COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

Selection of Data Mining Algorithm for Masked Feature Network Intrusion Detection on Real World Data with Missing Value: The case of Ethiopian Institutes of Agricultural Research

BY

Kassahun Admkie Tekle

January, 2018

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

A Thesis Submitted to

The School of Information Science of Addis Ababa University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Science

BY

Kassahun Admkie Tekle

January, 2018

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

Selection of Data Mining Algorithm for Masked Feature Network Intrusion Detection on Real World Data with Missing Value: The case of Ethiopian Institutes of Agricultural Research

BY

Kassahun Admkie Tekle

Name and Signature of Members of the Examining Board

Name	Title	Signature	Date
_____	**Chairperson**	_____	_____
Dr. Workshet Lamenew	**Advisor**	_____	_____
Dr. Tibebe Beshah	**Examiner**	_____	_____
Dr. Gashaw Kebede	**Examiner**	_____	_____

DECLARATION

I, the undersigned, declare that this thesis is my original work and has not been presented as a partial degree requirement for a degree in any other university and that all sources used for the thesis have been duly acknowledged.

Kassahun Admkie Tekle

2018

The thesis has been submitted for examination with my approval as University Advisor

---

Dr. Workshet Lamenew

2018

DEDICATION

It is a lifetime opportunity for me to dedicate this thesis work to my father Ato Admkie Tekle, my mother W/r Gete Kebede and my family for their love of education, guidance, motivation and scarification for me to reach at this level. Dear Father, may God rest your soul in peace. Your love and motivation is always with me and I would be very happy, if you were alive and harvest your fruit as you like. Dear Mother, I know all the ups and downs you passed through, your pray and strength makes me strong always. Long live for you! Dear my proud family, I have no words to say about your effort for me. You are my engine for motivation. It is a great opportunity for me to have such a blessed family. Thanks God and stay blessed!

++

ACKNOWLEDGMENT

First and foremost extraordinary thanks go for my Almighty God and His Mother Saint-Merry. It is with immense gratitude that I acknowledge the support and help of my advisor, Dr. Workshet Lamenew. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. He taught me how to question thoughts and express ideas. His patience, advises and support helped me overcome challenging situations and finish this thesis, plus his comments added great value for my future career.

I would like to acknowledge Dr. Tibebe Beshah, who taught me Data and Web Mining course, together with Dr. Workshet Lamenew who taught me Design and Administration of Systems & Networks. It is their methodology and approach initiated me and showed the direction of my thesis during their lecture hours.

Special thanks to all staff member of School of Information Science for their contribution in the special arena for the success of my study.

Most importantly, none of these would have been possible without the love and patience of my family. My father and my mother, to whom this thesis is dedicated to, have been a constant source of love, concern, support and strength all the past years. In addition, I would like to express my heart-felt gratitude to my wife.

Finally, I share the credit of my work for the rest of my brother, my son, my daughter, friends, ICT stuff of EIAR and DZARC, stuff of Debre Zeit Agricultural Research Center and Ethiopian Institute of Agricultural Research that I have not mentioned their name here.

++

“ወለዘሂ ፍርሃተ ፍርሁ ፣ ወለዘሂ ክብረ አክብሩ።።” ሮሜ ፲፫፣7

“መፈራት ለሚገባው መፈራትን ፣ ክብር ለሚገባው ክብርን ስጡ።።” ሮሜ ፲፯.፣7

“Fear to whom fear; honor to whom honor.” Romans 13፣7

++

Table Of Content

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMES

APEC	Asia-Pacific Economic cooperation
ARFF:	Attribute Relation File Format
CFS:	Correlation Feature Selection
CRISP-DM:	Cross Industry Standard Process for Data Mining
CSV:	Comma Separated Values
DARPA:	Defense Advanced research Project Agency
DM:	Data Mining
DOS:	Denial of Services
EIAR:	Ethiopian Institute of Agricultural Research
FP:	False Positive
Http:	hypertext transfer protocol
ICMP:	Internet control message protocol
ICT:	Information Communication Technology
IDPS:	Intrusion Detection Prevention Systems
IDS:	Intrusion Detection System
IGR:	Information Gain Ratio
IPS:	Intrusion Prevention Systems
KD:	Knowledge Discovery

KDD:	Knowledge Discovery in Database
MIT:	Massachusetts Institute of Technology
NBA:	Network Behavior Analysis
NIDS:	Network intrusion detection system
NSL:	Network Simulation Language
num_failed_logins:	number of failed logins
R2L:	Remote to Local
SEMMA:	Sample, Explore, Modify, Model, and Assess
SF:	Source Flag
SSL:	Semi-Supervised Learning
TCP:	Transmission Control Protocol
TP:	True Positive
U2R:	User to Remote
UDP:	User Datagram protocol
VPN:	Virtual Private Network
WEKA:	Waikato Environment for Knowledge Analysis
YATSI:	Yet Another Two Stage Idea

ABSTRACT

Intrusion detection has become a critical component of network administration due to the vast number of attacks persistently threating our computer system. As network attacks have increased in number and severity over the past few years, intrusion detection system (IDS) is increasingly becoming a critical component to secure the network. Due to large volumes of security audit data as well as complex and dynamic properties of intrusion behaviors, optimizing performance of IDS becomes an important open problem that is receiving more and more attention from the research community. Recently there has been much interest in applying data mining to computer network intrusion detection. Many methods have been developed to secure computer networks and communication over the Internet. However, none of the existing methods developed by different researches have an accuracy of detecting attacks with high detection rate and low false alarm rate. Moreover intruders can also chat the system by masking their some features to attack the system. The other thing is most deal with single detection approach with high number of features which is challenging and time consuming to implement.

This thesis work is devoted to solve those problems of Ethiopian Institute of Agricultural Research (EIAR) using intrusion detection system architecture that is based on semi-supervised collective classification algorithm of meta.Filtered Collective Classifier that can promptly detect and classify attacks, whether they are known or never seen before, even they mask their some features by using missing value dataset.

The data set in this study is taken from EIAR data center. After taking the data, it has been preprocessed. In the preprocessing activities, removing outliers and resolving inconsistencies tasks are taken place. The researcher has taken the dataset initially had 25192 records but after the preprocessing stage, it was reduced to 28 attributes and 12596 records which are labeled as Normal, DOS, U2R, Probe and R2L. For supervised modeling, the 6965 records are taken. For building a predictive model for intrusion detection semi-supervised collective classification meta.Filtered Collective Classifier and ordinary J48 decision tree algorithms have been tested as a classification approach by using unlabeled class with missing value and with no missing value dataset.

The model that was created using the Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test set showed the best classification accuracy of 96.2% by using the first dataset, with missing value and the ordinary J48 tree with its default 10-fold cross validation showed better performance of 100% accuracy by using the second dataset, with no missing value to classify the new instances as Normal, DOS, U2R, Probe and R2L classes. The findings of this study have shown that the data mining methods generates interesting rules that are crucial for intrusion detection in the networking industry. Future research directions are forwarded to come up an applicable system in the area of the study.

Keywords: - Intrusion detection, Data mining, Semi-Supervise Learning, Collective classifier, Missing value dataset, Masked feature Intrusion detection.

CHAPTER ONE

INTRODUCTION

1.1. BACKGROUND OF THE STUDY

An intrusion is a type of attack on information assets in which the instigator attempts to gain entry into a system or disrupt the normal operations of a system [1]. According to APEC [1], intrusions are actions that attempt to bypass security mechanisms of computer systems.

According to Marinova-Boncheva [2], some specific examples of intrusions that concern system administrators include:

- Unauthorized modifications of system information in network components (e.g. modifications of router tables in an internet to deny use of the network);
- Unauthorized modifications of system files so as to facilitate illegal access to either system or user information;
- Unauthorized access or modification of user files or information;
- Unauthorized use of computing resources (perhaps through the creation of unauthorized accounts or perhaps through the unauthorized use of existing accounts).

Among the varieties of security measures one is IDS/IPS. Intrusion detection starts with instrumentation of a computer network for data collection. Pattern-based software 'sensors' monitor the network traffic and raise 'alarms' when the traffic matches a saved pattern. Security analysts decide whether these alarms indicate an event serious enough to warrant a response. A response might be to shut down a part of the network, to phone the internet service provider associated with suspicious traffic, or to simply make note of unusual traffic for future reference [3].

An extensive growth in using Internet in social networking (e.g., instant messaging, video conferences, etc.), health care, e-commerce, bank transactions, and many other services are being witnessed nowadays. These Internet applications need a satisfactory level of security and privacy. On the other hand, our computers are under attacks and vulnerable to many threats. There is an increasing availability of tools and tricks for attacking and intruding networks. An intrusion can

be defined as any set of actions that threaten the security requirements (e.g., integrity, confidentiality, availability) of a computer/network resource (e.g., user accounts, file systems, and system kernels) [4].

In the era of information society, computer networks and their related applications are becoming more and more popular, so does the potential thread to the global information infrastructure. To defend against various cyber-attacks and computer viruses, lots of computer security techniques have been intensively studied in the last decade, namely cryptography, firewalls and anomaly intrusion detection. Among them, Data mining network intrusion detection (NID) has been considered to be one of the most promising methods for defending complex and dynamic intrusion behaviors. [5].

Intrusion detection includes identifying a set of malicious actions that compromise the integrity, confidentiality, and availability of information resources. Traditional methods for intrusion detection are based on extensive knowledge of signatures of known attacks. Monitored events are matched against the signatures to detect intrusions. These methods extract features from various audit streams, and detect intrusions by comparing the feature values to a set of attack signatures provided by human experts. The signature database has to be manually revised for each new type of intrusion that is discovered. A significant limitation of signature-based methods is that they cannot detect emerging cyber threats, since by their very nature these threats are launched using previously unknown attacks. In addition, even if a new attack is discovered and its signature developed, often there is a substantial latency in its deployment across networks. These limitations have led to an increasing interest in intrusion detection techniques based upon data mining [6].

Data mining is the process of discovering interesting patterns (or knowledge) from large amounts of data. The data sources can include databases, data warehouses, the Web, any other information repositories or data that are streamed into the system. Data mining is also called KDD (Knowledge Discovery in Databases). The goal of data mining process is to extract information from the dataset and it is changed into an understandable structure.

Data mining techniques can be used for misuse and anomaly intrusion detection. Misuse refers to known attacks and harmful activities that exploit the known sensitivities of the system. In misuse detection, each instance in a data set is labeled as ‘normal’ or ‘intrusion’ and a learning algorithm is trained over the labeled data. Anomaly means unusual activity in general that could indicate an

intrusion. An advantage of misuse detection techniques is their high degree of accuracy in detecting known attacks and their variation. [6]

We consider the general problem of learning from labeled and unlabeled data, which is often called semi-supervised learning or transductive inference [39]. The data set in this study is taken from Ethiopian Institute of Agriculture (EIAR) data center.

More than 80% of Ethiopian economy depends on Agriculture and Agricultural research takes a vital role to support the sector by releasing new technologies including improved varieties of crop, improved livestock and ideas etc.

The EIAR is one of the oldest and largest agricultural research system in Africa. Ethiopian Agricultural Research System (EARS) has evolved through several stages since its first initiation during the late 1940s, following the establishment of agricultural and technical schools at Ambo and Jimma. In 1955, a full-fledged agricultural experiment station was established at Debre Zeit (now named Debre Zeit Agricultural Research Center) under the then Imperial College of Agricultural and mechanical Arts (now called Haramaya University) and had been continued as the major research entity until the mid-1960s. In 1966, Institute of Agricultural Research (IAR) was established as the first nationally coordinated agricultural research system in Ethiopia. Institute of Agricultural Research (IAR) was established with a mission to formulate national agricultural research guidelines, coordinate National Agricultural Research System, and undertake research in its centers and sub-centers located in various agro ecological zones of Ethiopia. As per this Proclamation, its objectives are

(1) To generate, develop and adapt agricultural technologies that focus on the needs of the overall agricultural development and its beneficiaries;

(2) To coordinate technically the research activities of Ethiopian Agricultural Research System;

(3) Build up a research capacity and establish a system that will make agricultural research efficient, effective and based on development needs; and

(4) Popularize agricultural research results [7].

This Research institute has different database and warehouse to store huge data. As mentioned above agricultural data are highly linked with life and some of these data are changed to

information and some are not, therefore these data need highly protected network environment, this is the motive of the researcher.

Many researches had been taken concerning data mining techniques for Network Intrusion Detection but specialized, specific and efficient NIDS based on data mining techniques with data having missing value was not identified. In this paper the researcher identified and selected an efficient data mining Model of Network Intrusion Detection System (NIDS) for agricultural researches.

1.2. Statement of the Problem

Intrusion detection problem is becoming a challenging task due to the proliferation 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 [8].

Hence, there is a need of effective and efficient system which allows protecting the network from intruders. To develop such kind of system there is a need to use methods like feature selection which is a growing field of interest about selecting proper features. This is because it is expensive to carry out the entire process and degrades the classification performance of data mining algorithms. Therefore, feature selection approaches reduce the complexity of the overall process by allowing the data mining system to focus on what are really important features [9].

Given the nature of this problem, the possible solution is data mining approach. Data mining approach for intrusion detection techniques generally fall into one of two categories; misuse detection and anomaly detection. In misuse detection, each instance in a data set is labeled as 'normal' or 'intrusion' and a learning algorithm is trained over the labeled data. These techniques are able to automatically retrain intrusion detection models on different input data that include new types of attacks, as long as they have been labeled appropriately [10].

Many researchers proposed different models for network intrusion detection system (NIDS). Adamu[10] has tried to study a machine learning IDS that investigated the application of cost sensitive learning by applying decision tree algorithm.

Zewdie [11] proposed an optimal feature selection for Network Intrusion Detection using indirect cost sensitive feature selection approach. It is a DM approach system that tried to investigate jointly cost sensitive learning and feature selection to advance the classification performance of algorithms that incorporate cost. In his study, Information Gain Ratio (IGR) and Correlation

Feature Selection (CFS) are investigated for ranking and selecting features using the proposed cost sensitive approach. Zewide has tried to investigate decision tree classification algorithms that used indirect cost sensitive feature ranking and selection algorithms. Zewdie used in his study only those records which are labeled. He did not consider those records which are not labeled.

Tigabu [9] has studied on constructing predictive model using a Semi Supervised approach for intrusion detection system that will enhance the network security system.

Tigabu[9], Adamu[10]and Zewdie[11] conducted their researches of the NIDS on more of a supervised approach.

Finally, As Tigabu [9] shown in scope and limitation of his study, the study was conducted on the dataset taken from the Massachusetts Institute of Technology (MIT) Lincoln lab. Therefor he recommended further research conducted on real life dataset from organizations that have their own network by combining the problem domain and the domain expert on the study process. In addition his study was carried out using simple K-means and classification algorithms such as J48 decision tree and Naïve Bayes algorithms. H.A. Nguyen and D. Choi [45] recommended also, however they are fully aware of problems that have been cited with the KDD99 dataset and strongly discourage its further use in developing network intrusion detection data mining algorithms. Moreover, although the real world data is not complete due to different factors and intruders can also chat the system by masking their some features to attack. No research is done on data having missing value to detect those masked features of intrusions.

So, this study attempted to construct a Semi-Supervised meta.Filtered Collective Classifier model by considering both labeled and unlabeled records of the real Agricultural research data and evaluate the performance of different data mining techniques including J48 to fill a gap which was recommended as future research direction by Tigabu [9] , H.A. Nguyen and D. Choi [45]. In addition to fill the gap of the above mentioned researchers, this research has been focused also to address the challenges of the real world missing value dataset and to detect those intrusion masking some features by using a data set with missing value for selecting better performed Data Mining algorithm for NIDS to predict the newly coming intrusion.

Therefore, this research proposed to get answers for the following research questions.

- Which Data Mining algorithm can be more efficient for the purpose of predicting Network Intrusions detection with missing value data set in real world Agricultural Research Institutes?

- Which type of dataset (with missing value/ with no missing value) is better performed for academic and research purpose?
- How can the NIDS classify intrusions correctly, in data with missing value and with no missing value?
- What is the purpose of using missing value dataset for a given networks signal is a normal packet or an intrusion?

1.3. Objective

1.3.1. General Objective

The general objective of this research is to select efficient data mining algorithm of masked network intrusion detection in EIAR for effective defense against network attacks by using dataset with missing value.

1.3.2. Specific Objective

- Evaluate different data mining algorithms for NID by dataset with missing value.
- Understand nature and behaviors of masked NIDS for dataset with missing value.
- Conduct a detail literature review of different researches on data mining algorithms for NIDS.

1.4. Scope and Limitation of the Study

This research is partly the extension of the thesis worked by [9]. As he mentioned the datasets that were taken from the Massachusetts Institute of Technology (MIT) Lincoln lab, not include data from network security organizations in Ethiopia. Therefore this research tried to show and revised Tigabu's results by using the real datasets and select efficient data mining techniques of Network Intrusion Detection and proposed best performed classifier model as compared to existing model for Ethiopian Institutes of Agricultural Research.

Due to limitation of budget and time the research also forced to address Head Office and only few Agricultural Research Centers like Debrezeit and Jima Agricultural Research Center. Therefore, further research have to be conducted including other research centers. This research also accept the results of others related researches and assume an Ordinary J48 algorithms as existed beter performed algorithm to predict whether a new instance is normal or attack. Beside this, because of time limitation this research also focused mainly on how to effectively detect attacks, not to

prevent them. The IDS model constructed in this thesis just notify for the administrators after detecting an attack and administrators have to manually take proper actions.

1.5. Significance of the Study

Network security is an increasing industry as more and more of the corporate workspace is converted to digital media. Because companies and home users keep sensitive information on their computers. Hence agricultural data are more sensitive, very critical problems will arise ones data are lost or distorted, there is a great need to protect that information from those who would exploit it. One way to help keep attackers is by using IDS, which are designed to locate and notify systems administrators about the presence of malicious traffic [12] .

Detecting intrusions allows administrators to identify areas where their defenses need improvement, such as by identifying a previously unknown vulnerability, a system that was not properly patched, or a user that needs further education against social engineering attacks [19]. The research is applicable where; a continuous monitoring of many parameters is conducted to detect malicious applications.

As I have tried to show above many researches had been taken concerning data mining techniques for Network intrusion detection but specific and efficient NIDS based on the organizational network using unlabeled class with missing value of real dataset was not identified for Agricultural researches Institutes. In addition to this an agri-net which helps to link different research centers in the country to share and store agricultural data, was deployed in the EIAR before six years ago but not used effectively due to different factors, it is again demonstrating on different centers. Therefore this research had its own significant importance to Network security part of this project. This research identified and selected the best NIDS by the effectiveness and efficiency using the real data available in the research organization. Though this research is done on the real dataset of EIAR it can also address the problems of masked features network attack of other businesses. Therefore this research will contribute its part for the Agricultural research centers of the country, Ethiopia and other business areas of the world as well.

CHAPTER TWO

LITERATURE REVIEW

This Chapter is the output of literature survey of intrusion detection systems and discuss the proposed algorithm. Mainly there are two approaches to Intrusion Detection that is Misuse detection and Anomaly detection. Both the approaches have advantages and disadvantages. So recently a new approach has come up which combines both the approaches and gives better results.

The dynamic change of the technology and number of hackers and crackers in networking industry enhanced to invent an increasing manner there should be a means to minimize or remove such challenges. Data mining is one of the technologies that is used for intrusion detection and prevention.

In spite of the wide growth of information technology, security has remained one challenging area for computer and networks. The numbers of hacking and intrusion incidents are increasing year on year as technology rolls out [13], [14]

Intrusion detection has improved through time together with the development of technology, but this improvement seems to be a continuous process as advancement in the technology opens the door with a loop-hole for intruders every time.

2.1. Network Intrusion Detection (NID)

With advancements in network technologies, Internet services have been also growing rapidly in network traffic, accompanied by an increasing number of anomalies such as Denial of Service (DoS) attacks, virus exploits, port scans, worms and misconfigurations. These anomalies represent a large fraction of the Internet traffic that is unwanted and prevent legitimate users from accessing network resources in an optimal manner [15].Therefore, detecting and diagnosing these threats are crucial tasks for network operators to ensure that the Internet resources remain available. Because legitimate traffic must be able to travel efficiently, quickly and accurately identifying anomalies in network traffic is important, requires development of good detection techniques. Anomalies are patterns of interest to network defenders, who want to extract them from vast amount of network traffic data.

Network Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent

threats of violation of computer security policies, acceptable use policies, or standard security practices [16], [9] incidents have many causes, such as malware (e.g., worms, spyware), attackers gaining unauthorized access to systems from the Internet, and authorized users of systems who misuse their privileges or attempt to gain additional privileges for which they are not authorized. Although many incidents are malicious in nature, many others are not; for example, a person might mistype the address of a computer and accidentally attempt to connect to a different system without authorization.

As discussed in [17] IDSs focus on identifying possible incidents. For example, an IDS could detect when an attacker has successfully compromised a system by exploiting vulnerability in the system. The IDS could then report the incident to security administrators, who could quickly initiate incident response actions to minimize the damage caused by the incident. The IDS could also log information that could be used by the incident handlers. Many IDSs can also be configured to recognize violations of security policies. For example, some IDSs can be configured with firewall rule like settings, allowing them to identify network traffic that violates the organization's security or acceptable use policies. Also, some IDSs can monitor file transfers and identify ones that might be suspicious, such as copying a large database onto a user's laptop. [23]

2.1.1. Categories of Network Intrusion Detection

There are several ways to categorize the types of IDS as shown in figure 1, depending on the kind of activities, transactions and traffics or systems differs. Intrusion Detection system can be categorized into Network based Intrusion Detection System (NIDS) and Host based Intrusion Detection System (HIDS).

Based on the different approaches to event analysis, IDS can also be distinguished between Signature based detection and Anomaly detection.

Each type of Intrusion Detection System has its own advantages and disadvantages. [18]

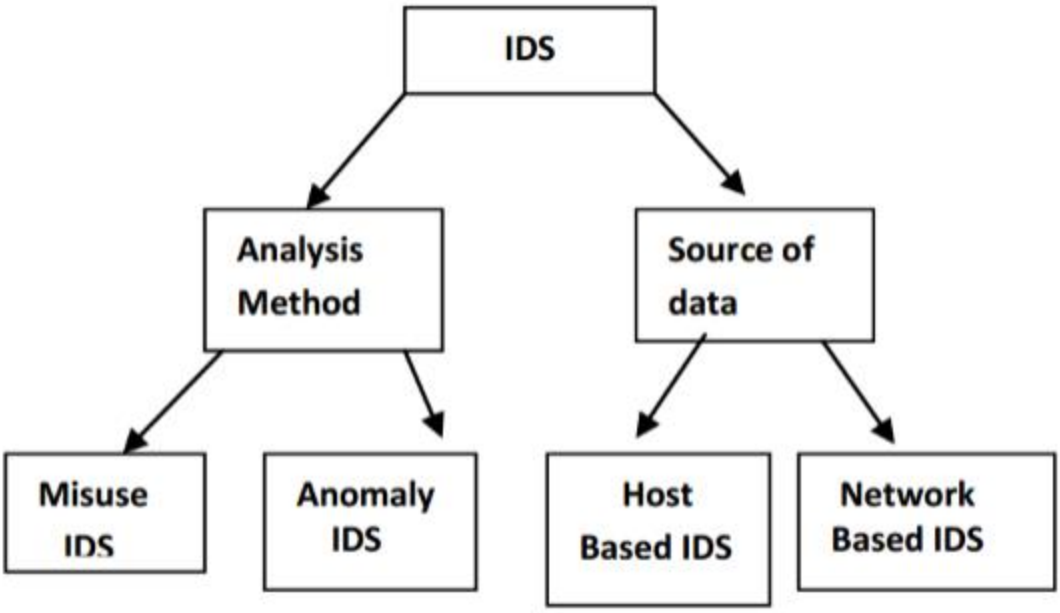

Figure 1 Taxonomy of IDS

2.1.1.1. Network based vs Host based intrusion detection systems

According to [19], [20] and [21] Network based IDS are best suited for alert generation of intrusion from outside the perimeter of the enterprise. NIDS are inserted at various points on Local Area Network (LAN) and observe packets traffic on the network information is assembled into packets and transmitted on LAN or Internet. NIDS are more valuable when they are placed outside the firewalls, thereby alerting personals of incoming packets that might get avoided to the firewall. Some NIDS allow taking input of custom signatures taken from user security policy which permits limited detection security policy violation. This limitation is due to packets traffic information that does not work well today in switched and encrypted environments, where packets analysis is weak in detecting, attacking or originating from authorized network users.

NIDS make use of raw network packets as the data source. The IDS typically use a network adapter in licentious mode that listens and analyses all traffic in real-time as it travels across the network.

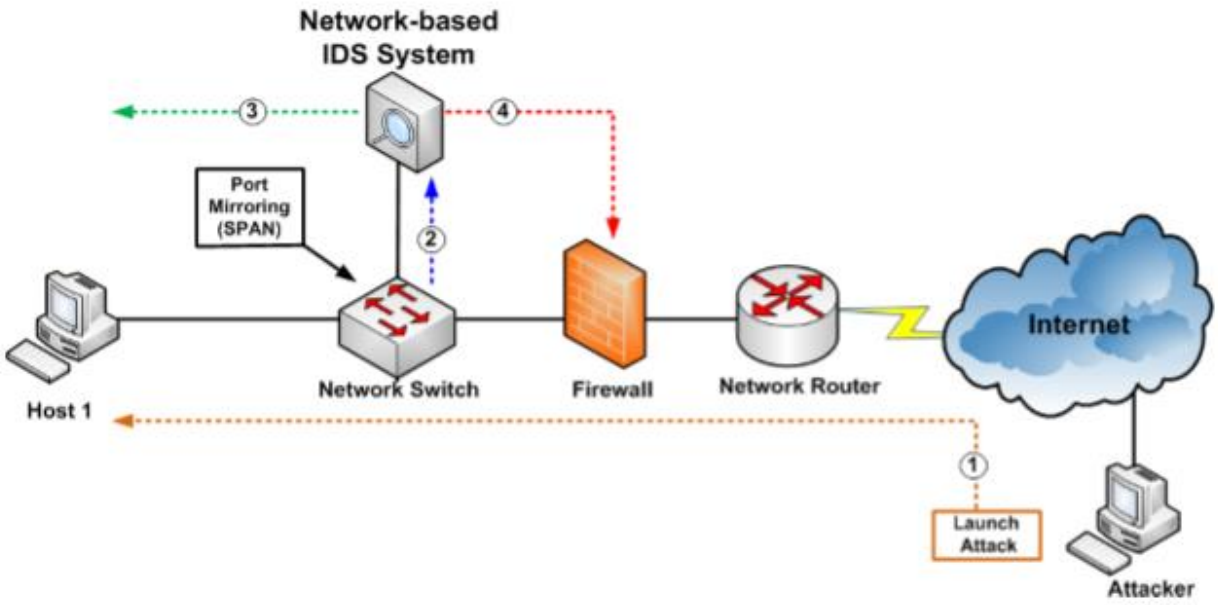

Figure 2 Network Based Intrusion Detection System **[22]**

In general, a NIDS (as shown in Figure 2) Network Based IDS are to detect malicious activities within the network and packets that are able to pass firewall which means it doesn‘t crosses the firewall. NIDS watches the traffic in real time and monitors traffic patterns, packet header, port number etc in order to detect security violation and threat. These types of IDS are valuable for detecting network based attacks like bandwidth based DDoS. Based on the IDS configuration, they react in real time to intervene the attacks before they are been successful. Most common responses are to drop the packets in order to avoid assembling of payload. There are other set of responses like disconnect sessions, reconfigure firewalls, quarantine intruder in honey-pots or padded cells.

According to [25], although there are still many issues in the research of NIDS, the following two issues appear to be the most challenging. The first challenge is the high false alarm rate in the anomaly NIDS, which has formed a hurdle for practical applications; therefore, reducing this high false alarm rate has become an intensive ongoing research topic. The second challenge is the detection of attacks that is likely to generate little network traffic and attacks originating from inside the protected network. There are some harmful network attacks that do not generate significant network traffic. [25]

- Advantages of IDS

Large networks can be monitored by deploying a few devices with a good network design. Ongoing network operations will not be disturbed by deploying NIDS, since they are passive devices. NIDSs are not vulnerable to direct attack, and may not be known by attackers.

- Disadvantages

NIDS may fail to recognize attack, when network volume becomes overwhelming. Since many switches have limited or no monitoring port capability, some networks are incapable of providing all the data for analysis by a NIDS. NIDS cannot analyze encrypted packets, making some of the traffic invisible to the process and reducing the effectiveness of NIDS. Attacks involving fragmented or malformed packets cannot be detected easily.

Host based intrusion detection system (HIDS) monitors incoming and outgoing activity on a particular system in the network. Specifically, it monitors the dynamic behavior and the state of the computer system. The administrator will be notified once an intrusion has been detected. An NIDS is usually used alongside a HIDS in order to identify any activities that HIDS overlooked. Host-based IDS places monitoring sensors also known as agents on network resources nodes to monitor audit logs which are generated by network operating system or application program. As per [22] and [25], audit logs contain records for events and activities taking place at individual Network resources. It is done because these HIDS can detect attacks that cannot be seen by NIDS such as Intrusion and can be misused by trusted insider. Host based systems utilize signature rule base which is derived from site-specific security policy. Host based can overcome the problems associated with Network based IDS immediately after alarming the security personnel who can locate the source provided by site security policy. HIDS also verifies if any attack is unsuccessful, either because of immediate response to alarm or any other reason. According to [28]

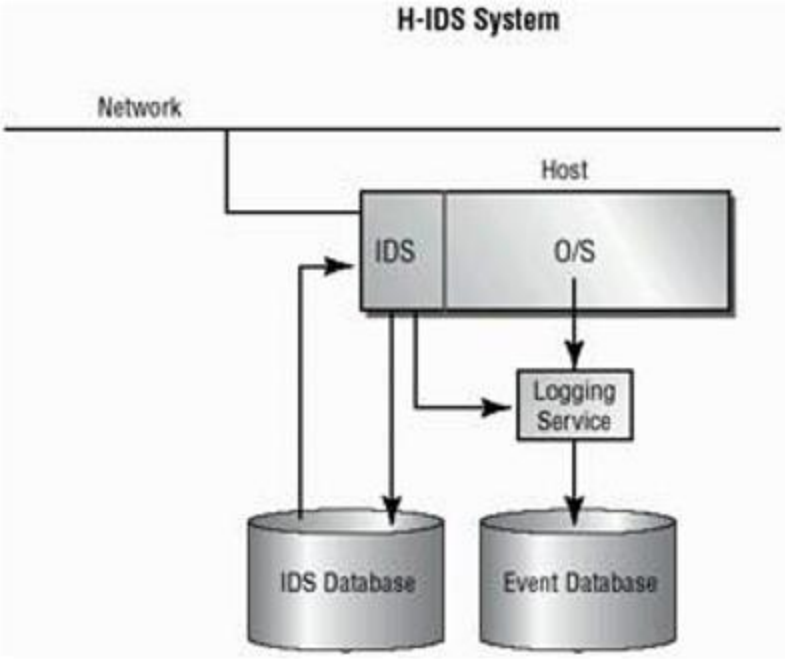

Figure 3 Host Based Intrusion Detection System **[28]**

2.1.1.2. Signature Based Detection Vs Anomaly Based Detection System

Detection of intrusion attacks is the most important issue in computer network security. Existing IDS can be divided into two classifies according to the detection approaches: anomaly detection and misuse detection. Approaches to anomaly detection have neural network, Statistics, Predictive pattern generation, and sequence matching and supervising. In misuse detection, there are state transition analysis, pattern matching, model-based, keystroke monitoring and Expert system [6]. The major category of IDS is known as Misuse Detection which is otherwise known as signature-based detection, since alarms are generated based on specific attack signatures [26], [27] and [29]. This type of signature attacks includes specific traffic or activity that is based on known intrusive activity. In general, Misuse Detection System assumes the abnormal behavior and activity which has a simple to define model. The primary advantage is its simplicity of adding known attacks to the model [25].

IDS can use signature-based detection, relying on known traffic data to analyze potentially unwanted traffic [27]. One of the advantages of Signature-based approaches result in fewer false alarms because they can be very specific about what it is and they are looking for. Because the IDS is looking for something known, a lot of information regarding what the misuse is, the

potential impact, and how to respond can be provided. This knowledge is extremely important in understanding what is occurring and effectively responding [23].

While the disadvantages of Signature based detection is that; Signature-based approaches can only detect misuse for which a signature exists. For a signature to exist, the form of misuse must be known about beforehand so it can be researched and programmatically identified. This means any new form of misuse will not be detected by a signature based system until it is identified, analyzed, and then incorporated into the product. Depending on the circumstances, this could be hours, days, weeks, or even months [27], [29] and [24] .

If the network is small and signatures are kept up to date, the human analyst solution to intrusion detection works well. But when organizations have a large, complex network the human analysts quickly become overwhelmed by the number of alarms they need to review. The sensors on the MITRE network, for example, currently generate over one million alarms per day. And that number is increasing. This situation arises from ever increasing attacks on the network, as well as a tendency for sensor patterns to be insufficiently selective (i.e., raise too many false alarms). Commercial tools typically do not provide an enterprise level view of alarms generated by multiple sensor vendors. Commercial intrusion detection software packages tend to be signature-oriented with little or no state information maintained. These limitations led us to investigate the application of data mining to this problem. [3]

Anomaly Based Detection on the other hand is based on defining the network behavior. Behavior of the network is the predefined one, when it is accepted or else it triggers the event in the anomaly detection. The accepted behavior of the network is prepared or learned by the specifications of the network administrators [30].

According to [25] the important phase in defining the network behavior is the IDS engine that is capable to cut through the various protocols at all levels. The engine is able to process the protocols and understand the goal. Though this protocol analysis is computationally expensive, the benefits it generates like increasing the rule set helps in less false positive alarms.

According to [27] the major drawback of Anomaly Detection is defining its rule set. The efficiency of the system depends on implementation and testing on all protocols that are available. Rule Defining Process is also affected by various protocols used by several vendors.

Intrusions can be detected by observing deviations from the expected behaviors of the system monitored. These "normal" behaviors can either correspond to some observations made in the past or to some forecasts made by various techniques. Everything that does not correspond to this "normal" pattern will be flagged as anomalous.

Anomaly detection technique is comparing user's current behavior with usual behavior which is already stored in the database. It used to detect the unknown attacks. By using statistical techniques to find patterns of activity that appears to be abnormal.

It is failed in high detection rate. But the goal of anomaly detection system is to find the intrusion in the system timely [26]. Therefore, the core process of anomaly detection is not to learn what is anomalous but to learn what is normal or expected [32], [26] [27]. The key advantage of anomaly-based detection over the other techniques is the ability to detect new attacks, attacks for which no signature or known protocol violation exists [29].

2.1.2. Components of Intrusion Detection System

Intrusion Detection Systems are generally made up of the following main types of components:-

Sensors:-these are deployed in a network or on a device to collect data. They take input from various sources, including network packets, log files, and system call traces. Input is collected, organized, and then forwarded to one or more analyzers.

Analyzers:-Analyzers in an IDS collect data forwarded by sensors and then determine if an intrusion has actually occurred. Output from the analyzers should include evidence supporting the intrusion report. The analyzers may also provide recommendations and guidance on mitigation steps. □

User interface:-the user interface of the IDS provides the end user a view and way to interact with the system. Through the interface the user can control and configure the system. Many user interfaces can generate reports as well.

Honeypot:-in fully deployed IDS, some administrators may choose to install a "honeypot," essentially a system component set up as bait or decoy for intruders. Honeypots can be used as early warning systems of an attack, decoys from critical systems, and data collection sources for attack analyses. Many IDS vendors maintain honeypots for research purposes, and to develop new intrusion signatures. Note that a honeypot should only be deployed when the organization has the

resources to maintain it. A honeypot left unmanaged may become a significant liability because attackers may use a compromised honeypot to attack other systems. [27]

2.1.3. Types of Attack

An IDS is an effective security tool which helps the users and administrators to prevent unauthorized access to network resources. There are various kinds of attacks that are most common in IDS. They are probe attacks, denial-of-service attacks (DoS), remote to local (R2L) and user to root (U2R) attacks [27], [22].

2.1.3.1. Probe Attacks

Probe attacks are aimed at acquiring information about the target network from a source that is often external to the network. The basic connection level features such as the "duration of connection" and "source bytes" are significant while features like "number of files creations" and "number of files accessed" are not expected to provide information for detecting probes. [25]

2.1.3.2. Denial-of-Service Attacks

The Denial-of-Service attack (DoS) attacks are meant to force the target to stop the service that is provided by flooding it with illegitimate requests [29]. Hence, for the DoS attack to be detected that the traffic features such as the "percentage of connections having same destination host and same service" and packet level features such as the "source bytes" and "percentage of packets with errors" are significant. Detecting the DoS attacks is not being important to know whether a user is "logged in or not." DoS is an attempt to make a computer resource unavailable to its intended users in computer security. Typically the targets are high-profile web servers and the attack attempts to make the hosted web pages unavailable on the internet. This is a cyber-crime, which violates the Internet proper usage policy as indicated by the Internet Architecture Board (IAB) [28], [25].

There are two main types of DoS attacks: Flooding and Flaw exploitations. Flooding attacks can be simply implemented. A DoS attack by just using the ping command is a typical example that results in sending the victim more number of ping packets. Moreover, when the attacker has access to greater bandwidth than the victim then this will easily and quickly grab the victim. [25].

DoS attacks have two general forms: [25].

- To force the victim computers to reset or consume its resources such that it can no longer provide its intended service.

- To obstruct the communication media between the intended users and the victim so that they can no longer communicate adequately.

A DoS attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include:

- Flooding a network, thereby preventing legitimate network traffic.
- Disrupting service to a specific system or person.
- Attacks can be directed at any network device, including attacks on routing devices, web, electronic mail, or Domain Name System Servers.
- Consumption of computational resources, such as bandwidth, disk space, or CPU time.
- Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.
- Disruption of state information, such as unsolicited resetting of TCP sessions.

2.1.3.3. Remote to Login (R2L) Attacks

The R2L Attacks are one of the most difficult ones to detect, as they involve the network level and the host level features. Therefore, selecting both the network level features such as the "duration of connection" and "service requested" and the host level features such as the "number of failed login attempts" among others for detecting R2L attacks [34], [25].

2.1.3.4. User to Root (U2R) Attacks

The U2R Attacks involve the meaningful details that are very difficult to capture at an early stage. Applications of these attacks are Content Based and Target Based. Then for U2R Attacks features such as "number of file creations" and "number of shell prompts invoked," are selected, while features such as "protocol" and "source bytes" are ignored [34], [25]. The different types of attacks and their categories discussed above summarized in table below: [30]

Attack type Category

Attack type	Category
buffer_overflow, httprunnel, loadmodule, perl, ps, rootkit, sqlattack, xterm	U2R
apacha2, back, land, mailbomb, netpune, pod, processtable, smurf, teardrop, udpstorm	DoS
ftp-write, guess password, imap, multihop, named, phf, sendmail, Snmpgetattack, Snmpguess, spy, warezclient, warezmaster, worm, Xlock, Xsnoop	R2L
ipsweep, Mscan, Namp, portsweep, saint, satan	Probing

Table 1 Categories of different types of attacks **[30]**

2.1.4. Types of Intrusion Detection

There are many types of IDPS technologies. For the purposes of this document, they are generally divided into the following two groups based on the type of events that they monitor and the ways in which they are deployed [23]. □

- Network-Based:- Monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity. It can identify many different types of events of interest. It is most commonly deployed at a boundary between networks, such as in proximity to border firewalls or routers, virtual private network (VPN) servers, remote access servers, and wireless networks.
- Host-Based:- which monitors the characteristics of a single host and the events occurring within that host for suspicious activity. Examples of the types of characteristics a host-based IDPS might monitor are network traffic (only for that host), system logs, running processes, application activity, file access and modification, and system and application configuration changes. Host-based IDPSs are the most commonly deployed on critical hosts such as publicly accessible servers and servers containing sensitive information.

2.1.5. Approaches of IDS

Currently there are two basic approaches to intrusion detection: The first approach, called anomaly detection, is to define and characterize correct static form and/or acceptable dynamic behavior of the system, and then to detect wrongful changes or wrongful behavior. It relies on being able to

define desired form or behavior of the system and then to distinguish between that and undesired or anomalous behavior. The boundary between acceptable and anomalous form of stored code and data is precisely definable. One bit of difference indicates a problem. The boundary between acceptable and anomalous behavior is much more difficult to define.

The second approach, called misuse detection, involves characterizing known ways to penetrate a system. Each one is usually described as a pattern. The misuse detection system monitors for explicit patterns. The pattern may be a static bit string, for example a specific virus bit string insertion. Alternatively, the pattern may describe a suspect set or sequence of actions. Patterns take a variety of forms as will be illustrated later.

According to Chang [36] view, no fundamentally different alternative approach has been introduced in the past decade. However, new forms of pattern specifications for misuse detection have been invented. The techniques for single systems have been adapted and scaled to address intrusion in distributed systems and in networks. Efficiency and system control have improved. User interfaces have improved, especially those for specifying new misuse patterns and for interaction with the system security administrator.

2.1.6. Uses of Intrusion Detection Prevention Systems (IDPSs) Technologies

As discussed by Karen and Peter [23] many IDPSs can also be configured to recognize violations of security policies. For example, some IDPSs can be configured with firewall rule set like settings, allowing them to identify network traffic that violates the organization's security or acceptable use policies. Also, some IDPSs can monitor file transfers and identify ones that might be suspicious, such as copying a large database onto a user's laptop.

Many IDPSs can also identify reconnaissance activity, which may indicate that an attack is imminent [29]. For example, some attack tools and forms of malware, particularly worms, perform reconnaissance activities such as host and port scans to identify targets for subsequent attacks. An IDPS might be able to block reconnaissance and notify security administrators, who can take actions if needed to alter other security controls to prevent related incidents. Because reconnaissance activity is so frequent on the Internet, reconnaissance detection is often performed primarily on protected internal networks.

In addition to identifying incidents and supporting incident response efforts, organizations have found other uses for IDPSs, including the following [23].□

- Identifying security policy problems:- An IDPS can provide some degree of quality control for security policy implementation, such as duplicating firewall rule-sets and alerting when it sees network traffic that should have been blocked by the firewall but was not because of a firewall configuration error.
- Documenting the existing threat to an organization:- IDPSs log information about the threats that they detect. Understanding the frequency and characteristics of attacks against an organization's computing resources is helpful in identifying the appropriate security measures for protecting the resources. The information can also be used to educate management about the threats that the organization faces.
- Deterring individuals from violating security policies. If individuals are aware that their actions are being monitored by IDPS technologies for security policy violations, they may be less likely to commit such violations because of the risk of detection. Because of the increasing dependence on information systems and the prevalence and potential impact of intrusions against those systems, IDPSs have become a necessary addition to the security infrastructure of nearly every organization.

2.1.7. Key Functions of IDPS

There are many types of IDPS technologies, which are differentiated primarily by the types of events that they can recognize and the methodologies that they use to identify incidents. In addition to monitoring and analyzing events to identify undesirable activity, all types of IDPS technologies typically perform the following functions [30] .

- Recording information related to observed events:- Information is usually recorded locally, and might also be sent to separate systems such as centralized logging servers, security information and event management (SIEM) solutions, and enterprise management systems.
- Notifying security administrators of important observed events:- This notification, known as an alert, occurs through any of several methods, including the following: emails, pages, messages on the IDPS user interface, Simple Network Management Protocol (SNMP) traps, and user-defined programs and scripts. A notification message typically includes

only basic information regarding an event; administrators need to access the IDPS for additional information.

- Producing reports:- Reports summarize the monitored events or provide details on particular events of interest.

Some IDPSs are also able to change their security profile when a new threat is detected. IPS technologies can respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which can be divided into the following groups: [23].

- The IPS stops the attack itself:- Examples of how this could be done are as follows:
  - Terminate the network connection or user session that is being used for the attack.
  - Block access to the target (or possibly other likely targets) from the offending user account, IP address, or other attacker attribute.
  - Block all access to the targeted host, service, application, or other resource.
- The IPS changes the security environment:-The IPS could change the configuration of other security controls to disrupt an attack. Common examples are reconfiguring a network device (e.g., firewall, router, switch) to block access from the attacker or to the target, and altering a host-based firewall on a target to block incoming attacks. Some IPSs can even cause patches to be applied to a host if the IPS detects that the host has vulnerabilities.
- The IPS changes the attack's content:-Some IPS technologies can remove or replace malicious portions of an attack to make it benign. A simple example is an IPS removing an infected file attachment from an e-mail and then permitting the cleaned email to reach its recipient. A more complex example is an IPS that acts as a proxy and normalizes incoming requests, which means that the proxy repackages the payloads of the requests, discarding header information. This might cause certain attacks to be discarded as part of the normalization process.

As discussed by Karen and Peter [23] another common attribute of IDPS technologies is that they cannot provide completely accurate detection. When an IDPS incorrectly identifies benign activity as being malicious, a false positive has occurred. When an IDPS fails to identify malicious activity, a false negative has occurred. It is not possible to eliminate all false positives and negatives; in most cases, reducing the occurrences of one increases the occurrences of the other. Many organizations choose to decrease false negatives at the cost of increasing false positives, which

means that more malicious events are detected but more analysis resources are needed to differentiate false positives from true malicious events. Altering the configuration of an IDPS to improve its detection accuracy is known as tuning.

Most IDPS technologies also offer features that compensate for the use of common evasion techniques [37]. Evasion is modifying the format or timing of malicious activity so that its appearance changes but its effect is the same. Attackers use evasion techniques to try to prevent IDPS technologies from detecting their attacks. For example, an attacker could encode text characters in a particular way, knowing that the target understands the encoding and hoping that any monitoring IDPSs do not. Most IDPS technologies can overcome common evasion techniques by duplicating special processing performed by the targets. If the IDPS can "see" the activity in the same way that the target would, then evasion techniques will generally be unsuccessful at hiding attacks.

2.2. Data Mining Techniques for Intrusion Detection and Prevention system

As Bloedorn discussed [3] data mining is, at its core, pattern finding. Data miners are experts at using specialized software to find regularities (and irregularities) in large data sets.

Data mining techniques have been popular in extracting the behaviors of these harmful patterns from large volumes of data in recent years. Data mining is used in many areas of application, e.g., the business world, medicinal sciences, physical sciences and engineering to make new discoveries [26], [27], [9], [20], [22]. Extensive studies have been performed in applying data mining techniques to network traffic anomaly detection, but the methods [20], [27] have limitations that notably discredit them from use in real environments. In this thesis, we explore the possibilities of integrating data mining techniques to identify network intrusions with significant performance improvement.

Much number of data mining techniques can be used in intrusion detection, each with its own specific advantage [25].

Data mining has gained a great deal of attention in the information industry and in the society as a whole in recent years due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. Based on [9], [20] and [30] data mining is defined as follows:

The term knowledge discovery in databases (KDD) refers to the process of converting raw data into useful information or knowledge. Data mining is a step in the KDD process, and applies a variety of algorithm for extracting patterns from data. In addition to this, the KDD process has additional steps including data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining to ensure useful knowledge is derived from the data [27] and [20].

2.3. Data Mining Tasks

There are two main tasks of data mining. These are: descriptive modeling and predictive modeling. The descriptive data mining tasks characterize the general properties of the data in the database, while predictive data mining tasks perform inference of the current data in order to make prediction. Descriptive data mining focus on finding patterns describing the data that can be interpreted by humans, and produces new, nontrivial information based on the available data set. Predictive data mining involves using some variables or fields in the data set to predict unknown or future values of other variables of interest, and produces the model of the system described by the given data set.

The goal of predictive data mining is to produce a model that can be used to perform tasks such as classification, prediction or estimation, while the goal of descriptive data mining is to gain an understanding of the analyzed system by uncovering patterns and relationships in large data sets.

The descriptive task encompasses methods such as clustering, summarization, association rules and sequence analysis. [22]

2.3.1. Descriptive Model

Descriptive data mining is normally used to generate frequency, cross tabulation and correlation. Descriptive method can be defined to discover interesting regularities in the data, to uncover patterns and find interesting subgroups in the bulk of data. In education, studies [31] used Descriptive to determine the demographic influence on particular factors. Summarization maps data into subsets with associated simple descriptions [32]. Basic statistics such as Mean, Standard Deviation, Variance, Mode and Median can be used as Summarization approach.

2.3.1.1. Clustering

Clustering is a data mining technique where data points are clustered together based on their feature values and a similarity metric. Frank [33] breaks clustering techniques into five areas:

hierarchical, statistical, exemplar, distance, and conceptual clustering, each of which has different ways of determining cluster membership and representation.

Clustering is an unsupervised learning process. A comprehensive survey of current clustering techniques and algorithms is available in [34].

Clustering is a tool for data analysis, which solves classification problems. Its objective is to distribute cases (people, objects, events etc.) into groups, so that the degree of similarity can be strong between members of the same cluster and weak between members of different clusters [9]. In clustering, there is no pre-classified data and no distinction between independent and dependent variables. Instead, clustering algorithms search for groups of records (the clusters composed of records similar to each other). The algorithms discover these similarities. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression [27]. Clustering is also called data segmentation in some applications because clustering partitions large datasets into groups according to their similarity. Clustering can also be used for outlier detection, where outliers (values that are "far away" from any cluster) may be more interesting than common cases [9].

Clustering provides some significant advantages over the classification techniques, in that it does not require the use of a labeled data set for training. [9]

K-means clustering: is a data mining/machine learning algorithm used to cluster observations into groups of related observations without any prior knowledge of those relationships. The k-means algorithm is one of the simplest clustering techniques [9]. The k-means algorithm is an evolutionary algorithm that gains its name from its method of operation. The algorithm clusters observations into k groups, where k is provided as an input parameter. It then assigns each observation to clusters based upon the observation's proximity to the mean of the cluster. The cluster's mean is then recomputed and the process begins again. According to [9] discussed below is how the k-mean algorithm works:

I. The algorithm arbitrarily selects k points as the initial cluster centers ("means").
II. Each point in the dataset is assigned to the closed cluster, based upon the Euclidean distance between each point and each cluster center.
III. Each cluster center is recomputed as the average of the points in that cluster. □

IV. Steps II and III repeat until the clusters converge. Convergence may be defined differently depending upon the implementation, but it normally means that either no observations change clusters when steps II and III are repeated or that the changes do not make a material difference in the definition of the clusters.

2.3.1.2. Association Rules

Associations or Link Analysis are used to discover relationships between attributes and items such as the presence of one pattern implies the presence of another pattern. i.e. to what extent one item is related to another in terms of cause-and-effect. This is common in establishing a form of statistical relationships among different interdependent variables of a model. Association Rules is a popular technique for market basket analysis because all possible combinations of potentially interesting product groupings can be explored [9].

2.3.1.3. Sequence Analysis

Sequence Analysis is used to determine sequential patterns in data [40]. The patterns in the dataset are based on time sequence of actions, and they are similar to association data, however the relationship is based on time. In Market Basket analysis, the items are to be purchased at the same time, on the other hand, for Sequence Analysis the items are purchased over time in some order. [9]

2.3.2. Predictive Model

The goal of the predictive models is to construct a model by using the results of the known data and is to predict the results of unknown data sets by using the constructed model. For instance a bank might have the necessary data about the loans given in the previous terms. In this data, independent variables are the characteristics of the loan granted clients and the dependent variable is whether the loan is paid back or not [35]. The model constructed by this data is used in the prediction of whether the loan will be paid back by client in the next loan applications.

2.3.2.1. Classification and regression

Classification and regression are two data analyzing methods which determine important data classes or may construct models which can predict future data trends. The classification model predicts the categorical values; the regression is used in the prediction of values showing continuity. For instance while the classification model is constructed to categorize whether the bank loan applications are safe or risky, the regression model may be constructed to predict the

spending of clients buying computer products whose income and occupation are given [36] and [43].

In the classification models the following techniques are mainly used [43]. Decision Trees, Artificial Neural Networks and Navie-Bayes.

2.3.2.1.1 Decision Trees

Decision trees technique is commonly used in data mining because their construction is cheap, their interpretation is easy [43]. Decision trees are trees that classify instances by sorting them based on feature values [37]. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values

2.3.2.1.2. Neural networks

Artificial neural network models have been studied for many years in the hope of achieving human-like performance in several fields such as speech and image understanding [45]. The networks are composed of many nonlinear computational elements operating in parallel and arranged in patterns reminiscent of biological neural networks. It is called back-propagation because the derivatives are computed first at the last layer of the network, and then propagated backward through the network, using the chain rule, to compute the derivatives in the hidden layers. For a multi-layer network, the output of one layer becomes the input of the following layer.

2.3.2.1.3. Bayesian Network (BN)

A BN is a graphical model for probability relationships among a set of variables. The BN structure S is a directed acyclic graph (DAG) and the nodes in S are in one-to-one correspondence with the features X. The arcs represent casual influences among the features while the lack of possible arcs in S encodes conditional independencies. Moreover, a feature (node) is conditionally independent from its non-descendants given its parents (X1 is conditionally independent from X2 given X3 if P (X1|X2, X3) =P (X1|X3) for all possible values of X1, X2, X3).

The most interesting feature of BNs, compared to decision trees or neural networks, is most certainly the possibility of taking into account prior information about a given problem, in terms of structural relationships among its features. This prior expertise, or domain knowledge, about the structure of a Bayesian network can take the forms [45].

2.4. Cost sensitive feature selection for IDs

A very important but often neglected facet of intrusion detection is its cost-effectiveness, or cost-benefit trade-off. An educated decision to deploy a security mechanism such as IDS is often motivated by the needs of security risk management [38]. The objective of IDS is therefore to provide protection to the information assets that are at risk and have value to an organization. An IDS needs to be cost-effective in that it should be cost no more than the expected level of loss from intrusions.

Currently these cost factors are, for the most part, ignored as unwanted complexities in the development process of IDSs [39]. This is caused by the fact that achieving a reasonable degree of technical effectiveness is already a challenging task, given the complexities of today's network environments and the manual effort of knowledge engineering approaches (e.g., encoding expert rules). Some IDSs do try to minimize operational cost. For example, the [40] scripting language for specifying intrusion detection rules does not support for-loops because iteration through a large number of connections is considered time consuming.

It is possible to develop a data mining framework for building intrusion detection models in an effort to automate the process of IDS development and lower its development cost. The framework uses data mining algorithms to compute activity patterns and extract predictive features, and then applies machine learning algorithms to generate detection rules [51].

2.5. Tasks to be performed by IDS.

The main task of intrusion detection systems is defense of a computer system by detecting an attack and possibly repelling it. Detecting hostile attacks depends on the number and type of appropriate actions (see Fig 4). Intrusion prevention requires a well-selected combination of "baiting and trapping" aimed at both investigations of threats. Diverting the intruder's attention from protected resources is another task. Both the real system and a possible trap system are constantly monitored. Data generated by intrusion detection systems is carefully examined (this is the main task of each IDS) for detection of possible attacks.

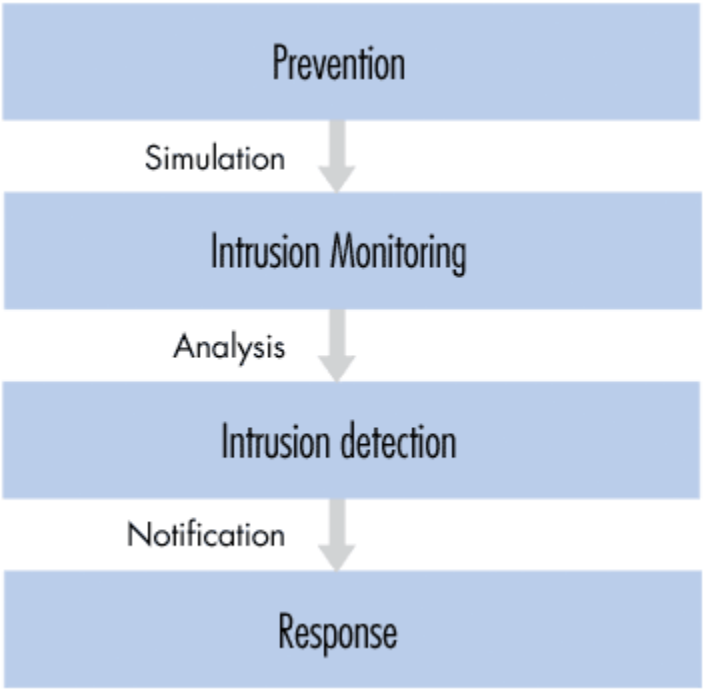

Figure 4 Intrusion detection system activities [65]

Once an intrusion has been detected, IDS issues alerts notifying administrators of this fact. The next step is undertaken either by the administrators or the IDS itself, by taking advantage of additional countermeasures (specific block functions to terminate sessions, backup systems, routing connections to a system trap, legal infrastructure etc.) – following the organization's security policy. An IDS is an element of the security policy.

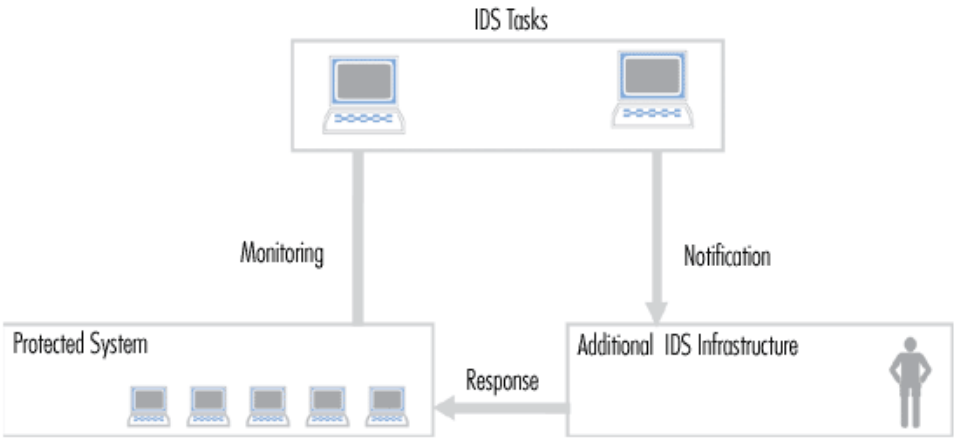

Figure 5 Intrusion Detection based on organizational security policy

Among the various IDS tasks, intruder identification is one of the fundamental one. It can be useful in the forensic research of incidents and installing appropriate patches to enable the detection of future attack attempts targeted on specific persons or resources.

Intrusion detection may sometimes produce false alarms, for example as a result of malfunctioning network interface or sending attack description or signatures via email.

2.6. Structure and architecture of intrusion detection systems

The figure below explains the Architecture of Data Mining based IDS consists of sensors, detectors, a data warehouse, and a model generation component. This architecture is capable of supporting not only data gathering, sharing, and analysis, but also data archiving and model generation and distribution.

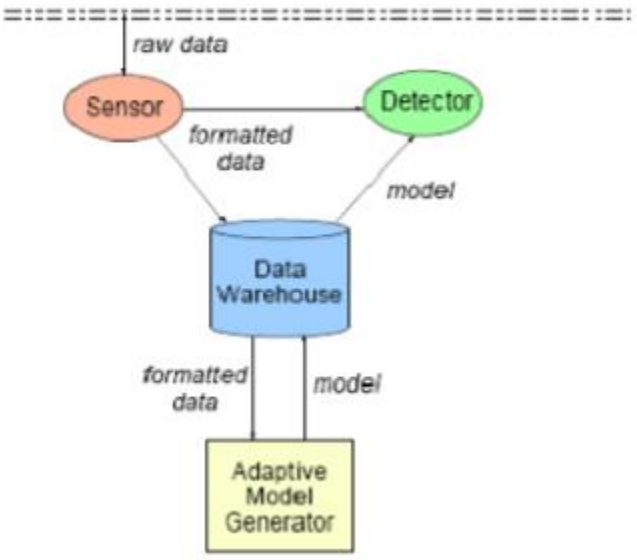

Figure 6 Architecture of Data Mining based IDS

The sensor is integrated with the component responsible for data collection of an event generator. The collection manner is determined by the event generator policy that defines the filtering mode of event notification information. The event generator (operating system, network, application) produces a policy-consistent set of events that may be a log (or audit) of system events, or network packets. This, set along with the policy information can be stored either in the protected system or outside. In certain cases, no data storage is employed for example, when event data streams are transferred directly to the analyzer. This concerns the network packets in particular.

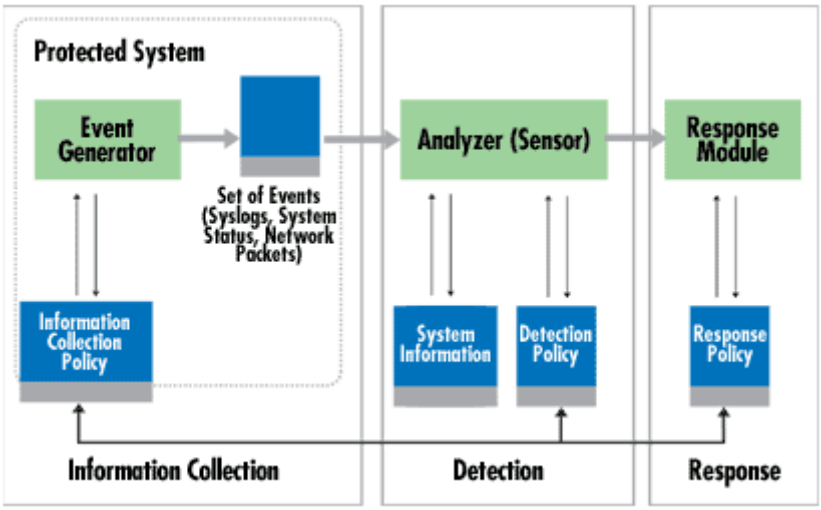

Figure 7 IDS components for data collection **[41]**

The role of the sensor is to filter information and discard any irrelevant data obtained from the event set associated with the protected system, thereby detecting suspicious activities. The analyzer uses the detection policy database for this purpose. The latter comprises the following elements: attack signatures, normal behavior profiles, necessary parameters (for example, thresholds). In addition, the database holds IDS configuration parameters, including modes of communication with the response module. The sensor also has its own database containing the dynamic history of potential complex intrusions (composed from multiple actions).

Intrusion detection systems can be arranged as either centralized (for example, physically integrated within a firewall) or distributed. A distributed IDS consists of multiple Intrusion Detection Systems (IDS) over a large network, all of which communicate with each other. More sophisticated systems follow an agent structure principle where small autonomous modules are organized on a per-host basis across the protected network. The role of the agent is to monitor and filter all activities within the protected area and depending on the approach adopted - make an initial analysis and even undertake a response action. The cooperative agent network that reports to the central analysis server is one of the most important components of intrusion detection systems. DIDS can employ more sophisticated analysis tools, particularly connected with the detection of distributed attacks **[57]**. Another separate role of the agent is associated with its mobility and roaming across multiple physical locations. In addition, agents can be specifically devoted to detect certain known attack signatures. This is a decisive factor when introducing protection means associated with new types of attacks **[57]**. IDS agent-based solutions also use less sophisticated mechanisms for response policy updating [57].

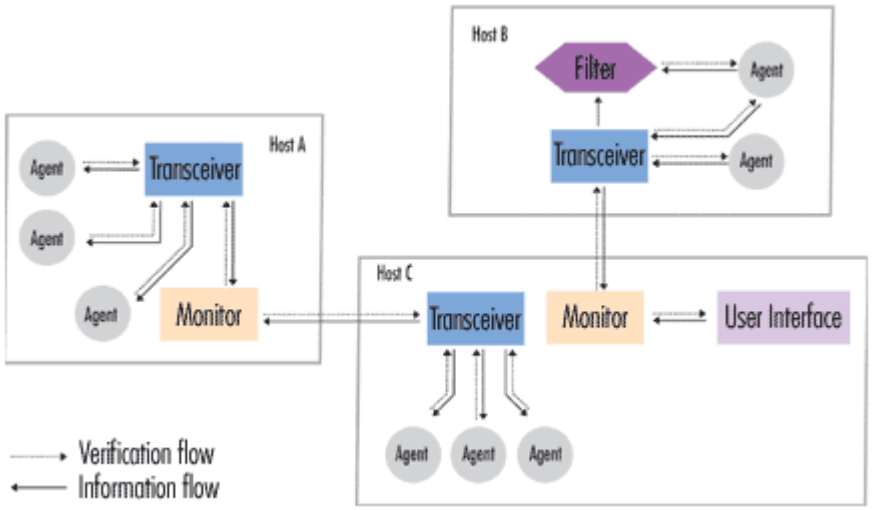

Figure 8 An AAFID compliant representation of an intrusion detection system employing autonomous agents

Basically, there are two approaches for intrusion detection design based on the uses of detection techniques: Knowledge based and behavior-based intrusion detection. Knowledge based intrusion detection is also called misuse detection. In principle, it is typically realized by modeling known attack behavior with prior understanding about specific attacks and system vulnerabilities. Afterward, the intrusion detection system compares network traffic data being observed with well-defined attack patterns for identifying the possible penetrations to the system. When the data is as same as one of the explicitly defined attack patterns, an alarm is raised. The defined attack patterns are frequently referred to as the signatures of intrusions. The signature could be a static string or a sequence of events. While knowledge-based intrusion detection is achieved by modeling known attack behavior, on the contrary, behavior based intrusion detection also known as anomaly detection models normal or expected behavior of computer users. It looks for malicious activities by comparing the observed data with these acceptable behaviors. If the data diverge from the learned normal behavior, an alarm is raised. In other word, anything will be suspected as an attack if its behavior is deviated from the previously learned behaviors. For developing intrusion detection systems, a large amount of traffic data is always necessary to be collected in advance for analysis with the use misuse detection or anomaly detection approaches. Based on the collected network audit trail, misuse detection techniques specify well defined attack signatures and anomaly detection techniques establish acceptable usage profiles to differentiate intrusions and normal activities from a future network traffic data stream.

Architecture for detecting computer attacks consists of several modules that will be executed by the agents in a distributed manner. Communication among the agents is done utilizing the TCP/IP sockets. Agent modules running on the host computers consist of data collection agents, data analysis agents, and response agents. Agents running on the secure devices consist of the agent control modules that include agent regeneration, agent dispatch, maintaining intrusion signatures and information of the features and feature ranking algorithms that help identify the intrusions.

2.7. Architecture of Ethiopian Institute of Agricultural Research

The Local Area Network that has been deployed for EIAR is based on collapsed core and access layer scenario. There are seven departments (Administration , management, finance & Audit, Training , ICD & Liberary , E_Oil , New Building) covered by the LAN. The Server room is located at ICD block. New Building, Administration, Finance & Training are connected to the Distribution/core via fiber link. Access switches are aggregated to the Distribution switch /Aggregator. There are three links subscribed from Ethiotelecom.

I. 512M VPN link for woredanet connection.

II. 1M VPN link for Agrinet connection

III. 40M Internet link.And a firewall is configured to protect the ministry network as well as woredanet & Agrinet against the external threats. The Agency network consists of one Core switch (S5300), one distribution switches (S5300), Eudemon E1000E firewall, Fourteen S3300 access switches and one AR29 router.

2.7.1. EIAR LAN Design

This topic provides introductory explanation to the approaches taken to design EIAR's Network. It also covers the interconnection between Networking Equipments used in the network. To further aid the discussion Figure 9- EIAR Physical Interconnection Architecture Design is used.

As depicted on the network architecture, there are three main connection details, the core/Distribution Design, Access Layer Design, and the Enterprise Edge Design.

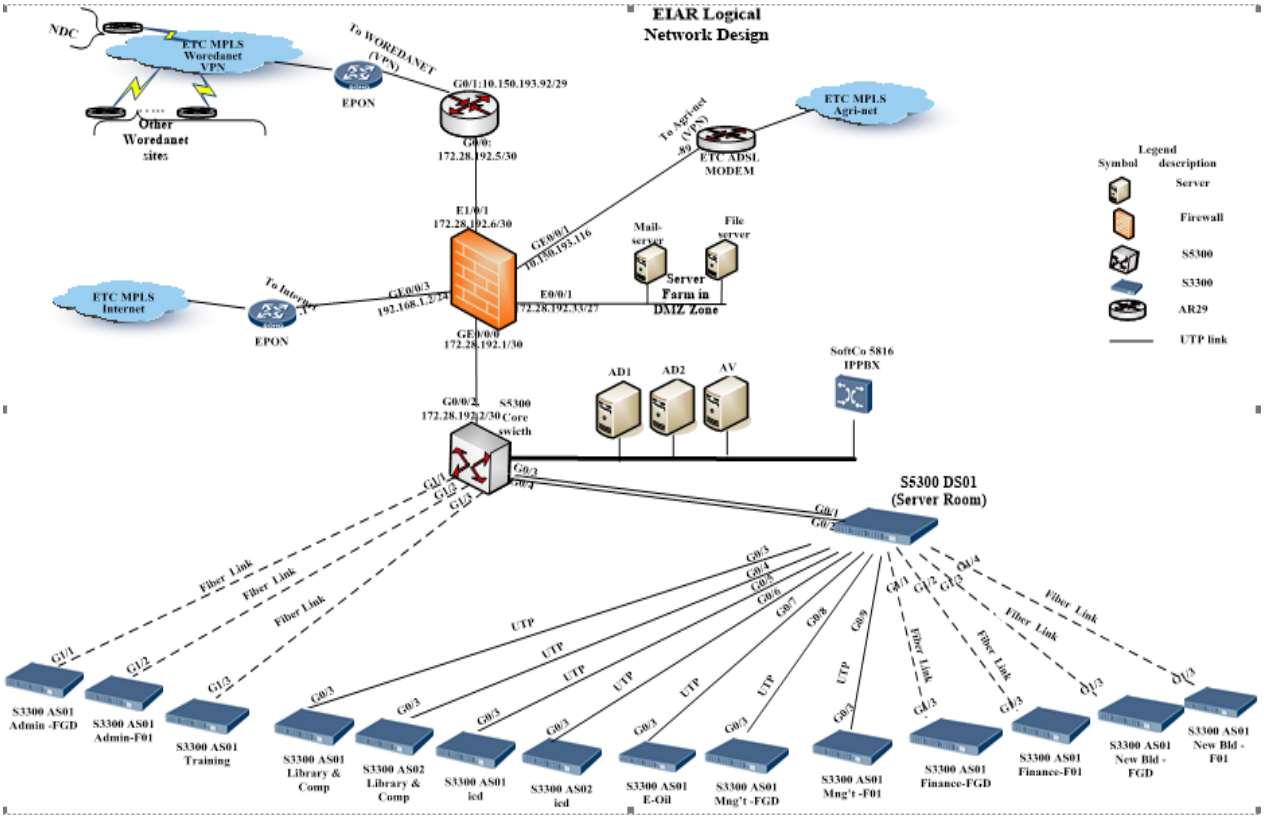

Figure 9 Generic LAN diagram for EIAR

2.7.1.1: Design Considerations

2.7.1.1.1. Design Objectives

The Objectives of this design in this architecture is to have successful implementation of the EIAR Enterprise Data Network and plan for future growth. The design:

- Provide a secure network infrastructure.
- Meets all of the customer functional requirements, scalable to the customer's defined growth levels.
- Provide redundancy, resilience, and economy in the system to minimize failure points and risks.
- Provide a high quality Voice & Data capable network.

The firewall will be configured in such a way that it has four Zones which are INTERNET, DMZ, WOREDANET, AGRINET and TRUST. Firewall inside is TRUST and the VPN connection to

NDC are in the WOREDANET zone, Internet connection is in the INTERNET zone, other EIAR-Branch offices are in AGRINET and Servers which are accessible from both WOREDANET, AGRINET and INTERNET are located in DMZ zone.

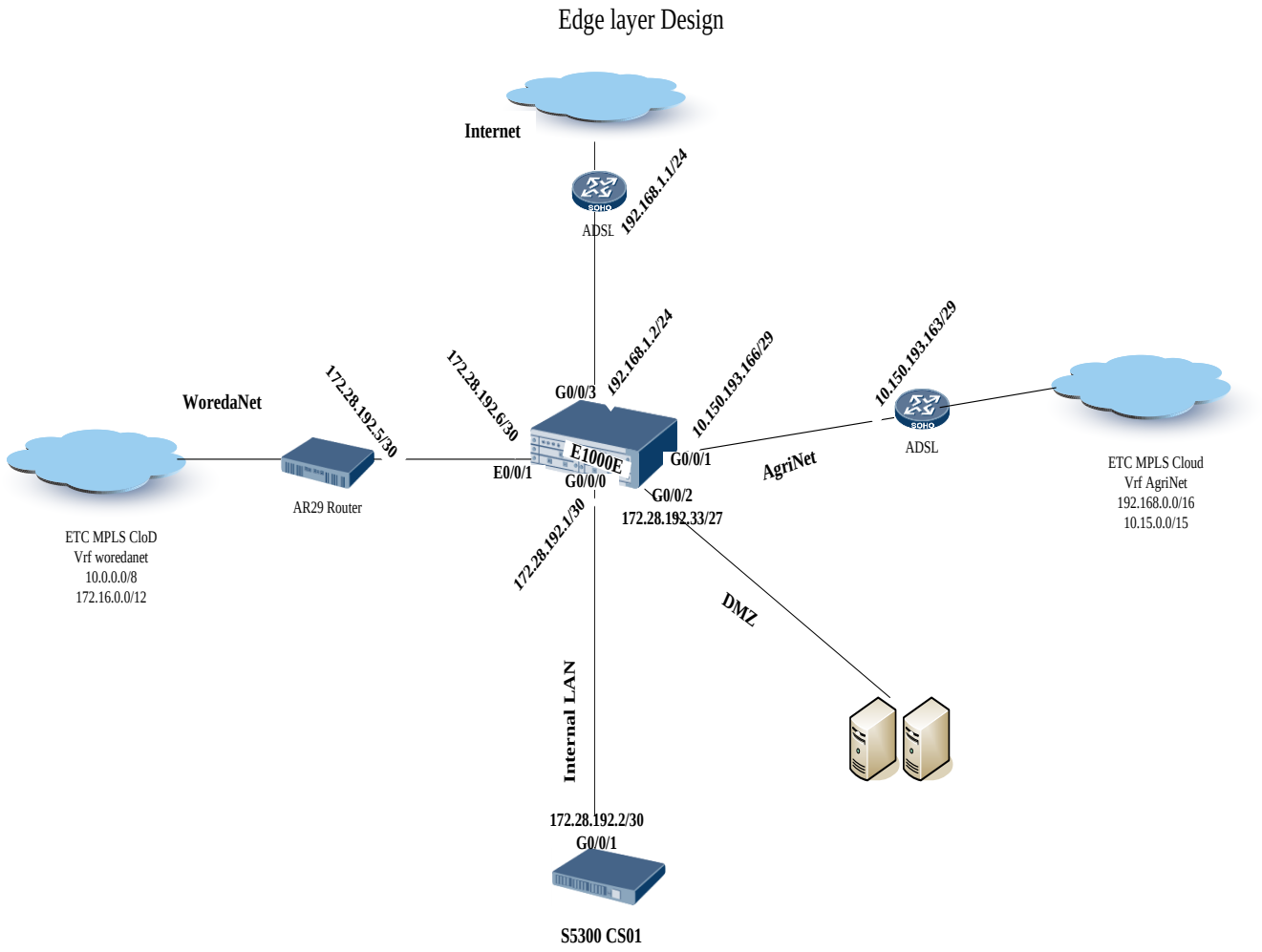

Figure 10 Edge layer design for EIAR

2.7.1.1.2. Packet filter:

Packet filter is a network security protection mechanism. It is used to control the inbound and outbound data between networks in different security levels. A series of filter rules are needed to filter data packets, which can be carried out by applying filter rules defined between different zones in the firewall.

2.7.1.1.3. Attack Defense

Normally, network attacks intrude or destroy network servers (hosts) to steal the sensitive data on servers or interrupt server services. There are also the network attacks that directly destroy network devices, which can make networks service abnormal or even out of service. The attack defense of the firewall can detect various types of network attacks and take the measures to protect internal networks from malicious attacks. As a result, the firewall can assure the normal operations of the internal networks and systems.

The Eudemon 1000E provides an abundant virus database. By comparing scanned files with the features in the virus database, the Eudemon 1000E identifies whether the files contain viruses. And then the Eudemon 1000E process the files infected with viruses according the processing modes configured in the AV policies.

AV processing can be implemented on the files transmitted through protocols such as SMTP, HTTP and POP3.But to apply for SMTP and POP3 protocols, it takes to much resource and affect the system performance. So we apply it on HTTP Protocol.

2.7.1.1.4. Routing:

For the network is simple, static route is configured in the router. Default static route to woredanet and one static route to internal network.

2.7.1.1.5. QOS:

The router is also configured with QOS to classify & prioritize traffic in such a way that Network control traffic gets the highest priority, then VoIP traffic then videoconference traffic and all other considered with default priority.

2.7.1.1.6 Traffic Flow:

The network is designed in such a way that traffic going out to MPLS network will go through EPON uplink as the primary link . All the traffic will automatic switch to wireless bridge in the case of EPON link failure .Traffic switch over will be done dynamically. Once EPON link failure is detected, it will route the traffic to wireless bridge and it will re-route back to EPON link once the link is back to normal.

2.8. Related works

The researcher review different research works which contains the following concepts. Network intrusion detection, data mining for network intrusion, and case based and rule based reasoning, integration of data mining with knowledge base system, integration of case based and rule based reasoning.

Tigabu [9] designed a semi-supervised intrusion detection system and he recommended designing knowledge base system which will add adaptability and extensibility features for intrusion detection. The experiments were conducted following the Knowledge Discovery in Database process model. The Knowledge Discovery in Database process model starts from selection of the datasets. The dataset used in this study has been taken from Massachusetts Institute of Technology Lincoln laboratory. The system has a prediction accuracy of 96.11% on the training datasets and 93.2% on the test dataset to classify the new instances as normal, DOS, U2R, R2L and probe classes. .

Azeb [46] attempted a research project for her master‘s thesis on how a rule based reasoning method can be integrated with the existing case based system called DrillEdge in order to improve the system‘s performance in handling complex situations in simpler and accurate ways. The research included studying how they can be integrated, and designing, implementing and testing of a demo system that demonstrates the integration. But Azeb [46] did not include the integration of data mining with the integrated reasoning approaches on her system. Abdulkerim [42] attempted to integrate data mining with knowledge based system to design a rule-based intrusion detection system. His system is aiming at utilizing hidden knowledge extracted by employing induction algorithm of data mining, specifically JRip from sampled KDDcup_99 intrusion data set. These system has scored 80.5% overall performance and the researcher recommends further investigations on different algorithms of the same case and the combination/integration of different reasoning systems to refine the knowledge base and boost the advantages of integrating data mining induced knowledge with knowledge based system.

Dawit [27] conduct a research on an integration of data mining with case based reasoning system for network intrusion detection. The system is aiming at utilizing hidden knowledge extracted by employing descriptive algorithm of data mining, specifically K-means cluster from sampled KDDcup‘99 intrusion dataset. Clustered case with centroid value is mapped to COLIBRI Studio

IDE and then the integrator application creates using Eclipse IDE with JDK 6. The system achieves 89.5% accuracy and further studies suggested to be done to improve the retrieval process by applying ontology based retrieval and integration of case based reasoning with rule based reasoning.

Jaiganesh [25] conducted a Ph.D dissertation on the Investigation of machine learning algorithms for network intrusion detection system, in this work; a data mining based hybrid intrusion detection system has been designed for distinguishing normal and intrusive events. The major contributions of this research work were the proposal of a hybrid architectural framework for effective intrusion detection. Therefore, this study focuses exploring a way to automatically use a knowledge acquired using descriptive and predictive mining models for network intrusion detection systems that combines case-based reasoning and rule-based reasoning.

2.9. Discussion of Proposed algorithm

SSL is a halfway method between supervised and unsupervised learning, which, in addition to unlabeled data, receives some supervision information such as the association of the targets with some of the examples. Collective Classification for Text Classification poses as an interesting method for optimizing the classification of partially-labelled data. Collective classification is a combinatorial optimization problem, in which we are given a set of documents, or nodes, D = {d1,...,dn} and a neighbourhood function N, where Ni ⊆D\{Di}, which describes the underlying network structure [12]. Being D a random collection of documents, it is divided into two sets X and Y where X corresponds to the documents for which we know the correct values and Y are the documents whose values need to be determined. Therefore, the task is to label the nodes Yi ∈ Y with one of a small number of labels, L = {l1,...,lq} [43] and [44].

- **The advantages of semi-supervised learning over supervised and unsupervised learning**

Obviously, we are working with a labeled dataset when we are building (typically predictive) models using supervised learning. The goal of unsupervised learning is often of exploratory nature (clustering, compression) while working with unlabeled data.

In semi-supervised learning, we are trying to solve a supervised learning approach using labeled data augmented by unlabeled data; the number of unlabeled or partially labeled samples is often larger than the number of labeled samples, since the former are less expensive and easier to obtain. So, our goal is to overcome one of the problems of supervised learning – having not enough labeled data. Adding cheap and abundant unlabeled data, we are hoping to build a better model than using supervised learning alone.

In collective classification the class labels of multiple instances are inferred simultaneously, assuming dependencies between these instances. Thus, the class label of a particular instance depends on the class labels and sometimes even attributes of the other related instances and not just on its own set of attributes. [46]

Meta Filtered Collective Classifier - A meta classifier that takes a filter and a collective classifier as input. The filter is only trained on the provided training set, but still applied to instances from the training and test set, as well as to any instance that gets passed to the meta classifier. [45]

CHAPTER THREE

METHODS AND APPROCHES

Today, intrusion detection is one of the high priority and challenging tasks for network administrators and security professionals. For an intrusion detection system, it is important to detect previously known attacks with high accuracy. However, detecting previously unseen attacks is equally important in order to minimize the losses as a result of a successful intrusion. It is also equally important to detect attacks at an early stage in order to minimize their impact.

In this study, supervised (which is created from descriptive data mining models) and unsupervised (which is created from predictive data mining models) learning models are selected and more performed network intrusion detection is designed.

3.1. Data Mining Functionalities

The kinds of patterns that can be discovered depend upon the data mining tasks employed. By and large, there are two types of data mining tasks:

- Descriptive data mining tasks that describe the general properties of the existing data, and
- Predictive data mining tasks that attempt to do predictions based on inference on available data.

The data mining functionalities and the variety of knowledge they discover are briefly presented in the following list:

• Characterization: It is the summarization of general features of objects in a target class, and produces what is called characteristic rules. The data relevant to a user specified class are normally retrieved by a database query and run through a summarization module to extract the essence of the data at different levels of abstractions. For example, one may wish to characterize the customers of a store who regularly rent more than 30 movies a year. With concept hierarchies on the attributes describing the target class, the attribute oriented induction method can be used to carry out data summarization. With a data cube containing summarization of data, simple OLAP operations fit the purpose of data characterization.

• Discrimination: Data discrimination produces what are called discriminant rules and is basically the comparison of the general features of objects between two classes referred to as the target class and the contrasting class. For example, one may wish to compare the general characteristics of the

customers who rented more than 30 movies in the last year with those whose rental account is lower than 5.

The techniques used for data discrimination are similar to the techniques used for data characterization with the exception that data discrimination results include comparative measures.

• Association analysis: Association analysis studies the frequency of items occurring together in transactional databases, and based on a threshold called support, identifies the frequent item sets. Another threshold, confidence, which is the conditional probability than an item appears in a transaction when another item appears, is used to pinpoint association rules. This is commonly used for market basket analysis.

• Classification: It is the organization of data in given classes. Classification uses given class labels to order the objects in the data collection. Classification approaches normally use a training set where all objects are already associated with known class labels. The classification algorithm learns from the training set and builds a model.

• Prediction: Prediction has attracted considerable attention given the potential implications of successful forecasting in a business context. There are two major types of predictions: one can either try to predict some unavailable data values or pending trends, or predict a class label for some data. The latter is tied to classification. Once a classification model is built based on a training set, the class label of an object can be foreseen based on the attribute values of the object and the attribute values of the classes. Prediction is however more often referred to the forecast of missing numerical values, or increase/ decrease trends in time related data. The major idea is to use a large number of past values to consider probable future values.

• Clustering: Similar to classification, clustering is the organization of data in classes. However, unlike classification, in clustering, class labels are unknown and it is up to the clustering algorithm to discover acceptable classes. Clustering is also called unsupervised classification, because the classification is not dictated by given class labels. There are many clustering approaches all based on the principle of maximizing the similarity between objects in a same class (intra-class similarity) and minimizing the similarity between objects of different classes (inter-class similarity).

• Outlier analysis: Outliers are data elements that cannot be grouped in a given class or cluster. Also known as exceptions or surprises, they are often very important to identify. While outliers can be considered noise and discarded in some applications, they can reveal important knowledge in other domains, and thus can be very significant and their analysis valuable.

• Evolution and deviation analysis: Evolution and deviation analysis pertain to the study of time related data that changes in time. Evolution analysis models evolutionary trends in data, which consent to characterizing, comparing, classifying or clustering of time related data. Deviation analysis, on the other hand, considers differences between measured values and expected values, and attempts to find the cause of the deviations from the anticipated values. [9] and [10]

3.2. Methods and Approaches of this research

As the researcher reviewed in section 2.1.1.2, based on the different approaches to event analysis, IDS can be classified as Signature based detection and Anomaly detection. In this research a hybrid NIDS approach was used.

A hybrid intrusion detection system is a system which is a combination of both signature based and anomaly based IDS. A signature-based IDS analyzes the network traffic looking for patterns that match a library of known signatures. These signatures are composed by several elements that allow identifying the traffic. On the other hand anomaly-based IDSs try to find suspicious activity on the system. In the initial phase, the IDS must be trained in order to get an idea about what is considered “normal” and “intrusion”. After that, the system will inform about any suspicious activity if there is a deviation from normal.

Since the researcher needs to develop better performed data mining model, therefore semi-supervised data mining technique is used to share advantage from unlabeled data set. WEKA 3.8 data mining tools with collective-classification WEKA package to execute semi-supervised techniques and expertise are utilized as means to address the research problem. The overall design issues are represented diagrammatically as shown in figure.

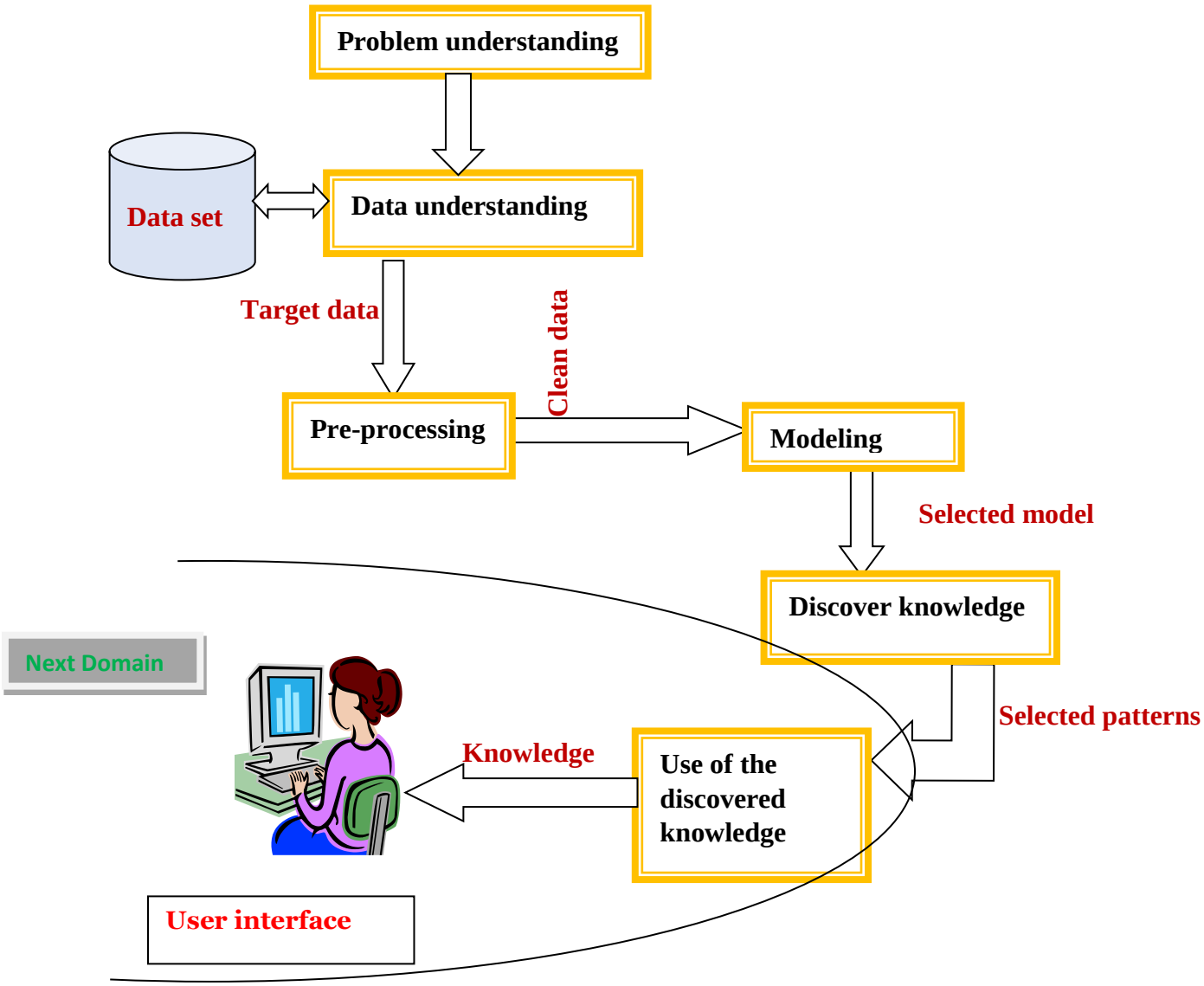

Figure 11. The overall design issues

3.2.1. Problem Understanding

This initial step has been thoroughly attempted to understand the driving force of Intrusion detection system. To accomplish this target, various tasks have been performed such as closely working with domain experts in order to define the problem and determine the research goals, identifying key people and learning about current solution to the problem, learning domain-specific terminology and preparation of a description of the problem are considered as a means of solving the problem.

3.2.2. Data understanding

Once the data is organized, a selection process occurs where some subset of this data becomes the target data upon which further analysis is performed. It is important when creating this target data

that the data analyst understands the domain, the end users' needs, and what the data mining task might be.

3.4.3. Pre-processing

Sometimes data is collected in an ad hoc manner. Data entry mistakes can occur and/or the data may have missing or unknown entries. During the data cleaning and preprocessing stage noise is removed from the data. Outliers and anomalies in the data can pose special problems for the data analyst during the data cleaning process. Care must be taken not to remove these types of outliers and anomalies. This step in the process can be the most time consuming. The data preprocessing step in this study was including basic operations, such as removing noise or outliers if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, and accounting for time sequence information and known changes, as well as deciding database management system issues, such as data types, schema, and mapping of missing and unknown values. Also, since a predictor can exploit only certain data features, it is important to detect which data preprocessing works best [46].

3.4.4. Transformation

Data Reduction and Coding step employs transformation techniques that are used to reduce the number of variables in the data by finding useful features with which to represent the data. The transformed data is used in the data mining step.

3.4.5. Choosing Data mining tasks

DM methods was applied for solving classification problems in many applications [47]. In DM, algorithms (learners) try to automatically filter the knowledge from example data (datasets).This knowledge can be used to make predictions about original data in the future and to provide insight into the nature of the target concept(s). According to Pradeep [13] the example data typically consists of a number of input patterns or examples to be learned. DM systems typically attempt to discover regularities and relationships between features and classes in learning or training phase.

For analyzing the data and classification of network attacks from a network environment, the two machine learning algorithms [48], the Semi-Supervised Learning (SSL) Collective classification model called meta.Filtered Collective Classifier and J48 decision tree classifiers are used in this thesis. Meta.Filtered Collective Classifier is the proposed collective classifier having both supervised and unsupervised characteristics that helps to give weight for the missing value for the masked feature and predict the unlabeled class. J48 decision tree classifier is an algorithms which was preferable existed

model to predict whether the newly coming instance is attack or not as compared to others ordinary classifiers.

The researcher tried to conduct experiments on semi-supervised collective classifier called YATSI (Yet Another Two State Idea), which is successful for missing value but can't use for predicting the unlabeled class. In addition to YATSI, Meta collective wrapper was also used for conducting the experiment, but its result is less than 60% accuracy, therefore these two algorithms are failed for further investigation.

3.4.6. Semi-Supervised Learning (SSL) meta.Filtered Collective Classifier model

SSL is a halfway method between supervised and unsupervised learning, which, in addition to unlabeled data, receives some supervision information such as the association of the targets with some of the examples.

3.4.6.1 meta.Filtered Collective Classifier model

Collective Classification for Text Classification poses as an interesting method for optimizing the classification of partially-labelled data. Collective classification is a combinatorial optimization problem, in which we are given a set of documents, or nodes, D = {d1,...,dn} and a neighborhood function N, where Ni ⊆D\{Di}, which describes the underlying network structure [12]. Being D a random collection of documents, it is divided into two sets X and Y where X corresponds to the documents for which we know the correct values and Y are the documents whose values need to be determined. Therefore, the task is to label the nodes Yi ∈ Y with one of a small number of labels, L = {l1,...,lq}. [43]

In collective classification the class labels of multiple instances are inferred simultaneously, assuming dependencies between these instances. Thus, the class label of a particular instance depends on the class labels and sometimes even attributes of the other related instances and not just on its own set of attributes. [46]

In this study for Semi-Supervised Learning (SSL) Collective classification model called meta.Filtered Collective Classifier modeling is used.

Meta.Filtered Collective Classifier – As the name implies a meta classifier that takes a filter and a collective classifier as input. The filter is only trained on the provided training set, but still applied to instances from the training and test set, as well as to any instance that gets passed to the meta classifier. Missing values are replaced with the Replace Missing Values filter. Since the collective

classifiers should get built using labeled and unlabeled dataset, they cannot be run in the usual Classify tab in the Explorer. Hence the package provides a custom tab, to perform experiments with the collective classifiers, called "Collective". It is a lot simpler compared to the Classify tab, but still offers the following evaluation options: Cross-validation, Percentage split (random or order preserved) and Supplied test set. [45].

3.4.6.1 Decision Tree

Decision tree is a predictive modeling technique most often used for classification in DM. The Classification algorithm is inductively learned to construct a model from the pre classified dataset. Each data item will defined by values of the attributes. The Decision tree will classify the data item using the values of its attributes. The decision tree is initially constructed from a set of pre-classified data. The main approach is to select the attributes, which best divides the data items into their classes [49]. In this study the J48 decision tree algorithms is used.

3.4.7. Modeling

It is in this step that the actual search for patterns of interest is performed. The search for patterns is done within the context of the data mining task and the representational model under which the analysis is being performed. The data mining task itself can be a classification task, linear regression analysis, rule formation, or cluster analysis.

3.4.8 Evaluation of the discovered knowledge

This step includes understanding of the results and checking the novelty and interestingness of the discovered knowledge as well as its impact. For performance evaluation: Accuracy and recall, full testing dataset of which is taken from the Ethiopian Institute of Agricultural Research pass though the developed model to detect the intrusions and find the detection error rate, precision, false positive rate, average misclassification cost and accuracy of the detection models but for comparison of models, misclassification cost, false positives rate and accuracy of detection is used as a major performance measurement.

3.4.9. Architecture of the study

Supervised intrusion detection approaches use only labeled data for training. To label the data however are often difficult, expensive, or time consuming as they require the efforts of experienced domain experts. Semi-supervised learning addresses this problem by using large amount of

unlabeled data, together with the labeled data, to build better classifiers and Semi-Supervised learning requires less human effort.

The architecture used for this thesis is shown in figure 12. This architecture was proposed by Pachghare [50] for the semi-supervised approach for intrusion detection system. As showed in figure 12, labeled data is used for training the system as supervised approach. After training, the system is tested using unlabeled data. The tested data will add to the training data so as to implement semi- supervised approach.

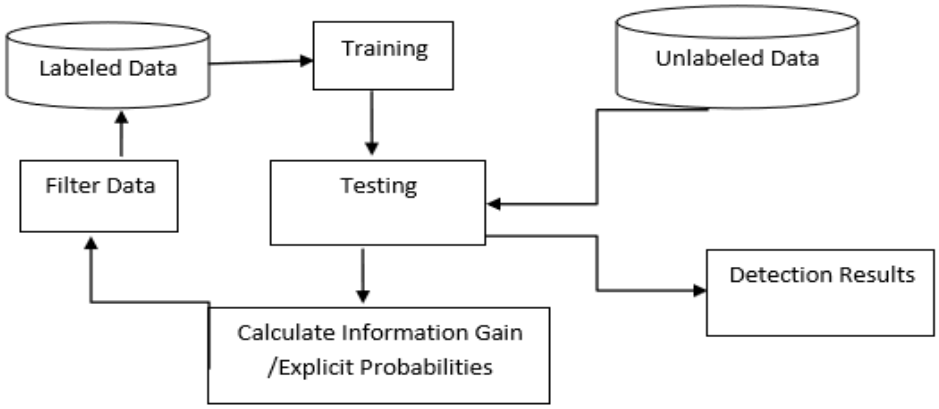

Figure 12 Architecture proposed for Semi-Supervised IDS

3.4.10. Implementation tool

The research was conducted for preparing the data from Ethiopian Institute Agricultural Research (EIAR) Information Communication Data Center and using WEKA(3.8.0) with Semi-supervised Collective classification package data mining tools which are powerful, user friendly and freely available for noncommercial purpose.

3.4.10.1. Installation of Semi-supervised Collective classification package

- First of all, you need to have last version of WEKA.
- Second, you have to download the collective package from the link. [51]
- Third, from WEKA GUI-> Tools -> Package manager, install it from the "Unofficial" section.
- Finally, restart WEKA to find the "Collective" tab displayed with other tabs.

3.4.11. Testing Procedure

The selected data was preprocessed and used the data mining software package and the network intrusion detection model was constructed. Finally the constructed model validated by feeding real life records into the data mining experimenter package.

CHAPTER FOUR

DATASET PREPARATION

4.1. Overview

In this chapter the researcher showed that how data is prepared for the purpose of the experiment. As showed in chapter one, under the methodology section of this thesis, the Data Mining process model selected is Semi-Supervised Learning (SSL) Collective classification model called meta.Filtered Collective Classifier data mining model which starts from understanding the business and then selection of data.

The data collected for this research is from Ethiopian Agricultural Research data center that was in comma separated value csv format. The dataset initially had 43 attributes and 25192 records but after the preprocessing stage, it was reduced to 28 attributes and 12596 records for building the predictive model. The data was extracted to Microsoft Excel-2013 for preprocessing purpose. Since data preparation is critical for the neatness and accuracy of the result I offered much more time and effort in this research.

4.2. Data Understanding and Cleaning

According to **[57]**, data understanding phase mainly focuses on creating a target dataset with selected sets of variables that is relevant to discovery process. Without understanding the existing data, it is difficult to draw the target dataset from the original, since the world data is unclean and not appropriate at the source to run mining process. In practice, it has been generally found that data cleaning and preparation takes approximately 80% of the total data engineering effort. Data preparation is, therefore, a crucial research topic. However, much work in the field of data mining was built on the existence of quality data [52]. The networking Attacks were classified as per the activities done by the attacker. Each attack type comes under one of the following four main categories [53]

1. Probe. Probing is a class of attacks where an attacker scans a network to gather information for the purpose of exploiting known vulnerabilities. An attacker with a map of machines and services that are available on a network can use the information to look for exploits. There are different types of probes: some of them abuse the computer's legitimate features; some of them use social

engineering techniques. This class of attacks is the most common and requires little technical expertise.

2. Denial of Service Attacks. Denial of Service (DoS) is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate users access to a system. There are different ways to launch DoS attacks: by abusing the computers' legitimate features; by targeting the implementation bugs; or by exploiting the system's misconfigurations. DoS attacks are usually classified based on the service(s) that an attacker renders unavailable to legitimate users.

3 User to Root Attacks. User to root or user to super-user (U2Su) exploits are a class of attacks where an attacker starts out with access to a normal user account on the system and then exploits vulnerability to gain root access. Most common exploits in this class of attacks are regular buffer overflows, which are caused by regular programming mistakes and incorrect environment assumptions.

4 Remote to User Attacks. Remote to local (R2L) is a class of attacks where an attacker sends packets to a machine over a network, then exploits the system's vulnerability to illegally gain local access as a user. There are different types of R2L attacks; the most common attacks in this class are done using social engineering.

5. Normal connections (Normal) are produced by pretending daily user behavior such as downloading files, and visiting web pages.

The researcher used available intrusion detection data sets from EIAR Data center. For the case of this research the researcher has taken 25192 records which are labeled. From this dataset the researcher used to filter by resample and prepared 55% of the total records, which is 12596 instances as unlabeled/test set.

The distributions of the data sets before preprocessing are shown in Table 2:

Dataset	Number of Records Collected
Normal	6730
Attacks	5866
Total	12596

Table 2 Distribution of the dataset

The distribution of attacks before processing is summarized in Table 3

Dataset Label	Number of Records Collected	Percent (%)
DOS	3529	60.16
R2L	1113	18.97
Probe	391	6.67
U2Su	86	1.47
Total	5866	

Table 3 Summary of the distribution of attacks before processing

The distributions of the data sets after processing are shown in Table 4:

Dataset	Number of Records Collected
Normal	3327
Attacks	2268
Unlabeled	7001
Total	12596

Table 4 Distribution of the dataset

4.3. Data Transformation and Feature Selection

Feature selection is the process of removing features from the data set that are irrelevant with respect to the task that is to be performed [54]. Feature selection can be extremely useful in reducing the dimensionality of the data to be processed by the classifier, reducing execution time and improving predictive accuracy (inclusion of irrelevant features can introduce noise into the data, thus obscuring relevant features). It is worth noting that even though some machine learning algorithms perform some degree of feature selection themselves (such as classification trees);

feature space reduction can be useful even for these algorithms. Reducing the dimensionality of the data reduces the size of the hypothesis space and thus results in faster execution time.

Effective feature (attribute) selection from intrusion detection datasets is one of the important research challenges for constructing high performance IDS. Irrelevant and redundant attributes of intrusion detection dataset may lead to complex intrusion detection model as well as reduce detection accuracy.

Feature selection methods are commonly used for cost insensitive learning and found to be very helpful because the elimination of useless features enhances the accuracy of detection while speeding up the computation, thus improving the overall performance of IDS [60].

This is to get a minimum set of best attributes for classification. Related attributes during problem understanding were selected for predictive model building. This is because considering the effects of predictors (independent variables) with predicted (outcome variable) whether they are linearly associated or not. As **[57]** datasets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task; these possibly lead to do with domain expert to pick out some of the useful attributes by ignoring the irrelevant attributes. These in fact, maximize the quality of mining result as well as speed up the algorithm in mining process. But the question is, how a set of a good subset of the original attributes one can possibly find. One can filter the best and worthy features using tests of statistical significance **[57]**. This means that selecting best features based on their significance level potentially explain the class attributes (Normal or Attack). If the attributes are dependent to each other, they may spoil the predictive accuracy of the model/classifier. Therefore, identifying how far the independent variables are correlated with each other is becoming a pre-requisite of intrusion mining process. Thus, attributes were checked through multi-co linearity analysis for their interdependences in order to make the outcome of interest free of confusion by which the outcome is really affected.

4.4. Evaluation Metrics

A cost matrix (C) is defined by associating classes as labels for the rows and columns of a square matrix **[57]**; in the current context for the dataset, there are five classes, {NORMAL, PROBE, DOS, U2Su, R2L},and therefore the matrix has dimensions of 5×5. An entry at row i and column j, C (i, j), represents the non-negative cost of misclassifying a pattern belonging to class i into class j. Cost matrix values employed for the dataset are defined **[57].** These values were also used for

evaluating results of the data set computation. The magnitude of these values was directly proportional to the impact on the computing platform under attack if a test record was placed in a wrong category. A confusion matrix (CM) is similarly defined in that row and column 5×5 matrix for the dataset. An entry at row i and column j, CM (i, j), represents the number of misclassified patterns, which originally belong to class i yet mistakenly identified as a member of class j.

The form of the cost matrix C will depend on the actual application. In general, it is reasonable to choose the diagonal entries equal to zero, i.e. C (i, j) = 0 for i = j, since correct classification normally incurs no cost as shown in table 4. In addition the size of the cost matrix should be the same as that of the confusion matrix.

	Observed vs. Predicted				
Cost(i,j)	**normal**	**probe**	**DOS**	**U2R**	**R2L**
Normal	0	2	2	2	2
Probe	2	0	2	2	2
DOS	2	2	0	2	2
U2R	2	2	2	0	2
R2L	2	2	2	2	0

Table 5 The 5X5 cost matrix used for the KDD 1999 winner result **[55]**

4.4.1. Standard Metrics to Evaluate Intrusion

To evaluate the approach, the four standard metrics of true positive, true negative, false positive and false negative developed for network intrusions, have been used. Table 6 shows these standard metrics.

Confusion metrics (standard metrics)		Predicted connection label	
		Normal	Intrusions (Attacks)
Actual Connection	Normal	True Negative (TN)	False Alarm (FP)
	Intrusions (attacks)	False Negative (FN)	Correctly detected Attacks(TP)

Table 6 Standard metrics for evaluations of Intrusions (attacks)

The representation of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are defined as follows:

- ✓ True Positive (TP): The number of malicious records that are correctly identified.
- ✓ True Negative (TN): The number of legitimate record that correctly classified.
- ✓ False Positive (FP): The number of records that are incorrectly identified as attacks however in fact they are legitimate activities.
- ✓ False Negative (FN): The number of records that are incorrectly classified as legitimate activities however in fact they are malicious.

4.4.2. Performance Measure

General performance of intrusion detection systems is measured in terms of numbers of selected features and the classification accuracies of the machine learning algorithms giving the best classification results. As discussed by [56] there are different techniques used for performance measuring of the IDS. Good IDS require high detection rate, low false alarm rate and lower average misclassification cost [62]. Thus during developing IDS; overall classification accuracy (OCA), detection rate (DR), false Positive rate (FPR), average misclassification cost (AMC), Error rate, and training and testing time are considered.

4.4.2.1. Error Rate

The error rate, which is only an estimate of the true error rate and is expressed to be a good estimate, if the number of test data is large and representative of the population, is defined as: [57]

Error Rate= [(Total Test samples - Total Correctly Classified Samples)*100%] /Total Test Samples ……. (4.1)

4.4.2.2. Accuracy

Overall Classification accuracy (OCA) is the most essential measure of the performance of a classifier. It determines the proportion of correctly classified examples in relation to the total number of examples of the test set i.e. the ratio of true positives and true negatives to the total number of examples. From the confusion matrix, we can say that accuracy is the percentage of correctly classified instances over the total number of instances in total test dataset, namely the situation TP and TN, thus accuracy can be defined as follows: [63]

$$\text{Accuracy} = [(TP+TN)*100\%]/TP+TN+FP+FN \quad \ldots\ldots. (4.2)$$

4.4.2.3 Detection Accuracy

Detection accuracy (rate) refers to the proportion of attack detected among all attack data, namely, the situation of TP, thus detection rate is defined as follows: [63]

$$\text{Detection Accuracy} = (TP*100\%)/ (TP+FN) \quad \ldots\ldots (4.3)$$

4.4.2.4. False Positive rate

False positive rate, also known as False Alarm Rate (FAR) measures the number of misclassified positive instances in relative to the total number of misclassified instances. It can be expressed also as the proportion that normal data is falsely detected as attack behavior, namely, the situation of FP. Thus false alarm rate is defined as follows: [58]

$$\text{False Positive Rate} = (FP*100\%)/ (FP+TN) \quad \ldots.. (4.4)$$

4.4.2.5. Precision and Recall

Recall and precision are two widely used metrics employed in applications where successful detection of one of the classes is considered more significant than detection of the other classes [59].

4.4.2.5.1. Precision

Precision is the number of class members classified correctly over the total number of instances classified as class members. Technically can be expressed as the attack has been occurred and the IDS detect correctly. [65]

$$\text{Precision} = (TP*100\%)/ (TP+FP)\ldots\ldots. (4.5)$$

4.4.2.5.2. Recall

Recall (also called True Positive Rate) (TPR)), Recall measures the number of correctly classified examples relative to the total number of positive examples. In other words the number of class members classified correctly over the total number of class members. [65]

$$\text{Recall} = (TP*100\%)/ (TP+FN) \quad \ldots\ldots. \quad (4.6)$$

4.4.2.6. Subjective evaluation

In addition to the above objective performance evaluation the researcher also used the domain experts as subjective evaluation measurement. For checking the rules and other related issues the researcher has consulted with EIAR department of ICT director and network experts. The domain experts agreed with the most determinant attributes which are selected during the design of intrusion detection model in this study.

CHAPTER FIVE

EXPERIMENTATION

In this chapter the researcher describes experimental study of the algorithms and procedures, which are discussed in the previous chapters. The researcher used both labeled and unlabeled data set. In this study different experiments were conducted using various data mining methods to derive knowledge from preprocessed data to predict unseen network attacks.

5.1. Experimentation Design

The data analysis and classification was carried out using a computer with the configurations Intel(R) Core(TM)i5 5200U CPU 2.2GHz, 4 GB RAM, and 64 bit operating system Microsoft Windows 8.1 Enterprise and a Weka (3.8.0 version) software environment with collective-classification-weka-package-master which is developed for evaluating Semi-Supervised Learning. Weka has collections of machine learning algorithms for data mining tasks that contain facilities for data preprocessing, classification, regression, clustering, association rules, and visualization. [60]. For WEKA the default memory value is 128m for maxheap. As an option it is possible to upgrade to 1Gig, because of the memory heap problem during the experiment.

The data set collected from EIAR Information Communication Data Center that was in comma separated csv format and the data was extracted to Microsoft Excel-2013 for preprocessing purpose. The dataset initially had 43 attributes and 25192 records but after the preprocessing stage, it was reduced to 28 attributes and 12596 records for building the appropriate predictive model. The dataset obtained after preprocessed were loaded in weka tool and filtered by resample and prepared 12596 instances for unlabeled/test. Therefore, the researcher used these clean, filtered and properly selected features for this research.

As described above the researcher used collective-classification-weka-package-master, which is developed by University of Waikato for SSL semi-supervised data evaluation. The package provides three options to partition the dataset. These are: preparing distinct files for training dataset and test dataset “Unlabeled” and “Test”; cross validation with possibility of setting variety number of folds (the default was 10 fold) and Random split. Ordinary J48 classification with 10-

fold cross validation has been also used for this research. In order to increase the accuracy of prediction and to reduce biasness. [61]

5.2. Semi-Supervised Modeling

We consider the general problem of learning from labeled and unlabeled data. Given a point set X = {x1,...,xl, xl+1,...,xn} and a label set L = {1,...,c}, the first l points have labels{y1,...,yl} ∈ L and the remaining points are unlabeled. The goal is to predict the labels of the unlabeled points. The performance of an algorithm is measured by the error rate on these unlabeled points only. Such a learning problem is often called semi-supervised or transductive. Since labeling often requires expensive human labor, whereas unlabeled data is far easier to obtain, semi-supervised learning is very useful in many real-world problems and has recently attracted a considerable amount of research [62] and [39].

Supervised intrusion detection approaches use only labeled data for training. To label the data however are often difficult, expensive, or time consuming as they need skilled man power. Meanwhile unlabeled data may be relatively easy to collect, but there has been few ways to use them. In this study semi-supervised learning addressed this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers. Semi-supervised learning (SSL) addressed in this thesis by allowing the model to integrate part or all of the available unlabeled data in its supervised learning [17].

The goal is to maximize the learning performance of the model through such newly-labeled samples while minimizing the work required of human effort.

In semi-supervised learning the training data (observations, measurements, etc.) are accompanied by both labels and unlabeled records. Label records indicating the class of the observations. Those unlabeled records show that the class is empty. New data is classified based on the training set. Classification is one of the categories under Semi-supervised learning.

In a classification task of machine learning each instance of a dataset is assigned into a particular class. A classification based IDS attempts to classify all traffic as either normal or nominal. The challenge in this case is to minimize the number of false positives (classification of normal traffic as nominal) and false negatives (classification of nominal traffic as normal). In this research, for the classification modeling experiments, meta.Filtered Collective Classifier is used.

5.2.1 Meta Filtered Collective Classifier modeling

Meta Filtered Collective Classifier - A meta classifier that takes a filter and a collective classifier as input. The filter is only trained on the provided training set, but still applied to instances from the training and test set, as well as to any instance that gets passed to the meta classifier. [15]

Since the collective classifiers should get built using labeled and unlabeled dataset, they cannot be run in the usual Classify tab in the Explorer. Hence the package provides a custom tab, to perform experiments with the collective classifiers, called "Collective". It is a lot simpler compared to the Classify tab, but still offers the following evaluation options:

• Cross-validation

• Percentage split (random or order preserved)

• Supplied test set. [15]

Fig 13 shows snap shoot of collective Classifier uploaded Weka (3.8.0) version.

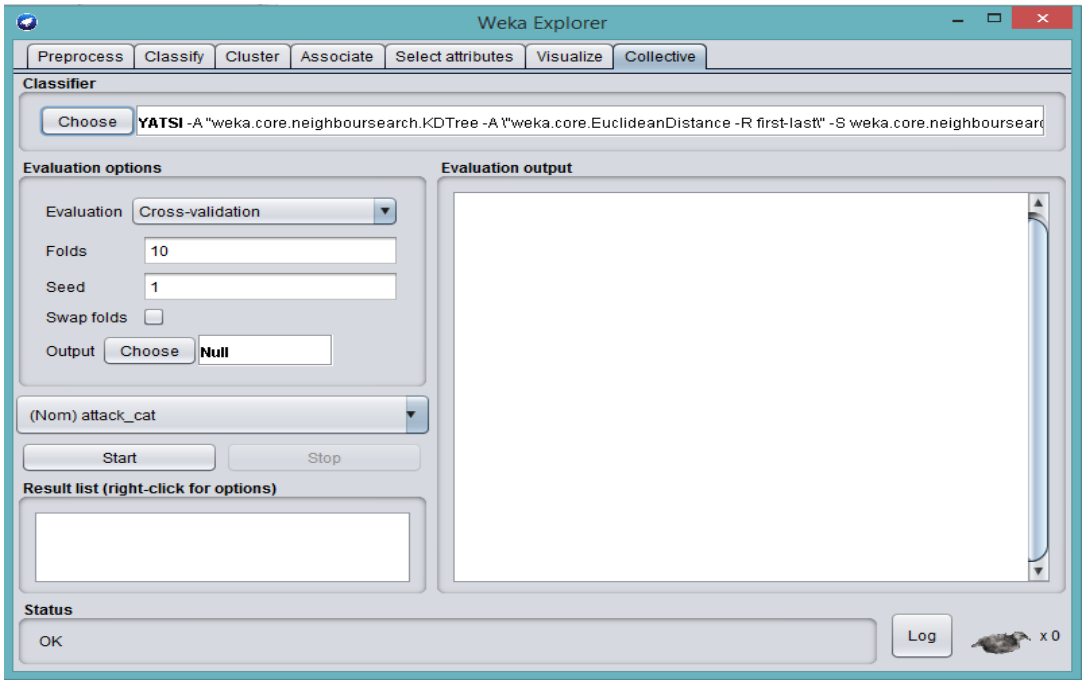

Figure 13 Collective Classifier uploaded Weka (3.8.0) version.

The package provides three options to partition the dataset. These are:

- Preparing distinct files for training dataset and test dataset "Unlabeled" and "Test".
- Cross validation with possibility of setting variety number of folds (the default was 10 fold) and
- Random split with different possibilities of testing sample (the default was 10% of selected sample training the remaining for testing including both labeled and unlabeled data).

In this research the researcher tried to show the performance of Cross validation and Random split of meta.Filtered Collective Classifier algorithm Verses Cross validation of the common types of j48 decision tree algorithm with their default value and more other option.

meta.Filtered Collective Classifier algorithm contains some parameters that can be changed to further improved classification accuracy. Initially the classification model is built with the default parameter values of the meta.Filtered Collective Classifier algorithm.

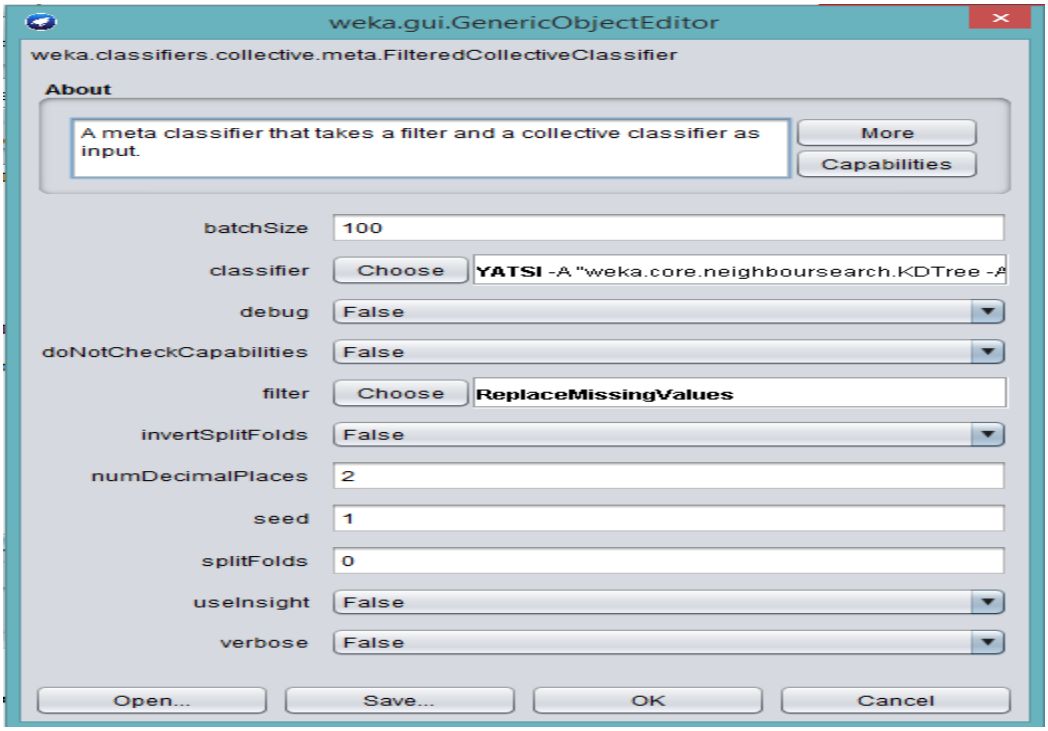

Figure 14 Default parameters with their values for the **meta.Filtered Collective Classifier** algorithm.

As shown on **Fig 14** meta.Filtered Collective Classifier algorithm uses YATSI as a classifier and replace missing values by filtering.

For conducting this research the researcher used two types of data set. The first data set is the data set of 55% of unlabeled class with missing value, all experiments are evaluated. The second data set is 65% of unlabeled class with no missing value, here the better performed proposed collective classifier and better performed existed ordinary j48 decision tree with 10 fold default value are evaluated.

The purpose of using missing value is to show the capability of the algorithm to detect those intrusion with masked features. For the case of this research the dataset has 12596 instances and each instances at a normal condition has 28 attributes, but 55% of the attributes for each instances are missing value including the class attribute, to detect those masked features (see Appendix 2).

On the other hand the purpose of using dataset of with no missing value is to evaluate the algorithms with full feature (see Appendix 3).

5.2.1.1. Experiment I:

The first experiment was performed with a meta.Filtered Collective Classifier default parameter of 10-fold cross validation test option.

meta.Filtered Collective Classifier collective classifier used more than 12 of the total 28 variables to generate the result. Some of these are: flag, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate,

diff_srv_rate, dst_host_serror_rate, dst_host_srv_serror_rate, and dst_host_rerror_rate. **Table 7** depicts the resulting accuracy matrix of this model.

Total number of instances (training sets)	Correctly classified Instances	Incorrectly Classified Instances
5595	5300	295

Table 7 Semi-supervised classification accuracy using meta.Filtered Collective Classifier parameters with 10 fold cross validation with seed=1.

As shown in the result, the meta.Filtered Collective Classifier algorithm scored an accuracy of 94.7 %. This result shows that out of the total datasets of 12596 only 5595(44%) instances are labeled and the other 7001(56%) instances are unlabeled "ignored". From 5595 known instances 5300 (94.7 %) records are correctly classified, while 295(5.3 %) of the records are incorrectly classified.

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.995	0.065	0.957	0.995	0.976	0.940	0.771	0.440	Normal
0.973	0.015	0.961	0.973	0.967	0.954	0.908	0.444	DoS
0.861	0.015	0.840	0.861	0.850	0.836	0.948	0.373	R2L
0.259	0.001	0.843	0.259	0.396	0.460	0.877	0.266	Probe
0.000	0.000	0.000	0.000	0.000	0.000	0.789	0.157	U2R
W. Avg. 0.947	0.044	0.937	0.947	0.937	0.913	0.828	0.428	

Table 8 Detail Accuracy by Class of Semi-supervised classification meta.Filtered Collective Classifier parameters of 10 fold cross validation with seed=1.

The researcher was also conducted the research by changing the default value of the above 10 fold cross validation seed to 2, the result also changed as shown below in **Table 9**

Total number of instances (training sets)	Correctly classified Instances	Incorrectly Classified Instances
5595	5300	295

Table 9 Semi-supervised using meta.Filtered Collective Classifier parameters with 10 fold cross validation with seed=2 accuracy.

As shown on the table 10 fold validation with changing seed=2, the meta.Filtered Collective Classifier scored similar result with seed = 1

Therefore, Semi-supervised meta.Filtered Collective Classifier parameters of 10 fold cross validation with default value was selected for further analysis. **Table 10** depicts the confusion matrix for meta.Filtered Collective Classifier parameters with 10 fold cross validation of the default value.

Classified as	Normal	DoS	R2L	Probe	U2R	Sum
Normal	3310	15	2	0	0	3327
DoS	42	1538	1	0	0	1581
R2L	45	13	409	8	0	475
Probe	37	11	75	43	0	166
U2R	23	23	0	0	0	46

Table 10 Confusion matrix for Semi-supervised meta.Filtered Collective Classifier parameters with 10 fold cross validation of the default value algorithm.

The above values of the confusion matrix in table 10 depicts, out of the total 5595 instances was provided to the algorithm, 5300 (94.7%) records were classified correctly and the remaining 295 (5.3%) were classified incorrectly. The result of this table also indicates that 15 and 2 records from actual class Normal were classified as DoS and R2L classes respectively, while 42 and 1 records

from class Normal and R2L respectively are wrongly classified as DoS class. 45, 13 and 8 records from Normal, Dos and probe respectively are wrongly classified as R2L. Again 37, 11 and 75 records from class Normal, DoS and R2L are wrongly classified as Probe. Also shows 23 and 23 records from class Normal and DoS respectively are wrongly classified as U2R. In addition to this the table shows, more misclassified records in all classes the reason is that there is unbalanced data in the data set.

5.2.1.2. Experiment I I:

The second experiment is conducted, by changing the default testing option (the 10-fold cross validation) with Random Split (Percentage Split). In this learning scheme a percentage split is used to partition the dataset into training and testing set. The purpose of using this parameter was to assess the performance of the learning scheme by increasing the proportion of testing dataset if it could achieve better classification accuracy than the first experimentation. In this research the researcher investigated in three different option of percentage split, the default percentage 10%, 50% and 70%.

The first research is conducted with the default percentage of 10%, which means the algorithm partition the data randomly in to 10% of the total data set as training.

Total number of labeled instances before training	Total number of labeled instances after training	Correctly classified Instances	Incorrectly Classified Instances
5595	5013	4672	341

Table 11 Semi-supervised classification accuracy using meta.Filtered Collective Classifier parameters with 10% Random split.

The result buffer shows the total number of instance before training and after training varies, because from the total data set with 10% randomly split and 5013 labeled instances are included for training. From 5013 labeled instances 4672(93.2%) instances are correctly classified and 341 instances are incorrectly classified (6.7%). In this case the unlabeled (unknown) instances are 6323.

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.996	0.106	0.932	0.996	0.963	0.907	0.758	0.424	Normal
0.920	0.013	0.965	0.920	0.942	0.920	0.885	0.427	DoS
0.809	0.013	0.852	0.809	0.830	0.815	0.919	0.336	R2L
0.367	0.003	0.764	0.367	0.495	0.520	0.805	0.245	Probe
0.073	0.000	1.000	0.073	0.136	0.269	0.730	0.241	U2R
W.Avg.0.932	0.068	0.930	0.932	0.925	0.886	0.809	0.410	

Table 12 Detail Accuracy of Semi-supervised classification meta.Filtered Collective Classifier parameters with Random split scheme of 10%.

As observed from **Table 12** the accuracy of using 10% Random selectin scheme of Semi-supervised classification meta.Filtered Collective Classifier is 93.2%.

The second option for investigation was changing the default value of meta.Filtered Collective Classifier parameters with Random split scheme of 50%.

Total number of labeled instances before training	Total number of labeled instances after training	Correctly classified Instances	Incorrectly Classified Instances
5595	2825	2678	148

Table 13 Semi-supervised classification accuracy using meta.Filtered Collective Classifier parameters with 50% Random split.

Here on the **Table 13** the result buffer shows the total number of instance after training decreases from 5013 to 2825 and the unlabeled (newly coming) instances are also decreases from 6323 to 3473. From the total data set with 50% randomly split 2825 labeled instances are included for training. From 2825 labeled instances 2678(94.8%) instances are correctly classified and 141 (5.2%) instances are incorrectly classified. Although the newly coming instances are decreased,

the accuracy of Random split meta.Filtered Collective Classifier parameters with 50% increased from 93.2% to 94.8%.

The other option for investigating the meta.Filtered Collective Classifier Random split scheme was 70%.

Total number of labeled instances before training	Total number of labeled instances after training	Correctly classified Instances	Incorrectly Classified Instances
5595	1663	1590	73

Table 14 Semi-supervised classification accuracy using meta.FilteredCollectiveClassifier parameters with 70% Random split.

Table 14 shows similarly, the total number of instance after training decreases from 2825 to 1663 and the unlabeled (newly coming) instances are also decreases from 3473 to 2116. From the total data set with 70% randomly split 1663 labeled instances are included for training. From 1663 labeled instances 1590(95.6%) instances are correctly classified and 73 instances are incorrectly classified (4.4%). The accuracy of Random split meta.Filtered Collective Classifier parameters with 70% is 95.6%. Therefore from the above experiment the researcher generalized as the random split percentage increases the labeled instances and the newly coming unlabeled also decreases, while the accuracy of the model increases.

Therefore Semi-supervised classification accuracy using meta.Filtered Collective Classifier parameter with 70% Random split having an accuracy of 95.6% is better performed as compared to the above experiments. So this model is selected for further analysis.

5.2.1.3. Experiment I I I:

The third experiment is conducted on Semi-supervised classification accuracy using meta.Filtered Collective Classifier with Unlabeled /Test set. In this learning scheme the researcher used full data set as training and test set. Therefore the total labeled classes used to train and helps to predict the newly coming unlabeled intrusion.

Total number of labeled instances	Correctly classified labeled Instances	Incorrectly Classified Instances
5595	5385	210

Table 15 Semi-supervised classification accuracy using meta.Filtered Collective Classifier parameters with fully Training/Test set.

Table 15 shows the total number of labeled instance are 5595. From the total number of training set 5385 instances are correctly classified and 210 instances are incorrectly classified. The detail accuracy of the model has been showed in the following **table 16.**

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.994	0.042	0.972	0.994	0.983	0.958	0.773	0.442	Normal
0.984	0.012	0.969	0.984	0.977	0.968	0.911	0.449	DoS
0.912	0.012	0.878	0.912	0.895	0.885	0.963	0.391	R2L
0.464	0.001	0.928	0.464	0.618	0.649	0.929	0.361	Probe
0.239	0.000	1.000	0.239	0.386	0.487	0.922	0.450	U2R
W.Avg 0.962	0.029	0.962	0.962	0.958	0.941	0.834	0.437	

Table 16 Detailed Accuracy by Class using Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test set.

As shown on the above table, the weighted average of TP(True Positive) rate and FP(False Positive) rate is 0.962 and 0.029 respectively. Which implies that the accuracy for Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test set is 96.2%.

Therefore, from the above experiments using Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test set has been chosen due to its better overall classification with accuracy 96.2% and FP(False Positive) rate is 0.029.

As reviewed on different literature many other researches are done on supervised data set and their result indicates ordinary J48 decision tree with default parameter of 10-fold cross validation was performed better to predict the newly coming intrusion. Therefore the researcher was needed to further investigation on the above better performed proposed Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test set and ordinary J48 decision tree with pruned parameter.

5.2.2. J48 decision tree modeling

5.2.2.1 Experimentation IV

In this experiment the ordinary J48- decision tree cross validation was tasted. In the experiment the cross validation with default value has been used. In this experiment the J48 decision tree with pruned and minobj value of 0.25 were used. From the experiment, using the default value, the tree was generated with number of leaves of 101 and size of the tree 192. The experimental result showed that the flag is the root node for the classification. This indicates that flag is the most determinant factor to identify the classes of a given intrusion. The accuracy of the model has been showed in the following **Table 17.**

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.998	0.143	0.911	0.998	0.953	0.881	0.773	0.441	Normal
0.937	0.005	0.986	0.937	0.961	0.947	0.912	0.456	DoS
0.674	0.006	0.907	0.674	0.773	0.765	0.967	0.398	R2L
0.440	0.000	1.000	0.440	0.611	0.658	0.964	0.300	Probe
0.413	0.000	1.000	0.413	0.585	0.641	0.941	0.384	U2R
W.Avg.0.932	0.087	0.935	0.932	0.927	0.881	0.836	0.437	

Table 17 Detailed Accuracy by Class using Semi-Supervised J48 algorithm parameters with their default values- 10 fold cross validation.

As showed in the above **Table 17**, the J48 learning algorithm scored an accuracy of 93.2 %. This result shows that out of the total number of training datasets 93.2 % records are correctly classified.

5.2.2.2 Experimentation V

Using the un-pruned and min-obj value of 0.2 the experimentation result has changed. In this experiment the number of leaves and the size of the tree were increasing to 1023 and 1220 respectively when we compared to the previous result. The accuracy of the model was shown in the following T**able 18.**

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.998	0.108	0.931	0.998	0.964	0.909	0.775	0.443	Normal
0.957	0.005	0.986	0.957	0.971	0.961	0.912	0.454	DoS
0.768	0.008	0.903	0.768	0.830	0.819	0.969	0.411	R2L
0.428	0.000	1.000	0.428	0.599	0.648	0.968	0.310	Probe
0.413	0.000	1.000	0.413	0.585	0.641	0.950	0.430	U2R
W.Avg. 0.945	0.066	0.947	0.945	0.941	0.906	0.837	0.439	

Table 18 Detailed Accuracy by Class using Semi-Supervised J48 algorithm parameters with Other Confidence factor -pruned with cf 0.2.

As shown in T**able 18**, the J48 learning algorithm scored an accuracy of 94.5 %. This result shows that out of the total datasets 94.5 % records are correctly classified and 5.5% records were not correctly classified.

In summary, when the J48 decision tree with the un-pruned parameter has been applied the performance of the models is increased. Hence, J48 with un-pruned creates a model with a better performance of 94.5 % with number of leaves of 1023 and size of the tree 1220. But, this result is worst as compared to the above experiment using Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test set having an accuracy of 96.2% and FP (False Positive) rate is 0.029.

5.2.3. Comparison of Semi-Supervised meta.Filtered Collective Classifier and J48 decision tree model

Comparing different classification techniques and selecting the best model for predicting the network intrusions is one of the aims of this study. Accordingly the Semi-Supervised Learning collective classifier particularly meta.Filtered Collective Classifier and the J48 decision tree classification approaches were used for conducting experiments.

Summary of experimental result for the two classification algorithms is presented in T**able 19** below:

Classifier/ Model	Test Mode	Accuracy	
		Correctly Classified	Incorrectly Classified
Semi-supervised meta.Filtered Collective Classifier	10-fold cross validation with Other default values.	94.7	5.3
	10-fold cross validation with seed value=2	94.7	5.3
	Random split with 10% values	93.2	6.8
	Random split with 50% values	94.8	5.2
	Random split with 70% values	95.6	4.4
	Fully Training/Test set value	96.2	3.8
J48 : Supervised	10-fold cross validation with Other default values(pruned) value	93.2	6.8
	10-fold cross validation with unpruned (-pruned)	94.5	5.5

Table 19 Summary of experimental result for the Semi-supervised meta.Filtered Collective Classifier and J48 decision tree model classification algorithms.

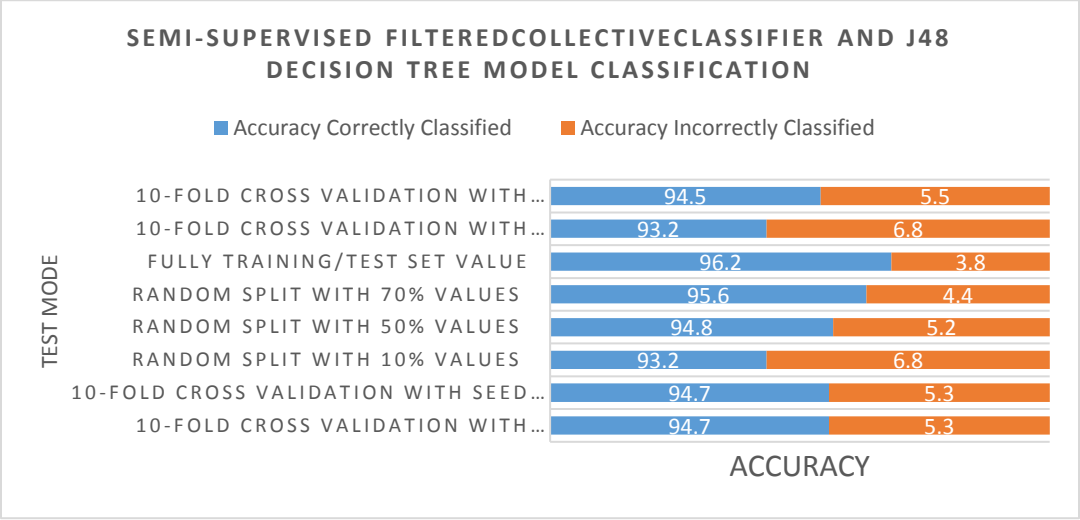

Figure 15 chart of accuracy result for the Semi-supervised **meta.Filtered Collective Classifier** and J48 decision tree model classification algorithms.

The detailed classification accuracy for the algorithms conducted in this Research, are shown in **Table 20** below.

Classifiers	Accuracy	Classes									
		Normal		DOS		R2L		Probe		U2R	
		TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
Filtered Collective Classifier with 10 fold cross validation	94.7	0.995	0.065	0.973	0.015	0.861	0.015	0.259	0.001	0.000	0.000
Filtered Collective Classifier with 10 fold cross validation with 2 seed	94.7	0.995	0.065	0.973	0.015	0.861	0.015	0.259	0.001	0.000	0.000
Filtered Collective Classifier with Random split 10%	93.2	0.996	0.106	0.920	0.013	0.809	0.013	0.367	0.003	0.073	0.000
Filtered Collective Classifier with Random split 50%	94.8	0.996	0.070	0.960	0.013	0.897	0.015	0.322	0.001	0.000	0.000
Filtered Collective Classifier with percentage split 70%	95.6	0.998	0.058	0.968	0.011	0.915	0.013	0.375	0.001	0.000	0.000
Filtered Collective Classifier full Training/Test	96.2	0.994	0.042	0.984	0.012	0.912	0.012	0.464	0.001	0.239	0.000
J48 with 10 fold cross validation	93.2	0.998	0.143	0.937	0.005	0.674	0.006	0.440	0.000	0.413	0.000
J48 with other confidence factors	94.5	0.998	0.108	0.957	0.005	0.768	0.768	0.428	0.000	0.413	0.000

Table 20 Comparison of the confusion matrix result for Semi-supervised meta.Filtered Collective Classifier and J48 decision tree model classification algorithms.

As shown on **Table 20** from all the above eight experiments, the Semi-Supervised meta.Filtered Collective Classifier parameters with full Training/Test set performed better for identifying intrusions either normal or attack (DOS, R2L, Probe and U2R).

The reason for the Semi-Supervised meta.Filtered Collective Classifier model parameters with fully Training/Test set performed better than J48 decision tree is due to the characteristics of the algorithm, which is the combination of different algorithms developed for solving semi-supervised learning. In addition to this, the proposed algorism had the capacity of filling the missing value by filtering and give weight by neighboring value. As shown in **Appendix 2**: Attributes relation, data declaration and sample of data, the data set that I have used was more than 55 % unlabeled class with missing value, therefore the experiment shows that for more unlabeled data set the more training data set is better. That is why Semi-Supervised meta.Filtered Collective Classifier parameters with full Training/Test set performed better. The selected model had highest accuracy of 96.2%, an average highest True Positive Rate (TP), an average lowest False Positive Rate (FP). A comprehensible segregation point that can be defined by the algorithm to predict the class of a particular network intrusion makes the model to be efficient.

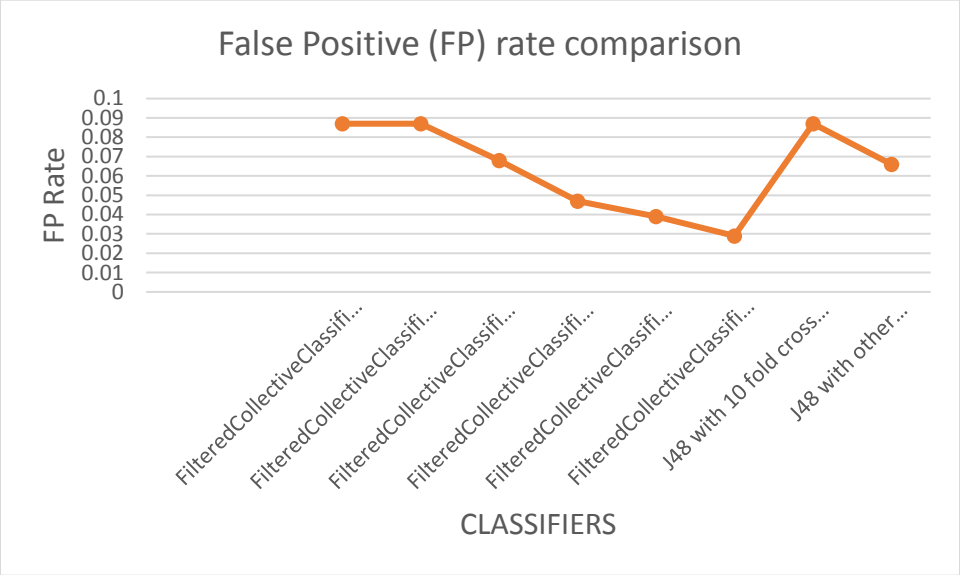

Figure 16 False Positive (FP) rate comparison of the YATSI collective classifier and J48 decision tree Algorithms.

5.2.4. Using the data set of 65% of unlabeled class with no missing value.

Here, the researcher needs further investigation on the above proposed best performed Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test and the existed ordinary J48 decision tree with pruned parameter by using the data set of 65% of unlabeled class

with no missing value to see the effect of the missing value and non-missing value dataset and how algorithms performance are varied to detect an intrusion with masked and full features.

5.2.4.1. Experimentation VI

In this experiment the researcher conducted by using unlabeled class with no missing value of the proposed Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test.

Total number of labeled instances	Correctly classified labeled Instances	Incorrectly Classified Instances
4426	4417	9

Table 21 Semi-supervised classification accuracy using meta.Filtered Collective Classifier parameters with fully Training/Test set using unlabeled class with no missing value.

Table 21 shows the total number of labeled instance are 4426. From the total number of training set 4417 instances are correctly classified and only 9 instances are incorrectly classified. The remaining 8170 instances are unlabeled, but the model uses the labeled instance and predict the majority instances with 99.8% accuracy. The detail accuracy of the model has been showed in the following **table 22.**

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	0.004	0.997	1.000	0.999	0.997	0.757	0.352	Normal
1.000	0.000	1.000	1.000	1.000	1.000	0.901	0.364	DoS
0.997	0.000	0.997	0.997	0.997	0.997	0.969	0.322	R2L
0.984	0.000	0.992	0.984	0.988	0.987	0.990	0.319	Probe
0.793	0.000	1.000	0.793	0.885	0.890	0.998	0.359	U2R
W.Avg. 0.998	0.002	0.998	0.998	0.998	0.997	0.824	0.352	

Table 22 Detailed Accuracy by Class using Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test set using unlabeled class with no missing value.

Therefore, from the above experiments Semi-Supervised meta.Filtered Collective Classifier parameters with fully Training/Test set using unlabeled class with no missing value has an. accuracy of 99.8% and FP (False Positive) rate is 0.002.

5.2.4.2. Experimentation VII

As shown on **Table 22** bellow, ordinary J48 classifier of default 10-fold cross validation by using unlabeled class with no missing value dataset, the tree has been generated with number of leaves of 11 and size of the tree 12. The experimental result showed that the flag is the root node for the classification. This indicates that flag is the most determinant factor to identify the classes of a given intrusion.

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1.000	0.000	1.000	1.000	1.000	1.000	0.757	0.352	Normal
1.000	0.000	1.000	1.000	1.000	1.000	0.901	0.364	DoS
1.000	0.000	1.000	1.000	1.000	1.000	0.969	0.321	R2L
1.000	0.000	1.000	1.000	1.000	1.000	0.989	0.314	Probe
1.000	0.000	1.000	1.000	1.000	1.000	0.998	0.337	U2R
W. Avg.1.000	0.000	1.000	1.000	1.000	1.000	0.824	0.352	

Table 23 Detailed Accuracy by Class Semi-Supervised J48 algorithm parameters with their default values- 10 fold cross validation using unlabeled class with no missing value.

As shown in T**able 23**, the J48 learning algorithm scored an accuracy of 100 %. This result shows that out of the total datasets all records are correctly classified and no records were incorrectly classified.

In summary, although the accuracy of the above two algorithms shows insignificant values, when the J48 decision tree with the default parameter has been applied the performance of the models is 100 % accurate. Therefore when the data set with no missing value (full features) used J48 algorithm is preferable than Semi-Supervised meta.Filtered Collective Classifier. But, it is obviously known that the real world data is not complete due to different factors.

5.3. Evaluation of the Discovered Knowledge

From all the experiments in this study, two models are achieved better classification performance. From those experiments the Semi-Supervised meta.Filtered Collective Classifier parameters with full Training/Test Set model gives a better classification accuracy of predicting newly arriving intrusions with their masked features in their respective class category by using

missing value real data set and ordinary J48 algorithm model with default value for full feature intrusions using no missing value data set. Some of the rules generated from the selected model, J48 algorithm model with default value for no missing value data set, are the following.

Rule 1. If flag = OTH or if flag = SF then it is Normal.

Rule 2. IF flag = REJ then the attack is R2L.

Rule 3. If flag = RSTO, if flag = RSTOS0 or if flag = RSTR then the attack is Probe.

Rule 4. If flag = S0 then the attack is DoS.

Rule 5. If flag = S1, if flag = S2, if flag = S3 or if flag = SH then the attack is U2R.

In this study as shown the above generated rules, most of the attributes used for constructing the selected predictive model are from flag. The others important attributes in this research which are mostly used in the construction of rules includes srv_rerror_rate, serror_rate, dst_host_srv_rerror_rate, dst_host_srv_serror_rate, srv_serror_rate, dst_host_serror_rate, dst_host_rerror_rate, rerror_rate, same_srv_rate, and count.

From the total of 28 attributes the rules generated by the selected model, J48 algorithm model with default value for no missing value data set, are more than 11 attributes are used for generating the tree. From these attributes flag is the most determinant variable. For checking the rules and other related issues the researcher has consulted with EIAR department of ICT director and network experts. The domain experts agreed with the most determinant attributes which are selected during the design of intrusion detection model in this study. The experts said that features like flag, protocol_type, and srv_rerror_rate, should have to be check in order to say a given network packet is either normal traffic or attack. As the domain experts said that, most of the time the attack is DOS if the protocol_type is icmp (Internet Control Message protocol) and the only additional attribute is service with any value.

In the EIAR firewall, the configured protocols are including Transmission Control Protocol (TCP), UDP (User Datagram Protocol), ICMP (Internet Control Message Protocol), IPSec (Internet Protocol Security), and Point to Point Transmission Protocol (PPTP).

From the above sample generated rules some of the rules are prevailing (that is known rules) and some of them are interesting rules (that is new rules). For deciding these rules, the researcher consulted the domain expert from EIAR.

In EIAR there is no a data mining intrusion detection system. EIAR has used firewalls as a network security tool for safe guarding their networks. That is why I have forced to modify my data from the data base of EIAR as a research requires.

The selected model, Semi-Supervised meta.Filtered Collective Classifier parameters with full Training/Test Set model has highest accuracy of 96.2 % for data set with 55% missing value and ordinary J48 algorithm with default value model records highest accuracy of 100% for data set with no missing value. This result showed, the proposed meta.Filtered Collective Classifier is better algorithm to scan masked intrusion and predict the newly coming instances weather it is masked or not. While the existed ordinary J48 algorithm is better for the full features intrusion, but intruders are always tried to mask some features of them to attack the system, therefore this research has significant impact for intrusion detection.

5.4. Summary

In spite of the wide growth of information technology, security has remained one challenging area for computer and networks. The numbers of hacking and intrusion incidents are increasing year on year as technology rolls out. With the Internet playing a vital role in incessant communication, its effectiveness can diminish owing to effects called intrusions. Intrusion is an activity that adversely affects the targeted system. An intrusion may compromise the integrity, confidentiality and availability of resources of the attacked system. There are different ways of detecting and preventing intruders in the network. Among the methods used to detect and Data Mining (DM) system is one of the most wildly used.

Intrusion detection systems are security management systems that are used to discover inappropriate, incorrect, or anomalous activities within computers or networks. With the rapid growth of Internet, these malicious behavior are increasing at a fast pace and can easily cause millions of dollar in damage to an organization. Hence, the development of intrusion detection systems has been set with the highest priority by government, research institutes and commercial corporations. During the past years, existing intrusion detection systems take a variety of approaches to the task of detecting intruders' activities. For developing the systems, data are

collected and then provided for the use of overall design process. However, these data sources do have some problems such as problem of irrelevant and redundant features, problem of uncertainty, and problem of ambiguity. These problems not only hinder the detection speed but also decline the detection performance of intrusion detection systems. Data mining is to identify valid, novel, potentially useful, and ultimately understandable patterns in massive data. It is demanding to apply data mining techniques to detect various intrusions.

In the last several years, some exciting and important advances have been made in intrusion detection using data mining techniques. Research results have been published and some prototype systems have been established. Inspired by the huge demands from applications, the interactions and collaborations between the communities of security and data mining have been boosted substantially.

In this study, attempts have been made to use DM technology with the aim of detecting and predicting intrusions in the networking industry. This study undertakes Semi-Supervised Learning of Collective-classifier, which is developed for unlabeled data Set. The data set in this study is taken from EIAR data center network appliance. After taking the data, it has been preprocessed. The major preprocessing activities include fill in missed values, remove outliers; resolve inconsistencies.

The model that was created using Semi-Supervised meta.Filtered Collective Classifier parameters with full Training/Test set showed the best classification accuracy of 96.2% for data set having unlabeled class with missing value. On the other hand the ordinary J48 algorithm parameters with their default values- 10 fold cross validation using unlabeled class with no missing value data set showed best classification accuracy of 100%. The findings of this study have shown that the data mining methods generates interesting rules that are crucial for intrusion detection and prevention in the networking industry.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The model that was created using Semi-Supervised meta.Filtered Collective Classifier parameters with full Training/Test set showed the best classification accuracy of 96.2% for dataset having unlabeled class with missing value. The algorithm has a capacity of unsupervised filtering and give weights for the missing value and train by using labeled data to predict and classify the unlabeled new instances as Normal, DOS, U2R, R2L and probe classes. On the other hand the ordinary J48 algorithm parameters with their default values- 10 fold cross validation using unlabeled class with no missing value data set showed best classification accuracy of 100%. Therefore, the proposed, Semi-Supervised meta.Filtered Collective Classifier model parameters with full Training/Test set is preferable for those newly coming instances for masking their feature and the existed J48 10-fold cross validation with default value is good if the newly coming instances are for full feature. The findings of this study have shown that the data mining methods generates interesting rules that are crucial for intrusion detection and prevention in the networking industry.

From the above experiments, the result gap of using missing value and non-missing value is narrow for the proposed algorithms as compared to ordinary J48. This is because the proposed algorithms doesn't leave or ignore each missing value, rather it calculate the neighborhood value and give weight for the missing value, then it predict the unlabeled class and records an accuracy of 96.2 % using the missing value and 99.8% using no missing value.

On the other hand the result gap of using missing value and non-missing value for ordinary J48 algorithm is wider as compared to the proposed algorithm. This is because when using missing value dataset the algorithm will ignore or give '0' for the missing value and predict the unlabeled class and records an accuracy of 94.7 % using the missing value and 100% using no missing value. That is why the proposed Meta filtered collective classifier is more performed than the existed J48 algorithm for masked feature intrusion detection.

Generally, the results from this study can contribute towards an improving the networking security of EIAR data center. The study has shown that it is promising to identify those network intrusions either normal or attacks (DOS, U2R, Probe and R2L) and put forward concrete mechanisms for detecting and preventing them using the appropriate Data mining approaches.

6.2. Recommendations

In this research two models are selected, Semi-Supervised **meta.Filtered Collective Classifie**r parameters with full Training/Test set has better result than other classification algorithms by using data set having unlabeled class with missing value and **ordinary J48** algorithm parameters with their default values- 10 fold cross validation using unlabeled class with no missing value data set.

1. I recommend Semi-Supervised meta.Filtered Collective Classifier parameters with full Training/Test set for real world Network Intrusion Detection, because real world data is not complete by its nature and intruders are also mask their feature to attack systems and files.
2. Since ordinary J48 algorithm parameters with their default values- 10 fold cross validation is efficient for unlabeled class with no missing value data set, it is recommended for research purpose and instances for full features.
3. This study was carried out using collective classification algorithms such as three types of meta.Filtered Collective Classifie**r** including selected model, meta.Filtered Collective Classifie**r** parameters with full Training/Test set, and J48 decision tree parameters with their default values- 10 fold cross validation. So further investigation needs to be done using other classification algorithms such as Others Collective Classification, Neural Networks and Support Vector Machine to explore to what extent the performance of NIDS improved.
4. Conduct similar researches to experiment on more size data set and more than 55% unlabeled and missing value data set.
5. Conduct similar researches on Ethiopian governmental and non-governmental organization do IDS research like INSA and others research institute to generalize about IDS in the country.
6. Further investigation should be done to change the network intrusion detection in to intrusion prevention system.
7. I recommend EIAR to implement the proposed, Semi-Supervised meta.Filtered Collective Classifier parameters with full Training/Test set NIDS model to detect the newly coming instances.

REFERENCES

[1] APEC, PEC Strategy to Ensure Trusted, Secure and Sustainable Online Environment, 2005.

[2] M. Boncheva, Problems of Engineering Cybernetics and - iit.bas.bg, 2007.

[3] Eric Bloedorn, Alan D. Christiansen, William Hill, Clement Skorupka, Lisa M. Talbot, Data Mining for Network Intrusion Detection: How to Get Started, Jonathan Tivel The MITRE Corporation 1820 Dolley Madison Blvd. McLean, VA 22102 (703) 983-5274.

[4] Emam, Ahmed Youssef and Ahmed, NETWORK INTRUSION DETECTION USING DATA MINING AND NETWORK BEHAVIOUR ANALYSIS, Department of Information Systems, King Saud University, Riyadh, KSA: DOI : 10.5121/ijcsit, 2011.

[5] Anh Nguyen and Huy Deokjai, Application of Data Mining to Network Intrusion Detection: Classifier Selection Model, Choi Chonnam National University, Computer Science Department, 300 Yongbong-dong, Buk-ku, Gwangju 500-757, Korea: Springer-Verlag Berlin H, 2008.

[6] Krishna Kant Tiwari, Susheel Tiwari and Sriram Yadav , Intrusion Detection Using Data Mining Technique, Millennium institute of technology, RGPV University,.

[7] EIAR, "www.eiar.gov.et," [Online]. Available: http://www.eiar.gov.et/index.php/about, 11/8/ 2017. [Accessed 11 December 2017].

[8] M., Helali, Data Mining Based Network Intrusion Detection System, A Survey in Novel Algorithms and Techniques in Telecommunications and Networking," Vols. PP.501-505, 2010.

[9] D., Tigabu, Constructing Predictive Model for Network Intrusion Detection, Addis Ababa University, Addis Ababa: Unpublished M.Sc thesis, 2012.

[10] T., Adamu, COMPUTER NETWORK INTRUSION DETECTION: MACHINE LEARNING APPROACH, Addis Ababa University, Addis Ababa: Unpublished Masters Thesis, 2010.

[11] M., Zewdie, Optimal feature selection for Network Intrusion Detection: A Data Mining Approch, Addis Ababa University, Addis Ababa, Ethiopia: M.Sc thesis, School of Information Science, 2011.

[12] B., Sterry, Data Mining Methods for Network Intrusion Detection, University of California, 2004.

[13] A., Meseret, A Combined Reasoning System For Knowledge Based Network Intrusion Detection, Addis Ababa University School Of Graduate Studies, School Of Information Science: MSc Thesis, 2016.

[14] Miller, S.K, An Introduction to Computer Security, *IEEE Computer,* vol. vol. 38, p. no.34, 2001.

[16] M., Karen S. and Peter, Guide to Intrusion Detection and Prevention Systems, National Institute of Standards and Technology, Department of Commerce, USA, 2007.

[17] F., Carl, Intrusion Detection and Prevention, McGraw-Hill, Osborne Media, 2003.

[18] Jaiganesh, Investigation on machine learning algorithms for network intrusion detection system, Department of Computer Science & Engineering, Manonmaniam Sundaranar University: Ph.D Dissertation, July, 2014.

[19] Abdilkerim M, Towards Integrating Data Mining with Knowledge Based System: The Case of Network Intrusion Detection, School of Information Science Addis Ababa University, Addis Ababa: MSc Thesis, 2013.

[20] H., Dawit, Integrating Descriptive Modelling with Case Based Reasoning in Network Intrusion Detection Machine Learning, School of Information Science: Addis Ababa University: M.Sc Thesis, 2015.

[21] K. Sujatha, Data mining approach for hybrid intrusion detection system, Anna University, Faculty of Information and Communication Engineering: Ph.D Dissertation , 2014.

[22] Jain, Sanket R, Information Security: Intrusion Detection and Prevention System, 2014.

[23] Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, A.J., and Tan, P.N., Data Mining for Network Intrusion Detection, IEEE, 1987.

[24] I. A. T. R. IATR, "Intrusion Detection Systems," 6th ed, New York, 2009.

[25] Khandelwal, Knowledge based systems, problem solving competency and learnability, Suresh Gyan Vihar University, Department of Computer Science, 2014.

[26] P.Kalarani1, Dr.S. Selva Brunda2 Assistant Professor, Department of CT and IT, Kongu Arts and Science College, Erode, Tamil Nadu, India1 Professor and Head, Department of CSE, Sasurie College of Engineering, Erode, Tamil Nadu, India2, A Survey on Efficient Data Mining Techniques for Network Intrusion Detection System(IDS), *International Journal of Advanced Research in Computer and Communication Emgineering,* vol. Vol. 3, no. Issue 9, September, 2014.

[27] Choi, Huy Anh Nguyen and Deokjai, Application of Data Mining to Network Intrusion Detection: Classifier Selection Model, Chonnam National University, Computer Science Department, 300 Yongbong-dong, Buk-ku, Gwangju 500-757, Korea.

[29] C. D. a. A. M., "DDoS attacks and defense mechanisms: classification and state-of-the-art of Computer Networks," *the International Journal of Computer and Telecommunications Networking,* vol. Vol. 44, no. No.5, pp. PP. 643 - 666, 2004.

[30] Crothers M., Implementing Intrusion Detection Systems, a Hands-On Guide for Securing the Network, USA, 2002.

[31] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Sch¨olkopf,, "Learning with Local and Global Consistency," Max Planck Institute for Biological Cybernetics, 72076 Tuebingen, Germany.

[32] H., Dunham, Data mining introductory and advanced topics, Upper Saddle River: NJ: Pearson Education, Inc, 2003.

[33] J., Frank, "Artificial intelligence and intrusion detection: Current and future directions," in *In Proc. of the 17th National Computer Security Conference*, Baltimore, MD National Institute of Standards and Technology (NIST), 1994.

[34] P., Berkhin, Survey of clustering data mining techniques, San Jose, CA: Tech. rep., Accrue Software, 2002.

[35] Berson A., Smith S. and Thearling K,, Building Data Mining Applications for CRM, New York, USA: McGraw Hill Professional Publishing, 2000.

[36] S., Chaudhuri, "Data Mining and Database Systems: Where is the Intersection?," *IEEE Bulletin of the Technical Committee on Data Engineering,* vol. Vol. 21, no. No.1, pp. PP. 4-8, 1998.

[37] P., Thair, Survey of Classification Techniques in Data Mining, Hong Kong: Proceedings of the International Multi-Conference of Engineers and Computer Scientists, 2009.

[38] D., Denning, Information Warfare and Security, USA: Addison Wesley, ACM Press, 1999.

[39] Amir A., Ahmad H. and Hadi B., "A New System for Clustering and Classification of Intrusion Detection System Alerts Using SOM," *International Journal of Computer Science and Security,* vol. Vol. 4, no. No. 6, pp. PP. 589-597, 2011.

[40] V., Bro, "System for detecting network intruders in real-time," in *In Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, 1998.

[41] E., SURAFEAL, CONSTRUCTING A PREDICTIVE MODEL FOR NETWORK INTRUSION DETECTION: THE CASE OF UNIVERSITY OF GONDAR,, UNIVERSITY OF GONDAR FACULTY OF NATURAL AND COMPUTATIONAL SCIENCES, DEPARTMENT OF COMPUTER SCIENCE, 2013.

[43] Carlos Laorden, Borja Sanz, Igor Santos, Patxi Gal´an-Garc´ıa, and Pablo G. Bringas,, Collective Classification for Spam Filtering, DeustoTech Computing - S3Lab, University of Deusto Avenida de las Universidades 24, 48007 Bilbao, Spain.

[44] DOBRA, AMIT DHURANDHAR and ALIN, Collective vs Independent Classification in Statistical Relational Learning, University of Florida.

[45] Bernhard Pfahringer, Kurt Driessens, and Peter Reutemann, Collective and Semi-supervised classification, February 27, 2015.

[46] Meera G., Gandhi and Srivatsa S, "Adaptive Machine Learning Algorithm (AMLA) Using J48," vol. Vol. 3, pp. PP. 291-304, 2010.

[47] S., Pradeep, "Comparing the Effectiveness of Machine Learning Algorithms for Defect Prediction," *International Journal of Information Technology and Knowledge Management,* Vols. Vol.2, No.2, pp. PP.481-483, 2005.

[48] I., Eibe F. and Witten, Data Mining–Practical Machine Learning Tools and Techniques, 2nd Edition, Elsevier, 2005.

[49] T., Kruegel C. and Toth, Using Decision Tree to Improve Signature Based Intrusion Detection, USA: 6thSymposium on Recent Advances in Intrusion Detection (REID), Lecture Notes in Computer Science, Spring Verlag, 2003.

[50] Pachghare V., Vaibhav K.,and Parag K., Performance Analysis of Supervised Intrusion Detection System, JICA Special Issue on Network Security and Cryptography, NSC, 2011.

[51] https://github.com/fracpete/collective-classification-weka-package, "www.weka.com," [Online]. Available: https://github.com/fracpete/collective-classification-weka-package. [Accessed 18 Sebtember 2017].

[52] SHICHAOZHANG and CHENGQIZHANG, DATA PREPARATION FOR DATA MINING, Sydney, Australia: Faculty of Information Technology, University of Technology, 2003.

[54] R., Selvakani S. and Rajiesh, "Genetic Algorithm for framing rules for intrusion detection," *IJCSNS International Journal of Computer Science and network Security,* vol. Vol.7, no. No.11, 2007.

[55] E., Charles, “The foundations of cost-sensitive learning”, *In Proceedings of the Seventeenth International Joint Conference of Artificial Intelligence, Morgan Kaufmann,* p. PP. 973–978, 2001.

[56] Cios J, Pedrycz W, Swiniarski R, Kurgan W, and Lukasz A, Data Mining: A knowledge Discovery approach., New York: Springer-Verlag Science Business Media, 2001.

[57] P., Mrutyunjaya, "Evaluating Machine Learning Algorithms for Detecting Network Intrusions," *International Journal of Recent Trends in Engineering,* vol. Vol. 1, no. No 1, p. PP. 472 477, 2009.

[58] Farhan A., Zulkhairi M., Dahalin and Shaidah J., "Distributed and Cooperative Hierarchical Intrusion Detection on MANETs," *International Journal of Computer Applications,* vol. Vol. 12, no. No.5, pp. PP. 33-40, 2010.

[59] M, Han J and Kamber, Data Mining: Concepts and Techniques., New York. USA: Morgan Kaufmann, 2001.

[60] Data Mining Machine Learning Software. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/]. [Accessed 8 December 2017].

[61] Ragsdale D, Carver C, Humphries J, and Pooh U, "Adaptation techniques for intrusion detection and intrusion response systems," *in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2000.

[62] Seeger, M., "Learning with labeled and unlabeled data," Technical report The University of Edinburgh, 2000.

APPENDIXES

Appendix 1: List of Selected Features Available in EIAR data set.

S. No	Feature Name	Description	Data type
1	protocol_type	type of the protocol, e.g. tcp, udp, etc.	String
2	service	network service on the destination, e.g., http, telnet, etc.	String
3	flag	normal or error status of the connection	Integer
4	land	1 if connection is from/to the same host/port; 0 otherwise	Integer
5	logged_in	1 if successfully logged in; 0 otherwise	Integer
6	is_host_login	1 if the login belongs to the "hot" list; 0 otherwise	Integer
7	is_guest_login	1 if the login is a ``guest"login; 0 otherwise	Integer
8	count	number of connections to the same host as the current connection in the past two seconds	Integer
9	srv_count	number of connections to the same service as the current connection in the past two seconds	Integer
10	serror_rate	% of connections that have ``SYN" Errors	Integer
11	srv_serror_rate	% of connections that have ``SYN" Errors	Double
12	rerror_rate	% of connections that have ``REJ" Errors	Integer
13	srv_rerror_rate	% of connections that have ``REJ" Errors	Double
14	same_srv_rate	% of connections to the same service	Integer
15	diff_srv_rate	% of connections to different services	Integer
16	srv_diff_host_rate	% of connections to different hosts	Double
17	dst_host_count	count of connections having the same destination host	Integer
18	dst_host_srv_count	count of connections having the same destination host and using the same Service	Integer
19	dst_host_same_srv_rate	% of connections having the same destination host and using the same Service	Double
20	dst_host_diff_srv_rate	% of different services on the current Host	Double
21	dst_host_same_src_port_rate	% of connections having the same destination host and using the same Service port	Double
22	dst_host_srv_diff_host_rate	% of connections to the same service coming from different hosts	Double
23	dst_host_serror_rate	% of connections to the current host that have an S0 error	Double
24	dst_host_srv_serror_rate	% of connections to the current host and specified service that have an S0error	Double
25	dst_host_rerror_rate	% of connections to the current host that have an RST error	Double
26	dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an RST Error	Double

Appendix 2: Attributes relation, data declaration and sample of unlabeled data set with missing value data.

```
@relation EIARDATA
@attribute protocol_type {tcp,udp,icmp}
@attribute service
{aol,auth,bgp,courier,csnet_ns,ctf,daytime,discard,domain,domain_u,ech
o,eco_i,ecr_i,efs,exec,finger,ftp,ftp_data,gopher,harvest,hostnames,ht
tp,http_2784,http_443,http_8001,imap4,IRC,iso_tsap,klogin,kshell,ldap,
link,login,mtp,name,netbios_dgm,netbios_ns,netbios_ssn,netstat,nnsp,nn
tp,ntp_u,other,pm_dump,pop_2,pop_3,printer,private,red_i,remote_job,rj
e,shell,smtp,sql_net,ssh,sunrpc,supdup,systat,telnet,tftp_u,tim_i,time
,urh_i,urp_i,uucp,uucp_path,vmnet,whois,X11,Z39_50}
@attribute flag {OTH,REJ,RSTO,RSTOS0,RSTR,S0,S1,S2,S3,SF,SH}
@attribute land {0,1}
@attribute logged_in {0,1}
@attribute is_host_login {0,1}
@attribute is_guest_login {0,1}
@attribute count numeric
@attribute srv_count numeric
@attribute serror_rate numeric
@attribute srv_serror_rate numeric
@attribute rerror_rate numeric
@attribute srv_rerror_rate numeric
@attribute same_srv_rate numeric
@attribute diff_srv_rate numeric
@attribute srv_diff_host_rate numeric
@attribute dst_host_count numeric
@attribute dst_host_srv_count numeric
@attribute dst_host_same_srv_rate numeric
@attribute dst_host_diff_srv_rate numeric
@attribute dst_host_same_src_port_rate numeric
@attribute dst_host_srv_diff_host_rate numeric
@attribute dst_host_serror_rate numeric
@attribute dst_host_srv_serror_rate numeric
@attribute dst_host_rerror_rate numeric
@attribute dst_host_srv_rerror_rate numeric
@attribute class {normal,anomaly}
@attribute attack_cat {Normal,DoS,R2L,Probe,U2R}

@data
udp,?,?,?,0,?,0,3,4,0,?,?,?,?,?,0.5,255,?,?,?,0.01,0,?,?,0,?,?,?
tcp,?,REJ,?,?,?,?,?,19,?,?,?,?,?,0.06,?,255,19,?,0.06,0,?,?,?,?,?,?,R2
L
tcp,http,?,?,?,?,0,6,?,?,?,0,0,1,0,?,?,229,?,0.01,0,0,?,0,?,0,?,Normal
udp,domain_u,?,0,?,0,0,?,17,?,?,0,?,1,?,?,187,?,0.98,0.01,?,0,?,0,0,?,
?,?
tcp,private,?,?,0,0,?,127,?,1,?,0,0,0.04,0.06,?,?,14,?,0.08,?,?,?,?,?,
?,anomaly,DoS
icmp,eco_i,SF,0,?,0,?,1,43,0,?,0,?,?,?,1,3,?,1,?,?,0.25,?,?,0,?,anomal
y,?
```

Appendix 3: Attributes relation, data declaration and sample of unlabeled data set with no missing value data.

```
@relation EIARDATA
@attribute protocol_type {tcp,udp,icmp}
@attribute service
{aol,auth,bgp,courier,csnet_ns,ctf,daytime,discard,domain,domain_u,ech
o,eco_i,ecr_i,efs,exec,finger,ftp,ftp_data,gopher,harvest,hostnames,ht
tp,http_2784,http_443,http_8001,imap4,IRC,iso_tsap,klogin,kshell,ldap,
link,login,mtp,name,netbios_dgm,netbios_ns,netbios_ssn,netstat,nnsp,nn
tp,ntp_u,other,pm_dump,pop_2,pop_3,printer,private,red_i,remote_job,rj
e,shell,smtp,sql_net,ssh,sunrpc,supdup,systat,telnet,tftp_u,tim_i,time
,urh_i,urp_i,uucp,uucp_path,vmnet,whois,X11,Z39_50}
@attribute flag {OTH,REJ,RSTO,RSTOS0,RSTR,S0,S1,S2,S3,SF,SH}
@attribute land {0,1}
@attribute logged_in {0,1}
@attribute is_host_login {0,1}
@attribute is_guest_login {0,1}
@attribute count numeric
@attribute srv_count numeric
@attribute serror_rate numeric
@attribute srv_serror_rate numeric
@attribute rerror_rate numeric
@attribute srv_rerror_rate numeric
@attribute same_srv_rate numeric
@attribute diff_srv_rate numeric
@attribute srv_diff_host_rate numeric
@attribute dst_host_count numeric
@attribute dst_host_srv_count numeric
@attribute dst_host_same_srv_rate numeric
@attribute dst_host_diff_srv_rate numeric
@attribute dst_host_same_src_port_rate numeric
@attribute dst_host_srv_diff_host_rate numeric
@attribute dst_host_serror_rate numeric
@attribute dst_host_srv_serror_rate numeric
@attribute dst_host_rerror_rate numeric
@attribute dst_host_srv_rerror_rate numeric
@attribute class {normal,anomaly}
@attribute attack_cat {Normal,DoS,R2L,Probe,U2R}

@data
udp,domain_u,SF,0,0,0,0,3,4,0,0,0,0,1,0,0.5,255,250,0.98,0.01,0.01,0,0
,0,0,0,normal,Normal
tcp,private,REJ,0,0,0,0,235,19,0,0,1,1,0.08,0.06,0,255,19,0.07,0.06,0,
0,0,0,1,1,anomaly,?
tcp,http,SF,0,1,0,0,6,6,0,0,0,0,1,0,0,255,229,0.9,0.01,0,0,0,0,0,0,nor
mal,?
udp,domain_u,SF,0,0,0,0,11,17,0,0,0,0,1,0,0.18,187,183,0.98,0.01,0.01,
0,0,0,0,0,normal,?
tcp,private,S0,0,0,0,0,127,5,1,1,0,0,0.04,0.06,0,255,14,0.05,0.08,0,0,
1,1,0,0,anomaly,DoS
```

Appendix 4: Sample of Prediction on test set by selected model, Semi-Supervised meta.FilteredCollectiveClassifier parameters with full Training/Test Set for data set with 55% unlabeled and missing value.

```
=== Predictions on test set ===

    inst#     actual  predicted error prediction
        1        1:?   1:Normal       1
        2      3:R2L      3:R2L       1
        3   1:Normal   1:Normal       1
        4        1:?   1:Normal       1
        5      2:DoS      2:DoS       1
        6        1:?   1:Normal       1
        7   1:Normal   1:Normal       0.927
        8        1:?      2:DoS       1
        9        1:?   1:Normal       1
       10   1:Normal   1:Normal       0.5
       11        1:?   1:Normal       1
       12        1:?   1:Normal       1
       13   1:Normal   1:Normal       1
       14   1:Normal   1:Normal       1
       15      2:DoS      2:DoS       1
       16   1:Normal   1:Normal       1
       17        1:?      2:DoS       1
       18   1:Normal   1:Normal       1
       19        1:?      2:DoS       1
       20      5:U2R   1:Normal   +   0.783
       21        1:?      3:R2L       0.691
       22        1:?      2:DoS       1
       23      2:DoS      2:DoS       1
       24   1:Normal   1:Normal       1
```

Appendix 5: Name of Selected model, Semi-Supervised meta.Filtered Collective Classifier parameters with full Training/Test Set for data set with 55% unlabeled and missing value and its detail results.

=== Model ===

FilteredCollectiveClassifier

Classifier.....: weka.classifiers.collective.meta.YATSI -A "weka.core.neighboursearch.KDTree -A \"weka.core.EuclideanDistance -R first-last\" -S weka.core.neighboursearch.kdtrees.SlidingMidPointOfWidestSide -W 0.01 -L 40 -N" -F 1.0 -K 10 -folds 0 -S 1 -W weka.classifiers.trees.J48 -- -C 0.25 -M 2

Filter.....: weka.filters.unsupervised.attribute.ReplaceMissingValues

Best model printed below...

=== Supplied test set ===

Correctly Classified Instances	5385	96.2466 %
Incorrectly Classified Instances	210	3.7534 %
Kappa statistic	0.9317	
Mean absolute error	0.0192	
Root mean squared error	0.1058	
Relative absolute error	8.5772 %	
Root relative squared error	31.6641 %	
Total Number of Instances	5595	
Ignored Class Unknown Instances	7001	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.994	0.042	0.972	0.994	0.983	0.958	0.773	0.442	Normal
0.984	0.012	0.969	0.984	0.977	0.968	0.911	0.449	DoS

	0.912	0.012	0.878	0.912	0.895	0.885	0.963	0.391	R2L
	0.464	0.001	0.928	0.464	0.618	0.649	0.929	0.361	Probe
	0.239	0.000	1.000	0.239	0.386	0.487	0.922	0.450	U2R
Weighted Avg.	0.962	0.029	0.962	0.962	0.958	0.941	0.834	0.437	

=== Confusion Matrix ===

a	b	c	d	e	<-- classified as
3308	13	5	1	0	a = Normal
24	1556	1	0	0	b = DoS
28	9	433	5	0	c = R2L
28	7	54	77	0	d = Probe
15	20	0	0	11	e = U2R

Appendix 6: Selected model, ordinary J48 algorithm parameters with their default values- 10 fold cross validation using unlabeled class with no missing value data set developed rules, number of leaves and size of tree.

=== Classifier model (full training set) ===

J48 pruned tree

---

flag = OTH: Normal (0.0)

flag = REJ: R2L (357.0)

flag = RSTO: Probe (46.0)

flag = RSTOS0: Probe (3.0)

flag = RSTR: Probe (73.0)

flag = S0: DoS (1284.0)

flag = S1: U2R (15.0)

flag = S2: U2R (3.0)

flag = S3: U2R (3.0)

flag = SF: Normal (2634.0)

flag = SH: U2R (8.0)

Number of Leaves : 11

Size of the tree : 12