



**A PREDICTIVE MODEL OF NETWORK INTRUSION
DETECTION SYSTEMS USING MACHINE LEARNING
APPROACH**

A Thesis Presented

by

KASSAHUN WORKU G/MICHAEL

to

The Faculty of Informatics

of

St. Mary's University

**In Partial Fulfillment of the Requirements
for the Degree of Master of Science**

in

Computer Science

January, 2023

ACCEPTANCE

A PREDICTIVE MODEL OF NETWORK INTRUSION DETECTION SYSTEMS USING MACHINE LEARNING APPROACH

By

KASSAHUN WORKU G/MICHAEL

Accepted by the Faculty of Informatics, St. Mary's University, in partial
fulfillment of the requirements for the degree of Master of Science in
Computer Science

Thesis Examination Committee:

Internal Examiner

Dr. Yihene Wondie

External Examiner

Dean, Faculty of Informatics

January 2023

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Kassahun Worku G/Michael

Full Name of Student

Signature

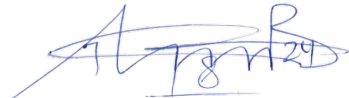
Addis Ababa

Ethiopia

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

Asrat Mulatu (Ph.D.)

Full Name of Advisor



Signature

Addis Ababa

Ethiopia

January 2023

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest thanks to the Almighty God;

My advisor Dr. Asrat Mulatu, I would like to express my sincere gratitude to his valuable time, guidance and comments which enabled me to gain good research experience;

My Parents, for their love, support, and sacrifices. Thank you for believing in me.

Table of Contents

ACKNOWLEDGEMENTS.....	i
LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
ABSTRACT	vii
CHAPTER 1: INTRODUCTION.....	1
1.1 Background of the study.....	1
1.2 Statement of the Problem	3
1.3 Motivation	5
1.4 Research question	5
1.5 Objectives	5
1.5.1 General Objective	5
1.5.2 Specific Objectives	6
1.6 Significance of the study.....	6
1.7 Scope	6
1.8 Organization of the Thesis Report.....	7
CHAPTER 2: LITRATURE REVIEW	8
2.1 Overview of Intrusion Detection System.....	8
2.2 Component of NIDS architecture	9
2.3 Types IDS.....	10
2.3.1 Host-based IDS (HIDS).....	10
2.3.2 Network-based IDS (NIDS)	11
2.3.3 An application-based IDS.....	11
2.3.4 Signature-based detection (misuse detection)	12
2.3.5 Anomaly detection	13
2.3.6 Hybrid based detection	13
2.4 Machine learning	14
2.4.1 Supervised algorithms	14
2.4.2 Unsupervised algorithms.....	14
2.5 Deployment Scenario for IDSs	14
2.5.1 Before the Firewall	15
2.5.2 Inside the Private Network.....	16
2.5.3 Behind the external firewall	17

2.5.4 Deployment on Individual Hosts	18
2.6 intrusion detection system challenges	18
2.7 Related Works	19
CHAPTER 3: METHODOLOGY	27
3.1 Research Design.....	27
3.2 Literature Review.....	27
3.3 Tools Used	28
3.4 Dataset Preparation.....	30
3.5 CIC-IDS2017 Dataset Description.....	32
3.5.1 Attack Types in CIC-IDS2017 Dataset	32
3.6 Data Preprocessing.....	37
3.7 Feature extraction	38
3.8 Performance metrics	38
3.9 Machine learning Algorithms	40
3.10 Ensemble Methods.....	41
3.10.1 Bagging	42
3.10.2 Boosting.....	43
3.10.3 Stacking (Blending)	43
CHAPTER 4: EXPERIMENT	44
4.1 Overview.....	44
4.2 System Configuration	44
4.3 Proposed ensemble machine Learning model	44
4.3.1 Sequence of Steps.....	46
CHAPTER 5: ANALYSIS AND RESULTS.....	47
5.1 Performance Evaluation Result	47
CHAPTER 6: CONCLUSIONS AND FUTURE WORKS	51
6.1 Conclusions.....	51
6.2 Future Works	51
References	52
APPENDICES.....	58
Appendix A: proposed model result summary.....	58
Appendix B: selected attribute for the model.....	59

LIST OF ABBREVIATIONS

ANIDS	Anomaly Network Intrusion Detection System
API	Application Programming Interface
CIC	Canadian Institute for Cyber security
CIC-IDS	Canadian Institute for Cyber security intrusion detection system
DDOS	Distributed Denial-of-Service
DOS	Denial of Service
FN	False Negative
FP	False Positive
FPR	False Positive Rate
HIDS	Host-based Intrusion Detection System
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection System
ML	Machine Learning
NIDS	Network Intrusion Detection System
ROC	Receiver Operating Characteristics
SIDS	Signature-based Intrusion Detection System
SIEM	Security Information and Event Management
WEKA	Waikato Environment for Knowledge Analysis
ROC	Receiver Operating Characteristics
SVM	Support Vector Machine
NSL-KDD	Network Simulation Language Knowledge Discovery in Database
CSV	Comma Separated Values
FI	Feature Importance
EM	Ensemble method
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol

LIST OF TABLES

<i>Table 3.1 Description of files containing CICIDS-2017 dataset</i>	<i>34</i>
<i>Table 3.2 Records in the CICIDS-2017 Dataset.....</i>	<i>34</i>
<i>Table 3.3 The Class wise instance occurrence of CICIDS-2017 dataset</i>	<i>35</i>
<i>Table 3.4 Distribution of selected dataset from CIC-IDS2017.....</i>	<i>35</i>
<i>Table 3.5 Features of CICIDS-2017 Dataset.....</i>	<i>36</i>
<i>Table 3.6 The distribution of training and testing dataset.....</i>	<i>38</i>
<i>Table 5.1 Performance Comparison of the five Classifiers algorithm on selected dataset ..</i>	<i>47</i>
<i>Table 5.2 Detailed Accuracy by classification using Ensemble model method.....</i>	<i>48</i>
<i>Table 5.3 Classification accuracy using Ensemble model.....</i>	<i>49</i>
<i>Table 5.4 Detection result of selected algorithm and ensemble model.....</i>	<i>49</i>

LIST OF FIGURES

<i>Figure 2.1 Intrusion Detection System architecture</i>	<i>10</i>
<i>Figure 2.2 IDS classification IDS based on their installation place</i>	<i>10</i>
<i>Figure 2.3 Classification of IDSs by detection method.....</i>	<i>12</i>
<i>Figure 2.4 Typical Network Scenario without IDS.....</i>	<i>15</i>
<i>Figure 2.5 Network IDS Placed Before the Gateway Firewall</i>	<i>16</i>
<i>Figure 2.6 IDS After Second Firewall</i>	<i>17</i>
<i>Figure 2.7 IDS in the DMZ</i>	<i>17</i>
<i>Figure 2.8 Host Based Intrusion Detection System</i>	<i>18</i>
<i>Figure 2.9 IDS challenges.....</i>	<i>19</i>
<i>Figure 3.1 WEKA GUI.....</i>	<i>29</i>
<i>Figure 3.2 WEKA ensemble method simulation GUI</i>	<i>30</i>
<i>Figure 4.1 proposed System Model Architecture.....</i>	<i>46</i>

ABSTRACT

Information and network security issues are very critical in this Era. Information is playing a vital role to realize informed and civilized society and to create democratic, transparent and accountable government, and to assure sustainable economic development. On the other hand, the reliance on information systems is increasing the vulnerability of organizations for cyber-attacks which are becoming highly complicated, dynamic and destructive. In order to protect organizations from cyber-attacks and minimize their impact, it is essential to ensure the security of information and information systems.

Machine learning techniques provide a promising result in improving Detection accuracy of intrusion detection system (IDS). A variety of machine learning techniques have been designed and integrated with IDSs. But Most of the Intrusion detection systems still have poor intrusion detection rate and high false positive rate. This thesis focused on ensemble method involving the integration of predictions by multiple individual classifiers.

Ensemble method enable to compensate for the weakness of individual classifiers and use their combined knowledge to enhance its performance, different ensemble methods in the field are analyzed, taking into consideration different types of ensembles and various approaches for integrating the predictions of individual classifiers for an ensemble classifier. This research has attempted to build a predictive ensemble ML model for intrusion detection using a new standard dataset from the Canadian Institute for Cyber security intrusion detection system (CIC-IDS2017) for performance evaluation. Simulation outcomes prove that the proposed ensemble model outperforms current IDS systems, attaining accuracy of up to 99%. The performance of this algorithm is measured using accuracy, precision, false positive, F1 score, and recall which found promising results for deployment on real network infrastructure.

Keywords: *Cyber Security, Intrusion Detection, Machine learning algorithms, ensemble model, CIC-IDS2017Datase.*

CHAPTER 1: INTRODUCTION

1.1 Background of the study

With the constant growth of the Internet, cyber-attacks are increasing not only in numbers and diversity, ransomware are on the rise like never before, and zero-day exploits become so critical that they are gaining media coverage. Antiviruses and firewalls are no longer sufficient to ensure the protection of a company network, which should be based on combined layers of security. One of the most important layers, designed to protect its target against any potential attack through a continuous monitoring of the system, is provided by an Intrusion Detection System (IDS).

Cyber security is a growing problem in modern times because of the rapid growth and technological advancement. The internet provides all knowledge that has been accumulated by man and with the advent of mobile computing at every person's finger tips, cyber-attacks and cybercrimes has become all too popular. A report from the anti-phishing working group has shown that about 227,000 malware detections occur daily which is linked to over 20 million new malwares daily [1]. There has been a straight forward method for dealing with malware in the past, but over the past two decades there has been an evolution in cyber-attacks and how exploits are carried out, as such cyber security techniques are also undergoing an evolution into more intelligent approaches.

Network and internet communication is rapid which uses of electronic devices like computers, laptops, mobiles... etc., to transfer, process, store and retrieve information. The security issue is very important. Nowadays, cyber-attacks are highly increasing all over the world including Ethiopia. Africa loses \$3.5b to cyber security attacks in 2020 [2]. In our country Ethiopia many people are connected to the internet and other networks, even the majority of the society did not get access yet. The last few years' cyber-attack is sharply increasing all over the world. According to report by information network security agency (INSA), Ethiopia was hit 1000 cyber-attacks within a quarter of 2021 and 1800 within six months of 2021. According to the Report, Ethiopia was one of the 11 African nations targeted by WannaCry ransom ware cyber-attack that hit 155 countries worldwide [3].

An activity known as a cyber-attack or network intrusion aims to jeopardize a computer network's regular operation. We must create an intrusion detection mechanism, which is a way to lessen or report these incursions, to protect against cyberattacks. Yet, with typical IDS, monitoring and detecting intrusions at very fast network speeds and during an uptick in Distributed Denial of Service (DDoS) attacks becomes challenging. There have been numerous attempts in recent years to design effective intrusion detection systems in order to address these difficulties (IDS). IDS is a program that keeps an eye out for unusual activities that could jeopardize the network's Confidentiality, Integrity, and Availability (CIA) qualities. It includes monitoring of unwanted utilization of the network resources, keeping it available for the legitimate users and in some cases preventing loss of information/data to the intruder.

The most efficient way to tackle this growing problem involves the use of machine learning algorithms to detect attacks before they attack a legitimate system. A lot of network logs (IDS Dataset) already exists from past attacks, this log file can be fed into the algorithm to train it to be able recognize attacks and send alert to system Administrator or intrusion prevention system (IPS) [4].

In order to detect these intrusions, various intrusion detection systems (IDSs) are implemented in many organizations networks. These systems are classified into host-based IDS, network based (NIDS), and hybrid IDS. HIDS monitors the system and looks for malicious activities, and NIDS examines the traffic payload in the network for suspicious events. [5] Based on detection methods, IDS are characterized into two types, namely signature-based IDS and anomaly-based IDS [6].

The lack of a comprehensive network-based data set that can depict contemporary network traffic scenarios, a wide range of low footprint intrusions, and deep structured information about the network traffic is one of the primary research difficulties in this subject [6]. A decade ago, the benchmark data sets KDD98, KDDCUP99, and NSLKDD were created in order to assess network intrusion detection systems research efforts. However, multiple recent research revealed that these data sets do not completely capture network traffic and contemporary low footprint assaults for the current network threat environment [7]. Countering the unavailability of network benchmark data set challenges, this paper will

examine a Canadian Institute for Cyber security intrusion detection system (CIC-IDS2017) dataset. This data set has a hybrid of the real modern normal and the contemporary synthesized attack activities of the network traffic [7]. Existing and novel methods are utilized to generate the features of the CIC-IDS2017 data set. This data set is available for research purposes and can be accessed freely.

The goal of this thesis is to suggest strategies for improving the detection quality of intrusion detection systems (IDS) utilizing machine learning approaches for implementation on real-world networks. This research has attempted to build ensemble ML model for intrusion detection using a new standard dataset from the Canadian Institute for Cyber security intrusion detection system (CIC-IDS2017) for performance evaluation.

1.2 Statement of the Problem

Machine learning techniques provide a promising solution for improving intrusion detection system (IDS) [8]. Machine learning techniques are classified into supervised and unsupervised learning techniques. Supervised learning needs a training dataset with labeled instances for normal as well as anomaly classes, whereas in unsupervised learning, the algorithm directly learns patterns from the data, without any human intervention. Developing IDS model with better accuracy and low false positive detection system has become an important solution to detect existing attacks and emerging attacks. Using ensemble machine learning algorithms for intrusion detection can improve its overall performance as the weaknesses of one algorithm might be complemented by the second one. Security system of enterprise network should be improved in line with technology advancement to enhance network security defences. This research has attempted to build ensemble machine learning algorithms models for intrusion detection using a new benchmark dataset Canadian Institute for Cyber security intrusion detection system (CIC-IDS2017) for performance evaluation. Applying a new dataset to meet the current significant advances in internet traffic diversity and emerging attacks types is mandatory.

Ensemble machine learning algorithms intrusion detection method is capable of detecting attacks with high accuracy [4]. Ensemble machine learning algorithms NIDS attempts to overcome the shortcomings of accuracy and false positive rate NIDSs [9] based on the literature review, many intrusion detections related papers specific to machine

learning-based IDSs have been developed. However, the existing IDSs still have their own shortcomings. Some of the shortcomings are, low detection rate, high training time, low processing speed, relatively high false alarm rate (FAR).

To address these issues, several attempts have been made in recent years to develop effective IDS. However, there is still room for development in these systems. The suggested study effort centered on constructing an intrusion detection model with a higher detection rate, reduced training time, and enhanced performance by parallelizing training and selecting the optimal parameters for models that can increase model performance.

To strengthen network security defenses, enterprise networks' security systems should be developed in step with technological innovation. The goal of this study was to develop a predictive ensemble ML model for intrusion detection utilizing the new benchmark dataset CIC-IDS2017 from the Canadian Institute for Cyber Security. To address the present major improvements in internet traffic diversity and emerging attack types, a new dataset must be used. High rates of false-positive alarms and low detection rates for zero-day assaults are the two main drawbacks of current intrusion detection systems. To overcome these problems, we need intrusion detection techniques that can learn and effectively detect intrusions. Ensemble methods based on machine learning techniques have been proposed by different researchers. These methods take advantage of the single detection methods and leverage their weakness [10].

The detection rate, false alarm rate, complexity, and evaluation of both known and new threats were generally neglected in earlier attempts on intrusion detection systems. The goal of this study is to close the gap between the aforementioned issues. The ensemble approach refers to the fusion of various machine learning algorithms. According to a review of the literature, machine learning's ensemble method lowers the rate of false positives. Basic Machine Learning Classifiers can be combined using four major techniques: Bagging, Boosting, Randomization, and Stacking. The implementation of an intrusion detection system using the bagging-based Ensemble approach is suggested in this research. In this study, the ensemble model and benchmark dataset Canadian Institute for Cyber security intrusion detection system (CIC-IDS2017) datasets are used for model evaluation. This ensemble model combines the output of several classifiers and produced a single composite classification.

The suggested Ensemble machine learning model consists of two distinct algorithms. The first is a random forest, while the second approach is bagging method found on WEAK tool, the proposed model trained using supervised data. This distinction can actually serve to improve overall detection rates since the shortcomings of one approach may be addressed by the other. So, the goal of this research will be to create a model that is more capable of recognizing diverse cyberattacks and has the potential to reduce the percentage of false positives by boosting the accuracy of detecting newer attacks.

1.3 Motivation

The motivations of this research work are application of intrusion detection system for different organization, availability of open sources and weakness of currently available network security tools with regard to detecting intrusion. Despite the fact that intrusion detection system applicable in different organization and used more than decade, there still exists many issues around IDS. Including false positive, low detection capacities.

1.4 Research question

There are a number of problems associated to IDS. In this research we will address the following Questions:

- How can we minimize intrusion?
- Which network intrusion detection dataset is better for simulation?
- Which detection technique is the best to use for detection rate enhancement?
- How can Detect unknown attacks and minimize false alarm rate?

1.5 Objectives

1.5.1 General Objective

The general objective of this study is to build ML model for network intrusion detection system (NIDS) using Ensemble approach that will enhance the computer network security system.

1.5.2 Specific Objectives

The specific objectives of this research study are:

- To study different types intrusion detection system of classification.
- To conduct training and testing of the predictive models using the new CIC-IDS2017 benchmark dataset.
- To design suitable machine learning logs classification and attack prediction model.
- To extract the most prominent features and apply classification techniques.
- To compare the accuracy rates of different classifiers.
- To apply pre-processing techniques on CIC-IDS2017 benchmark dataset
- To interpret and analyse the results of the selected model

1.6 Significance of the study

- Improve intrusion detection system (IDS). System administrator and IPS uses the result as an input, it helps the prevention mechanism to be proactive rather than reactive. The technique that an attack occurs and responses applied this indicates reactive approach, whereas made prevention before attack occurred and build a model is proactive approach.
- Reduces cyber security risk impact, knowing the behaviors of users is very important to prevent assets from damage early by applying IPS based on specified features.
- Reduce the staff cost and misconfiguration of using SIEM system.
- Reduce cyber-attack and its impact such as financial, political, and social.

1.7 Scope

In this thesis we will design ensemble intrusion detection system. It focuses on identifying possible cyber incidents, and reporting them to the security administrators or IPS. This system is designed to increase detection rate and reduce false positive rate and applicable in any organization's network. The data used for this thesis obtained from publicly available dataset state of the art IDS dataset. But We believe it is better using local and real log files to predict and classify cyber-attacks. Because, using local data helps us to know attack targeted our country Ethiopia and cyber security status at real situation, but due to confidentiality issue we did not use local and recent IDS log files.

One of the limitations of this research work is that the dataset is used from Canadian Institute for Cyber security organization, and cannot directly implement the trained model to specific organization network. Which is due to the network infrastructure and configuration of one organization is different from the others.

1.8 Organization of the Thesis Report

The following is an overview of the structure of this thesis. First chapter gives an introduction to this research giving statement of the problem, thesis objectives, motivation and the scope of this work. This is followed by second chapter introduces the conceptual information on intrusion detection and related works in the field of machine learning based intrusion detection system using different detection techniques. It also discusses how intrusion detection systems are classified. The third chapter introduces the research methods, algorithms and dataset to use in this paper. The fourth chapter introduces the research experiment and which explores the study done including evaluation setup, The fifth chapter includes performance analysis of the selected algorithms. and chapter six introduces concluding remarks and present ideas for improvements and recommendations for future research are forwarded.

CHAPTER 2: LITRATURE REVIEW

2.1 Overview of Intrusion Detection System

Monitoring network traffic and computer events to detect malicious or unauthorized activities is a process called “intrusion detection”. Every device or software application whose goal is to conduct an intrusion detection is considered as an Intrusion Detection System (IDS). These IDS alarms are then reported either to an administrator or collected centrally using a Security Information and Event Management (SIEM) system. A SIEM system provides real-time analysis of the outputs of multiple sources to correlate the different alerts and show a comprehensive view of IT security [11]

IDSs are sometimes confused with two other security tools: firewalls and Intrusion Prevention Systems (IPSs). These three security mechanisms are designed to protect systems within a network but use different means. For instance, firewalls look outwardly for intrusions in order to stop them before they enter the protected network. They analyze packet headers to filter incoming and outgoing traffic based on predetermined rules (protocol, IP address, port number...) [12]. On the other hand, IDSs are able to monitor activities within the protected network and not just at its perimeter. Unlike a firewall, IDSs only have a monitoring role they cannot take action to block suspicious activities and therefore need an administrator or IPS to process their alerts. This is not the case with IPSs, which function as an IDS but are able to proactively block a detected threat. This automation adds a layer of complexity since an inappropriate response can cause additional problems on the network [13].

Intrusion detection systems are methods and applications that are designed to evaluate and defend computers, networks, programs, and data against assaults, unauthorized access, unlawful read/write, and deletion/corruption. Depending on where and how it is installed, they might be classified as host or network. Intrusion detection systems should not be confused with other types of security measures found in a network or system, such as firewalls and antivirus software. Intrusion detection systems function in tandem with these since they are more intimately associated with detecting and raising alerts. Intrusion detection systems (IDS) are a type of computer security management system. An Intrusion

Detection System collects and analyzes data from certain regions of a network or computers in order to detect potential security breaches.

There are two types of IDS classification methods [14]. Detection-based method and data source-based methods. Depending on how the intrusion is detected, there are two different types of IDS: signature-based (misuse) IDS (SIDS) and anomaly detection-based IDS (ADIDS). SIDS [15] is based on pattern matching techniques to find a known attack; these are also known as Knowledge-based Detection or Misuse Detection.

2.2 Component of NIDS architecture

The architecture of a generic NIDS contains these components:

Data gathering sensors

They are used to monitor the infrastructure where data collection takes place and to watch certain activities or protocols. They use the data acquired from their location to produce a primary categorization.

Detector engine

This module compares the collected data to the set of rules that has been established.

When IDS detects a divergence from the typical status, it generates an alarm.

Storage Module

It contains the rule sets of the IDS, which the detector uses when comparing the received data.

Response

When an alarm sounds, it triggers a predefined action. Depending on the type of alert, there may be an action in which the IDS performs a specified action, such as discarding the malicious packets. Sometimes a passive response is appropriate, such as documenting the behavior and letting the human factor decide on the action.

A typical IDS architecture is shown below:

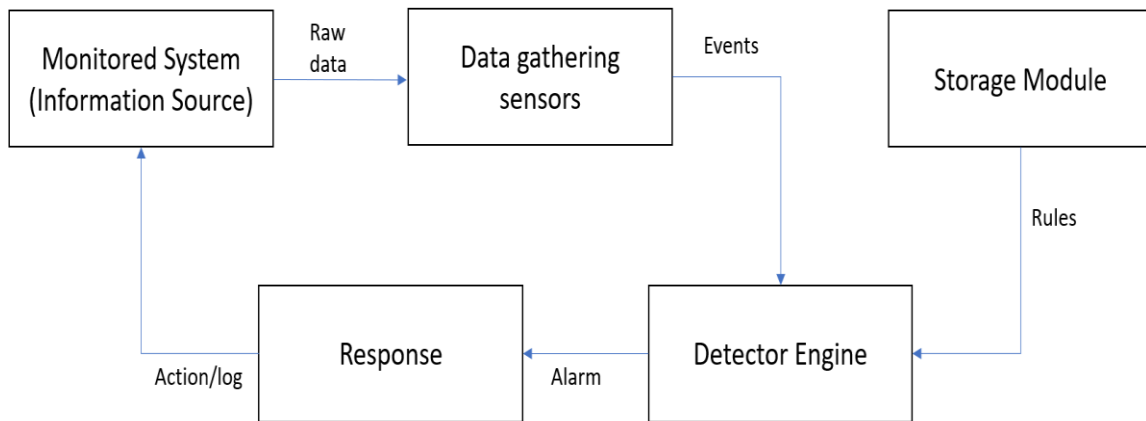


Figure 2.1 Intrusion Detection System architecture

2.3 Types IDS

This section will go through the many types of intrusion detection systems and detection methodologies. IDS are classified into two categories based on where they are installed. host-based IDS and network-based IDS. IDS may be divided into three types based on the type of activity being monitored: network-based IDS, host-based IDS, and application-based IDS.

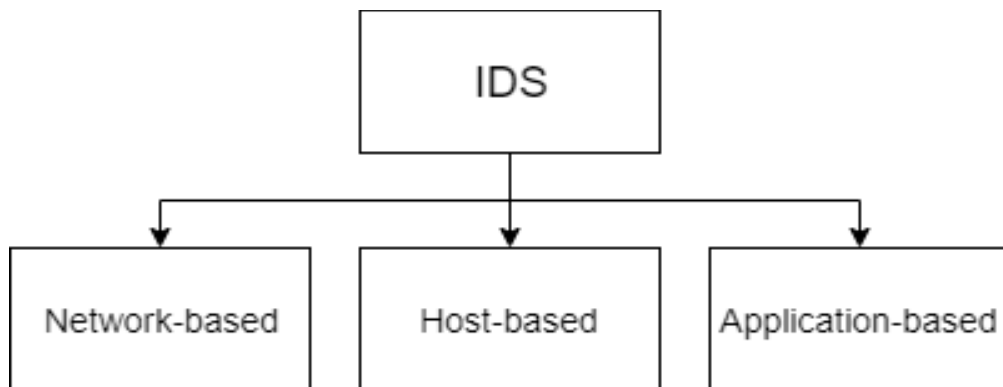


Figure 2.2 IDS classification *IDS based on their installation place*

2.3.1 Host-based IDS (HIDS)

HIDS is an agent installed on individual hosts, which analyses their activity, files, processes, system logs, etc. HIDSs have multiple resources at their disposal. Snapshots of the system can be compared to check for the presence. A HIDS determines if a system has been

compromised by inspecting the full communication stream and warning administrators accordingly, i.e., it can detect a rogue program that suspiciously accesses a system's resources or discovers that a program has modified the registry in a harmful way [16]. the diagram below depicts the classification of intrusion detection systems depending on their installation technique.

IDS classification based on assessed activity. in the case of unlawful or questionable behavior. Multiple unsuccessful logins attempts and unusually high CPU utilization over an extended period of time are indicators of a possible assault. Some HIDSs may identify kernel-based threats by examining system calls and modifications to system binaries. They might also be used to spy on people by tracking their activity.

2.3.2 Network-based IDS (NIDS)

A network-based intrusion detection system (NIDS) often employs sensors located throughout the network. the traffic is analyzed either locally by the sensor or remotely by a central controller. Because NIDSs are more scalable and cross-platform than HIDSs, they are more often used to safeguard a company's IT equipment. However, these technologies may be used in tandem to provide a better level of protection.

A network-based intrusion detection system (NIDS) functionality is to monitor and analyze the network traffic. Its prominent role is to protect the system from network-based threats by discovering unauthorized malicious access to a LAN and exploring traffic that traverses the wire, multiple hosts. Detection algorithms read inbound and outgoing packets and searches for any suspicious patterns, so an alert generated by NIDS notifies the administrator [17].

2.3.3 An application-based IDS

An application-based intrusion detection system (IDS) is a sort of HIDS that is designed to monitor a specific application. Application-based intrusion detection systems (IDSs) examine user-application interactions such as file executions or edits, logs, authorizations, and any other potentially suspicious activity. they can create profiles of individual users in order to detect suspicious events. Some application-based intrusion detection systems (IDSs) can also access data before it is encrypted, functioning as a middleman between the application and the encrypted data for storage.

IDSs can also be classified according to the detection method they use. They fall into three categories: signature-based detection, anomaly-based detection, and hybrid detection.

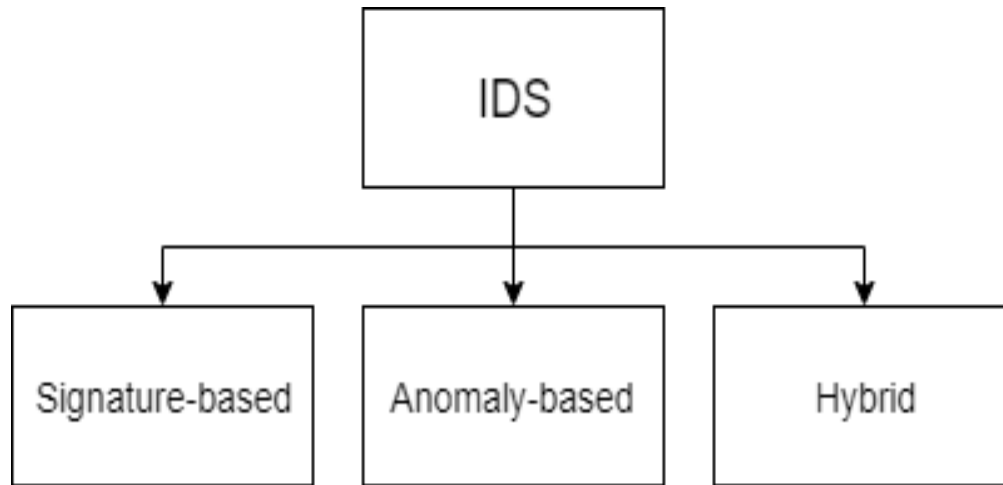


Figure 2.3 Classification of IDSs by detection method.

2.3.4 Signature-based detection (misuse detection)

A database of known attack signatures is included with signature-based detection. It checks observed data against the signature database. A misuse detection IDS, like a traditional antivirus, examines the input stream for the existence of an attack pattern. This signature can be expressed as a series of bytes or characters, but more complicated patterns are frequently represented as a branching tree diagram. To be effective, the database of this type of IDS must be updated on a regular basis. However, even with the most recent patches, this approach can only identify known threats.

Misuse detection or signature-based detection relies on an extensive database that contains previous well-known vulnerabilities registered in it. The system looks for a unique pattern called signature and matches the existing ones in the database in the occurring attacks.

Pranggono et al. [17]

2.3.5 Anomaly detection

Anomaly detection attempts to understand the system's "normal" or "expected" behavior. any divergence from this pattern is regarded as a potential intrusion and will raise an alarm. this technique does not need any updates or the presence of a database. It can detect unknown threats, but it also generates a large number of false positives that are difficult to handle. It is also more difficult to gather information about the intrusion since it lacks a distinct signature.

Behavior-based detection simulates the predicted system and network behavior. It notifies the administrator when the behavior deviates from the preset threshold. These methods, unlike signature-based detection, may identify zero-day assaults since no rules must be created. This detection approach makes it more difficult for attackers to understand the capabilities of IDSs. However, they have a high false alarm rate and have difficulty determining the sort of attack. Several scholars have integrated various strategies and assets to address these shortcomings. When utilized alone, however, they have a poor detection rate. Another problem is that it is difficult to define the ruleset.

2.3.6 Hybrid based detection

Hybrid detection combines the two solutions to mitigate weaknesses of each category: anomaly detection then misuse detection, misuse detection then anomaly detection, or both at the same time. The goal is to detect known attacks with their signatures, and to use anomaly detection to identify unknown intrusions.

Machine learning approaches enable the combination of behavior-based and signature-based detection, resulting in a new type of detection known as hybrid detection or specification-based detection. this combination decreases the previously described detection gaps by lowering both false negative and false positive rate warnings. According to surveys, applying machine learning algorithms has a positive influence on cyber-attack detection and is considered as a viable strategy to improving cyber security.

2.4 Machine learning

Machine learning algorithms can improve network security by making necessary calculations and decisions such as recognizing the type of packets in the traffic. Machine learning algorithms are categorized as follows.

2.4.1 Supervised algorithms

Fully labeled class data is required for supervised algorithms. dataset separated into two parts is required for network intrusion detection using supervised machine learning algorithms: training data and testing data. the basic goal is to develop a model by training algorithms using labeled data. the trained model is then used to forecast the 'unknown' in test data. Supervised learning is used for detecting existing attacks, but not for detecting new ones zero-day attack.

2.4.2 Unsupervised algorithms

Clustering techniques are used in unsupervised learning to create a model from unlabeled data. They are able to differentiate between malicious inputs and host logs or network traffic in this way. Unsupervised methods randomly and without any prior knowledge examine the data attributes in accordance with their statistical properties. Arguments to highlight that unsupervised approaches do not require the time-consuming data training stage are made in Nisioti et al. [18].

The actual methods employed by intrusion detection systems to address challenges including real-time detection, poor detection accuracy, and erratic detection rates were explored in [19] and [20]. In order to solve these issues, hybrid machine learning methods had to be created. Reduced false-negative and false-positive alarms are the key goals of hybrid machine learning methods. Hybrid methods used for intrusion detection combine supervised and unsupervised machine learning approaches to improve the system's performance.

2.5 Deployment Scenario for IDSs

There are several methods for incorporating IDS instruments into our network, each with pros and cons. the optimum option would be a balance between cost and desired properties, while retaining a high level of benefits and a limited number of drawbacks, all in line with the organization's demands. As a result, the IDS placements inside a network give varied

features. Then we shall explore many alternatives inside the same network. Assume we have a network with a firewall separating the Internet from the demilitarized zone (DMZ). Demilitarized Zone), and another that separates the DMZ from the organization's intranet, as indicated in the following figure. the DMZ is the space between the Internet and the local network.

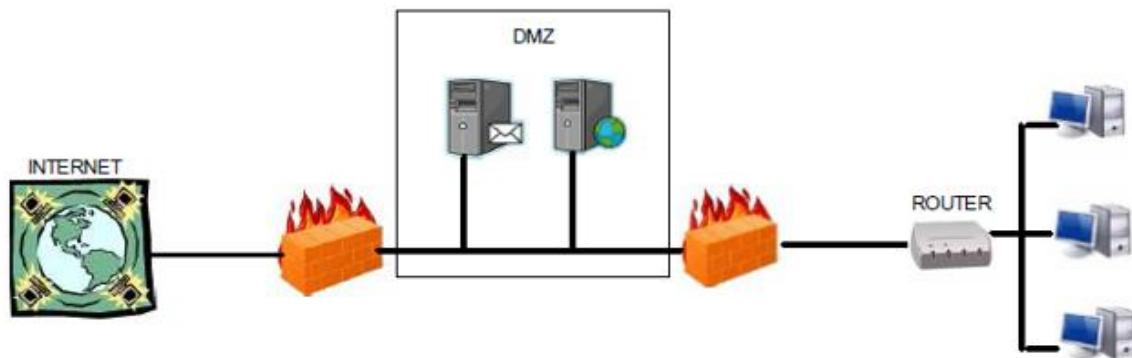


Figure 2.4 Typical Network Scenario without IDS

2.5.1 Before the Firewall

In this point, the NIDS can keep track of all network events of interest, even those attacks which subsequently may fail. as it has to handle large traffic, NIDS ought to be installed on a faster machine so that analysis is done in real time. also, it has to be configured correctly and number of false alarms can be reduced. Figure 2.4 shows how to deploy IDS before firewall [18].

In this role, the IDS will record all incoming and outgoing network traffic, allowing it to monitor the quantity and kind of assaults against the organization's infrastructure and the external firewall. Because of the significant number of false alerts in this area, IDSs should be designed with a low sensitivity.

The main drawbacks of this location are that the IDSs can't detect attacks using in their communications some methods to hide information, such as encryption algorithms, and that

in this location the traffic rate is usually so high that the IDSs can't monitor all the packages [19].

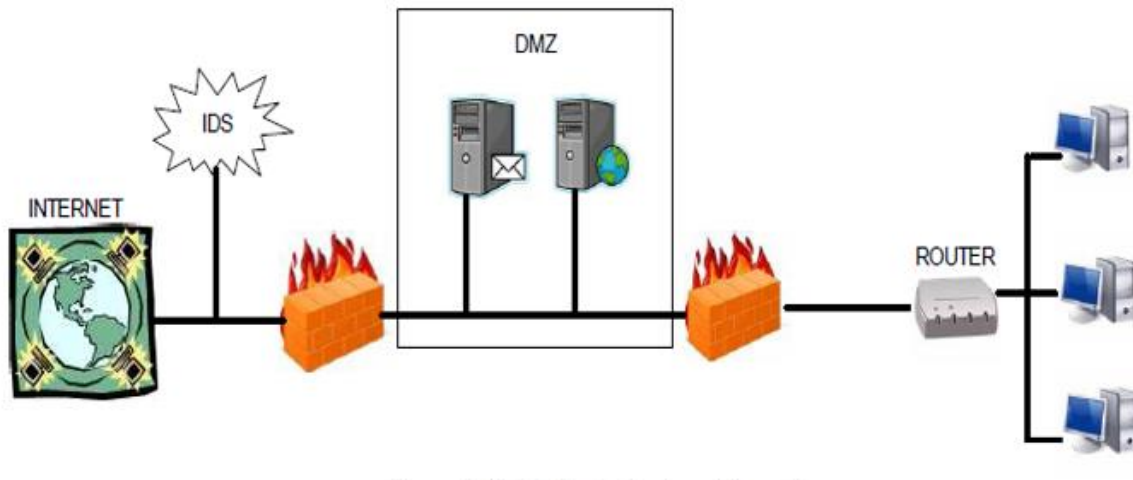


Figure 2.5 Network IDS Placed Before the Gateway Firewall

2.5.2 Inside the Private Network

Another location for NIDS is within the company network, as seen in Figure 2.6. This site is intended to detect attack emanating from local networks as well as those relayed via firewall. Because the number of attack conceivable in this location is lower than in the prior situations, the application demands are lower. In this situation, IDS creates a small number of false alerts. Because the scope of visibility is confined to the business network, the failed intrusion will not be detected as in the prior examples.

In this instance, the IDS is situated in-between the internal network and the second firewall. It won't listen to any internal traffic because it is outside the internal network. As there is currently less traffic, this IDS ought to be weaker than those mentioned previously. Any unusual traffic that appears here must be treated with hostility. Since there won't be as many false alarms at this stage of the network, any IDS alarm should be investigated right away. These systems are especially vulnerable to attacks because of this implementation, both from the outside and from within their own infrastructure. While installing an intrusion detector in this space, it is imperative to bear this in mind, in order to detect attacks produced from within the network itself, such as those launched by internal staff [19].

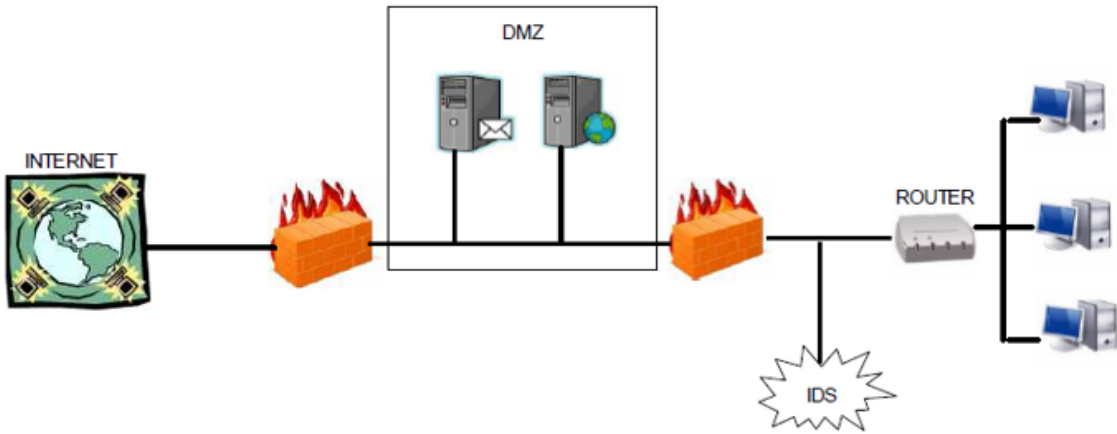


Figure 2.6 IDS After Second Firewall

2.5.3 Behind the external firewall

Another choice is to position the IDS between the firewalls in the demilitarized zone. Monitoring is done on intrusions that get past the main firewall. It is possible to identify attacks on the servers in this subnet that offer public services. By identifying the most frequent threats, the primary firewall configuration can be improved to better block them in the future. Similar to the prior instance, the disadvantages are related to encrypted attacks and the saturation of the IDS as a result of large traffic volume. As only access to our servers should be permitted at this time, this area experiences fewer false alarms. [20].

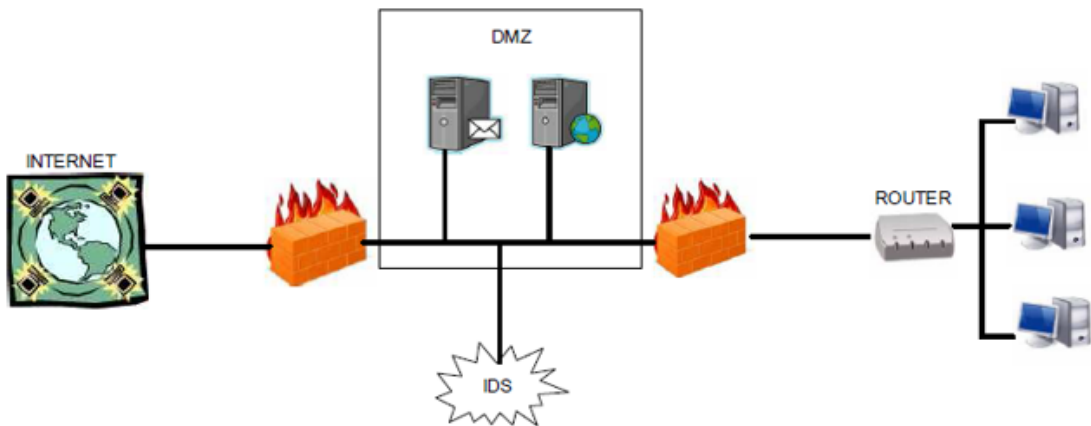


Figure 2.7 IDS in the DMZ

2.5.4 Deployment on Individual Hosts

In this case host-based IDS will be installed to hosts and gathers information either the operating system audit trails or system logs of host which it has been installed. It does not only monitor the communication traffic in and out of a single computer but also checks the integrity of the system file. Figure 2.4 shows sample host-based IDS [21]

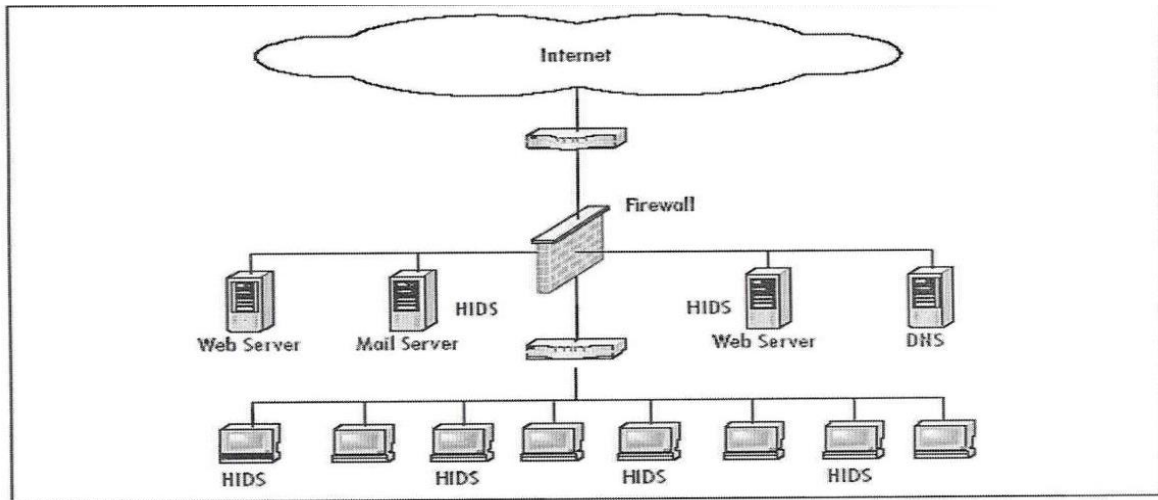


Figure 2.8 Host Based Intrusion Detection System

2.6 Intrusion detection system challenges

Several strategies for intrusion detection system have been devised and implemented by specialists. They have explored more accuracy, quicker training time, and scalable approaches for IDS. However, the variety and amount of intrusion detection system challenges are projected to grow. Figure 2.9 summarizes the issues in IDS. These difficulties include false alarm rates, low detection rates, imbalanced datasets, and reaction time.

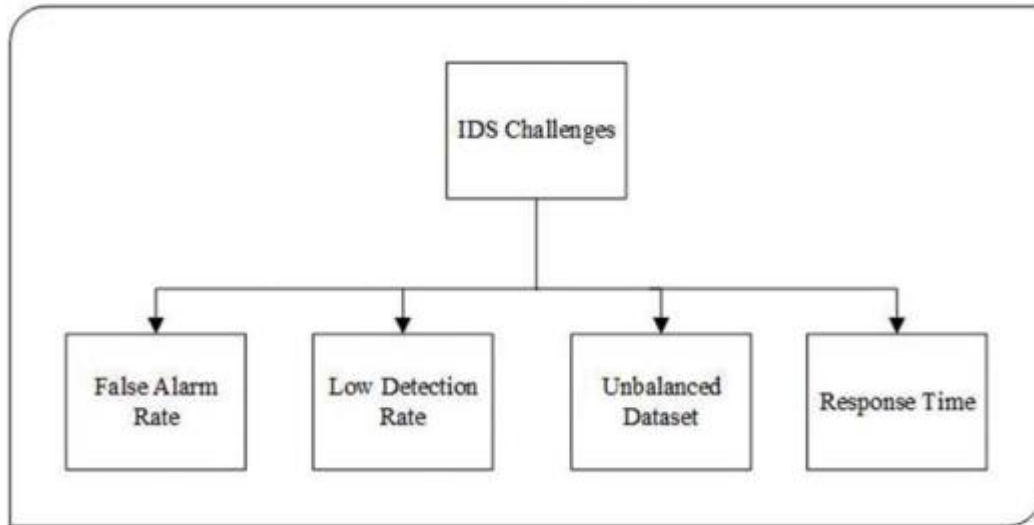


Figure 2.9 IDS challenges

2.7 Related Works

A variety of tools have been developed for the purpose of network intrusion detection and detect anomalies by matching the traffic pattern or the packets using a set of predefined rules that describe characteristics of the anomalies. The work in [22] uses the open-source network intrusion detection tool which is SNORT and tests it on high-speed network in the campus network. Their work concentrates on ICMP flood attack and they detected 12 signatures among which ICMP PING has maximum alert. But this work is not efficient on detecting novel attack and the SNORT was down during heavy traffic.

[23] Mainly concentrated on implementing IDS on wireless LAN. They proposed a new Network Intrusion Detection System (NIDS) that will mainly detect the most prominent attack of Wireless Networks, i.e., DoS attack and Man-in-the-Middle attack. This proposed IDS operates as lower layer of firewall which means after checking from the database using IP address and system content and if intruder is discovered it sends it to the firewall for blocking. This work assumed that it is part of firewall system. The fundamental issue with this system is that it simply analyzes the database, focuses only on two types of attacks, and delivers firewall requests for decision-making. It is also only controlled by decisions to allow and deny, with no alerting or signature generation for new assaults.

In anomaly-based intrusion detection systems the frequently used method is to set up a statistical model of normal network traffic. A deviation from the model will be marked as suspicious. Those statistical models are built from different perspectives of the traffic

analysis. Current network anomaly systems such as ADAM [15] and SPADE [24] belong to this category. ADAM is a statistical anomaly-based IDS developed at George Mason University, which uses a rule-based data mining technique for network intrusion detection. ADAM builds a normal profile by mining attack free data and it also has a rule set, which is initialized by user defined abnormal patterns and is constantly updated with new rules. ADAM obtains the good result when applied to DARPA evaluation data set. However, it is highly dependent on the training data even if they use pseudo-Bayesian system to avoid dependency.

On the basis of IP addresses and ports, SPADE creates a statistical model. For instance, SPADE generates its usual traffic model using SNORT as its engine based on the quantity of connections seen from particular IP and port pairs. It is more likely that the less often used IP port pair will be marked as suspicious. The disadvantage of SPADE is that it frequently generates false alarms for traffic from less frequently used IP-port pairings.

In [25] three separate statistical techniques were employed by the researchers: chi-square, Gaussian mixture distribution, and Principal Component Analysis. By creating performance logs from hosts, they evaluated the effectiveness of these three strategies. The average values for each column of the table are computed after the performance logs have been generated for each day and separated into 4 groups. The values are kept in a different table after being determined as the average values. These numbers serve as a standard data set. They obtained results in terms of detection rate and false alarm for PCA and Gaussian 97.5% and 2.5% respectively but for chi-square 90% and 10%. So this work shows that most statistical methods generate false alarm which degrades the performance of the intrusion detection system.

In [26] the authors did a survey on incremental method for anomaly detection and tried to evaluate the problems of anomaly detection which are high false alarm rate, non-scalability issue and not fit for high-speed networks. Most of the works evaluated are based on the benchmark of KDD dataset which is not updated and they said that they observe good result in most of the work but it does not have any idea whether it works good for real network data or not and even they suggested another technique to be combined with the incremental approach to have better performance.

The authors conducted a survey of incremental anomaly detection methods and attempted to assess the issues with anomaly detection, including its high false alarm rate, lack of scalability, and unsuitability for high-speed networks. The majority of the works that were evaluated were based on the benchmark KDD dataset, which has not been updated. They claimed that they saw good results in the majority of the work, but they did not know whether or not it would perform well with real network data. They even suggested adding another technique to the incremental approach to improve performance.

The work in [27] SNORT and an anomaly detection system were combined to form a hybrid intrusion detection system. Unlike the other hybrid IDSs, their strategy only uses SNORT to provide alerts, and anomaly detection is only utilized to produce SNORT automatically. To accomplish this, regular traffic is sent to the anomaly system to generate a standard profile of frequently recurring episode regulations. Real-world traffic will be injected into the system following the training period. If FERs generated from actual traffic do not match any of the FREs in the standard profile, they will be compared to the standard profile and considered suspicious. The system adds the matched rule to the SNORT when it exceeds the threshold and records it as an anomaly. Some other hybrid systems combine different anomaly detection systems depending on predetermined criteria, taking into account that each anomaly detection methodology has different detection capabilities, as compared to combining signature detection and anomaly detection methodologies. This kind of hybrid system's major goal is to reduce the excessive number of false alarms generated by traditional anomaly detection systems while maintaining a respectable detection rate. According to the test results, the HIDS has a detection rate of 60%, as opposed to the SNORT and Bro systems' 30% and 22%, respectively. The main drawback of this study is that they used restricted Internet trace data for training purposes in real time, despite the fact that the performance of an IDS depends on the training data, and they did not implement any module that explains the countermeasures once the intrusion was discovered. Many IDS has been proposed by many researches using machine learning techniques to improve the IDS from different perspectives which is described in terms of detection of novel attack, detection accuracy, reduce false alarm rate, and time consuming. Authors in [28] has proposed ensemble approach intrusion detection using Alternating Decision Tree (ADTree) and K-Nearest Neighbor (KNN) classifiers. An strategy known as an ensemble classifies new

data points by using a weighted vote of the predictions made by the classifiers to create a collection of classifiers. As a result of the mutual combination of many classifiers, it has been particularly effective in creating high performance IDSs. Using the input feature space as well as extra characteristics acquired via k-means clustering, individual classifiers are constructed. The outcome demonstrates that the approach performs better than all of the tests, especially when it comes to the R2L, U2R, and Probe classes.

Researchers in [29] have proposed hybrid prediction model by applying different classification and clustering algorithms. Their experiments were conducted using KDD cup99 and NSL-KDD dataset. In [30], Hybrid approach by combining different ML techniques has been implemented which aimed to increase attack detection rate while minimizing high false alarm rate. J48 (C45) with Random Tree, J48 (C45) with Random Forest, and Random Forest with Random Tree classifiers were used and the result have shown combining J48 with Random Tree improves performance of intrusion detection rate and false alarm rate. Author in [31] has conducted a comparative study on efficient intrusion detection system using hybrid approach which is by combining supervised and unsupervised DM techniques. Four data mining techniques, K-mean, fuzzy C mean, naïve Bayes and support vector machine are applied. The researcher discovered Fuzzy C-Mean (FCM) is a better classification technique and SVM in terms of accuracy and detection rate. The authors in [32] has proposed a hybrid machine learning technique for network intrusion detection based on combination of K-means clustering and support vector machine classification. The results have shown that the proposed technique has achieved a positive detection rate and reduce the false alarm rate. Hybrid intrusion detection method is proposed by [32] using C4.5 and support vector machine algorithms. The result demonstrate that the proposed method is better than the conventional methods in terms of the detection rate for both unknown and known attacks while it maintains a low false positive rate too.

In order to develop an intrusion detection systems (IDS) model that matches the effectiveness of real-time traffic, a reliable and substantial amount of data must be available. An intrusion detection system can only benefit from training and testing on a dataset with a broad and huge amount of high-quality data that resembles real-time traffic. The first benchmark dataset to evaluate a recently proposed intrusion detection method was KDD. It

was developed by the Defense Advanced Research Projects Agency (DARPA) and other universities to serve as a benchmark dataset for evaluating a proposed intrusion detection system. For many years, the research community utilized this standard IDS dataset as a baseline for evaluating intrusion detection systems. For the purpose of evaluating the applicability of this dataset in machine learning research, the author of [22] examined IDS studies finished between 2010 and 2015. (MLR). The bulk of research that have been published, according to this study, have employed the KDD99 dataset, which has established itself as the most popular dataset in the IDS and machine learning disciplines. In addition to KDD99, several academics in this field of study used other datasets. [6] has made an effort to research a machine learning IDS that looked into the use of decision tree algorithm-based cost sensitive learning. He made no comparisons between the outcome and other predictive modeling approaches, such as neural networks, Naive Bayes, and others.

Another research was undertaken by [7] they performed statistical analysis on the KDD99 dataset and found two key flaws that significantly impacted the performance of the evaluated IDS model, leading to a subpar assessment of anomaly detection approaches. It has a lot of duplicate records; in the train and test sets, respectively, 78% and 75% of the records are duplicated. As a result, learning algorithms will favor records that occur more frequently. In order to analyze the records in the KDD data set, the researcher used seven ML models, each of which was trained three times. 98% and 86% of the records in the train and test sets, respectively, were correctly identified. This is because random samples from the KDD train set are utilized as test sets, making it impossible to compare IDS models. To address the aforementioned issues, NSL-KDD, a condensed version of the KDD99 dataset, was developed.

The new KDD data set, NSL-KDD, was developed to fix the issues with KDD99, although it still has some issues that make it not a perfect representation of the real networks that are currently in use. Numerous academics have harshly criticized the KDD98 family evaluation dataset and proposed a new IDS dataset to address the flaws.

Utilizing evaluation criteria, a thorough analysis of the available datasets was conducted, and a technique for evaluating the IDS dataset was created. The researchers came to the conclusion that a new dataset is necessary because the existing datasets do not accurately

reflect current real-world traffic, suffer from a lack of traffic diversity and volumes, some do not cover a variety of attacks, and lack feature set and metadata. According to this evaluation framework, a complete dataset should cover eleven features or criteria. Consequently, both datasets contain issues.

According to the evaluation findings from [36], the cutting-edge IDS benchmark datasets KDD and others are no longer dependable since they fall short of the standards set by recent advancements in computer technology. The University of New Brunswick Institute of Cyber Security was driven to create an alternative dataset that matched contemporary computer technology and the associated changes in cyber-attack by the necessity to provide a trustworthy alternative IDS benchmark dataset. As a result, a new CIC-IDS2017 benchmark dataset for the assessment of IDS models based on machine learning was created in 2017. The aforementioned issues can be resolved using the current datasets.

[33] used an indirect cost sensitive feature selection approach to suggest the best feature selection for network intrusion detection. The system used a DM method and attempted to examine cost sensitive learning and feature selection concurrently in order to improve the classification performance of cost-incorporating algorithms. His research examines the Information Gain Ratio (IGR) and Correlation Feature Selection (CFS) for ranking and choosing features using the suggested cost-sensitive methodology. [33] has tried to investigate decision tree classification algorithms that used indirect cost sensitive feature ranking and selection algorithms. [26] used in his study only those records which are labeled. He did not consider those records which are not labeled. Both [6] and [33] conducted the NIDS on a supervised approach.

[34] using a Random Forest (RF) classifier to suggest a detection framework. He used data mining to eliminate aspects that were unnecessary and, as a result, increase accuracy. He used RF to train the data and was able to identify the DOS assault with up to 99.67% accuracy.

The author [35] has suggested an ensemble classifier using Bayesian net and Artificial neural network classifiers to detect assaults in the most recent NSL KDD and previous KDD cup 99 datasets. The next section will show how many other studies were undertaken to increase the detection rate of attacks using machine learning and deep learning techniques. They

examined a number of classification approaches, including ensemble methods and CART, ANN, and Bayesian Net. For the 70% partition of the data, the experiment produced the maximum accuracy rates of 97.53% for ANN and Bayes Net and 99.41% for KDD cup 99.

Qassim et al. [36] developed an anomaly-based network intrusion classifier to automatically classify activities from the Internet. The authors used a packet header-based anomaly detection system for network traffic collection. From the collected network traffic, they extracted features but the methods used for feature extraction were not stated. After that, the network anomalies classifier was applied to the extracted features. They used random committee, rotation forest, RF, and random tree as ML classifiers for intrusion detection using the WEKA tool. Their report showed that they compared the performance of the algorithms on five datasets and they found random committee and random tree performed better with similar accuracy than the other algorithms. Out of the two high performed algorithms the authors report showed that random committee gained an accuracy of 96.61%, 99.70%, 98.45%, 98.09%, and 98.18% for dataset A, B, C, D, and respectively. The proposed classification model was able to classify malicious activities having large samples effectively, but it was not effective in detecting a small number of training samples.

Karami created a solution that uses ML approaches to precisely identify anomalies and give end users graphical information. The author's major objective was to increase the detection rate while minimizing false alarms. Two self-organizing map (SOM) methods were utilized by the author: a standalone SOM with three stages and a fuzzy SOM. Outliers that were benign were found in the initial stage. SOM was performed over those chosen data in the second step and again in the third stage. Using MATLAB R2016b on the Windows platform, the author conducted the experiments. On the datasets NSL-KDD, UNSW-NB15, AAGM, and VPN-nonVPN, the suggested model was assessed. Last but not least, the experimental findings demonstrated that the suggested approach offered superior lattice adjustment with a limited number of overlapped connections among neighbors. The connections between nodes and their neighbors from the other two techniques, however, were unsuccessful.

Kshirsagar and Joshi [36] Employing data mining frameworks, a rule-based classification approach for intrusion detection was proposed. The primary objective of the suggested research project was to evaluate various rule-based classifiers for IDS and choose the

optimal one. Initial preprocessing of the audit data was done as ASCII new packet information, which was later condensed into connection records. After then, a data mining method was used to construct rules for the connection records. The suggested approach was tested on four different forms of attacks, including DoS, probing, U2R, and R2L, using the KDD CUP 99 dataset. The experiment was carried out using the WEKA tool, and according to the authors, the suggested model generated high detection accuracy for known attacks. But the work was not able to classify new attacks (attacks in the test dataset but not available in the training dataset).

IDSs can be categorized in a variety of ways depending on several factors, including information source, analysis type, response type, and detection time. The majority of IDS researches, as seen in the literature review, used the KDD and its improved version NSL-KDD for performance evaluation. It is crucial to stop using this benchmark dataset and to begin using the recently introduced benchmark dataset in order to evaluate machine learning-based intrusion detection systems effectively. The new CIC-IDS2017 benchmark dataset will be used for this study's performance assessment of IDS models.

CHAPTER 3: METHODOLOGY

This section discusses the research methodology and tools, methods, and techniques to achieve the objective of this research and used to carry out the research under consideration. The research work was developed by discovering existing findings of several kinds of literature as well as comprising our imagination. This thesis covers all parts of a standard research methodology approach that starting from problem identification to find the solution and from implementation to validation of final results.

The methodologies to be used in conducting this research are described as follows.

3.1 Research Design

The research method used in this study is based on the design science paradigm to address its general and specific objectives. It is a design science-based research study that will try to design and develop a model that will best suit simulation. Design science is chosen as a methodology because it takes in to account the action, occasion or problem that comprises a real or hypothetical condition you would meet in the work place. So, this will help to see how the difficulties of real life impact the decisions or solution proposed.

3.2 Literature Review

This project began with a study of the literature in the focus area. a research gap was discovered as a result of the review. Following that, some sort of examination of potential solutions was undertaken. This study proposes to design a intrusion detection system because to the high importance it has in the defense of a network-based system.

Several data were collected and several experimentations were carried out in order to build the proposed IDS solution. The studies took into account ML concepts and activities such as feature selection, parameter adjustment, and cross validation. the materials needed, the general actions taken, and the accompanying rationale for selecting these resources, as well as other relevant principles included into the task, are covered in the following sections of the chapter.

3.3 Tools Used

Weka is collection of different machine learning algorithms which can be used for data mining [37] It is written in Java and is especially used for educational research purposes. Weka is a platform independent, open source, easy to use, data processing tool, flexible for scripting experiments and graphical user interface tool. Weka contains different tools and algorithms for regression, classification, pre-processing and clustering. It is supportable on different platforms such as Mac OS, Linux and Windows. When dealing with large data sets, it is best to use a CL based approach as Explorer tries to load the whole data set into the main memory causing performance issues. We have used a standard dataset, CIC-IDS2017 intrusion detection system dataset which is prepared by Canadian Institute for Cyber security.

The WEKA tool incorporates the following steps [37]

- Analysis and pre-processing of the features in the database and assessing the correctness of the data.
- Definition of the class attributes which divide the set of instances into the appropriate classes.
- Extraction of the potential features to be used for classification.
- Selection of a subset of features to be used in the learning process.
- Investigation of a possible imbalance in the selected data set and how it may be counteracted.
- Selection of a subset of the instances, i.e., the records that learning is to be based on.
- Application of a classifier algorithm for the learning process.
- Decision on a testing method to estimate the performance of the selected algorithm.

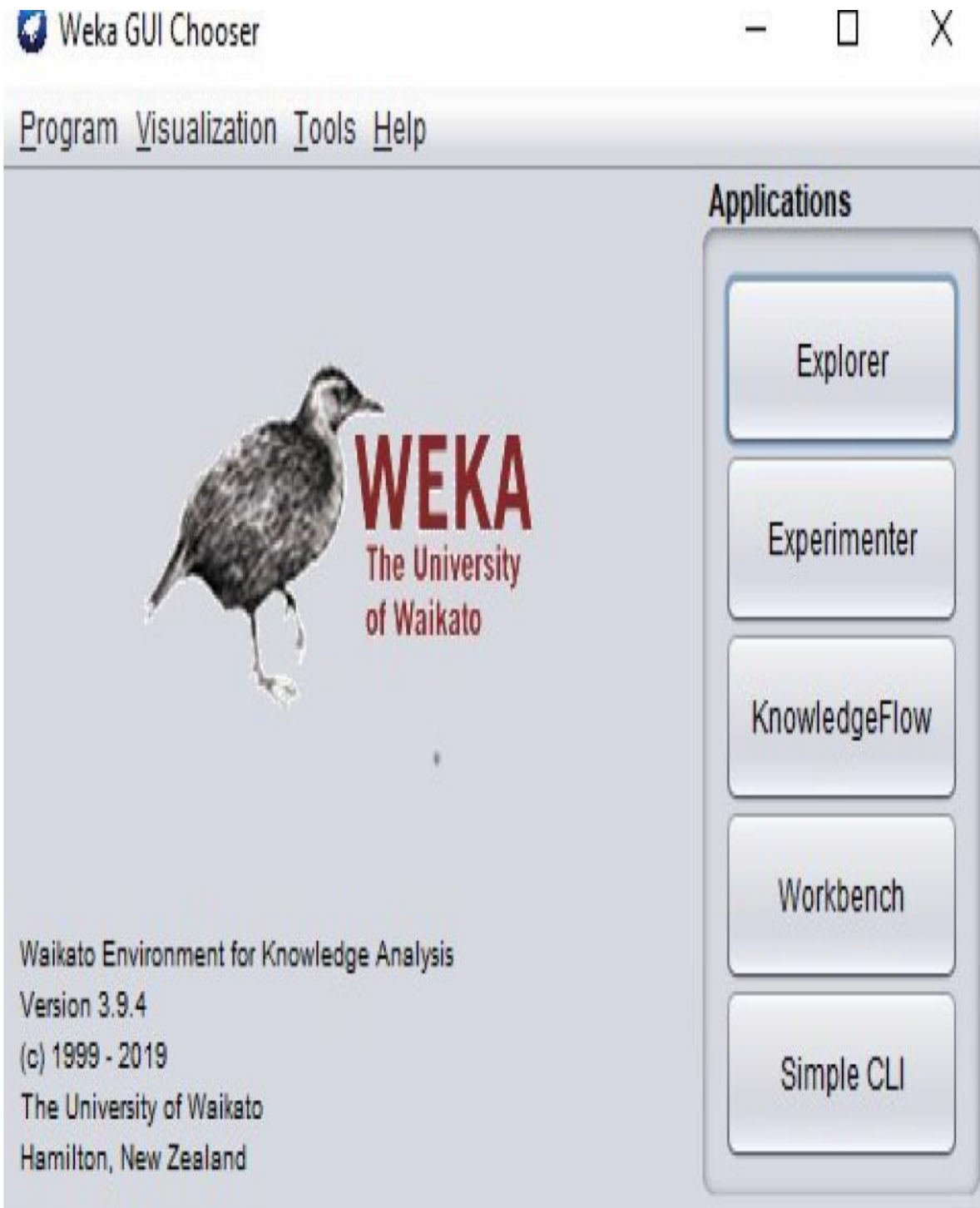


Figure 3.1 WEKA GUI

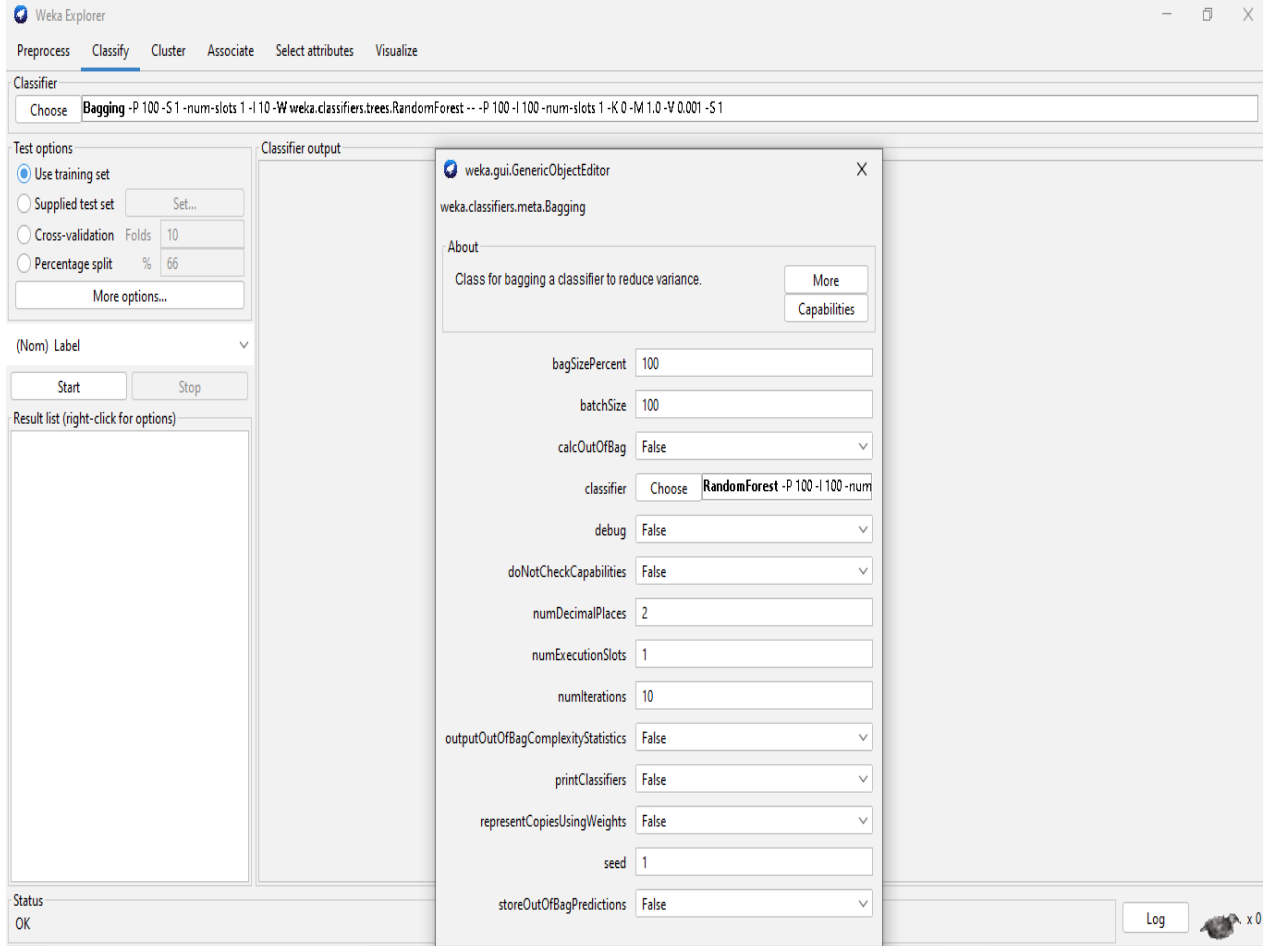


Figure 3.2 WEKA ensemble method simulation GUI

3.4 Dataset Preparation

A novel intrusion detection assessment dataset is discovered through a review of the literature. the CIC-IDS 2017 dataset was created by the University of New Brunswick Institute of Cyber Security. This is a new benchmark dataset that has been provided for the effective and robust evaluation of machine learning-based intrusion detection systems. this study uses the new CIC-IDS2017 benchmark dataset for evaluating the performance of IDS models.

Analysis of earlier datasets, such as KDD-Cup99 and NSL-KDD, revealed shortcomings, opening the path for newer datasets that addressed the concerns discovered. CIC-IDS2017 is one of the most current datasets for network intrusion detection. It has the benefit of being available as raw data as well as flow-based features in CSV files. CICIDS2017 is a dataset created by the Canadian Institute for Cybersecurity for IDSs and IPSs.

This study uses of Machine Learning CSV data from the CICIDS-2017 dataset from the ISCX Consortium. Machine Learning CSV is made up of eight (8) traffic monitoring sessions, each in the form of a comma separated value (CSV) file. this file contains both benign and malicious communication. The attack traffics are described in further depth in the third column of Table 3.1. This dataset contains 14 different forms of intrusions in addition to typical and benign traffic.

The eight CSV files in Table 3.1 are concatenated into a single CSV file. this CSV file is then turned into an ARFF file in order to process the dataset with Weka software. The experiment makes use of the whole Wednesday's traffic collection from Machine Learning CSV data. was chosen since it is the major session dealing with various forms of attacks. Furthermore, the attack in Friday's capture included.

The dataset has two characteristics or columns called "Fwd Header Length," which are redundant, thus one of them must be eliminated. After deleting the duplicate characteristics, there are only 77 features left to examine. As mentioned in the CICIDS-2017 data, data prone to high-class imbalance will have a negative influence on detection accuracy and false alarm. by adopting solution suggested by Karimi *et al.* [38] and Panigrahi and Borah [39] a new labeling attack traffic is introduced as listed in Table 3.4. The 77 features are already in numerical data type, so no data transformation is required to feed the data into Weka software.

After relabeling the attack classes, the whole Wednesday traffic acquisition of Machine-Learning CSV data is divided into 80% and 20%. As shown in Table 3.6, 80% of the data is utilized for training, while the remaining 20% is used for testing. the data component was split 80:20.

A traffic capture was performed for five days, from Monday to Friday, during the production of the whole CICIDS2017 dataset. The dataset supports twelve different types of attacks. DoS attack traffic gathering was mostly done on Wednesday's traffic capture session. There are four DoS attack families: DoS Hulk, DoS Slowhttptest, Slowlories, and Slowlories, this session has included functionality for DoS GoldenEye attack. this session also includes a traffic capture for the Heartbleed attack. Additional DDoS assaults are performed, and the

associated traffic capture is included in Friday's release. CICIDS-2017 features more complicated sorts of attacks, as shown in Table 3.1, and the rationale for using it is to have a dataset that closely mimics current real-world network traffic in the trials.

3.5 CIC-IDS2017 Dataset Description

The CICIDS2017 dataset closely resembles real-world network data. PCAP (research group create a free and open resource Machine Learning dataset) repository extracts 78 features and 79 with labels using CICFlowmeter-V3.0. As indicated in Table 2, this dataset contains the abstract characteristic attitudes of 25 users according to the HTTP, HTTPS, FTP, SSH, and email protocols. Data is collected during working hour periods. The cyberattacks in this dataset, according to the 2016 McAfee Report, are classified as brute force FTP, brute force SSH, DoS, Heartbleed, web, infiltration, botnet, and DDoS attacks, which are not seen in any of the previously stated datasets. CICIDS2017 uses the B-Profile system to accomplish abstract characteristic profiling of human interactions and the Alpha profile to simulate various multi-stage attack scenarios.

The CICIDS2017 dataset comprises benign and up-to-date common attacks, and it closely reflects authentic real-world data. It also includes the findings of the network traffic analysis performed using Cyclometer, which included labeled flows based on the time stamp, source and destination IPs, source and destination ports, protocols, and attack vectors (CSV files). the extracted features definition is also accessible.

3.5.1 Attack Types in CIC-IDS2017 Dataset

Malicious network traffic is here defined as any traffic produced from an attack with the intent of doing harm or intrusion to a computer system or network. The datasets selected for this research contain the following classes of malicious traffic.

- 1 BENIGN:** proposes a realistic background traffic produced B-Profile system. the B-Profile is in charge of profiling abstract human interaction behavior and generating naturalistic innocuous background traffic. It is built by applying the abstract features of human and attack behavior to a varied set of network protocols with varying topologies.
- 2 Denial of service (DoS):** Is a form of attack that aims to shut down a system or network, rendering it inaccessible to its intended users, by flooding the targeted machine in an

attempt to overwhelm it. DoS attack normally do not result in the theft or loss of major information or other assets, but they can cost the victim a substantial amount of time and money to handle. Apache2, Mail bomb, SYN Flood, ICMP Flood, and Ping of Death are some examples.

- 3 Distributed Denial of service (DDoS):** DDoS enables exponentially more requests to be delivered to the target, boosting attack strength. It also makes attribution more difficult because the real source of the assault is more difficult to determine.
- 4 Brute Force Attack:** This is one of the most common assaults, which may be used not only to break passwords, but also to find hidden pages and information in an online application. It is essentially attack using software tools to break password by guessing repeatedly using different password combination.
- 5 Botnet attacks:** They use a network of Bots (Zombies), which are malware-infected computers that may be triggered to undertake attacks on other devices, such as spam e-mail or DoS.
- 6 Port scanning:** When an attacker sends probe packets to a network or system, he or she gathers intelligence from the responses. the attacker can determine which ports are open as well as whether or not susceptible services are operating on those ports.
- 7 SQL-injections** are fraudulent database queries that are frequently designed to extract large amounts of data from a database. They can be injected, for example, through poorly secured forms on web pages.
- 8 Cross-site-scripting (XSS)** Malicious scripts are included in the HTML content of a web page to carry out attacks. It may be used to steal a user's cookies in order to impersonate them.
- 9 Heartbleed** is a flaw identified in prior Open-SSL versions of the Heartbeat Protocol in 2014.

Table 3.1 Description of files containing CICIDS-2017 dataset

Name of Files	Day Activity	Attacks Found
Monday-WorkingHours.pcap_ISCX.csv	Monday	Benign (Normal human activities)
Tuesday-WorkingHours.pcap_ISCX.csv	Tuesday	Benign, FTP-Patator, SSH-Patator
Wednesday-workingHours.pcap_ISCX.csv	Wednesday	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Thursday	Benign, Web Attack – Brute Force, Web Attack – Sql Injection, Web Attack – XSS
Thursday-WorkingHours-Afternoon-Infiltration.pcap_ISCX.csv	Thursday	Benign, Infiltration
Friday-WorkingHours-Morning.pcap_ISCX.csv	Friday	Benign, Bot
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	Friday	Benign, PortScan
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Friday	Benign, DDoS

Table 3.2 Records in the CICIDS-2017 Dataset

Type of Attack	Day	Total Records
Benign	Monday	529918
Brute Force Attack	Tuesday	445909
Heartbleed Attack/ DoS Attack	Wednesday	692703
Web Attack	Thursday (Morning)	170366
Infiltration Attack	Thursday (Afternoon)	288602
Botnet Attack	Friday (Morning)	191033
Port Scan Attack	Friday (Afternoon)	286467
DDoS Attack	Friday (Afternoon)	225745

Table 3.3 The Class wise instance occurrence of CICIDS-2017 dataset

Class Labels	Number of instances
BENIGN	2359087
DoS Hulk	231072
PortScan	158930
DDoS	41835
DoS GoldenEye	10293
FTP-Patator	7938
SSH-Patator	5897
DoS slowloris	5796
DoS Slowhttptest	5499
Botnet	1966
Web Attack – Brute Force	1507
Web Attack – XSS	652
Infiltration	36
Web Attack – Sql Injection	21
Heartbleed	11

Table 3.4 Distribution of selected dataset for this study from CIC-IDS2017

	Dataset label	Number of Records selected
1	BENIGN	220103
2	DoS slowloris	2898
3	DoS slowhttptest	2749
4	DoS Hulk	115536
5	DoS GoldenEye	5146
6	Heartbleed	5
7	Portscan	124
8	Web Attack	6
9	Total record	346567

Table 3.5 selected Features of CICIDS-2017 Dataset

Features no	Features	Feature no.	Features
1.	Destination Port	41.	Packet Length Mean
2.	Flow Duration	42.	Packet Length Std
3.	Total Fwd Packets	43.	Packet Length Variance
4.	Total Backward Packets	44.	FIN Flag Count
5.	Total Length of Fwd Packets	45.	SYN Flag Count
6.	Total Length of Bwd Packets	46.	RST Flag Count
7.	Fwd Packet Length Max	47.	PSH Flag Count
8.	Fwd Packet Length Min	48.	ACK Flag Count
9.	Fwd Packet Length Mean	49.	URG Flag Count
10.	Fwd Packet Length Std	50.	CWE Flag Count
11.	Bwd Packet Length Max	51.	ECE Flag Count
12.	Bwd Packet Length Min	52.	Down/Up Ratio
13.	Bwd Packet Length Mean	53.	Average Packet Size
14.	Bwd Packet Length Std	54.	AvgFwd Segment Size
15.	Flow Bytes/s	55.	AvgBwd Segment Size
16.	Flow Packets/s	56.	Fwd Header Length
17.	Flow IAT Mean	57.	FwdAvg Bytes/Bulk
18.	Flow IAT Std	58.	FwdAvg Packets/Bulk
19.	Flow IAT Max	59.	FwdAvg Bulk Rate
20.	Flow IAT Min	60.	BwdAvg Bytes/Bulk
21.	Fwd IAT Total	61.	BwdAvg Packets/Bulk
22.	Fwd IAT Mean	62.	BwdAvg Bulk Rate
23.	Fwd IAT Std	63.	SubflowFwd Packets
24.	Fwd IAT Max	64.	SubflowFwd Bytes
25.	Fwd IAT Min	65.	SubflowBwd Packets
26.	Bwd IAT Total	66.	SubflowBwd Bytes
27.	Bwd IAT Mean	67.	Init_Win_bytes_forward
28.	Bwd IAT Std	68.	Init_Win_bytes_backward
29.	Bwd IAT Max	69.	act_data_pkt_fwd
30.	Bwd IAT Min	70.	min_seg_size_forward
31.	Fwd PSH Flags	71.	Active Mean
32.	Bwd PSH Flags	72.	Active Std
33.	Fwd URG Flags	73.	Active Max
34.	Bwd URG Flags	74.	Active Min
35.	Fwd Header Len	75.	Idle Mean
36.	Bwd Header Length	76.	Idle Std
37.	Fwd Packets/s	77.	Idle Max
38.	Bwd Packets/s	78.	Idle Min
39.	Min Packet Length	79.	Label
40.	Max Packet Length		

3.6 Data Preprocessing

Basic steps in data cleaning and preparation include removing noise or outliers as necessary, acquiring the necessary data to model or account for noise, selecting methods for handling missing data fields, and taking into consideration time sequence information and known changes. Preprocessing is the process of preparing the raw dataset for experimentation and analysis in order to obtain the suggested results.

[44] suggests categorizing system events and making predictions about the future based on the observations of the past. IDS Datasets often contain massive volumes of log entries. However, many of the log entries include duplicate information since a single root cause may activate several components, which may output distinct log messages for the same root cause. However, repeated log entries convey the same information and are useless for machine learning. We used a mix of log message normalization and filtering to eliminate redundant log events. another aim for preprocessing activities was to avoid missing values and incomplete information. Log messages can be normalized using transformations that comprise the following steps:

- Extracting log files from zip file
- Removing incomplete log records in rows
- Parsing log files in to suitable format for analysis
- Deleting some missing values from record
- Replacing words into numbers
- Removing duplicate and redundant values
- Exporting text log files into excel format
- Converting string values into integer
- Labeling the class values for all records by representing 1 for attack and 0 for normal (non-attack) we labeled by following some attack word hints like the attribute value worm propagation attempt, network scan, privacy violation, etc. we considered above terms as attack and the rest as normal.
- Converting excel log file into SCV file format
- converting arff file format for WEKA implementation.

Table 3.6 The distribution of training and testing dataset

Dataset label	Instances of Training dataset	Instances of Testing dataset
BENIGN	176097	44006
DoS slowloris	2321	577
DoS slowhttpptest	2177	572
DoS Hulk	92478	23058
DoS GoldenEye	4077	1069
Heartbleed	3	1
Portscan	97	27
Web Attack	5	1
Total record	277254	69313

3.7 Feature extraction

Information Gain is the most used feature selection technique. It is a filter-based feature selection [40], [41]. Information Gain uses a simple attribute rank and reduces noise that caused by irrelevant features then detects a feature that have most of information base in specific class. The best feature is determined by calculating feature's entropy. Entropy is a measure of uncertainty that can be used to infer the distribution of features in a concise form [42].

Information Gain in the WEKA environment is the feature selection method we choose for our thesis because it is a filtered-based method that produces more stable sets of selected features due to its robustness against overfitting. Filter-based techniques have an overall computational cost of $O(mn^2)$, where m is the number of training data and n is the number of attributes/features. In comparison to embedded and wrapper-based approaches, it is less [48]. Wrapper-based approaches run a considerable risk of overfitting due to their complexity. Hence, the execution time of the classification algorithms employed in the attack detection process will be decreased by applying feature selection techniques that create meaningful, relevant, a smaller number of characteristics, and less computing complexity.

The IDS assigned to the features range from 1 to 77. The Information Gain ranks the features based on their weight values, and the minimal weight is manually established using a trial

and error technique. the researchers suggest in this study to rank and arrange the characteristics based on the minimal weight values. as a result, feature groups are formed, with each feature group containing a varied number of characteristics, as illustrated in Table 3.5. Furthermore, all feature groups will be evaluated using the five classifier algorithms to identify whether feature groups are successful enough to be utilized for categorization of attack types. The dataset is reduced to 77 characteristics once the unnecessary columns are removed.

3.8 Performance metrics

In this work, we compared performance measures like as accuracy, precision, recall, and F score, which are defined as follows: Accuracy: This is defined as the percentage of correct predictions; that is, the percentage of anomalous traffic that is correctly categorized. It is the ratio of accurate detections to the total number of records in the dataset, and it may be calculated as follows:

- Detection rate (or “true positive rate”, “recall”, “sensitivity”) is the proportion of attacks that are correctly detected.

$$\text{Detection rate} = \text{TP} / (\text{TP} + \text{FN})$$

- False positive rate (or “false alarm rate”) is the proportion of normal traffic incorrectly flagged as attack.

$$\text{False positive rate} = \text{FP} / (\text{TN} + \text{FP})$$

- Accuracy is the fraction of correctly identified results (attack and normal traffic). In multiclass classification, accuracy is equal to the Jaccard index, which is the size of the intersection divided by the size the union of the label sets.

$$\text{Accuracy} = \text{TP} + \text{TN} / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

- Precision (also called positive predictive value) is the proportion of identified attacks that are indeed attacks.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

- Recall: Recall is the ratio of the number of records correctly classified to the number of all corrected events, and can be computed as follows:

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

- F1-score is the harmonic mean of precision and recall (previously called “detection rate”).

$$\text{F1-Score} = 2TP / (2TP + FN + FP)$$

3.9 Machine learning Algorithms

Machine learning algorithms used in this thesis are listed here with a short conceptual description.

Random Forest (RF)

Random Forest is one of the ensemble classifier methods Random Forest (RF) is a machine learning (ML) classifier that uses numerous decision trees on distinct subsets of a dataset and averages the results to enhance prediction accuracy. A condition is compared with one or more properties of the incoming data at each tree node. RF aggregates predictions from numerous trees to produce the outcomes rather than relying on a single decision tree. Each tree votes for a certain class, and the class with the most votes is the projected class. The number of classifications performed by RF on unbalanced datasets is relatively [43].

Decision Tree

A classifier defined as a recursive instance space split is a decision tree [50]. By constructing a tree from training instances with class labels on the leaves, a decision tree does classification. Since the decision tree's nodes form a rooted tree, it is a directed tree without any incoming edges. There is only one incoming edge for each other node. One with outgoing edges is referred to as an internal or test node. Leaves refer to all further nodes (also known as terminal or decision nodes). According to a discrete function of the input attribute value values, each internal node in a decision tree divides the instance space into two or more sub-spaces. In the most basic and common instance, each test considers a single attribute, such that the instance space is partitioned according to the attribute's value. In the case of numeric attributes, the condition refers to a range.

Bayes Network (BN)

A Bayesian Network (BayesNet) is a probabilistic model that employs a graphical structure to solve complex problems. It provides knowledge about a domain in which each node represents a set of random variables, and each edge represents the statistical relationship

between these variables [44]. In addition, each node has a conditional probability distribution associated with the corresponding random variables. The random variable can be conditions or states. BayesNet is used to represent probabilistic causal relationships [45].

J48

The C4.5 method is a classification technique that uses information theory to generate decision trees. It is an adaptation of Ross Quinlan's older ID algorithm, also known as J48 in Weka. C4.5's decision trees are utilized for classification, and as a result, C4.5 is frequently referred to as a statistical classifier. Accounting for missing data, decision tree pruning, continuous attribute value ranges, rule generation, and other features are included in the J48 version of the C4.5 algorithm. J48 is an open-source Java version of the C4.5 algorithm in the WEKA data mining tool. J48 supports categorization using decision trees or rules derived from them [46]

OneR

OneR, short for "One Rule", is a simple, yet accurate, classification method that creates one rule for each predictor in the data, then picks the rule with the least overall error as its "one rule". To establish a rule for a predictor, we build a frequency table for each predictor against the goal. It has been demonstrated that OneR provides rules that are only slightly less accurate than state-of-the-art classification algorithms while also providing rules that are straightforward for people to comprehend. [47]

3.10 Ensemble Methods

Ensemble methods are techniques that aim at improving the accuracy of results in models by combining multiple models instead of using a single model. The combined models increase the accuracy of the results significantly. This has boosted the popularity of ensemble methods in machine learning. Ensemble methods aim at improving predictability in models by combining several models to make one very reliable model [48]

Ensemble methods are divided into two categories: sequential ensemble techniques and parallel ensemble approaches. Sequential ensemble approaches, such as Adaptive Boosting, produce base learners in a sequential order (AdaBoost). The successive production of base learners fosters reliance among the base learners. The model's performance is then enhanced

by giving bigger weights to previously misrepresented learners. Parallel ensemble approaches create base learners in a parallel fashion, such as random forest. Parallel techniques make use of the parallel creation of base learners to develop independence among the base learners. The independence of base learners considerably lowers the inaccuracy caused by the use of averages [49].

[50] In base learning, the majority of ensemble strategies use a single algorithm, resulting in homogeneity across all base learners. Homogenous base learners are base learners of the same kind with comparable characteristics. Other approaches make use of heterogeneous base learners, resulting in heterogeneous ensembles. Heterogeneous base learners are different sorts of learners. The following are the main type of ensemble methods:

3.10.1 Bagging

Bagging (Bootstrap Aggregating) is an ensemble approach that produces independent samples of the training dataset and creates a classifier for each sample. The findings of these many classifiers are then integrated (such as averaged or majority voting). The secret is that each sample of the training dataset is unique, providing each trained classifier a somewhat different emphasis and viewpoint on the problem [51]. Bagging, short for bootstrap aggregation, is mostly used in classification and regression. It improves model accuracy by using decision trees, which greatly minimizes variance. The decrease of variance improves accuracy by removing overfitting, which is a problem for many prediction models [52].

Bootstrapping and aggregation are the two categories under which bagging is categorized. Bootstrapping is a sampling strategy where samples are taken utilizing the replacement procedure from the entire population (set). The sampling with replacement method aids in the randomization of the selection process. The process is finished by applying the base learning algorithm to the samples [60]. In bagging, aggregation is used to combine all possible outcomes of the prediction and randomize the output. Predictions will be inaccurate without aggregation since all outcomes will be ignored. As a result, the aggregate is based on probability bootstrapping processes or on all predictive model outputs. Bagging is useful because it combines weak base learners to generate a single strong learner that is more stable than single learners. It also removes any variance, which reduces model overfitting. Bagging

has the disadvantage of being computationally costly. When the right bagging technique is not followed, it might lead to increased bias in models [53].

3.10.2 Boosting

Boosting is an ensemble approach that begins with a basic classifier that has been trained on training data. A second classifier is then developed behind it to focus on the cases in the training data that the first classifier missed. The process of adding classifiers continues until a limit in the number of models or accuracy is achieved [54]. Boosting is an ensemble strategy that uses prior predictor failures to improve future predictions. The strategy merges numerous weak base learners into a single strong learner, considerably boosting model predictability. Boosting works by placing weak learners in a sequential order so that weak learners can learn from the next learner in the sequence, resulting in improved prediction models [55].

Gradient boosting, Adaptive Boosting (AdaBoost), and XGBoost are all types of boosting (Extreme Gradient Boosting). AdaBoost employs weak learners in the form of decision trees, which typically have only one split, also known as decision stumps. AdaBoost's core decision stump consists of observations with equal weights [56]. XGBoost uses decision trees with boosted gradients to increase speed and performance. It is strongly reliant on the computing speed and performance of the target model. Model training needs to be done in a certain order, which slows down the implementation of gradient boosted machines [57].

3.10.3 Stacking (Blending)

Blending is an ensemble approach in which many separate algorithms are trained on training data and a meta classifier is trained to take the predictions of each classifier and generate correct predictions on unseen data [58].

In general Ensemble approaches are good for minimizing model variance and hence enhancing prediction accuracy. When many models are integrated to generate a single forecast that is chosen among all other potential predictions from the combined models, the variance is removed. An ensemble of models mixes different models to guarantee that the resultant forecast is the best possible based on all predictions [59].

CHAPTER 4: EXPERIMENT

4.1 Overview

We described the design of the suggested ensemble intrusion detection system in this Chapter. Different components of the planned ensemble IDS are outlined, along with their significance and the strategies to be used in their construction. The architecture is presented in this chapter, along with the algorithms that have been proposed.

4.2 System Configuration

All simulations in this experiment are executed on a computer with specification of Intel Core i7 processor with 2.9 GHz 8 GB RAM, running Windows 10 as Operating System. For analysis purposes, the Weka 3.8 with heap size of 3072 MB, as machine learning software is used.

4.3 Proposed ensemble machine Learning model

The use of various ML models to solve problems and make data-driven decisions has become the most significant topic. There are three types of methodologies that are widely utilized to create ML models. The first is that just one ML method, either supervised or unsupervised learning, is used. The second method is hybrid, which employs both supervised and unsupervised learning algorithms. The goal of this strategy is to have both algorithms complement each other in order to increase the model's performance on a certain job. The third strategy is known as ensemble learning, and it involves the use of numerous ML algorithms to form an ML model; in our work, we employed ensemble learning algorithms.

The ensemble machine Learning (EML) approach generates several instances of classic Machine Learning methods and combines them to produce a single optimal solution to a problem. When compared to the old technique, this approach produces superior predictive models. The most common reasons for using the EML approach are when there are uncertainties in data representation, solution objectives, modeling methodologies, or the presence of random beginning seeds in a model. The instances or candidate methods are referred to as base learners. Each base learner operates separately like a standard ML approach, and the final results are integrated to form a single robust output. For regression

and classification techniques, the combination might be done using either of the averaging (simple or weighted) approaches and voting (majority or weighted).

The random forest algorithm is a well-known machine learning algorithm. It is an ensemble model using bagging as the ensemble technique and decision tree as the individual model, which means that during the training phase of the algorithm, it makes many decision trees (thus the name forrest) and then delivers the most popular output as its categorization. One of the most significant advantages of the random forest is its speed; categorization occurs very rapidly [60].

Random forest generates a forest of independent dataset subsets. The optimal split is discovered by randomly picking n variables at each node. In general, ensemble learning is a model that produces predictions based on a number of distinct models. When different models are combined, the ensemble model becomes more flexible (less biased) and less data-sensitive (less variance) [61]. We employ a Random Forest-based ensemble machine learning strategy in this study.

Meta algorithms are used to merge numerous models into a single one, with the goal of improving the performance of the machine learning model throughout the dataset. Bagging is utilized to minimize variance in our model, and the random forest bagging approach is employed. Bagging-based ensemble approaches generally function in two steps. The first involves applying several ML models to a subset of the dataset, while the second involves aggregating multiple ML models into a single integrated ML model.

Our goal is to create an ensemble learning model that is more accurate, has less false alarms, and capable of sniffing and identifying unexpected attacks. As a result of the literature study, we can infer that a single algorithm cannot identify all forms of intrusions with outstanding performance. Some may excel in detecting one form of intrusion but fall short on another. This is why the ensemble approach was proposed. Since our primary goal from the beginning has been to boost detection capabilities while decreasing false rate.

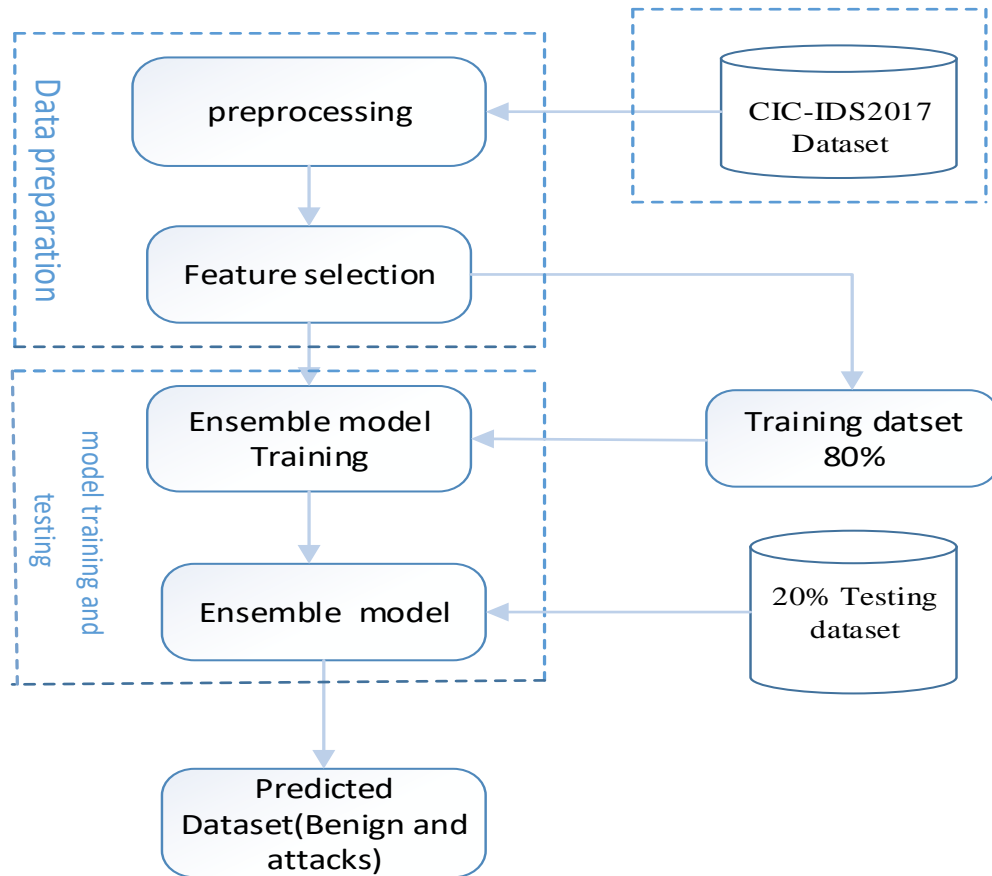


Figure 4.1 proposed System Model Architecture

4.3.1 Sequence of Steps

First, the CIC-IDS2017 dataset is loaded and pre-processed, data cleaning, imputation and feature scale or normalization are performed which is explained in detail in the methodology section. Feature selection techniques are applied to identify contributing features for the model. After the important features are selected using the feature selection techniques, Next, we train the ensemble model using selected. Then The complete ensemble ML model is built using bagging method and random forest-based ensemble classifiers, finally model is evaluated by its performance in terms of Accuracy, Precision, Fmeasure, and Recall.

CHAPTER 5: ANALYSIS AND RESULTS

5.1 Performance Evaluation Result

The selected six intrusion detection algorithms which were built using training dataset in the previous section were also evaluated on the testing dataset. To compare performance of the models' parameters such as, accuracy, precision, True Positive Rate (TP) which is the detection rate, Recall and False Positive Rate (FP) also known as False Alarm Rate of each algorithm on a specific attack category were recorded. These evaluation parameters are the most important criteria for the algorithms to be considered as the best models for the given attack category [62]. Experiment results are given in the Table 5.1. The first column in the table is the algorithms used to detect the intrusion, the last seven contains the criteria used to measure the performance of the highest of the accuracies between all algorithms is highlighted in bold.

To analyze the performance of the feature selection performed by Information Gain and the six (6) classifier algorithms, seven (7) measurement metrics are used, they are: True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, Accuracy, percentage of incorrectly classified and execution time [63] The execution time is measured during the training time (the time measured from the classification process starts until the classification process stops). In the experiment, each feature subset is classified by RT, BN, RF, OneR, DecisionStump and J48 classifier.

Table 5.1 Performance Comparison of the five Classifiers algorithm on selected dataset

Classifier	Accuracy	FP Rate	Precision	Recall	F-Measure	ROC Area
Random Tree	99%	0.01 %	99 %	99%	99.9 %	99.9 %
BayesNet	98.7%	0.7%	98.4%	98.7%	99%	99%
J48	97.9	0.1%	99%	99%	99%	99%
RandomForest	99.9%	0.001%	100%	99%	99%	98%
OneR	93.7%	8.5%	95%	93%	94%	92%
DecisionStump	85%	24.1%	82%	85%	78%	81%

The evaluation result, in Table 5.1, shows that for a given attack category, certain algorithms demonstrate superior detection performance compared to others. The algorithm Random Forest Classifier has the highest detection accuracy, false positive with 99.9% and 0.01% respectively, while J48 algorithm outperforms the others with its TP rate at 98 % in detecting DDos. And for the case of Heartbleed attacks, which is one of the rare attack types, OneR is the best classifier with 93% detection rate.

From the experiment result we can see that all algorithms were able to detect DoS attack types with high detection performance, the highest being Random Forest with 99.12% detection rate. J48 surpasses the others with 98.51% while detecting portscan, followed by Random Forest with 98.27% and 93.1% respectively. Rare attack types like Heartbleed were only detected with higher detection rate by Random Forest. OneR algorithm has the second highest detection rate for DDos attack types but it has the highest false alarm rate compared with the others. Random Forest performed well in detecting most attack types with relatively low FAR in all attack category. But most importantly the experiment illustrates no single algorithm could detect all attack categories with a high probability of detection and a low false alarm rate. This observation strengthens the thinking that the combination of different algorithms should be used to deal with different types of network attacks.

Table 5.2 Detailed Accuracy by classification using Ensemble model method.

Class	TP Rate	FP Rate	Precision	Recall	F Measure	ROC Area
BENIGN	0.999	0.001	1.000	0.999	0.999	1.000
DoS slowloris	0.995	0.000	0.997	0.995	0.996	1.000
DoS Slowhttptest	0.993	0.000	1.000	0.993	0.996	0.999
DoS Hulk	0.999	0.001	0.998	0.999	0.999	1.000
DoS GoldenEye	0.993	0.000	0.998	0.993	0.995	0.999
Heartbleed	0.667	0.000	1.000	0.667	0.800	1.000
PortScan	1.000	0.000	1.000	1.000	1.000	1.000
Web Attack	1.000	0.000	1.000	0.000	1.000	1.000

Confusion Matrix of Ensemble model

=== Confusion Matrix ===

	a	b	c	d	e	f	g	h	<-- classified as
43974	0	0	0	32	0	0	0	0	a = BENIGN
3	574	0	0	0	0	0	0	0	b = DoS slowloris
2	2	568	0	0	0	0	0	0	c = DoS Slowhttptest
10	0	0	23046	2	0	0	0	0	d = DoS Hulk
4	0	0	4	1061	0	0	0	0	e = DoS GoldenEye
1	0	0	0	0	0	2	0	0	f = Heartbleed
0	0	0	0	0	0	0	27	0	g = PortScan
1	0	0	0	0	0	0	0	0	h = Web Attack

Table 5.3 Classification accuracy using Ensemble model

Model	No of test dataset instances	Correctly classified	Incorrectly Classified	% Correctly classified	% incorrect classification
Ensemble Model	69313	69252	61	99 %	0.01%

As shown in the resulting confusion matrix, the Ensemble model has classified 69252 dataset records correctly and 61 dataset records incorrectly. Thus, EM scored an accuracy of 99% while 0.01% of the records are incorrectly classified. The performance result of this model derived and the confusion matrix is presented in Appendix.

Table 5.4 Detection result of selected algorithm and ensemble model

Classifier	Accuracy	TP Rate	FP Rate	Precision	Recall	F Measure	ROC Area
Ensemble method	99 %	99%	0.01%	99.9%	99%	99.9 %	100%
Random Tree	98.8 %	98.9 %	0.1 %	97.9 %	98.9 %	98.9 %	97.9 %
BayesNet	98.7%	98.7%	0.7%	99.4%	98.7%	99%	99%
J48	97.9	97%	0.01%	98%	98%	98%	98%
RandomForest	98.8%	99%	0%	100%	100%	100%	98%
OneR	93.7%	93%	8.5%	95%	93%	94%	92%
DecisionStump	85%	85%	24.1%	82%	85%	78%	81%

As described in the table above in ensemble method performance all the other attack types are classified with higher detection rate, precision, recall and lower false positive rate by the proposed ensemble algorithm. The accuracy of ensemble method is 99%, false positive rate of 0.01%, Recall 99%, precision 99%.

In general, when comparing the random forest-based bagging ensemble method and with single classifier algorithms applied previously on the same dataset, using bagging ensemble method enhance the accuracy of intrusion detection and also was able to lower false positive rate and also perform well in all performance metrics for attack found in CISIDS2017 Dataset even if it still needs improving in its false alarm rate. The proposed ensemble method combines the advantages bagging technology detection approaches, which manages to improve the accuracy of the system significantly, when compared to the basic single algorithm systems.

CHAPTER 6: CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

Nowadays machine learning techniques were becoming preferable to protect information and assets from cyber-attack. It used large amount of data to learn the machine and predict the attack based on behavior it is proactive approach. In this paper, we have examined various classification algorithms and ensemble model algorithms based on random forest-based ensemble method is proposed for detecting network intrusion with the help of WEKA on CICIDS-2017 dataset and have shown the best performance in terms of accuracy, precision, Recall, F-measure and time to build a model. Simulation results shows that the use of feature selection algorithms with reduces the dimension of dataset, time to build a model, false alarms and induces high performance results.

6.2 Future Works

In the future work,

- The data used for this thesis obtained from publicly available data source, we recommend for future work it is better to evaluate the model using real time traffic data in real environment to predict and classify cyber-attacks.
- The combined intrusion detection system can be extended as an intrusion prevention system to enhance the performance of the system.
- we will consider the idea of advanced machine learning, deep learning and hybrid algorithms to detect network intrusion and anomaly with WEKA on CICIDS-2017 dataset to achieve a higher level of performance.

References

- [1] global-digital-report,"wearesocial.com,"2018.[Online].Available: <https://wearesocial.com>. [Accessed 2022].
- [2] "cyber-security-attacks- in-2017-survey.," *apanews.net*, p. [Accessed 28 01 2012]., 2017.
- [3] INSA, "esega.com," INSA, 10 2019 Accessed 28 01 2022. [Online]. Available: <https://www.esega.com/News/NewsDetails/4202/INSA-Reports- Ethiopia-Hit-by-256-Cyber-Attacks-in-Six-Months>". [Accessed 10 jan 2022].
- [4] Atawodi and S. Ilemona , "A Machine Learning Approach to Network Intrusion Detection System Using K Nearest Neighbor and Random Forest," *Master's Theses*, p. 651, 2019.
- [5] pankaj, "geeksforgeeks.org," geeksforgeeks, 17 Jan 2022. [Online]. Available: <https://www.geeksforgeeks.org/intrusion-detection-system-ids/>. [Accessed 10 dec 2022].
- [6] W. Meng, W. T. Elmar , Qingju Wang,, Yu Wang and h. Jinguang, "When intrusion detection meets blockchain technology: a review," *Ieee Access*, vol. 6, p. 10179–10188, 2018.
- [7] www.unb.ca, "www.unb.ca," Canadian institute for cyber security , 10 January 2018. [Online]. Available: <https://www.unb.ca/cic/datasets/ids-2018.html>. [Accessed 20 januanry 2022].
- [8] Cuelogic Technologies, "medium.com," Cuelogic Technologies, 13 May 2019. [Online]. Available: <https://medium.com/cuelogic-technologies/evaluation-of-machine-learning-algorithms-for-intrusion-detection-system-6854645f9211>. [Accessed 20 Decemebr 2022].
- [9] tech-faq, "www.tech-faq.com," tech-faq, 10 dec 2020. [Online]. Available: <https://www.tech-faq.com/network-attacks.html>. [Accessed 2022].
- [10] M. Alsallal, "securityintelligence.com," securityintelligence, 17 Jan 2017. [Online]. Available: <https://securityintelligence.com/applying-machine-learning-to-improve-your-intrusion-detection-system/>. [Accessed 20 dec 2022].
- [11] S. Cooper, "www.comparitech.com," www.comparitech, 4 November 2022. [Online]. Available: <https://www.comparitech.com/net-admin/network-intrusion-detection-tools/>. [Accessed 10 dec 2022].
- [12] A. Yadav, "resources.infosecinstitute.com," INFOSEC, 4 August 2020. [Online]. Available: <https://resources.infosecinstitute.com>. [Accessed 10 january 2022].

- [13] V. Kanade, "www.spiceworks.com," spiceworks, 24 march 2021. [Online]. Available: <https://www.spiceworks.com/>. [Accessed 15 decemeber 2022].
- [14] H. Liu and B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," *Applied Sciences*, vol. 9, pp. 43-96, 2019.
- [15] A. Khraisat, I. Gondal , P. Vamplew and J. Kamruzza, "Survey of intrusion detection systems: techniques, datasets and challenges," 2019.
- [16] W. Brown and . L. Stallings , Computer Security, Principles and Practice, 4 th ed. Pearson, 2018, pp. 1-192.
- [17] P. Bernardi and M. Kieran , Intrusion detection systems for critical infrastructure, ENGLAND: Press, 2014, p. 150–170.
- [18] E. S. a. M. Endorf C., Intrusion Detection, MvGraw Hill, 2004.
- [19] Munish S. and Anuradha., "Network Intrusion Detection System for Denial of Service Attack Based on Misuse Detection," *IJCEM International Journal of Computational Engineering & Management*, vol. Vol. 12, April 2011.
- [20] R. Suman and S. Vikram , "SNORT: An Open Source Network Security Tool for Intrusion Detection in Campus Network Environment," *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, vol. 2, 2010.
- [21] . J. Beale, . J. C. Faster and J. Posluns , "SNORT 2.0 Intrusion Detection," *Syngress Publishing*, 2003.
- [22] S. Eltanbouly and M. Bashendy, "Machine Learning Techniques for Network Anomaly," *IEEE International Conference on Informatics, IoT, , 2020*.
- [23] M. S. a. Anuradha., "Network Intrusion Detection System for Denial of Service Attack Based on Misuse Detection," *IJCEM International Journal of Computational Engineering & Management*,, vol. 12, 2011.
- [24] W. N. a. J. S. Barbar D., "ADAM: Dectecting Intrusions by Data Mining," *Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security*, 2001.
- [25] Hari and H. Tanmoy , "Statistical Techniques in Anomaly Intrusion Detection System," *International Journal of Advances in Engineering & Technology*, 2012.
- [26] B. D. a. K. J. Bhuyan M., "Anomaly Based Intrusion Detection Using Incremental Approach A Survey," *University of Colorado, Colorado Springs*, march 2012.
- [27] H. Kal and C. Min , "Hybrid Intrusion Detection with Weighted Signature Generation Over Anomalous Internet Episodes," *IEEE Transactions on Dependable and Secure Computing*, p. 1–55, 2007.
- [28] M. E, "Network Security a Beginners Guide," *Mc Graw Hill Professional*, 2002.

- [29] a. H. C. Brenton C., *Mastering Network Security*, 2nd Edition ed., Sybex Incorporated , 2002.
- [30] S.Defense,"SPADE,"[Online]. Available: <http://www.silicondefense.com/software/spice>. [Accessed january 2022].
- [31] Endorf C *Intrusion Detection*, MvGraw Hill, 2004.
- [32] T. M, *Information Assurance Tools Report – Intrusion Detection*, 2009.
- [33] Z. M, "Optimal Feature Selection For Network Intrusion Detection: A Data Mining Approach," *Unpublished Masters Thesis*, Addis Ababa University.
- [34] A. M. Z. a. M. J. A. A. Qais Saif Qassim, "Qais Saif Qassim, Abdullah Mohd Zin, and Mohd Juzaidin Ab Aziz, "Anomalies Classification Approach for Network-based Intrusion Detection System," *International Journal of Network Security*, p. Vol 18, 2016.
- [35] K. Dewangan, "An ensemble model for classification of attacks with feature selection based on kdd99 and nsl-kdd data set," *International Journal of Computer Applications*, 99(15):8–13, 2014..
- [36] M. S. Vivek kshirsagar, "Rule Based Classifier Models for Intrusion Detection System," *International Journal of Computer Science and Information Technologies*, vol. 7, 2016.
- [37] A. L. S. Saabith, "Comparative Analysis of Various Tools for Data Mining and Big Data Mining," *International Journal of Engineering Research* , vol. 7, Dec 2018.
- [38] Z. Karimi, M. Mansour , R. Kashani and . A. Haroun, "Feature ranking in intrusion detection dataset using combination of ltering methods," *Int. J. Comput. Appl*, Vols. vol. 78, no. 4, pp. 21-27, Sep. 2013.
- [39] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," *Int. J. Eng. Technol*, vol. 7, pp. 479-482, 2018.
- [40] T. A, M. Alhaj, A. Siraj , H. Zainal, T. Elshous and F. Elhaj, "Feature selection using information gain for improved structural-based alert correlation," *PLoS ONE*, Vols. 11, no. 11, p. Art. no. e0166017., 2016.
- [41] Z. Karimi, M. Mansour , K. Riahi and A. Haroun, *Int. J. Comput. Appl*, Vols. 78, no. 4, pp. 21-27, , Sep. 2013.
- [42] P. Bereziński, . B. Jasiul and . M. Szyrka, "An entropy-based network anomaly detection method," *Entropy*, Vols. 17, no. 4, , pp. 2367-2408, 2015.
- [43] A. Mouhammd , . A.-N. Ghazi, H. Ahmad and M. Mohammad, "Detecting distributed denial of service attacks using data mining techniques," *International Journal of Advanced Computer Science and Applications.*, vol. 7, pp. 436-445, 2016.
- [44] L. Sukhan and S. Shunichi , "BAYESNET: Bayesian classification network based on

- biased random competition using Gaussian kernels," *IEEE International Conference*, pp. 1354-1359, Mar. 1993.
- [45] C. Eugene , "Bayesian networks without tears.," *AI Magazine*, vol. 12, pp. 50-50, Dec 1991.
- [46] N.Khanna,"medium.com,"18Aug,2021.[Online].Available:
<https://medium.com/@nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell-24c50d20658e>. [Accessed 15 dec 2022].
- [47] "www.saedsayad.com,"www.saedsayad.com,[Online].Available:
<https://www.saedsayad.com/oner.htm>. [Accessed 10 Dec 2022].
- [48] "https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/,"
[Online].Available:<https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/>. [Accessed january 2022].
- [49] CFI Team , "corporatefinanceinstitute.com," CFI, 13 December 2022. [Online]. Available: <https://corporatefinanceinstitute.com>. [Accessed 3 jan 2023].
- [50] "https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/,"
[Online].Available:<https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/>.
- [51] J. B. PhD, "machine learning mastery," 27 April 2021. [Online]. Available: <https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/>. [Accessed 20 December 2022].
- [52] C. Team, "corporatefinanceinstitute.com," 13 Decemebr 2022. [Online]. Available: <https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/>. [Accessed 20 january 2022].
- [53] C. Team, corporatefinanceinstitute.com, 13 decemebr 2022. [Online]. Available: <https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/>. [Accessed 15 jan 2022].
- [54] J. B. PhD, "machinelearningmastery.com," machinelearningmastery, 27 April 2021. [Online]. Available: <https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/>. [Accessed 4 jan 2022].
- [55] CFI Team, "corporatefinanceinstitute.com," corporatefinanceinstitute, 13 December 2022.[Online].Available:<https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/>. [Accessed 5 january 2022].
- [56] "corporatefinanceinstitute.com,"orporatefinanceinstitute,[Online].Available:
<https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/>.
- [57] "corporatefinanceinstitute.com,"corporatefinanceinstitute,5/10/2020.[Online]. Available:<https://corporatefinanceinstitute.com/resources/data-science/ensemble->

methods/. [Accessed 8 11 2022].

- [58] [Online]. Available: <https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/>.
- [59] [Online]. Available: <https://corporatefinanceinstitute.com/resources/datascience/ensemble-methods/>.
- [60] J. B. PhD, "machinelearningmastery.com," machinelearningmastery, 3 Dec 2020. [Online]. Available: <https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/>. [Accessed 10 dec 2022].
- [61] [Online]. Available: <https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/> .
- [62] X. Geerinck, "medium.com," 3 jan 2020. [Online]. Available: <https://medium.com/>. [Accessed 10 dec 2022].
- [63] D. Sharma, "Accuracy Performance Measures in Data Science: Confusion Matrix," *Towards Data Science*, Sep 17, 2019.
- [64] A. Nisioti, "From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods," *IEEE Communications Surveys & Tutorials*, p. 3369–3388, 2018 .
- [65] C. M, "Implementing Intrusion Detection Systems", a Hands-On Guide for Securing the Network, USA, 2002.
- [66] A. Qais Saif Qassim, " Anomalies Classification Approach for Network".
- [67] [Online]. Available: <https://corporatefinanceinstitute.com/resources/datascience/ensemble-methods/>.
- [68] [Online]. Available: <https://corporatefinanceinstitute.com/resources/datascience/ensemble-methods/>.
- [69] "corporatefinanceinstitute.com," [Online]. Available: <https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/>.
- [70] O. G. Yalçın, "towardsdatascience.com," towardsdatascience.com, Nov 2015. [Online]. Available: [https://towardsdatascience.com.](https://towardsdatascience.com/) [Accessed 5 dec 2022].
- [71] L. Taranenko, "MobiDev.biz," MobiDev, August 2020. [Online]. Available: <https://mobidev.biz/blog/>.
- [72] F. SCHULZE, "Predicting website exits with machine learning," *Master in Computer Science*, 2018.
- [73] v. H. and H. Teerat Pitakrat, "A Framework for System Event Classification and

Prediction by Means of Machine Learning," *Framework*, pp. 174-176.

- [74] T. Hastie, R. Tibshirani and J. Friedman, "The elements of statistical learning: Data mining, inference, and prediction," *in Printing. New York, NY, USA:Springer*, 2017.
- [75] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov 2015.

APPENDICES

Appendix A: proposed model result summary

=== Summary ===

Correctly Classified Instances	69252	99.912 %
Incorrectly Classified Instances	61	0.088 %
Kappa statistic	0.9982	
Mean absolute error	0.0006	
Root mean squared error	0.0143	
Relative absolute error	0.4623 %	
Root relative squared error	5.7926 %	
Total Number of Instances	69313	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.999	0.001	1.000	0.999	0.999	0.998	1.000	1.000	BENIGN
	0.995	0.000	0.997	0.995	0.996	0.996	1.000	0.995	DoS slowloris
	0.993	0.000	1.000	0.993	0.996	0.996	0.999	0.997	DoS Slowhttptest
	0.999	0.001	0.998	0.999	0.999	0.998	1.000	1.000	DoS Hulk
	0.993	0.000	0.998	0.993	0.995	0.995	0.999	0.998	DoS GoldenEye
	0.667	0.000	1.000	0.667	0.800	0.816	1.000	1.000	Heartbleed
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	PortScan
	0.000	0.000	1	0.000	1	1	1.000	1.000	Web Attack
Weighted Avg.	0.999	0.001	0.999	0.999	0.999	1	1.000	1.000	

=== Confusion Matrix ===

	a	b	c	d	e	f	g	h	<-- classified as
43974	0	0	0	32	0	0	0	0	a = BENIGN
3	574	0	0	0	0	0	0	0	b = DoS slowloris
2	2	568	0	0	0	0	0	0	c = DoS Slowhttptest
10	0	0	23046	2	0	0	0	0	d = DoS Hulk
4	0	0	4	1061	0	0	0	0	e = DoS GoldenEye
1	0	0	0	0	0	2	0	0	f = Heartbleed
0	0	0	0	0	0	0	27	0	g = PortScan
1	0	0	0	0	0	0	0	0	h = Web Attack ♦ Brute Force

Appendix B: selected attribute for the model.

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter

Choose **None** Apply Stop

Current relation
 Relation: Wednesday-workingHours.pcap_ISCX-weka.filters.unsupervised.instance.Randomiz... Attributes: 77
 Instances: 346567 Sum of weights: 346567

Selected attribute
 Name: Label Type: Nominal
 Missing: 0 (0%) Distinct: 8 Unique: 0 (0%)

Attributes

All None Invert Pattern

No.	Name
60	<input type="checkbox"/> Bwd Avg bulk Rate
61	<input type="checkbox"/> Subflow Fwd Packets
62	<input type="checkbox"/> Subflow Fwd Bytes
63	<input type="checkbox"/> Subflow Bwd Packets
64	<input type="checkbox"/> Subflow Bwd Bytes
65	<input type="checkbox"/> Init_Win_bytes_forward
66	<input type="checkbox"/> Init_Win_bytes_backward
67	<input type="checkbox"/> act_data_pkt_fwd
68	<input type="checkbox"/> min_seg_size_forward
69	<input type="checkbox"/> Active Mean
70	<input type="checkbox"/> Active Std
71	<input type="checkbox"/> Active Max
72	<input type="checkbox"/> Active Min
73	<input type="checkbox"/> Idle Mean
74	<input type="checkbox"/> Idle Std
75	<input type="checkbox"/> Idle Max
76	<input type="checkbox"/> Idle Min
77	<input checked="" type="checkbox"/> Label

Remove

No.	Label	Count	Weight
1	BENIGN	220103	220103
2	DoS slowloris	2898	2898
3	DoS Slowhttptest	2749	2749
4	DoS Hulk	115536	115536
5	DoS GoldenEye	5146	5146
6	Heartbleed	5	5
7	PortScan	124	124
8	Web Attack i;/: Brute Force	6	6

Class: Label (Nom) Visualize All

Status
OK Log x 0