	tcp	http	SF	246	2090	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	16	16	0	0	0	0	1	0	0	35	255	1	0	0.03	0.05	0	0	0	0	normal
20	0	udp	private	SF	45	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	505	505	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	0	normal
21	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	204	18	0	0	0	1	1	0	0	255	18	0.07	0.07	0	0	0	0	1	1	DoS
22	0	tcp	ldap	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	118	19	0	0	1	1	0	0	0	255	19	0.07	0.05	0	0	0	0	1	1	DoS
23	0	tcp	pop_3	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	255	87	0.34	0.01	0.01	0	1	1	0	0	mscan
24	0	tcp	http	SF	196	1823	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	17	17	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	0	normal
25	0	tcp	http	SF	277	1816	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	17	18	0	0	0	0	1	0	0.11	36	255	1	0	0.03	0.02	0	0	0	0	normal
26	0	tcp	courier	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	116	8	0	0	1	1	0	0	0	255	8	0.03	0.06	0	0	0	0	1	1	DoS
27	0	tcp	discard	RSTO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	273	13	0	0	1	1	0	0	0	255	13	0.05	0.06	0	0	0	0	1	1	DoS
28	0	tcp	http	SF	294	6442	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	22	46	0	0	0	0	1	0	0.11	180	255	1	0	0.01	0.01	0	0	0	0	normal
29	0	tcp	http	SF	300	440	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	7	7	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	0	normal
30	0	icmp	eco_i	SF	520	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	511	511	0	0	0	0	1	0	0	46	59	1	0	1	0.14	0	0	0	0	DoS
31	0	udp	private	SF	54	51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	511	511	0	0	0	0	1	0	0	255	255	1	0	0.83	0	0	0	0	0	normal
32	805	tcp	http	RSTR	76944	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	12	12	0	0	1	1	1	0	0	241	238	0.99	0.01	0	0	0	0	0.1	0.1	apache
33	0	tcp	smtp	SF	720	281	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	1	0	0	1	82	0.52	0.06	0.01	0	0.01	0	0	0	normal
34	0	tcp	http	SF	301	19794	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	15	15	0	0	0	0	1	0	0	20	255	1	0	0.05	0.02	0.05	0	0	0	normal
35	0	udp	private	SF	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	3	0	0	0	0	0	0	0	255	3	0.01	0.58	0.99	0	0	0	0	0	normal



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ
1	Duration	Protocol_type	Service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_shell	su_attempted	num_root	num_file_creations	num_shells	num_access_files	num_outbound_cmds	is_host_login	is_guest_login	count	srv_count	serror_rate	srv_serror_rate	rerror_rate	srv_rerror_rate	same_srv_rate	diff_srv_rate	srv_diff_host_rate	dst_host_count	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	class	
2	0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	150	25	0.17	0	0.17	0	0	0	0.05	0	normal	
3	0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	1	0	0	0	0	0.1	0.15	0	255	1	0	0.6	0.88	0	0	0	0	0	normal	
4	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	6	1	1	0	0	0.1	0.07	0	255	26	0.1	0.1	0	0	1	1	0	0	neptune	
5	0	tcp	http	SF	232	8153	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	5	5	0.2	0.2	0	0	1	0	0	30	255	1	0	0.03	0.04	0.03	0.01	0	0.01	normal	
6	0	tcp	http	SF	199	420	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	30	32	0	0	0	0	1	0	0.09	255	255	1	0	0	0	0	0	0	0	normal	
7	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	121	19	0	0	1	1	0.2	0.06	0	255	19	0.07	0.1	0	0	0	0	1	1	neptune	
8	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	9	1	1	0	0	0.1	0.06	0	255	9	0.04	0.1	0	0	1	1	0	0	neptune	
9	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	117	16	1	1	0	0	0.1	0.06	0	255	15	0.06	0.1	0	0	1	1	0	0	neptune	
10	0	tcp	remote_j	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	270	23	1	1	0	0	0.1	0.05	0	255	23	0.09	0.1	0	0	1	1	0	0	neptune	
11	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133	8	1	1	0	0	0.1	0.06	0	255	13	0.05	0.1	0	0	1	1	0	0	neptune	
12	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	205	12	0	0	1	1	0.1	0.06	0	255	12	0.05	0.1	0	0	0	0	1	1	neptune	
13	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	3	1	1	0	0	0	0.06	0	255	13	0.05	0.1	0	0	1	1	0	0	neptune	
14	0	tcp	http	SF	287	2251	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	7	0	0	0	0	1	0	0.43	8	219	1	0	0.12	0.03	0	0	0	0	normal	
15	0	tcp	ftp_data	SF	334	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	1	0	0	2	20	1	0	1	0.2	0	0	0	0	warezclient	
16	0	tcp	name	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	233	1	1	1	0	0	0	0.06	0	255	1	0	0.1	0	0	1	1	0	0	neptune	
17	0	tcp	netbios_	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96	16	1	1	0	0	0.2	0.05	0	255	2	0.01	0.1	0	0	1	1	0	0	neptune	
18	0	tcp	http	SF	300	13788	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	8	9	0	0.11	0	0	1	0	0.22	91	255	1	0	0.01	0.02	0	0	0	0	normal	
19	0	icmp	eco_i	SF	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	16	1	0	1	0	0	0	0	0	ipsweep	
20	0	tcp	http	SF	233	616	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	1	0	0	66	255	1	0	0.02	0.03	0	0	0.02	0	normal	
21	0	tcp	http	SF	343	1178	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	9	10	0	0	0	0	1	0	0.2	157	255	1	0	0.01	0.04	0	0	0	0	normal	
22	0	tcp	mtp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	223	23	1	1	0	0	0.1	0.05	0	255	23	0.09	0.1	0	0	1	1	0	0	neptune	
23	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	280	17	1	1	0	0	0.1	0.05	0	238	17	0.07	0.1	0	0	0.99	1	0	0	neptune	
24	0	tcp	http	SF	253	11905	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	8	10	0	0	0	0	1	0	0.2	87	255	1	0	0.01	0.02	0	0	0	0	normal	
25	##	udp	other	SF	147	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	1	0	0.9	1	0	0	0	0	0	normal	
26	0	tcp	mtp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	248	2	1	1	0	0	0	0.06	0	255	2	0.01	0.1	0	0	1	1	0	0	neptune	
27	507	tcp	telnet	SF	437	14421	0	0	0	0	1	3	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	0	0	255	25	0.1	0.1	0	0	0.53	0	0.02	0.16	normal	
28	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	279	7	1	1	0	0	0	0.06	0	255	13	0.05	0.1	0	0	1	1	0	0	neptune	
29	0	tcp	http	SF	227	6588	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	5	22	0	0	0	0	1	0	0.18	43	255	1	0	0.02	0.14	0	0	0.56	0.57	normal	
30	0	tcp	http	SF	215	10439	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	14	14	0	0	0	0	1	0	0	255	255	1	0	0	0	0	0	0	0	normal	
31	0	tcp	http	SF	241	1400	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	33	33	0	0	0.03	0	1	0	0	255	255	1	0	0	0	0	0	0	0	normal	
32	0	icmp	eco_i	SF	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	12	0	0	0	0	1	0	1	1	53	1	0	1	0.51	0	0	0	0	ipsweep	