Performance Evaluation of Machine Learning Algorithms for Detection of SYN Flood Attack: The case of ethio telecom

PREPARED BY: WASSIHUN BEYENE

ADVISER: YALEMZEWD NEGASH (PHD)

A Thesis submitted to
School of Electrical and Computer Engineering
Addis Ababa Institute of Technology

In Partial Fulfillment of the Requirements for the Degree of Master of Telecommunication Engineering (TIS)

ADDIS ABABA UNIVERSITY
Addis Ababa, Ethiopia
February 28, 2020

Declaration of Originality

I, the undersigned, declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research. I trully acknowledged and referred every materials which used in this thesis work.

WASSIHUN BEYENE

Name

Signature

ADDIS ABABA UNIVERSITY

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Thesis on

Performance Evaluation of Machine Learning Algorithms for Detection of SYN Flood Attack: The case of ethio telecom

By: **WASSIHUN BEYENE**

Signed by :

Adviser	Yalemzewd Negash (PhD)	Signature	_____	Date	_____
Evaluator	_____	Signature	_____	Date	_____
Evaluator	_____	Signature	_____	Date	_____

ABSTRACT

Telecom service providers operate and control complex network infrastructure used for data transmission. However, security issues have been among the most serious problems for service providers in general and ethio telecom in particularly. One of the main security problems that become the hardest and most serious threat is called Distributed Denial of Service (DDoS) attacks specifically Synchronize (SYN) flood attack. Nowadays, different researchers to detect and prevent SYN flood attack recommended several statistical detection methods. However, due to the dynamic behavior of attack has been challenged to detect using existing detection approaches.

This research focused on the performance evaluation classification machine learning (ML) algorithms for detection SYN flood attack. The classification models trained and tested with packet captured (PCAP) dataset has been used and gathered from ethio telecom network by generated and captured using Hping3 and Wireshark tools respectively. This dataset has been further preprocessed and evaluated using four classification ML algorithms and three training approaches. The implementation has been performed using WEKA (Waikato Environment for Knowledge Analysis) data mining tool.

The experimental results show J48 algorithm performs with 98.57% of accuracy and AdaBoost, Naïve Bayes and ANN algorithms with 98.52%, 95.31% and 94.85% of accuracy respectively. The first reason was that the J48 algorithm is more efficient than the other algorithms; it has been used as a pruning technique in order to reduce the complexity of the final classifier and to prevent over fitting the data. The second reason was the ability to learn mechanisms. Therefore, based on the performance evaluation result model with J48 algorithm has been recommended for SYN attack detection.

KEYWORDS

Adaptive Booster, ANN, Distributed denial of service attack, Denial of service attack, Hping3, J48, Naive Bayes, SYN flood attack, WEKA and Wireshark

ACKNOWLEDGMENTS

First and foremost, I would like to thank God for giving me the courage and strength to finish this study. I am grateful to my company ethio telecom, and Addis Ababa University's School of Electrical and Computer Engineering, for making it possible for me to study here.

Next, my best gratitude goes to my advisor Yalemzewd Negash (PhD) for his, unreserved comments, encouragement, guidance and motivation he gave me to accomplish this thesis. Besides my advisor, I would also like to thank my thesis examiners: Ephrem Teshale (PhD), and Murad Ridwan (PhD), for their encouragement, insightful comments and suggestions.

I sincerely thank my whole family; without their support and encouragement, I would have not been here today. They have always been with me supporting, helping and cherishing in my journey to be a better man.

In addition, I would like to take this opportunity to thank Mehadi Ali who consulted me on different issues regarding the thesis.

In addition, I also would like to thank the contributions of various experts in the field network security and data collection support staffs in ethio telecom.

Finally, I would like to thank, H/Meskel G/Tsadik, Surafel Fikire, Derebe Tekeste, and Wubishet Abebe all my colleagues and friends. I am grateful for being with me during my difficult moments and giving me continuous support and encouragement.

CONTENTS

1 INTRODUCTION 1

1.1 Background . . . 1
1.2 Statement of the problem . . . 3
1.3 Objectives of the Research . . . 4
1.4 Scope of the Research . . . 5
1.5 Contributions of the Research . . . 5
1.6 Related Works . . . 5
1.7 Methodology of Research . . . 7
1.8 Research Organization . . . 8

2 LITERATURE REVIEW 9

2.1 Intrusion Detection System . . . 9
2.1.1 Signature-Based IDS . . . 9
2.1.2 Anomaly Based IDS . . . 10
2.1.3 Machine Learning-Based IDS . . . 10
2.2 Intrusion Prevention System . . . 11
2.3 Network Security Attacks . . . 11
2.4 Denial of Service Attacks . . . 12
2.5 Distributed Denial of Service Attacks . . . 13
2.5.1 SYN Flood Attack . . . 14
2.5.2 ICMP Flood Attack . . . 15
2.5.3 UDP Flood Attack . . . 16

3 MACHINE LEARNING 17

3.1 Data Mining . . . 17
3.2 Machine Learning Algorithm . . . 17
3.2.1 Supervised Learning Technique . . . 18
3.2.2 Unsupervised ML Technique . . . 22

3.2.3 Semi-supervised Learning Technique . . . 23
3.2.4 Reinforcement ML Technique . . . 24
3.3 Tools used for the Experiments . . . 24
3.3.1 Packet Generating Tool . . . 24
3.3.2 Packet Capturing Tool . . . 25
3.3.3 Packet Analyzing Tool . . . 25
3.3.4 Kali Linux Operating System . . . 25

4 EXPERIMENTAL ANALYSIS 26
4.1 System Model for Detection of SYN flood Attack . . . 26
4.2 Experimental Setup . . . 27
4.3 Data Source . . . 28
4.3.1 SYN Flood Packet Generation . . . 28
4.3.2 Normal Packet Capturing . . . 30
4.3.3 Data Description . . . 30
4.4 Data Preprocessing . . . 31
4.4.1 Data Cleaning . . . 31
4.4.2 Data Transforming . . . 31
4.4.3 Normalization . . . 32
4.5 Feature Selection . . . 33
4.5.1 Manual Feature Selection . . . 33
4.5.2 Feature Worthiness Evaluation . . . 34
4.6 Data Formatting . . . 38
4.7 Model Evaluation Metrics . . . 38
4.8 Training and Testing Datasets . . . 41
4.9 Model Experimentation . . . 42
4.9.1 Model with AdaBoost Experimentation . . . 42
4.9.2 Model with Naïve Bayes Experimentation . . . 43
4.9.3 Model with Artificial Neural Network Experimentation . . . 45
4.9.4 Model with J48 Decision Tree Experimentation . . . 46

5 RESULTS AND DISCUSSION 48
5.1 Experimental Results . . . 48
5.2 Discussion . . . 52

6 CONCLUSION AND FUTURE WORK 58
6.1 Conclusion 58
6.2 Recommendation for future works 60

BIBLIOGRAPHY 61

LIST OF FIGURES

Figure 1.7.1	Research Methodology	8
Figure 2.4.1	Structure of a Denial of Service Attack [15]	13
Figure 2.5.1	Structure of a Distributed Denial of Service Attack [27].	13
Figure 2.5.2	Structure of Normal TCP 3- Way Handshake [27]	14
Figure 2.5.3	SYN Flood Attack Scenario [27]	15
Figure 2.5.4	ICMP-Flood Attack Scenario [30]	16
Figure 3.2.1	Decision Tree Structure [38]	20
Figure 3.2.2	MPL Architecture of ANN [40]	22
Figure 4.1.1	System Model for SYN Flood Attack Detection	27
Figure 4.3.1	Lab Experiment for SYN Flood Attack [27]	29
Figure 4.3.2	SYN Flood Packets Generation Kali Linux Command	29
Figure 4.5.1	Evaluation of Feature Subset using Information Gain FS	34
Figure 4.5.2	Evaluation of Feature Subset using Gain Ratio FS	35
Figure 4.5.3	Evaluation of Feature Subset using CFS	36
Figure 5.1.1	Comparison of Accuracy and FPR	49
Figure 5.1.2	Comparison of ROC and RMSE models performance	50
Figure 5.1.3	Time Taken to Build Models	50
Figure 5.1.4	Time Taken to Evaluate Models	51
Figure 5.1.5	Comparison with three Performance Metrics of Models	51
Figure 5.1.6	Comparison of ROC Curve	52

LIST OF TABLES

Table 4.4.1	Data Transforming Nominal to Numeric	32
Table 4.5.1	Ranking of Selected Feature Subset	36
Table 4.7.1	Confusion Matrix Value (2X2)	39
Table 4.9.1	Confusion Matrix of AdaBoost Algorithm	43
Table 4.9.2	Classification Accuracy of AdaBoost Algorithm	43
Table 4.9.3	Confusion Matrix of Naïve Bayes Algorithm	44
Table 4.9.4	Classification Accuracy of Naïve Bayes Algorithm	44
Table 4.9.5	Confusion Matrix of ANN Algorithm	45
Table 4.9.6	Classification Accuracy of ANN Algorithm	45
Table 4.9.7	Confusion Matrix of J48 Algorithm	46
Table 4.9.8	Classification Accuracy of J48 Algorithm	47
Table 5.1.1	Comparison of Selected Algorithms Performance	49

ACRONYMS

AdaBoost Adaptive Booster

ACK Acknowledgment

ANN Artificial Neural Network

ARFF Attribute Relation File Format

CFS Correlation based Feature Selection

CSV Comma Separated Values

DDoS Distributed Denial of Service

DoS Denial of Service

FNR False Negative Rate

FPR False Positive Rate

TPR True Positive Rate

GR Gain Ratio

ICMP Internet Control Message Protocol

IDS Intrusion Detection System

IG Information Gain

IPS Intrusion Prevention System

ML Machine Learning

MLP Multilayer Perceptron

PCAP Packet Capture

RMSE Root Mean Square Error

ROC Receiver Operating Characteristic

SYN Synchronize

TCP Transmission Control Protocol

UDP User Datagram Protocol

WEKA Waikato Environment for Knowledge Analysis

1

INTRODUCTION

This chapter has provides the background of this study and describes statement of the problem, objectives, scope, limitation and contribution of the research. Moreover, it discusses the methodology used and briefly describes reviewed literature which is related to the study. It also outlines the whole contents of the thesis in last section of this chapter.

1.1 BACKGROUND

Telecommunications sector plays a crucial role in the modern lifestyle and has the highest contribution over economic development [1]. Telecom service providers operate and manage complex network infrastructures used to data transmission, they communicate and store large amounts of sensitive data. However, security issues have been one of the serious problems for service providers as attackers also changed dynamically [2]. Distributed Denial of Service (DDoS) attack is a serious threat to telecom service providers around the world with an active online business.

Because of the volume sizes of DDoS continued to grow up at a faster rate than over the past year, and it is also increased the frequency and complexity [3]. Denial of Service (DoS) attack can be launched from either a single source or multiple sources, multiple-source DoS are called DDoS attacks [4]. DDoS attacks are performed from multiple agents towards a single victim. Essentially, all attacking mediators generate multiple packets towards the victim to overwhelm it with requests, thereby overloading the resources of the victim. Such attacks suffer a huge

financial loss to respective companies. The impact of a DDoS attack should not be underestimated.

The DDoS attack can degrade the network capacity, reduced performance of the network, improved traffic exchange costs, interrupt service accessibility and even bring down the Internet access [4]. Currently, the DDoS attack open-source software tools have grown into automated and sophisticated, by this means allowing attackers to implement all or a few of the steps are automatically with minimal human effort.

There are different forms of DDoS attacks which target unrelated services, some of them are Synchronize (SYN) flood, Internet Control Message Protocol (ICMP) flood and User Datagram Protocol (UDP) flood attacks. Those attacks are sending many malicious packets to overwhelm the victim's network resources such as central processing unit (CPU), memory and node [5].

The SYN is one of the most common types, dangerous and easiest to implement a DDoS attack [6]. The aims to make a server inaccessible to legitimate traffic by consuming the entire available resources and caused due to the drawback that comes with the *"three-way handshake"* of Transmission Control Protocol (TCP) connection sequence. Which means that the SYN flood attack, the the requester continually sends SYN packets several SYN request to the server but later does not respond to the server with Acknowledgment (ACK) approval after receiving SYN-ACK request. Also, the attack can be done by sending the request from a false source IP address.

The Intrusion Detection System (IDS) is process of examining the events happening in telecom networks and detecting the actions that make an attempt to compromise the confidentiality, availability, and integrity of a resource. This detection approach is very effective to detect known attacks. However, the DDoS attack behavior is not statics and their character will change frequently. So, this approach could be ineffective to detect such type of attacks [7].

Data mining-based approaches are another paradigm for building IDS. The main benefit of these methods is that they leverage the generalization ability of data mining methods and in order to detect new and unknown attacks. A data mining-

based IDS uses machine learning and data mining algorithms on a large set of system audit data to build detection models. These models have been established to be very effective [8].

This study focuses on the use of Machine Learning (ML) algorithm for detecting SYN flood attack. The dataset gathered from ethio telecom network by generated and captured packet dataset using Hping3 and Wireshark tools respectively. The proposed method has been tested using four classification machine learning algorithms, such as Naive Bayes, Adaptive Booster (AdaBoost), J48 and Artificial Neural Network (ANN). The proposed model implemented using the Waikato Environment for Knowledge Analysis Waikato Environment for Knowledge Analysis (WEKA) data mining tool.

1.2 STATEMENT OF THE PROBLEM

Among different types of network attacks, DDoS attack is one of the most serious and frequent threats for telecom service providers [9]. The main objective is to interrupt the server's availability and suspend the user's access to a telecom network. DDoS attacks are performed from multiple agents towards a single victim [4]. SYN flood, UDP flood, and ICMP flood are typical flooding attack of DDoS attacks [10]. The SYN flood DDoS attack can cause due drawback of the TCP "3-way handshake" activity in order to begin a connection between the source and the target victim machine, thus making the system interrupt to legitimate users [11].

Nowadays, different service provider including ethio telecom uses different types of firewall and signature-based IDS for the purpose of detecting network intrusion. These types of detection approach effective against known attacks but mostly ineffective against novel and statics behaviour attacks [12]. Moreover, this method requires prior knowledge of attack signatures, and signatures must be hand-coded. According to this, a human dependent process it requires several man-hours to test, create and deploy those signatures and again create a new signature for unknown attacks, which is expensive as well as an error-prone job.

However, the DDoS and SYN flood attack behaviour is statics and their behaviour will change frequently and the attacker can pass simply to networks [5]. The impact of such types of attacks can interrupt service accessibility, degrade the network capacity, reduced performance and even bring down Internet access [13].

This research introduces a solution that contributes in overcoming detection of SYN flood attack using data mining technique specifically classification ML approach. Classification based data mining technique is one of the methods with a capability to learn from the situation and improve its performance through learning which is reached by an iterative process. This detection process automatically acquire knowledge from network traffic packet captured dataset, gathered from ethio telecom, by generating and capturing the dataset. Finally, this study explores and finds to get answers for the following research questions.

Q1. Which machine learning algorithm can be more efficient for the purpose of detecting SYN flood attack?

Q2. Which traffic dataset features are relevant for the purpose of detecting SYN flood attack?

Q3. Which model training approach is better in order to detecting SYN flood attack?

1.3 OBJECTIVES OF THE RESEARCH

General Objective

The general objective of this study is to compare the performance of four classification ML algorithms (AdaBoost, Naives Bayes, ANN and J48) for detection of SYN flooding attack with a better performance and minimum false alarm rate.

Specific Objectives

In order to achieve the general objective, the specific objectives of this study are:

- To identify algorithms and tools for detection of SYN flood attack
- To generate and capture network traffic dataset from ethio telecom network

- To select relevant features to visualize patterns of dataset for its processing.
- To test and evaluate the performance of selected ML algorithms
- To give recommendations based on the findings.

1.4 SCOPE OF THE RESEARCH

The scope of this thesis focused on performance evaluation of ML algorithm for detection of SYN flood attack, that will use generating and capturing SYN flood packet datasets using Hping3 and Wireshark open-source tools respectively. In addition to this show the efficiency of SYN flood attack detection, recommend better detection model and algorithm.

1.5 CONTRIBUTIONS OF THE RESEARCH

At the end of this research, the result will have both practical and theoretical contribution, as per our knowledge, there is no specific study focus on SYN flood attack detection using ML based. This research helps ethio telecom to detect SYN flood attack from normal network traffic in addition to the existing security mechanism. It also provides a suitable ML algorithm and model that can detect SYN flood attack with best detection performance and minimum false alarm rate. In addition, this study will motivate future researchers to work on the other intrusion attacks by packet generate and capture network traffic dataset and by using ML techniques.

1.6 RELATED WORKS

Different researchers have been studied in network IDS and intrusion prevention system field using different tools and ML approach. In this section, the researchers

review some of the non-parametric detection approaches found in recent literature that is related to SYN flood attack are reviewed in the next five paragraphs.

Authors of [14], proposed an approach to defense against the TCP SYN flooding attacks. A swarm intelligence-based ant colony optimization is used in proposed work which tries to reduce the share of attack half-open requests from the buffer of TCP. The primary objective of the proposed work is to defend against the TCP SYN flooding attack on wired networks. The proposed scheme is improving the results by decreasing the TCP connection loss and the share of attack requests from the buffer space.

Hussain *et al.* [15], proposed a three-way counter algorithm has been presented to mitigate SYN flooding attack. This algorithm is based on windows advanced firewall rules. This work is the enhancement of firewall capabilities to identify SYN flooding attack. The proposed work evaluates in DDoS environment, the result shows that 97.5% identification, detection and mitigation of SYN flood attack in DDoS environment. The result shows that through this technique become able to detect and mitigate SYN flooding attack successfully. They concluded that enhancement of the firewall capabilities to identify SYN flooding attack.

Bogdanski *et al.* [16], proposed a novel method consisting of five modules which can be used for mitigation and protection against the considered SYN flood attack, as well as other similar flooding-based attacks. The researcher uses a simulation conducted against the production web server using a virtual machine with the following Linux script that spoofs a random IP address and source port. The paper gives some simulation results which show the effects of this attack.

Authors of [17], the primary purpose of this manuscript is to characterize the defense mechanisms and compare the technical details involved in defense mechanisms of attacks due to the SYN flood. This will help the researchers to propose improved and more efficient defense mechanisms. Subsequently, the study compares performance of the Victim Side SYN flood (VSSF) attack protection system implemented using a general-purpose processor, with the VSSF attack protection system implemented using Nios core processor. They concluded that the Field Pro-

grammable Gate Array (FPGA) based SYN food attack protection system supports faster data transfer than the software-based system.

Villing, J [18], introduced and compared several SYN flood mitigation approach as well as discussing challenges of their detection. Some techniques like increasing the backlog or reducing the timeout focus on delaying the exhaustion of the backlog while others try to prevent it from getting full, like SYN agent or SYN cookies. Additionally, recycling and random drop take no precautions to prevent the backlog from being filled, instead, they simply drop a connection from the backlog if it is fully based on the respective heuristic algorithm. Finally, they concluded SYN Cookies are the best choice if the mitigation technique must be ready for immediate use because this method is included in Linux.

1.7 METHODOLOGY OF RESEARCH

This section describes the methodology of this research. In order to accomplish the objectives of this study and to answers the research questions. This thesis follows the experimental research methodology followed by the researchers. Figure 1.7.1 shows the research methodology to be followed in this thesis.

First, conducting of a comprehensive literature, which is necessary to acquire a deeper understanding of the research area and its problem domains and state of art designing SYN flood attack detection. Several books, journals, articles, and papers from the Internet were consulted to assess the importance and applications of SYN flood attack detection with the machine learning approach.

Second, select the appropriate and well-known classification algorithms and data mining, network traffic generates and capture open source tools. After that by generating and capturing network traffic collected SYN flood and normal Packet Capture (PCAP) dataset, from ethio telecom network. Then, in order to improves the detection performance and minimize the time taken to build and evaluate models, the data preprocessing task have been done.

Third, after the data preprocess performed, the next major role was selection of a relevant feature subset by removing redundant and irrelevant features. Which is helps to improve the accuracy of a classification algorithm and minimize false alarm rate and to reduce the time taken to build and evaluate models. Finally, make it ready for training and testing the models experiments.

Fourth, after data processing and feature selection, trained selected classification ML algorithms by training dataset. Then, we will evaluate the detection models using some metrics like accuracy, false positive rate, building time and so on. Finally, discussed the results and give recommendation based on findings.

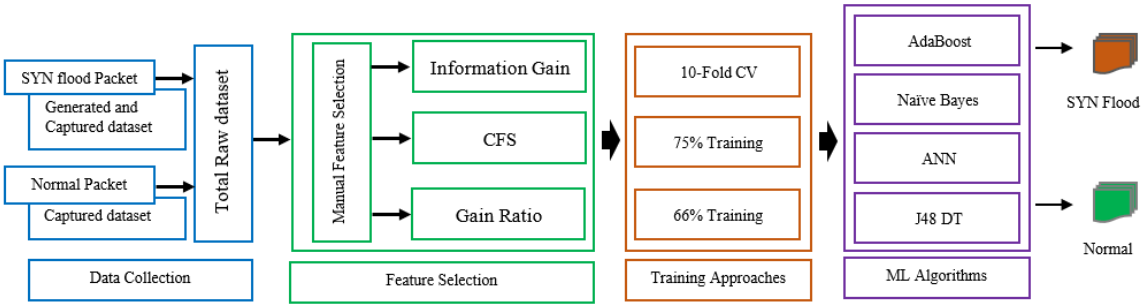


Figure 1.7.1: Research Methodology

1.8 RESEARCH ORGANIZATION

This research work is structured into six chapters and the rest of organized as follows: Chapter 2 focuses on the literature review to gives a fundamental understanding of the intrusion detection techniques employed in IDS, provide some information about DoS and DDoS intrusion attack types which is SYN flood. Chapter 3: deals with the concepts of ML techniques used in the surveyed which are used in this study. Chapter 4: deals about SYN flood packet generation and capturing process, dataset preprocessed, feature selection and models experimentation will be discuss. Chapter 5: focuses on experimental results and discussing the outcomes of experiment. Finally, Chapter 6: Provides the concludes of study and suggests recommendations for further work.

2

LITERATURE REVIEW

This chapter, focuses on the state-of-the-art and literature review of related work for this research. It gives an overview of background about network IDS followed by a description of two major IDS approaches, anomaly-based and signature-based. It also describe ML based IDS presented. Finally, provides some information about denial of service and distributed denial of service and SYN flood attack.

2.1 INTRUSION DETECTION SYSTEM

Intrusion Detection System (IDS) is a software or hardware implement for detected anomalous behaviors in a network [7]. An abnormal pattern covers many definitions. In this study, it is likely described as malicious, unwanted and misuse activity happening within a network. IDS is a standard component of a security infrastructure that allows network administrators to detect policy violations, check all incoming and outgoing network activity and determine suspicious patterns that indicate attacks from people trying to break or compromise the system. In general, IDS are classified into two major groupings based on detection methods: anomaly-based and signature-based [19].

2.1.1 *Signature-Based IDS*

Signature-based intrusion detection examines system activity, examining for events or sets of events that match a predefined pattern of events that describe a known attack [20]. As the patterns corresponding to known attacks are called signatures. signature-based detection is sometimes known as misuse intrusion detection. The

most common form of misuse IDS used in commercial products specifies each pattern of events corresponding to an attack as a separate signature. This type of detection is very fast and easy to configure. However, an attacker can slightly modify an attack to render it undetectable by signature-based IDS [20].

2.1.2 *Anomaly Based IDS*

In the contrast to signature-based IDS, anomaly-based IDS does not need signatures to detect intrusion. Additionally, an anomaly-based IDS can identify unknown network intrusion attacks depending on the similar behavior of other network intrusions attacks. According to [21], anomaly-based IDS can detect intrusions when the current behaviors move away statistically from the normal behavior model. An IDS that looks at network traffic and detects data that is incorrect, not valid, or generally abnormal is called anomaly-based detection. This method is useful for detecting unwanted traffic that is not specifically known. The main disadvantage of anomaly based is no clear-cut method for defining normal behavior. So, any deviations from this model are rare and potentially might be a result of intrusive activity [21].

2.1.3 *Machine Learning-Based IDS*

According to [22] data mining is the process of analyzing data from different perspectives and summarizing it into useful information. It is the process of finding correlations or patterns among lots of fields in large relational databases. Network traffic is huge, and information comes from different sources, so the dataset for IDS becomes large. Hence the analysis of data is very hard in case of a large dataset. Data mining techniques are applied to IDS because it can extract the hidden information and deals with a large dataset. Technologically, data mining techniques play a vital role in IDS. By using data mining techniques, IDS help to detect abnormal and normal patterns [22].

2.2 INTRUSION PREVENTION SYSTEM

The Intrusion Prevention System (IPS) is proactive in nature and tries to prevent an intrusion to occur in network security technologies [19]. Its goal is the process of performing intrusion detection and attempting to stop detected possible incidents. IPS is applied by some recent IDS. Instead of analyzing the traffic logs, which lies in discovering the attacks after they took place, IPS tries to warn against such attacks. While the systems of IDS try to give the alert, IPS block the traffic rated dangerous.

2.3 NETWORK SECURITY ATTACKS

Network security attacks are unauthorized actions against private, corporate or governmental IT assets in order to destroy, modify them or steal sensitive data [23]. As more enterprises invite employees to access data from mobile devices, networks become vulnerable to data theft or total destruction of the data or network. Every security control and every vulnerability can be viewed in light of one or more of these key concepts. For a security program to be considered comprehensive and complete, it must adequately address the entire confidentiality, integrity and availability (CIA) triad.

Confidentiality means that data, objects and resources are protected from unauthorized viewing and other access. Integrity data is safe from Unlawful changes to ensure that it is reliable and correct. Availability means that authorized users have access to the systems and resources they need. Network security attack can be categories into two, such as passive and active attacks [24].

Passive attack attempts to learn or make use of information from the system however, does not affect resources. Passive Attacks are eavesdropping on or monitoring of transmission. The goal of opponent is to obtain information is being transmitted. The names of some passive attacks are traffic analysis, Eavesdropping, and Monitoring [24].

Active attack attempts to alter system resources or affect their operations. The active attack involves some modification of the data stream or the creation of false statement. Types of active attacks are as following: spoofing, wormhole, modification, DoS, sinkhole, and Sybil attacks [24].

2.4 DENIAL OF SERVICE ATTACKS

A DoS attacks is very common in the world of Internet today and active attacks [25]. It is a critical attack that can disable service, or downgrade service performance by exhausting resources for providing services. When legitimate users are unable to access information systems, devices, or other network resources due to the actions of a malicious cyber threat actor. Services affected may include email, websites, on-line accounts, or other services that rely on the affected computer or network. Figure 2.4.1 shows the simple architecture of the DoS attack. It is an action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as central processing unit (CPU), memory and bandwidth. The DoS attacks can be categorized in the following three parts [26][15] .

Connection flooding: The attacker bogs down the host by establishing a large number of TCP connections at the targeted host. These fake spoof IP address block the network and make it unavailable to legitimate users.

Vulnerability Attack: The attacker by sending a few well-crafted messages to the vulnerable operating system or application running on the targeted host stops the service or make it worse to the extent that host crashes.

Bandwidth flooding: The attacker prevents legitimate packets from reaching the server by sending a flood of packets. Those are large in number so that the target's link gets blocked for others to access.

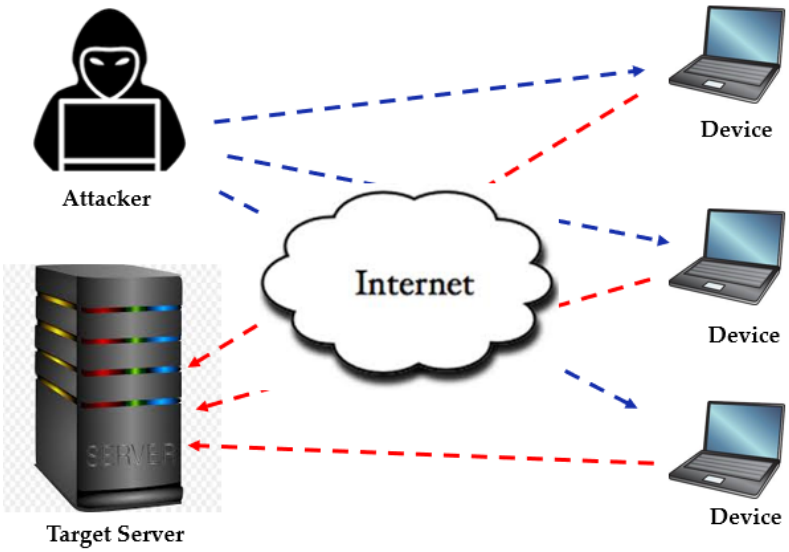


Figure 2.4.1: Structure of a Denial of Service Attack [15]

2.5 DISTRIBUTED DENIAL OF SERVICE ATTACKS

A DDoS attacks is a subclass of DoS attack. It involves multiple connected on-line devices, collectively known as a botnet, which are used to overwhelm a target website with fake traffic IP address [26]. Also, a DDoS attack is a complex version of a DoS and is much harder to detect and defend compared to a DoS attack. The DDoS attacker uses multiple compromised systems to target a single DoS attack targeted system. Figure 2.5.1 shows the simple architecture of the DDoS attack.

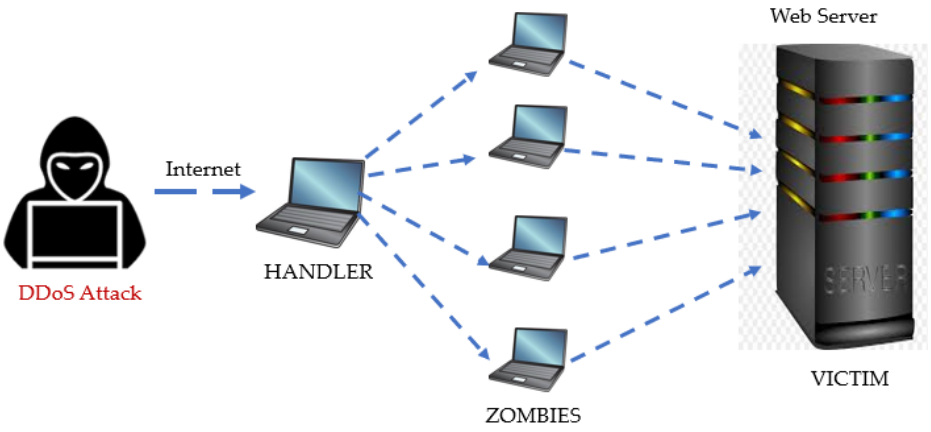


Figure 2.5.1: Structure of a Distributed Denial of Service Attack [27].

DDoS attacks are degraded or completely disrupt services to legitimate users by expending communication and/or computational resources of the target. It is also an amplified form of DoS attacks where attackers direct hundreds or even thousands of compromised hosts called zombies against a single target. According to [28] there are different types of DDoS attacks, under two broad headings: network/transport layer and application-layer attacks. There are many types of DoS or DDoS attacks. The SYN, ICMP and UDP flood attacks are discussed in the next section.

2.5.1 *SYN Flood Attack*

The SYN flood is a stateful protocol attack and requester sends numerous SYN request to the server nevertheless does not respond to the server with ACK confirmation after receiving SYN-ACK request. Also, the attack can be done by sending the request from a spoofed Internet protocol (IP) address. In both scenarios, the server keeps waiting for the acknowledgment of "three-way handshake" and its resources is used resulting in the server to deny all the new request of TCP connection [15].

Figure 2.5.2, shows a successful connection establishment, SYN and ACK are transferred between the client and server.

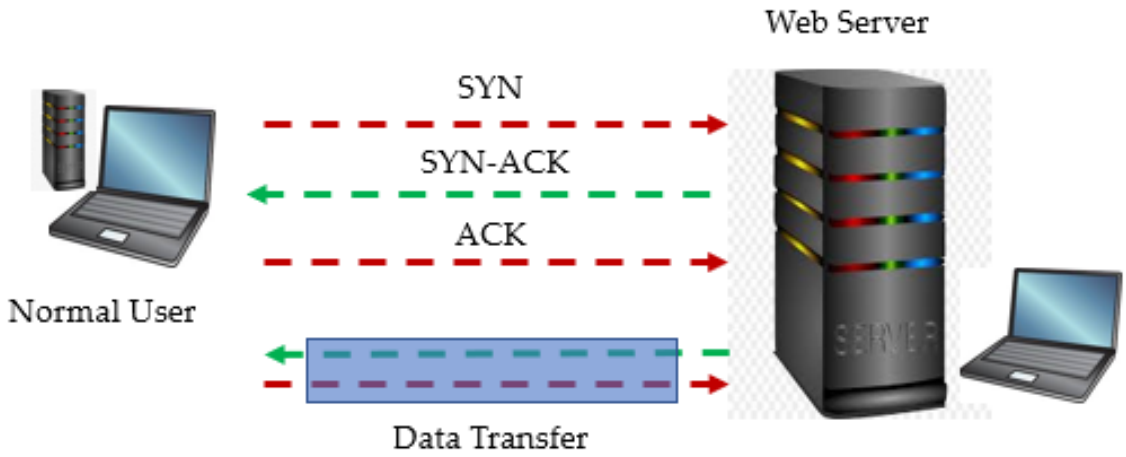


Figure 2.5.2: Structure of Normal TCP 3- Way Handshake [27]

Figure 2.5.3 shows a half-open state where an attacker sends SYN to the web server, it sends SYN-ACK back to client assuming that attacker exists however the web server never gets back the ACK from attacker and goes to the half-open state. While the request is waiting to be confirmed from the attacker, it remains in the web-server queue. Each half-open connection will remain in the memory queue until it times out, it will retransmit the SYN-ACK, doubling the timed-out value after each retransmission [18][16].

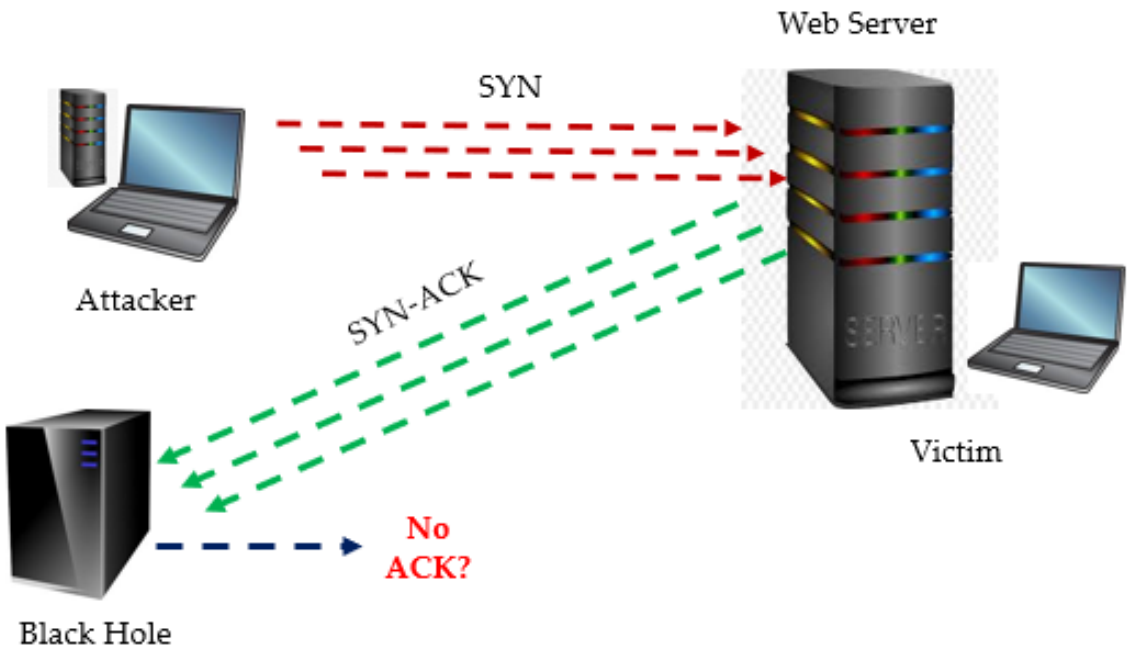


Figure 2.5.3: SYN Flood Attack Scenario [27]

Whereas, to create DoS attack, an attacker exploits the fact that after an initial SYN packet has been received, the web server will respond back with one or more SYN-ACK packets and wait for the final step in the handshake.

2.5.2 *ICMP Flood Attack*

The ICMP flood attacks, the attacker overwhelms the victim machine with ICMP echo request (ping) packets, large ICMP packets, and other ICMP types to significantly saturate and slow down the victim's network infrastructure [29]. As

shown that in Figure 2.5.4. ICMP is mostly used in networking technology, it is a connectionless protocol and mainly used for diagnostic purposes, error reporting or querying any server still now attackers are using ICMP protocol for sending payloads [30].

More specifically during a DoS ICMP flood attack, the attacker sends large volumes of ICMP request packets ping to the victim machine. These request a reply from the victim machine and this has, as a result, the saturation of bandwidth of victim's machine network connection. During an ICMP flood attack, the source IP address may be spoofed. The attacker uses IP spoofing in order to hide their true identity, and this makes the traceback of DoS attacks even more difficult [30].

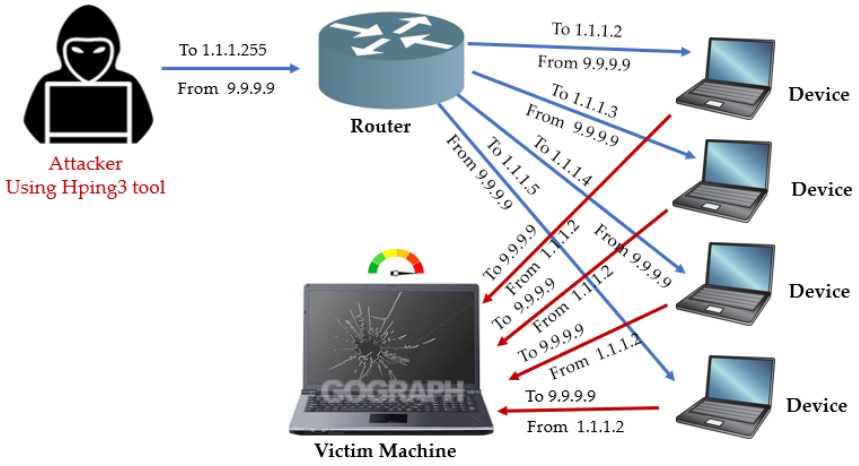


Figure 2.5.4: ICMP-Flood Attack Scenario [30]

2.5.3 *UDP Flood Attack*

A UDP flood is a type of DoS attack in which a large number of UDP packets are sent to a targeted server with the aim of overwhelming that device's ability to process and respond [31]. It is very similar to the SYN flood in that an attacker uses a botnet to send a significant amount of traffic to the target server. UDP packets become one of the most well-known and compelling methods for DoS and DDoS attack. UDP can be constructed as a very small packet so that the attacker can easily send a high volume of small-sized UDP packets which causes forwarding issues for network.

3

MACHINE LEARNING

This chapter will cover an introduction to data mining technique, machine learning types and selected three classification algorithms by introducing and definition. In addition, this chapter cover introduction to three feature selection technique in Section 3.2. And describes different types of tools used to generate, capture and data analysis for SYN flood attack detection in Section 3.3.

3.1 DATA MINING

Data mining is the process of assessing data to uncover patterns and variations as well as defining any changes or events that have taken place within the data structure. It is extraction of useful patterns from data sources (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data. It is an emerging multi-disciplinary field for ML, statistics, databases, information retrieval and visualization. It provides an extra level of intrusion detection by identifying the boundaries for usual network activity so it can distinguish common activities from uncommon. Data mining significantly improves intrusion detection using a variety of different methods [32].

3.2 MACHINE LEARNING ALGORITHM

Machine Learning (ML) is a keystone when it comes to artificial intelligence and large data investigation. It is a system capable of obtaining and integrating the knowledge automatically. The capability of the systems to learn from experience, training, analytical observation, and other means, results in a system that can

continuously self-improve and thereby exhibit efficiency and effectiveness [22]. A ML system usually starts with some knowledge and a corresponding knowledge organization so that it can interpret, analyze, and test the knowledge acquired [33]. So ML algorithm is a program with a specific way to adjusting its own parameters, given feedback on its previous performance making predictions about a dataset. There are four main classes of ML techniques, as discussed next subsection: (1) Supervised learning, (2) Semi-supervised learning, (3) Unsupervised learning, (4) Reinforcement learning

3.2.1 *Supervised Learning Technique*

Supervised ML can be defined as learning from the labeled training dataset. It learns the training dataset and produces a predictive function [34]. The predictive function will be used to decide the class label for unseen instances. It uses classification and regression techniques to develop predictive models.

Classification techniques predict discrete responses—for example, whether an email is genuine or spam. Classification models classify input data into categories. Typical applications include medical imaging and speech recognition [34].

Regression techniques predict continuous responses— for example, changes in temperature or fluctuations in power demand. Typical applications include electricity load forecasting and algorithmic trading [34].

A. AdaBoost Algorithm

AdaBoost is short for Adaptive Boosting. Fundamentally, AdaBoost was the first successful boosting algorithm developed for binary classification. Also, it is the best starting point for understanding boosting. It can be used to boost the performance of any machine learning algorithm. It is best used with weak learners. It takes a training dataset consisted of labeled instances and initially gives them equal weight, each instance in the training dataset is weighted (W) and calculated as W = 1/n where n is the number of instances in the training dataset [35].

The important advantages of AdaBoost algorithm are low generalization error, easy to implement, works with a wide range of classifiers, no parameters to adjust. Especial attention is needed to data as this algorithm is sensitive to outliers. It can be used for both classification and regression problem. AdaBoost is best used to boost the performance of decision trees on binary classification problems. It is a classifier in an iterative fashion. In each iteration, it calls a simple learning algorithm that returns a classifier and assigns a weight coefficient to it. The final classification will be decided by a weighted "vote" of the base classifiers. The smaller the error of the base classifier, the larger is its weight in the final vote [35].

B. Naive Bayes Algorithm

The Navies Bayes algorithm is a simple technique for constructing classifiers. It is a supervised ML algorithm based largely off Bayes Theorem. It uses conditional probability and output is the posterior probability of being a certain class label given the instance features as denoted in Equation (3.1). Naive Bayes algorithm is relatively simple to understand and build, easily trained, fast and not sensitive to irrelevant features. In this study, to build the Naïve Bayes classifier model, software package is used in WEKA, and it employs the Naïve Bayes Simple algorithm in developing the model [36].

$$P(A/B) = \frac{P(B/A)XP(A)}{P(B)} \tag{3.1}$$

Where

- P(A/B) posterior probability of being a class "B" given the Feature value "A"
- P(B/A) is probability of the feature value "B" when the given class is "A"
- P(A) is prior probability of the class label "A"
- P(B) is prior probability of the attribute value "B"

The Naïve Bayes model is a heavily simplified Bayesian probability model. The Naïve Bayes classifier operates on a strong independence assumption [36]. This means that the probability of one attribute does not affect the of other. To calculate the probability that an intrusion is occurring based on some dataset by first

calculating the probability that some previous dataset was part of them and then multiplying by the probability of intrusion occurring.

C. J48 Decision Tree Algorithm

The J48 decision tree algorithm is one of the predictive model techniques used in data mining, machine learning and statistics for classifier. J48 creates univariate decision trees. It is based used attribute correlation based on entropy and information gain for each features [37]. It has been used in many fields of study, such as machine learning, data mining, information extraction and pattern recognition. The J48 algorithm has many advantages, some of them are it can deal with different input data types: nominal, textual and numeric. J48 algorithm has an advantage over Iterative Dichotomiser 3 (ID 3) in that it can build small trees because the J48 decision tree is an extension of the algorithm ID3. The J48 is an open-source Java implementation of the C4.5 algorithm.

Figure 3.2.1 shows that a typical decision tree structure. The tree is constructed by following these three main steps: A decision tree consists of several elements: root, internal nodes, and leaves. The Leaf nodes represent decision or the class. Internal nodes perform the conditions in which the value of parameters will be tested. Based on these values and condition, the flow of tree will be decided along which branch the decision tree must go [37].

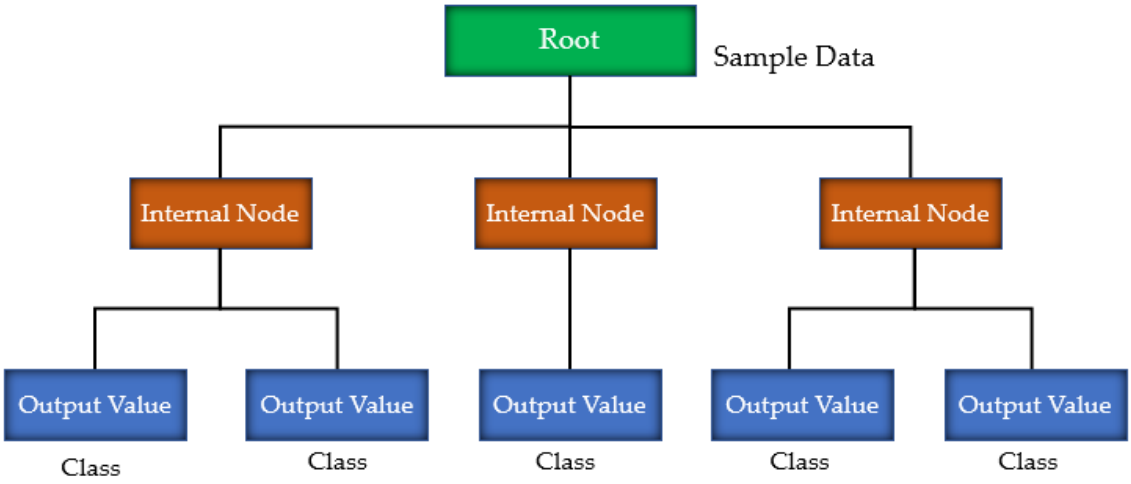


Figure 3.2.1: Decision Tree Structure [38]

The entropy comes from information theory and indicates the amount of information that is held. In other words, the higher entropy, the more information content there is. Entropy of H(X) is define by Equation (3.2):

$$H(X) = -\sum P(X)\ log2\ P(X) \tag{3.2}$$

Where P(X) is the probability of class 'X'.

Information gain expresses the importance of feature or attribute, and it determines which feature is the most important one for distinguishing between the classes to be knowledgeable. This piece of information is calculated using training data and by this Equation (3.3). Information gain can help in choosing the best split; if it has a high value then this split is good, otherwise, the split is not good enough. Information gain can be calculated by the data achieved from entropy.

$$IG = Entropy\,(\text{parent}) - [\text{Average Entropy}\ \ (\text{children})\,] \tag{3.3}$$

D. Artificial Neural Network Algorithm

Artificial Neural Network (ANN) is an algorithm which is inspired by the function and structure of biological neural networks, the central nervous system in a human's brain [39]. ANN include an accumulation of processing elements interconnected together and aimed to transform some inputs to desired outputs. It is used for both classification and regression tasks. In a neural network algorithm, the structure consists of a network with many processors (neurons). The processor receives data from outside or from other neurons. Then the data received earlier will be accepted through a link called weights.

ANN has been involved in many applications to solve real-world problems. In industry, engineers can apply ANN to solve many engineering problems such as classifications, prediction, pattern recognition, and non-linear problems where the issues are very difficult or might be impossible to solve through normal mathematical processes. Advantages of ANN has self-learning capability, performs tasks that a linear program cannot, and a neural network learns and does not need to be reprogrammed. In ANN, the perceptron is used to classify linearly separable

classes. while, MultilayerPerceptron(MLP) is used for classes which cannot be separated using a linear function.

MLP is a well-known type of ANN, which is usually used in classification problems. It is widely used neural network classifier based on several classes (output) and several hidden layers, MLP uses weights for every node at the neural network, most effective attributes will get large weights conversely attributes not affect in predictive class. MLP always takes the largest time for training, however it has quick time for testing. Figure 3.2.2 shown that MLP the neurons are grouped in one hidden layer.

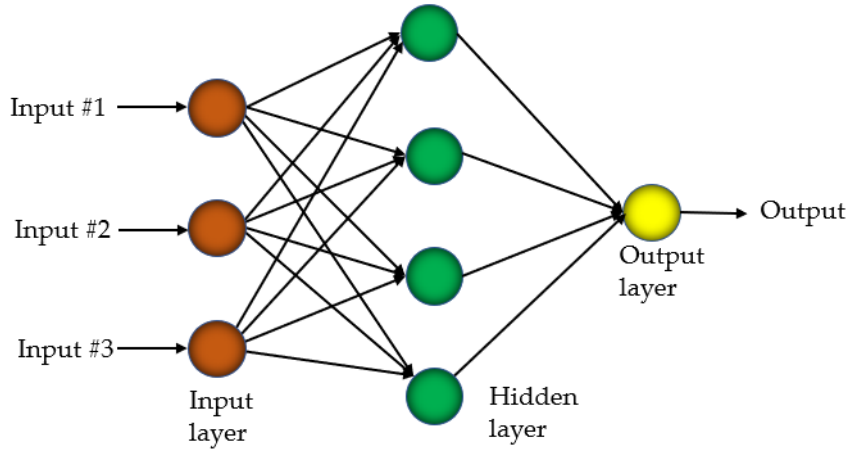


Figure 3.2.2: MPL Architecture of ANN [40]

3.2.2 *Unsupervised ML Technique*

In contrary to supervised ML, unsupervised ML is learning from unlabeled training data. It is a type of ML algorithm that brings order to the dataset and makes sense of data. It is used to group unstructured data according to its similarities and distinct patterns in dataset. The main objective of unsupervised learning is to model the underlying structure or distribution the data in order to learn more about the data. It can be further grouped into clustering and association ML [41].

Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior. Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns in data. The common clustering ML algorithms used in IDS are K-Means and Expectational Maximization [41].

A. K-Means Algorithm

The K-Means clustering algorithm is seen as one of the simplest and most classical approaches to data clustering and remains one of the most widely used in practice, largely due to its simplicity [42]. In the K-Means approach, the optimal clustering model is defined as the set of K centroids which minimizes the sum of squared euclidean distances between each datum and its cluster centroid. Centroid-based clustering defines clusters in relation to single representative points [42].

B. Expectational Maximization Algorithm

Expectational Maximization (EM) Algorithm is a clustering technique that can derive the maximum likelihood estimates in the presence of missing data [43]. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step.

3.2.3 *Semi-supervised Learning Technique*

Semi-supervised ML is a combination of supervised and unsupervised ML methods. It is learning from both labeled and unlabeled dataset. Labeled data is often expensive and difficult to obtain. While, the unlabeled dataset is relatively easy to collect, however, it is contains less information than labeled data. By using a large amount of unlabeled data with a small amount of labeled data, semi-supervised learning requires less human effort and tends to higher accuracy [44].

3.2.4 *Reinforcement ML Technique*

It is the branch of ML algorithm relating to learning in successive decision-making settings. Reinforcement technique ML a mapping from circumstances to actions by trial-and-error interactions with a dynamic environment. The objective of reinforcement ML technique is defined using the concept of a reinforcement function, which is the exact purpose of future reinforcements the agent seeks to maximize.

3.3 TOOLS USED FOR THE EXPERIMENTS

In this study, three different types open source tools have been used for the development of SYN flood attack detection, such as packet generation tool (Hping3), packet capturing tool (Wireshark) and packet analyzing tool (WEKA) were discussed in the next subsection.

3.3.1 *Packet Generating Tool*

The Hping3 is a free packet generator and analyzer pre-installed package on Kali Linux. It is a command-line based packet analyzer. It can be used for Firewall testing, advanced port scanning, network testing using different Internet protocols, advanced traceroute, TCP/IP protocol [45]. With Hping3 options users can specify the target server, a number of packets send to the target port, spoofing attack source, selecting a random source, destination, flooding to send requests to the target as fast as possible, protocol types such as TCP, UDP, ICMP and many more options. We have used Hping3 to launch UDP and TCP flood on the server running on another machine.

3.3.2 *Packet Capturing Tool*

Wireshark is an open-source application that captures and displays data traveling back and forth on a network and capture live data flowing in network interface. It is a very popular packet analyzer for various platforms, and it supports a lot of protocols. It is used for network troubleshooting, analysis and protocol development. It is a great tool for capturing traffic on a single interface or system, but it is not designed to handle large volumes of traffic. It is open-source, which is responsible for real-time PCAP and operates in a Windows and Linux operating systems [46].

3.3.3 *Packet Analyzing Tool*

WEKA is a data mining tool written in Java code [47]. It is an open-source tool, developed at the University of Waikato, New Zealand. It is also containing a collection of data preprocessing, classification, regression, clustering, association rules and visualization. All the selected algorithms are supported in WEKA (version 3.8.3). It has a graphical user interface as well as a command-based interface which makes it attractive to be used in this research. It requires file formats such as Comma Separated Values (CSV) and Attribute Relation File Format (ARFF).

3.3.4 *Kali Linux Operating System*

Kali Linux (version 19.3) is the world's most powerful and popular penetration testing platform, used by security professionals in a wide range of specializations, including penetration testing, forensics, reverse engineering, and vulnerability assessment [48]. It is open-source and free of cost tool. It is a Linux distribution that contains its own collection of hundreds of software tools specifically tailored for their target user's penetration testers and other security professionals.

4

EXPERIMENTAL ANALYSIS

This chapter focuses on how the dataset is collect and prepare for the purpose of experimental analysis of SYN flood attack detection. In the first section, describe the system model, the process of network traffic packet generated and captured. In the second section, discuss understanding the dataset, data description, preparation, integration, transformation and feature selection up to data formatting. In the third section present feature selection process and defines all metrics used in our experimental for the next chapter. Finally, analysis performs.

4.1 SYSTEM MODEL FOR DETECTION OF SYN FLOOD ATTACK

This section describes the system model for detection of SYN flood attack as shown in Figure 4.1.1. In order to accomplish the objectives of this study and answers the research questions. This system model indicates that how the thesis work was structured from data collection until the model experimentation.

In the first step: Generate network traffic dataset using Hping3 utilization tool in Kali Linux environment using ethio telecom network.

In the second step: Captured real network traffic and generated SYN flood dataset by using Wireshark open source tool, all captured data have been in PCAP format and passes through different ethio telecom network infrastructure;

In the third step: Both captured network traffic PCAP dataset load into the Structured Query Language (SQL) database to easily managed and preprocessed the datasets.

In the fourth step: To improves the detection performance and minimize the time taken to build models, the data preprocessing, cleaning and normalization are performed. After that, feature selection techniques are applied to identify relevant features for the model, using manual and three feature selection techniques.

In the fifth step: After data processing and feature selection, trained selected classification ML algorithms by training dataset and testing using test dataset. Then, evaluate the detection models using some metrics such as accuracy, False Positive Rate (FPR), building time and so on.

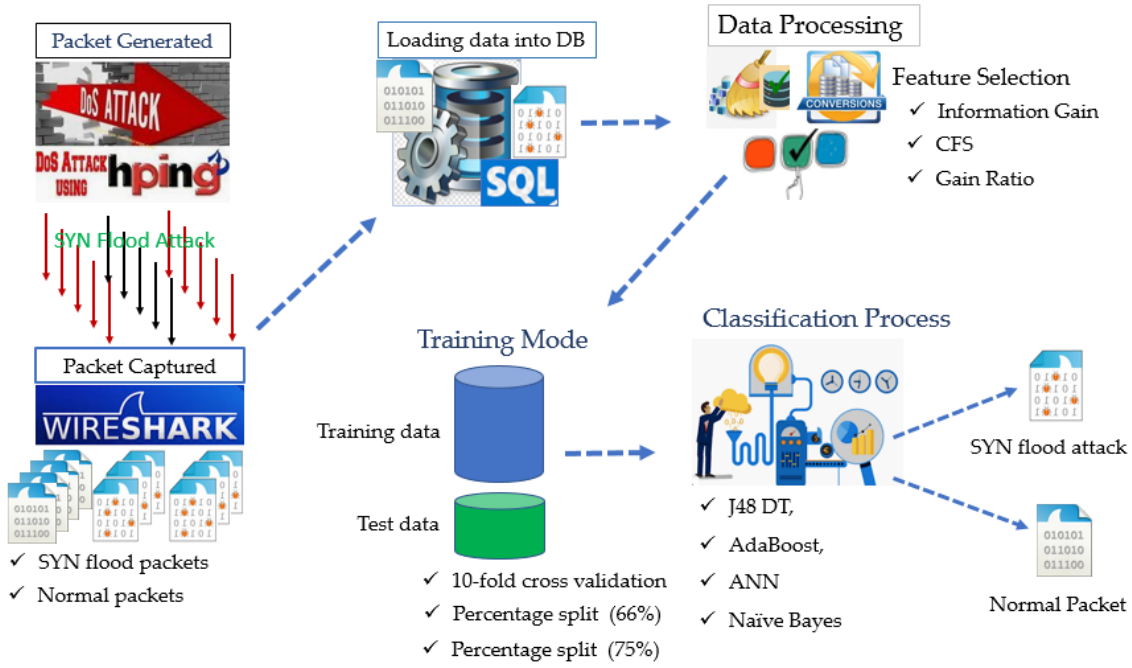


Figure 4.1.1: System Model for SYN Flood Attack Detection

4.2 EXPERIMENTAL SETUP

All experiments in this study were performed using a hardware specification of Intel Core i7-4610M CPU @ 3.00G Hz processor, 8 GB RAM with the operating system Windows 10, 64bit and Kali Linux (version 19.3). Experiments were done using the WEKA (version 3.8.3) data mining tool used for data preprocessing, classification and visualization.

4.3 DATA SOURCE

Dataset is always required in ML to train selected algorithms to gain knowledge. Accordingly, in this study network traffic dataset has been used, which was obtained from ethio telecom network. Hence, generated and captured SYN flood PCAP dataset using Hping3 and Wireshark open-source tools. The normal traffic data for this study separately captured from ethio telecom real network traffic, using Wireshark tool and then combined them together.

The generated and captured dataset was PCAP format and contains data from second to the seventh layer of the Open Systems Interconnection (OSI) model information. For this specific study only have been used IP header from layer 3 and TCP header from layer 4 of the OSI model. To detect SYN flood attack using ML algorithms, network traffic dataset was required. The process of SYN flood packet generation and capturing and normal network traffic captured from ethio telecom network. The following two subsection are explained the process.

4.3.1 *SYN Flood Packet Generation*

To generate SYN flood PCAP format dataset, have been used three laptop machines. The first laptop is used for attacking the victim machine, the second laptop was used as victim of attack and the third laptop is used as the observed and captured network traffic dataset, as shown in Figure 4.3.1 The attackers have been used spoof IP addresses. All these three Laptop machines are installed in the same Local Area Network (LAN) environment. The total generated and captured SYN flood contains 2 million packets and 20 basic feature subsets.

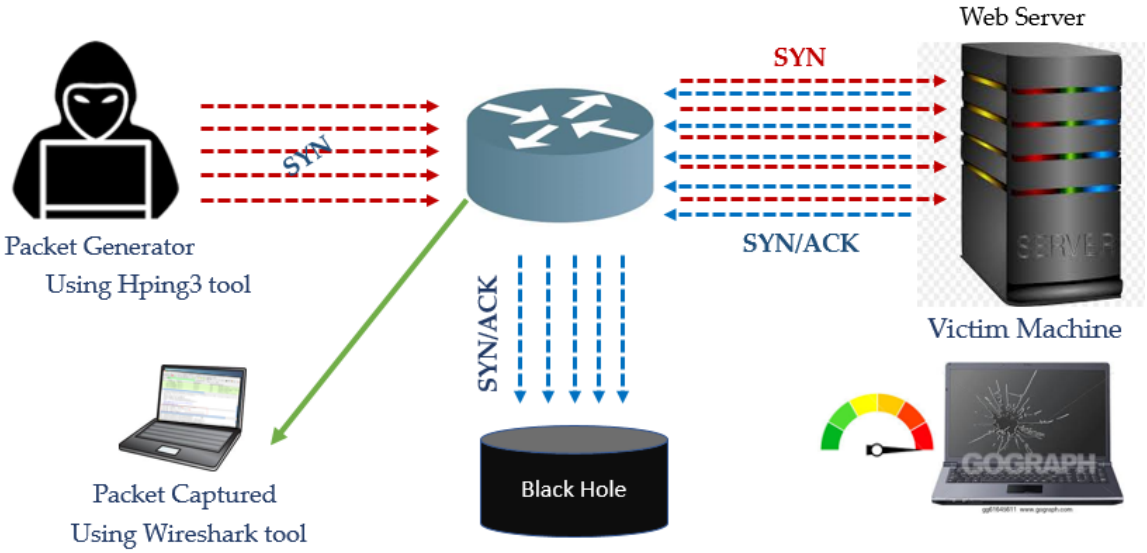


Figure 4.3.1: Lab Experiment for SYN Flood Attack [27]

For this experiment, the researcher has been used the following hardware and software specifications. Dell Intel Core i7-4610M CPU @ 3.00 GHz processor with 8.00 GB of Random Access Memory (RAM) and Windows 10, 64-bit operating system used for victim and packet observer. The attacking machine Kali Linux (version 19.3) and the host Debian was installed in the virtual environment, and both machines were set up for demonstration.

The Kali Linux Hping3 command as shown in Figure 4.3.2. It is a TCP/IP packet generator and analyzer tool. It is commonly used for generating network traffic packets. Because of its characteristic functionality, many attackers utilize Hping3 for denial of service attacks for flooding. This study has been used Kali Linux Hping3 for SYN flood packet generate.

```
root@kali:~# hping3 -S -p 80 --flood --rand-source 192.168.103.130
HPING 192.168.103.130 (eth0 192.168.103.130): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.103.130 hping statistic ---
8571113 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~#
```

Figure 4.3.2: SYN Flood Packets Generation Kali Linux Command

Where

Hping3 = name of packet generation application
-c = Number of packets (15000)
-d = Data size
-S = SYN packets only
-p 80 = Destination port (example http port number = 80)
-s = Source port
-a = spoofed hostname
–flood = Sent packets at fast as possible and don't show replies, non-stop, rapid transmission. The third laptop is used as the observe communication between the attacker and victim machine, finally captured using Wireshark.

4.3.2 *Normal Packet Capturing*

To captured normal network traffic dataset, have been used one laptop machines, using Wireshark open source tool, from ethio telecom real network traffic. All capturing raw datasets were in PCAP format and usually done by mirroring ports on the network. This dataset is PCAP format and contains data from second to the seventh layer of the OSI model information. The total captured normal traffic dataset contains 2 million and 20 basic feature subsets.

4.3.3 *Data Description*

According to the above two data collection technique have been collected around 4 million connection records and 20 features. The datasets composed of two types and labeled as either normal and SYN flood packets. It is obtained directly from the packet header and payload. The raw data consists of packet-level transmission data including the following features subset. The list of feature contains both numeric and textual types.

4.4 DATA PREPROCESSING

The main objective of data preprocessing is used to obtaining clean dataset from the raw input data or reducing dimensionality of the dataset by keeping relevant information. In this study, in order to improve efficiency, reduce computational time and ease the data mining process, according to data behaviour the following major data preprocessing task were performed. Such as data cleaning, transformation, normalization and finally the data formatting process has been done. Since the real-world dataset has a tendency to be incomplete, noisy and inconsistent, thus the data preprocessing stage is one of the major tasks in the knowledge discovery process.

4.4.1 *Data Cleaning*

The data cleaning process is used for making sure that the dataset is free from different types of errors and in order to make the data complete. In this study, the data cleaning process has been done by removing outliers, filling missed values in a dataset, redundant and smoothing noisy dataset. Data cleaning is typically required to prepare data for use with ML algorithms. Microsoft excel 365 and Structured Query Language (SQL) database used for further studied, selected, preprocessed and transformed to the appropriate format [49].

4.4.2 *Data Transforming*

The generated and captured dataset contains symbolic, continuous, and binary values. For instance, the feature 'protocol type' datasets include symbolic values such as TCP, UDP, and ICMP. As many ML algorithms accept only numerical values, the converting process is considered vital and has a significant impact on SYN flood attack detection accuracy. According to this, as per Table 4.4.1 shown the transformation value is presented. [49].

Table 4.4.1: Data Transforming Nominal to Numeric

Nominal Value	Numeric Value	Description
0x002	2	SYN
0x004	4	RST
0x010	10	ACK
0x011	11	(FIN, ACK)
0x014	14	(RST, ACK)
0x018	18	(PSH, ACK)
0x019	19	(FIN, PSH, ACK)
0x000	0	No flag

4.4.3 *Normalization*

In this research packet generated and capture dataset have been used. This dataset contains numeric features that are characterized over ranges with different magnitude. This disparity in magnitude may lead to the emergence of a bias towards certain features. Features that take on large numeric values can dominate the classifier's model relative to features with relatively small numeric values. Hence, It is a method that is used to ensure that the features are defined over a common range, with no bias towards certain features, as a result of disparity of scale [50].

There are several normalization techniques that can be used to normalize numerical features include [50]. In this research, the maximum-minimum normalization technique is applied to the numeric features of the generated and captured dataset. The technique rescales each numeric feature such that it has a mean value of zero and a unity standard deviation when calculated over all samples of the dataset. It is expressed as shown Equation (4.1)

$$Normalization = \frac{Xo - Xmin}{Xmax - Xmin} \tag{4.1}$$

Where

- **Xo** is the old value of each entry in data.
- **Xmin** is the minimum absolute value
- **Xmax** is the maximum absolute value

4.5 FEATURE SELECTION

Feature selection is the process of gaining a subset of related attributes or features to be used in constructing a model [51]. It helps to improve the detection performance, minimize false alarm rate and improve the model building and evaluation time taken. The required dataset collected from ethio telecom network by generating and capturing. The captured dataset has 20 basic features. For this specific study, not all feature subset is relevant, so to identify the relevant feature, a feature selection technique is mandatory. In order to select the relevant feature by remove redundant and irrelevant features. This study has been used manual feature selection and evaluate the worthiness of feature subset techniques are discussed in the next subsection.

4.5.1 *Manual Feature Selection*

The first feature selection method has been identified as the relevant features by manual selection with the help of ethio telecom domain expert support and conducting a different literature review, related to network intrusion detection. As a result, the following 11 features were selected from 20 basic features, such as *duration, dst_add, flag_types, dst_types, count, srv_count, src_bytes, dst_bytes, packet_size, protocol_type and src_port.*

4.5.2 *Feature Worthiness Evaluation*

The second feature selection method was evaluated the worthiness of each feature in relation to the classification class. In this study to evaluate the worthiness of manually selected feature subset have been used three well-known feature selection and reduction techniques, such as information gain, gain ratio and correlation based feature selection are discussed in the next subsection.

i. Information Gain Feature Selection

Information Gain (IG) is commonly used to measure in the fields of information theory and machine learning [52]. It evaluates the worth of a feature based on the information gain with respect to the class. IG feature selection ranks the attribute based on the separating classes of the training samples. For each feature, a score is obtained based on how much more information about the class is gained when using that feature.

The result obtained through information gain feature selection methods as shows in Figure 4.5.1. The feature subset (protocol type and source port) are perform zero values, and the others feature subset perform better result, and the importance of a feature subset decreased from top to bottom for this specific study. The rank of each feature subset as shown in Table 4.5.1.

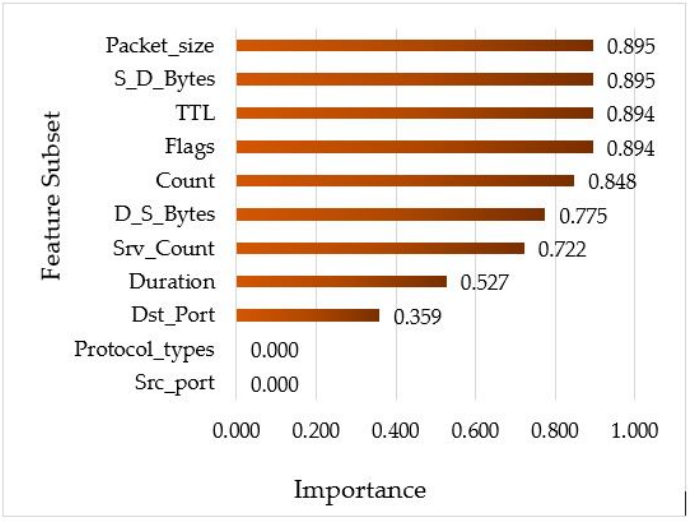


Figure 4.5.1: Evaluation of Feature Subset using Information Gain FS

ii. Gain Ratio Feature Selection

Gain Ratio (GR) is a modification of the information gain that reduces its bias. It takes the number and size of branches into account when choosing a feature subset. It corrects the information gain by taking the intrinsic information of a split into account. It is the entropy of distribution of instances into branches (i.e. how much info do we need to tell which branch an instance belongs to). Value of attribute decreases as intrinsic information gets larger [52].

The result obtained through GR feature selection methods as shown in Figure 4.5.2. The feature subset (protocol type and source port) are performed zero values, and the others feature subset perform better result, and the importance of a feature subset decreased from top to bottom for this specific study. The rank of each feature subset as shown in Table 4.5.1.

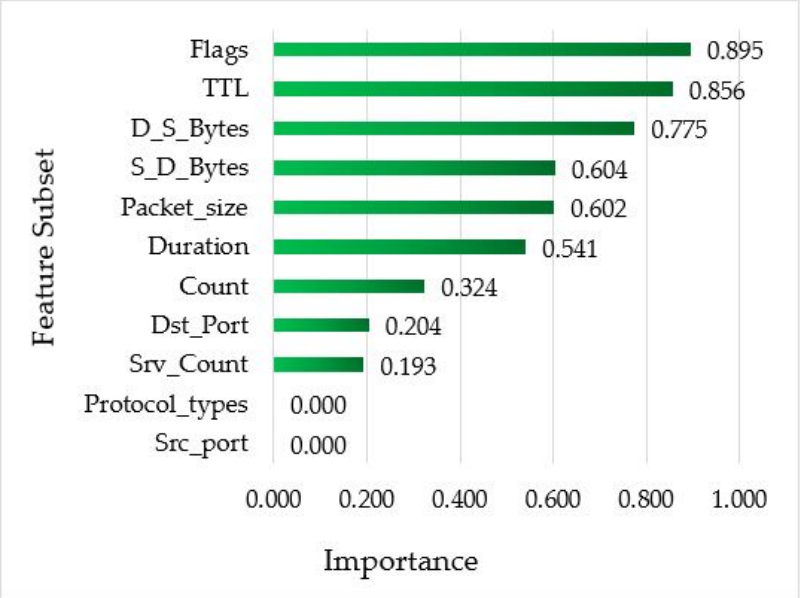


Figure 4.5.2: Evaluation of Feature Subset using Gain Ratio FS

iii. Correlation-based Feature Selection

Correlation based Feature Selection (CFS) for classification tasks in machine learning can be accomplished based on the correlation between features and that such a feature selection procedure can be beneficial to common machine learning algorithms. Correlation is a well-known similarity measure between two features. One is the feature-classification correlation, and another is the feature-feature. These

two concepts are based on the following hypothesis: "Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other" [53].

The result obtained through CFS methods as shown in Figure 4.5.3. The feature subset (protocol type and source port) are performed zero values, and the others feature subset perform better result, and the importance of a feature subset decreased from top to bottom for this specific study. The rank of each feature subset as shown in Table 4.5.1.

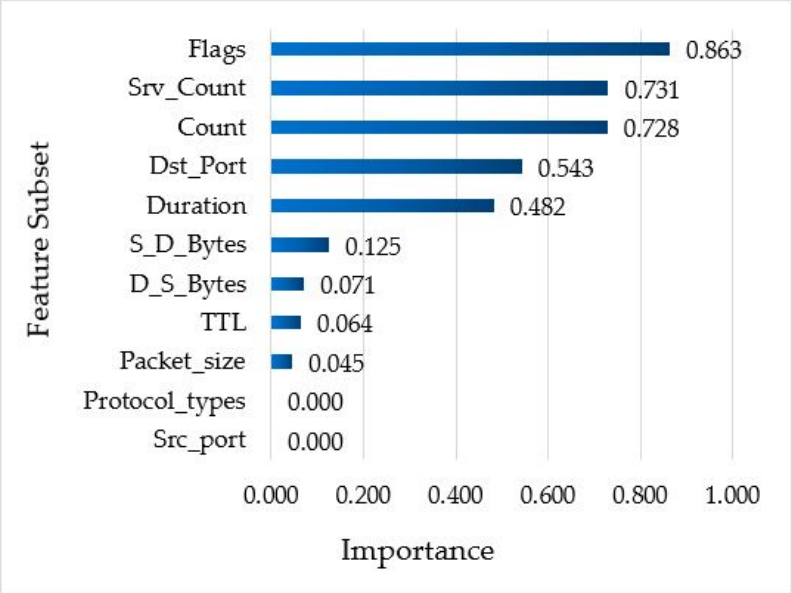


Figure 4.5.3: Evaluation of Feature Subset using CFS

Table 4.5.1 shows the rank of each features, based on three feature selection methods. From this learn goes from left to right importance of the feature decrease for SYN flood attack detection.

Table 4.5.1: Ranking of Selected Feature Subset

Method	Subset of Features selected										
GR	7	9	11	10	8	1	2	4	6	5	3
Correllation	7	6	2	4	1	10	11	9	8	5	3
IG	10	8	9	7	2	11	6	1	4	3	5

The summery of three-feature selection techniques result, both feature methods selected 9 common types of feature subset and two feature subsets were zero value. As a result, 'protocol type' and 'Src_port' (Source port) features had a value of zero. Which means that manually selected two features were irrelevant, and not provide significant information for SYN flood attack detection. Hence, it is removed from the selected feature list and the remaining. Whereas, the other 9 features were relevant and important to this specific attack detection. According to this from 20 basic 9 features subset are selected and describes each with their corresponding data type are listed in the following.

- **Duration:** Packet time interval or length of the connection (in sec) its data type is numeric.
- **Protocol_type:** Type of the protocol (TCP, UDP, ICMP, ...) which is used for communicating over a network and its data types is nominal/Textual.
- **Service:** Also known as destination port number Network service on the destination (port number 21, 80, 22, 443 and 25) and its data type is numeric.
- **Src_bytes:** It is source bytes, which is the number of bytes sent from the source system to the destination system and its data type is numeric.
- **Dst_bytes:** It is destination bytes, which is the number of bytes sent from the destination system to the host system and its data type is numeric.
- **Flag:** It is connection status flag, which indicate the packet is normal or error status of the connection (SYN, ACK, Finished (FIN), ...) and its data types is nominal.
- **TTL:** It is time to live, which indicate the amount of time or "hops" that a packet is set to exist inside a network and its data type is numeric.
- **Dst_IP_addr:** It is a destination IP address and its data types is numeric.
- **Packet size:** each packet includes a source and destination as well as the content being transferred and its data type is numeric.
- **Count:** It is the sum of connections to same destination IP address within 2 sec. The term 'same host' refers to connections in the past two seconds that

have the same destination host as the current connection and its data type is numeric.

- **Srv_Count:** It is service count and the sum of connections to the same port number within 2 second. The term 'same service' refers to connections in the past two seconds that have the same service as the current connection and the data type is numeric.

4.6 DATA FORMATTING

After data preprocessed and feature selection, the dataset has been ready for model training and testing purpose. In this study all experimental analysis implemented in WEKA data mining tool. This tool accepts file formats such as comma delimited CSV and ARFF. Hence, converted all excel format into CSV format.

4.7 MODEL EVALUATION METRICS

In this study, for running the experimental analysis, WEKA data mining tool has been selected due to the ease of varying different parameters to observing the results. The results will be presented in two different methods namely confusion matrix and Receiver Operating Characteristic (ROC) [54].

i. ROC Curve: It help to decide where to draw the line between normal network traffic and SYN flood attack. Moreover, the process of evaluation was done and shown by comparing with the existing intelligent approach for SYN flood attack detection 4.7.1.

ii. Confusion Matrix: It is used to evaluate the performance of each classifier and combination methods are derived from the confusion matrix results of each fold of each experiment. In this study each classifier is making a binary decision for each connection, SYN flood attack or normal network traffic, therefore a (2 x 2) confusion matrix is used [54]. As shown in the Table 4.7.1 below.

Where

- **True Positive (TP)**: Attack occur, and the alarm raised
- **True Negative (TN)**: No attack and no alarm
- **False Positive (FP):** No attack however alarm raised
- **False Negative (FN):** Attack occurs however no alarm

Table 4.7.1: Confusion Matrix Value (2X2)

		Predict Result	
		Positive	Negative
Actual Result	Positive	TP	FN
	Negative	FP	TN

Accuracy: The accuracy of an ML model in a classification problem corresponds to the correct (either positive or negative) predictions made overall predictions. As mentioned earlier effectiveness of the proposed IDS is measured in terms of accuracy in which it identifies how much do IDS classify the coming packet as normal and attack. The accuracy of detection approach is calculated using Equation (4.2).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.2}$$

Precision: It is used to measure the proportion of positive data instances that a model classified as positive. The precision metric ignores the capabilities of a model to recognize negative classes. Technically can be expressed as the attack has occurred and IDS detects correctly. The precision of detection approach is calculated using Equation (4.3).

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

Recall: It is a model proportion of true positives that the model identified. As such, a model that yields no false negative (FN) has a recall of value well be one. Precision and recall are normally inversely proportional to each other. Which means one is improved, the other is degraded. The recall of detection approach is calculated using Equation (4.4)

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

F-Measure: It is a combination and harmonic mean of precision and recall. The f-measure of detection approach is calculated using Equation (4.5)

$$F - Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.5}$$

Root Mean Square Error (RMSE): It is a quadratic scoring rule which measures the average magnitude of the error. It is the average of squares difference between forecast and corresponding observed values, and the square root of the average is taken. Since the errors are squared before they are averaged, it gives a relatively high weight to large errors. This means it is most useful when large errors are particularly undesirable. The RMSE of detection approach is calculated using Equation (4.6)

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(d_i - f_i)^2} \tag{4.6}$$

ROC Curve: It is an essential graph for diagnostic test evaluation. It is a graph of FPR against $TruePositiveRate$(TPR) area measures the ability of classifier to correctly classify the test data. It shows the performance of models across all possible thresholds. A model that covers a larger area in the plot has better classification.

False Negative Rate : It is defined as the ratio of false negative samples to total positive samples. In attack detection, the False Negative Rate (FNR) is also called the missed alarm rate. It is calculated using Equation (4.7)

$$FNR = \frac{FN}{TP + FN} \tag{4.7}$$

False Positive Rate : It is defined as the ratio of false positive samples to predicted positive samples. In attack detection, the FPR is also called the false alarm rate, and it is calculated using Equation (4.8)

$$FPR = \frac{FP}{TP + FP} \tag{4.8}$$

4.8 TRAINING AND TESTING DATASETS

In this study, network traffic dataset have been generated, and captured for evaluating the appropriate ML algorithm for the detection of SYN flood attack. The selected modeling algorithms, that is AdaBoost, J48, Naïve Bayes and ANN are used for training and their respective resulting models are used for detection. All dataset is labeled as SYN flood and normal, contains 40,832 and 41,383 packets respectively. Training data is used for building classification model and testing data is used for measuring the performance of model.

Training and testing data should be different, mutually independent and created by random sampling. In this research, all experiments have been used 10-fold cross-validation and percentage split training and testing mode because it reduces the variance of estimate. In addition to this, to ensure a balanced level of algorithms comparison, recommended default parameters are used.

In order to show the impact of training dataset size on the out put performance, this study have taken the following training and testing options as followed.

- The first option has been used trained and tested using the 10-fold cross validation testing model.
- The second option has been used 75% of the total dataset for our training and the remaining 25% for testing the model, which means three-fourth of dataset for training and one-fourth for testing.
- The third option has been used 66% of the total dataset for training to build the model and remaining 34% would be used for testing of the model, which means two-third of dataset for training and one-third for testing.

4.9 MODEL EXPERIMENTATION

The main objective of this thesis work is to evaluate the performance of machine learning algorithm for the detection of SYN flood attack. To achieve the main objectives and answer the research questions, the collected dataset have been pre-processed and selected relevant features. In addition to this, in order to show the impact of training dataset, set three training approaches and four classification ML algorithms. Accordingly, there are 12 models of experimental analysis were performed in the next subsection.

4.9.1 *Model with AdaBoost Experimentation*

The first experiment has been implemented model with AdaBoost classifier. As described in Chapter 3. AdaBoost algorithm is a classification in an iterative fashion. In each iteration, it calls a simple learning algorithm that returns a classifier and assigns a weight coefficient to it. The final classification will be decided by a weighted "vote" of base classifiers. The experimentation of this model has been done by employing the following three training approaches (10-fold cross-validation and (66% and 75% training) percentage split approach).

Table 4.9.1: Confusion Matrix of AdaBoost Algorithm

Predict →	10-fold		66% Training		75% Training	
Actual ↓	**N**	**S**	**N**	**S**	**N**	**S**
N	41188	185	14034	75	10315	54
S	1028	39814	372	13472	273	9912

Where "**N**" represent Normal network traffic and "**S**" SYN flood attack.

Table 4.9.1 shows the resulting of confusion matrix for AdaBoost algorithm experiments in both training option. The number of correctly and incorrectly classified instances are recorded as shown in the Table 4.9.2.

Table 4.9.2: Classification Accuracy of AdaBoost Algorithm

Training Mode	**Classification**		**Time (sec)**	
	Accuracy	**FPR**	**Build**	**Evaluate**
Cross Validation	98.52%	2.5%	1.77	198
Percentage Split (75%)	98.41%	2.7%	0.02	74
Percentage Split (66%)	98.40%	2.7%	0.03	85

According to the above three experimental results, model with AdaBoost algorithm using 10-fold cross-validation approach attends the highest classification accuracy of 98.52% and minimum FPR of 2.5%. However, the time taken to build a model percentage split 75% training approach is better performance than 10-fold and percentage split 66% training approach. From this result, when training dataset decreases the accuracy of algorithm also reduced.

4.9.2 *Model with Naïve Bayes Experimentation*

The second experiment, has been implemented model with Naïve Bayes classifier. As described in Chapter 3, It is assumes that the attributes are conditionally

independent and thus tries to estimate the class conditional probability. It often produces good results in the classification where there exist simpler relations. It also requires only one scan of the training data and thus it eases the task of classification a lot. The experimentation of this model has been done by employing the following three training approaches (10-fold cross-validation and (66% and 75% training) percentage split approach).

Table 4.9.3: Confusion Matrix of Naïve Bayes Algorithm

Predict →	10-fold		66% Training		75% Training	
Actual ↓	**N**	**S**	**N**	**S**	**N**	**S**
N	38460	2103	13193	710	9677	692
S	2100	40113	600	13415	600	9936

Where "**N**" represent Normal network traffic and "**S**" SYN flood attack.

Table **??**latexïveBayestab : ConfusionMatrixofNaïveBayeswstheresultingofconfusionma

Table 4.9.4: Classification Accuracy of Naïve Bayes Algorithm

Training Mode	**Classification**		**Time (sec)**	
	Accuracy	**FPR**	**Build**	**Evaluate**
Cross Validation	94.92%	5.0%	0.20	94
Percentage Split (75%)	93.82%	5.7%	0.12	82
Percentage Split (66%)	95.31%	4.3%	0.18	80

According to the above three experimental results model with Naive Bayes using percentage split 66% training approach attends the highest classification accuracy of 95.31% and minimum FPR of 4.3%. However, the time taken to build a model percentage split 66% training approach is better performance than the other two approaches. From this result, when the training dataset decreases accuracy of the algorithm increase.

4.9.3 *Model with Artificial Neural Network Experimentation*

The third experiment, has been implemented model with ANN classifier. As described in Chapter 3, It is most commonly used for a wide variety of problems based on a supervised procedure and comprise three layers: input, hidden, and output. It has the ability to learn so quickly is what makes them so powerful and useful for a variety of tasks. The experimentation of this model has been done by employing the following three training approaches (10-fold cross-validation and (66% and 75% training) percentage split approach).

Table 4.9.5: Confusion Matrix of ANN Algorithm

Predict →	10-fold		66% Training		75% Training	
Actual ↓	**N**	**S**	**N**	**S**	**N**	**S**
N	41355	100	13930	938	10360	9
S	4280	36627	256	8054	1352	8833

Where "**N**" represent Normal network traffic and "**S**" SYN flood attack.

Table 4.9.5 shows the resulting of confusion matrix for ANN algorithm experiments in both training option. The number of correctly and incorrectly classified instances are recorded as shown in the Table 4.9.6.

Table 4.9.6: Classification Accuracy of ANN Algorithm

Training Mode	**Classification**		**Time (sec)**	
	Accuracy	**FPR**	**Build**	**Evaluate**
Cross Validation	94.68%	10.5%	3.61	236
Percentage Split (75%)	93.38%	13.3%	3.54	225
Percentage Split (66%)	94.85%	3.1%	3.57	230

According to the above three experimental results, model with ANN algorithm using percentage split 66% training approach attends the highest classification

accuracy of 94.85% and minimum FPR of 3.1%. However, the time taken to build a model percentage split 66% training approach is better performance than the other two approaches. From this results, when the training dataset decreases accuracy of the algorithm also reduced.

4.9.4 *Model with J48 Decision Tree Experimentation*

The last experiment, have been implemented model with J48 machine learning algorithm. As described in Chapter 3, the J48 Decision Tree is used for building the decision tree model using a given dataset to find the optimal decision tree by minimizing the generalization error. The experimentation of this model has been done by employing the following three training approaches (10-fold cross-validation and (66% and 75% training) percentage split approach).

Table 4.9.7: Confusion Matrix of J48 Algorithm

Predict →	10-fold		66% Training		75% Training	
Actual ↓	**N**	**S**	**N**	**S**	**N**	**S**
N	41209	164	14043	66	10325	44
S	1008	39834	367	13477	271	9914

Where "**N**" represent Normal network traffic and "**S**" SYN flood attack.

Table 4.9.7 shows the resulting of confusion matrix for J48 algorithm experiments in both training option. The number of correctly and incorrectly classified instances are recorded as shown in the Table 4.9.7.

Table 4.9.8: Classification Accuracy of J48 Algorithm

Training Mode	Classification		Time (sec)	
	Accuracy	FPR	Build	Evaluate
Cross Validation	98.57%	2.4%	1.40	139
Percentage Split (75%)	98.47%	2.7%	1.40	142
Percentage Split (66%)	98.45%	2.7%	1.40	144

According to the above three experimental results, model with J48 decision tree algorithm using 10-fold cross-validation approach attends the highest classification accuracy of 98.57% and minimum FPR. However, the time taken to build a model all training approach achieved equal time taken to build a model. From this result, when the training dataset decreases the accuracy of algorithm also reduced.

5

RESULTS AND DISCUSSION

This chapter presented all the results obtained from testing and evaluating the selected four classification ML algorithms and three training approaches for the detection of SYN flood attack. The previous chapter experiments were conducted to find out best detection performance and minimum false alarm rate. Moreover, this chapter discusses the experimental results of each model and compared the best model using different performance metrics. The results will be presented in two different methods namely Confusion Matrix and Receiver Operating Characteristic curve. Based on these experimental analyses discussed the outcomes of experiments.

5.1 EXPERIMENTAL RESULTS

In this section, review the final stage of selected four classification models which were built using the training dataset and evaluated on the testing dataset in the previous chapter. To evaluate the performance of each model's parameters such as, TPR which is the detection rate and FPR represents records misclassified as attacks divided by the kind records in the dataset. These evaluation parameters are the most important criteria for the machine learning algorithms to be considered as the best models for SYN flood attack category. The experimental results are shown in Table 5.1.1.

Table 5.1.1: Comparison of Selected Algorithms Performance

Selected Algorithm	Test Model	Accuracy	FPR
AdBoost	10-fold	98.52%	2.5%
Naïve Bayes	Persentage (66%)	95.31%	4.3%
ANN	Persentage (66%)	94.85%	3.1%
J48 DT	10-fold	98.57%	2.4%

Table 5.1.1 presents the evaluation result of J48 classifier has the highest detection rate for SYN flood attack with classification 98.57% of accuracy and 2.4% of FPR in 10-fold cross-validation training approach. Whereas ANN has the lowest classification of 95.15% of accuracy and 3.1% of FPR in percentage split 66% training approach. The AdaBoost algorithm has good accuracy and minimum FPR than Naive Bayes and ANN. The graphical representation as shown in the Figure 5.1.1

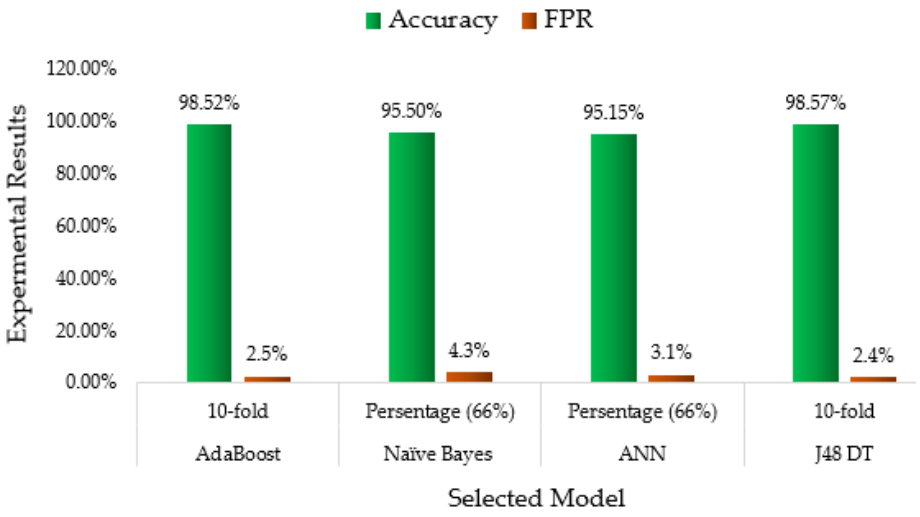


Figure 5.1.1: Comparison of Accuracy and FPR

Figure 5.1.2 shows the graphical representation of ROC and RMSE classification models performance. According to this AdaBoost and J48, algorithms have the lowest RMSE (error rate) which is 0.12 while ANN has scored the highest RMSE (error rate) value which is 0.67. The ROC results of J48, AdaBoost and Naïve Bayes were comparatively better than ANN model, which is 98.57%, 98.52% and 95.31%

respectively. Whereas ANN algorithm presented the comparatively lower value of ROC and RMSE result in this specific study.

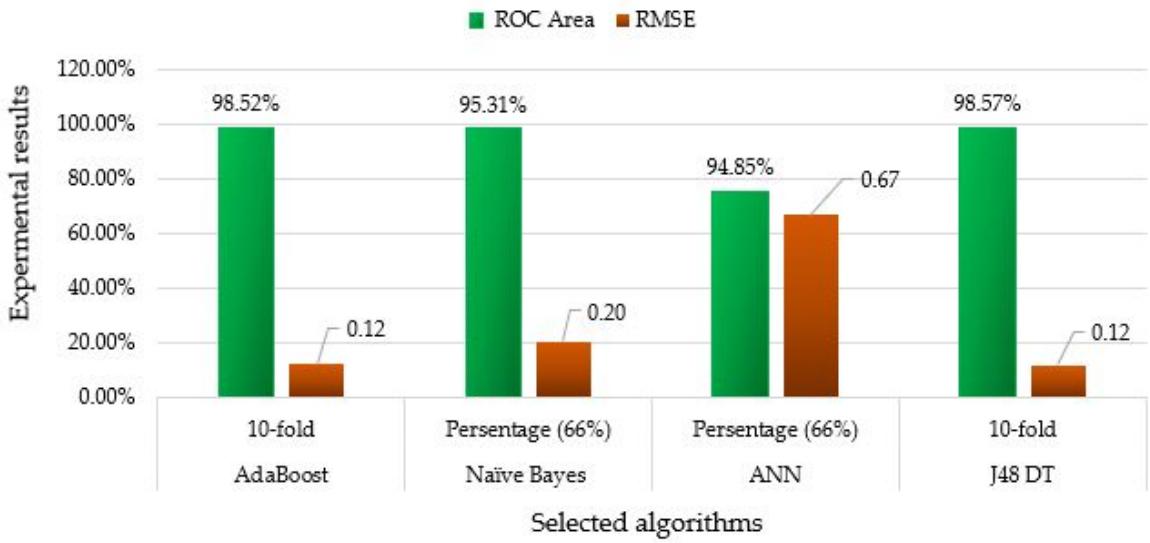


Figure 5.1.2: Comparison of ROC and RMSE models performance

Figure 5.1.3 and Figure 5.1.4 shows the graphical representation models classification performance , based on the time taken to build and evaluate. Accordingly, the Naïve Bayes classification model has a minimum building and evaluation time taken, whereas the ANN classification model performs maximum time is taken in both building as well as evaluation time. J48 and AdaBoost classification models presented the comparatively medium value of time taken in both cases.

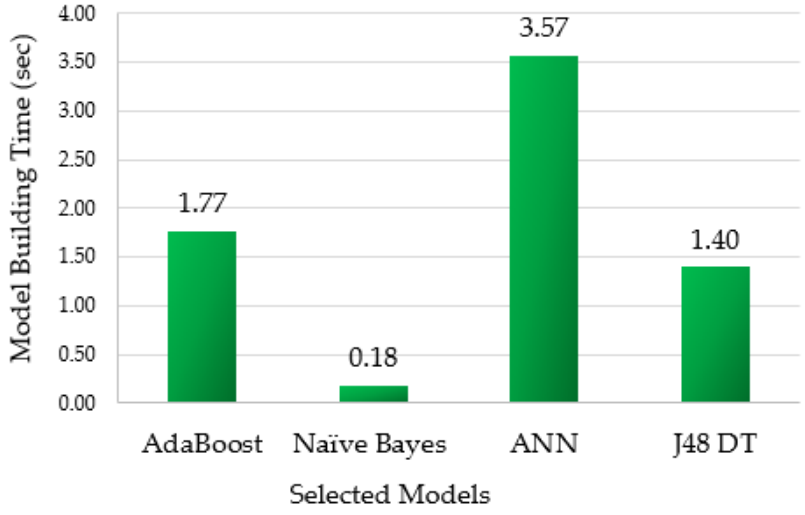


Figure 5.1.3: Time Taken to Build Models

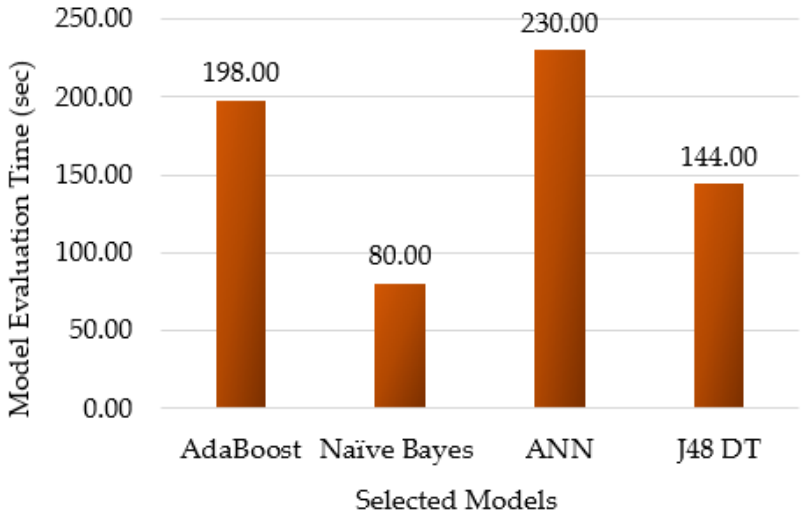
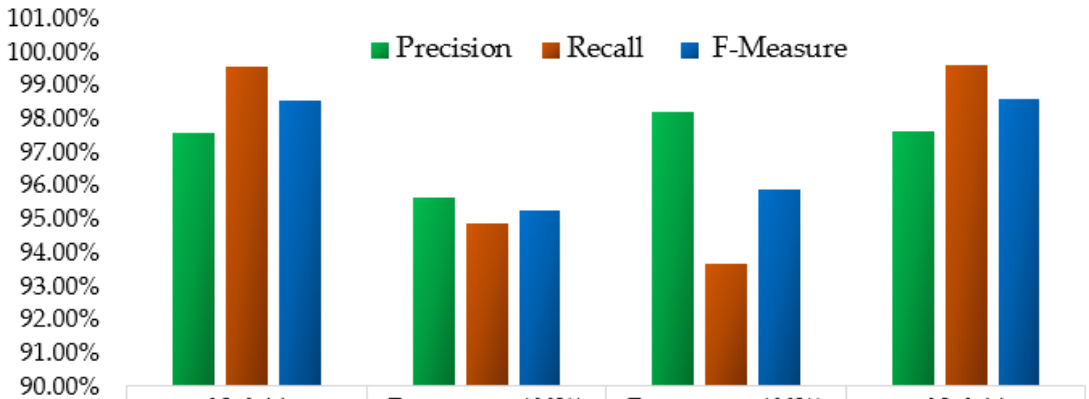


Figure 5.1.4: Time Taken to Evaluate Models

Figure 5.1.5 shows that four classification models performances used in this thesis and detail value of performance metrics like precision, recall and F-measure. J48 decision tree model have a better accuracy as compared to the others three models. AdaBoost classification model was also have a better accuracy than Naïve Bayes and ANN model. ANN classification model has been lower accuracy as compare to the others three classification models.



	10-fold AdaBoost	Persentage (66%) Naïve Bayes	Persentage (66%) ANN	10-fold J48 DT
Precision	97.56%	95.65%	98.20%	97.61%
Recall	99.55%	94.89%	93.69%	99.60%
F-Measure	98.55%	95.27%	95.89%	98.60%

Figure 5.1.5: Comparison with three Performance Metrics of Models

Figure 5.1.6 shows the performance comparison four classification model represented using the ROC curve. By observing the graphs, it can be concluded that the J48, AdaBoost and Naïve Bayes classification model has reached the same curve. These models have the lowest FPR and highest TPR in identifying SYN flood attacks. However, the ANN classification model comparatively the minimum performer for the detection of SYN flood attack in this study.

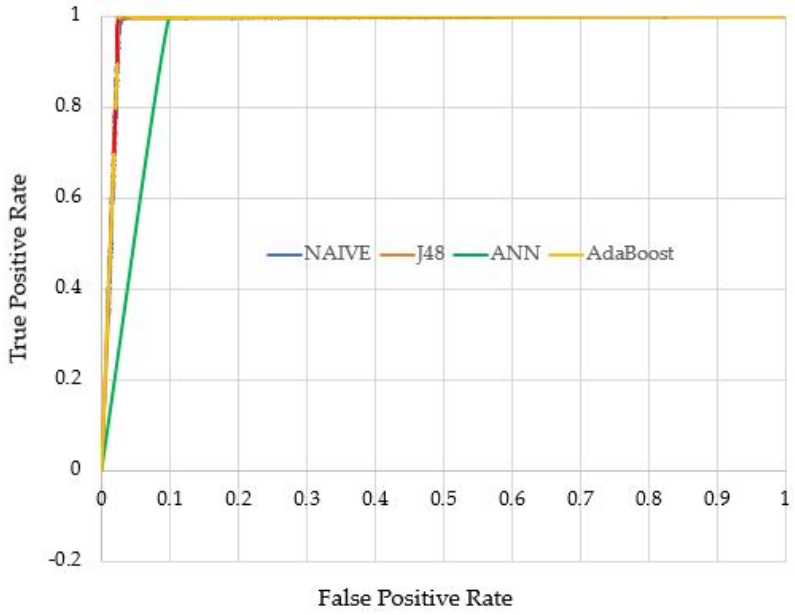


Figure 5.1.6: Comparison of ROC Curve

In summary, the classification models building experiment was accomplished in the previous chapter. Experiments were done using the WEKA data mining tool. Finally, the model which is developed with the J48 with 10-fold cross-validation classification approach is considered as the selected working model for SYN flood attack detection.

5.2 DISCUSSION

In this thesis work, the aim was to detect SYN flood attack and minimizing false alarm rate from telecom network, and the main objective was to evaluate the performance of machine learning algorithm as describe in Chapter 1. To achieve this objective and answer to the research question, the network traffic datasets have

been generated and captured from ethio telecom network and the data preprocessed task have been done. In addition to this, the classification machine learning algorithms (Naïve Bayes, AdaBoost, J48 decision tree and ANN) have been used from different literature reviews recommendation on the field of network intrusion detection. Moreover, this study has been used three types of training approaches, such as 10-fold cross-validation and percentage split 66% training and 34% test, 75% training and 25% test and implemented on WEKA data mining tool.

The first experiment has been performed model with AdaBoost classifier algorithm through three types of training approaches. The result in Table 4.9.2 shown that all models have equivalent performance measures. However, the AdaBoost classification model 10-fold cross-validation achieves the best performance as compared to the other two models and its value is 98.57% of accuracy and 2.5% of FPR. From this experimental analysis, we can see that, when the number of training dataset increase and the test dataset decrease, the performance of a model increased and FPR was minimum. Whereas the time taken to build and evaluate the model were percentage split approach has been better than 10-fold cross-validation.

The second experiment has been performed model with Naïve Bayes classifier algorithm through three types of training approaches. The result in Table 4.9.4 shown that all models have equivalent performance measures. However, the Naïve Bayes classification model with percentage split 66% training and 34% testing) approach achieves the best performance as compared to the others two models and its value is 95.31% of accuracy and 4.3% of FPR. From this experimental analysis, we can see that, when the number of training dataset decrease and the test dataset increase, the performance of a model increased and FPR was maximum. Whereas the time taken to build and evaluate the model were percentage split approach was better than 10-fold cross-validation.

The third experiment has been performed model with ANN classifier algorithm through three types of training approaches. The result in Table 4.9.6 shown that all models have equivalent performance measures. However, the ANN classification model with percentage split 66% training and 34% testing) approach achieves the

best performance as compared to the others two models and its value is 94.85% of accuracy and 3.1% of FPR. From this experimental analysis, we can see that, when the number of training dataset decrease and test dataset increase, the performance of a model increased and FPR was maximum. Whereas the time taken to build and evaluate the model were percentage split approach is better than 10-fold cross-validation.

The last experiment has been performed model with J48 classifier algorithm through three types of training approaches. The result in Table 4.9.6 shown that all models have corresponding performance measures. However, J48 classification model with 10-fold cross-validation approach achieves the best performance as compared to the other two models and its value is 98.57% of accuracy and 2.4% of FPR. From this experimental analysis, we can see that, when the number of training dataset increase and the test dataset decrease, the performance of a model increased and FPR was minimum. Whereas the time taken to build and evaluate the model was similar in both approaches.

Accordingly, the large value of classification model accuracy indicates the ability of a model to detect SYN flood attack while the lower value classification presents the weakness of a model. In addition to this, FPR measures the number of misclassified positive instances in relative to the total number of misclassified instances of the normal data is falsely detected as SYN flood attack. Therefore, according to the above experimental results, J48 decision tree and AdaBoost classification models were achieved best performance value J48 is 98.57% of accuracy and 2.4% of FPR; the performance value of AdaBoost is 98.52% accuracy and 2.5% FPR.

In addition to the above performance evaluation metrics, the quality of the classification models is identified by plotting the ROC curve. In each of these ROC plots, the x-axis is the false alarm rate, calculated as the percentage of normal connections classified as an SYN flood attack; the y-axis is the detection rate, calculated as the percentage of SYN flood attack detected. A data point in the upper left corner corresponds to optimal performance, which means the high detection rate with a low false alarm rate.

According to Figure 5.1.6 shows the experimental result of the ROC curve of classification models, it is possible to make a comparison between them based on the area under the ROC curve (AUC) that can display the performance accuracy of a binary classifier. In this study, we evaluated the performances of both ML algorithms using the area under AUC. The larger area under the curve, better accuracy detection can be achieved. The three classification models overlap each other, which indicate a slightly difference between them and achieve maximum performance. Whereas, the ANN classification model comparatively the minimum performer for the detection of SYN flood attack.

When we evaluate the classification, models based on their RMSE (error rate) for the detection of SYN flood attack. RMSE presents the difference between actual and desired outputs based on confusion matrix. The model which has lower RMSE is more efficient than a model having a larger RMSE. Accordingly, the J48 and AdaBoost classifier models have the minimum error rate for SYN flood attack with result 0.12 of RMSE. Whereas ANN has the highest error rate of 94.85% RMSE. Naïve Bayes model has a moderate error rate result in 0.20 than ANN model. The graphical representation as shown in Figure 5.1.2

When evaluate the classification, models based on their time taken for the detection of SYN flood attack. The small-time is taken to build and evaluate indicates the ability of a model to detect SYN flood attack while the higher time taken presents the weakness of a model. Figure 5.1.3 shown that the time took to build the classification model for the SYN flood attack. Hence, the Naïve Bayes classification model archives best performance. AdaBoost and J48 classifiers achieve moderate time took to build and evaluate models. Whereas the ANN classification model has poor detection performance, FPR and time took to building and evaluation model for this specific task.

Model building and evaluation time are dependent on the type of machine used in the classification process, the size and type of used dataset, the internal working principle of selected models and the way of training mode and size. In this study as shown in Figure 5.1.4, experimental result all experiments were performed in the same laptop and have been using the same data size and training mode.

However, the Naïve Bayes classification algorithm archive better time taken in both building and evaluation. size of the data to be classified, the types of laptop used in the classification process number of folds used in cross-fold validation is also another factor. The time achieved by these algorithms in this research could have been different if the number of folds and the size of dataset is changed.

Finally, as per shown the results obtained from the above experiments, the J48 classification model has better performance with 10-fold cross-validation approach. The algorithm works very well as compared to the other three model and values of 98.57% accuracy and 2.4% of FPR which is the sign of a good ML model. The ROC and RMSE result of a model were very good. The time taken to build a model and evaluate was 1.40 seconds and 1.39 seconds respectively were better.

This thesis work is answering the research questions and meets the objectives stated in Chapter 1. As a reminder, the primary objective of this study was to compare the performance of four classification ML algorithms for detecting SYN flood attack.

"Which type of ML algorithm can be effective in order to detect SYN flood attacks?"

In order to answer this question, several experiments were performed in Chapter 4 Based on the experimental result for this specific type of attacks detection, J48 classification algorithm was the best as compared to the other three classification algorithm.

"Which traffic dataset features are relevant in order to detect SYN flood attacks?"

In order to answer this question, as shown in Chapter 4 in detail useful dataset features were identified manually and evaluated using IG, CFS and GR feature selection algorithm, derived and applied in the experiment. According to this, the relevant feature for this specific type of intrusion detection were duration, dst_add, flag_types, dst_types, count, srv_count, src_bytes, dst_bytes, packet_size.

"Which training/test approach is better in order to detect SYN flood attack?"

In order to answer this question, several experiments were performed in Chapter 5. Based on the experimental result for this specific type of attacks detection, 10-fold cross-validation was the best as compared to percentage split approaches.

In general, when comparing four classifier algorithms applied previously on the same dataset, by generated and captured from ethio telecom network. The J48 classier enhance the accuracy of SYN flood attack and improving in its false alarm rate. Likewise, the experiment result of AdaBoost model detection performance for SYN flood attack was slightly lower than the J48 classifier model which has the highest rate among the Naïve Bayes and ANN classification models.

6

CONCLUSION AND FUTURE WORK

The main objective of this thesis was to develop model for detection SYN flood attack using ML algorithms. To achieve this objective network traffic PCAP dataset by generated, captured and preprocessed. Models were built, evaluated, and those with better performance were recommend. This chapter discusses on the outcomes and findings of the research. According to these results derived a conclusion and provided recommendations.

6.1 CONCLUSION

The telecom service providers manage and operate the complex network infrastructures used for data and voice transmission, they communicate and store huge amounts of sensitive data. However, security issues have been one of the most serious problems for service providers as attackers also changed dynamically. The DDoS attacks have become a most serious threat for telecom service providers around the world. The main objective is to interrupt the server's availability and suspend the user's access to a telecom network.

The SYN flood attack is one of the most dangerous and easiest to perform a DDoS attack. The aims to make a server unavailable to legitimate traffic by consuming all available server resources and caused due to the drawback that comes with the "three-way handshake" of TCP connection sequence. DDoS attack behavior is not statics and their behavior will change frequently. Hence, it difficult to detect using traditional security detection mechanism. Such as anti-virus, anti-Spyware, firewalls, signature-based IDS and Virtual Private Network (VPN).

The main objective is to enhance the detection performance of SYN flood attack while reducing the number of false alarms. The dataset used in this study has been taken from ethio telecom network by capturing packet. Additionally, packet data was generated using ethio telecom network. This research work began by conducting of a literature review, which is necessary to understand significance of the problem, identifying its limitations of existing intrusion detection and state of the art designing of SYN flood attack.

In order to increasing the performance of SYN flood attack detection four well-known classification algorithms (Naïve Bayes, ANN, AdaBoost and J48) were selected based on recommendation of different literature studies in the area of intrusion detection. The models were implemented using WEKA data mining tool.

Following selection of ML algorithms and datasets collection, the dataset was preprocessed and made suitable for data mining experiments. Accordingly, 4 million generated and captured datasets were preprocessed to gets 41,383 normal traffic and 40,832 of SYN flood attack instances. The datasets contain a total of 20 attributes, in the first step 11 relevant attributes were selected manually. In the second step 9 feature were selected based on worthiness using three feature selection methods (IG, CFS and GR) in the WEKA data mining tool. The final the dataset was converted into suitable CSV format. In order to show the impact of training dataset size, we have taken the following training and testing options as followed.

The first option has been used trained and tested using the 10-fold cross validation testing model. The second option has been used 75% of the total dataset for our training and the remaining 25% for testing the model. The third option has been used 66% of the total dataset for training to build the model and remaining 34% would be used for testing of the model.

In this study all experiments were carried out by identical dataset with three training and testing approaches, then comparing with four classification machine learning algorithms. It gives a promising solution for detecting SYN flood attack and a valuable contribution to the field of network intrusion detection in telecom service providers. The following are findings of this study:

SYN flood attack detection using traditional approaches is not much efficient. These days machine learning based techniques for detection of SYN flood attack has received much attention.

In this research work have been used four machine learning algorithms for analysis, such as (AdaBoost, J48, ANN and Naïve Bayes). The evaluation results show that J48 algorithm performs with 98.57% of accuracy and AdaBoost, Naïve Bayes and ANN algorithms with 98.52%, 95.31% and 94.85% of accuracy respectively. The model with J48 classification algorithm achieves best in all evaluation performance metrics, except the time taken to build a classification model. AdaBoost comparatively fewer results as compared to Naïve Bayes and ANN for the detection of SYN flood attack. However, in the case of time taken to build a model, J48 is less than Naïve Bayes model and higher than the AdaBoost and ANN models.

Finally, deployment of machine learning detection mechanism for SYN flood attack is recommend for ethio telecom to improve its security defiance as the current is signature-based IDS. As a result, it would also help the company reduce maintenance cost, increase revenue and minimize human intervention.

6.2 RECOMMENDATION FOR FUTURE WORKS

The following areas are recommended for future work.

- More investigation needs to be done using other data mining techniques using hybrid form
- More investigations on combination of IDS with an IPS to raise the achievement of network security
- Further investigation extending the work using other protocols (ICMP, UDP, etc.) and service types.
- Involve using a larger volume of real-world network traffic dataset.

BIBLIOGRAPHY

[1] S. Sharif, "Telecommunication and its impact over the economic development of saarc countries," *Available at SSRN 2810057*, 2016.

[2] R. Vaidya, "Cyber security breaches survey 2019," 2019.

[3] D. Larson, "Distributed denial of service attacks–holding back the flood," *Network Security*, vol. 2016, no. 3, pp. 5–7, 2016.

[4] S. D. Kotey, E. T. Tchao, and J. D. Gadze, "On distributed denial of service current defense schemes," *Technologies*, vol. 7, no. 1, p. 19, 2019.

[5] K. P. Prakash, T. Nafis, and D. S. Biswas, "Preventive measures and incident response for locky ransomware," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, pp. 392–395, 2017.

[6] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (ddos) resilience in cloud: Review and conceptual cloud ddos mitigation framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147–165, 2016.

[7] H. Hindy, D. Brosset, E. Bayne, A. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, "A taxonomy and survey of intrusion detection system design techniques, network threats and datasets," *arXiv preprint arXiv:1806.03517*, 2018.

[8] R. Marwaha, "Intrusion detection system using data mining techniques–a review," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7, no. 5, 2017.

[9] F. SHAAR and A. EFE, "Ddos attacks and impacts on various cloud computing components," *Int. Journal of Information Security Science*, vol. 7, pp. 26–48, 2018.

[10] P. Dzurenda, Z. Martinasek, and L. Malina, "Network protection against ddos attacks," *International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems*, vol. 4, no. 1, pp. 8–14, 2015.

[11] R. Mohammadi, R. Javidan, and M. Conti, "Slicots: An sdn-based lightweight countermeasure for tcp syn flooding attacks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 487–497, 2017.

[12] A. Ganesan, P. Parameshwarappa, A. Peshave, Z. Chen, and T. Oates, "Extending signature-based intrusion detection systems withbayesian abductive reasoning," *arXiv preprint arXiv:1903.12101*, 2019.

[13] R. R. Zebari, S. R. Zeebaree, and K. Jacksi, "Impact analysis of http and syn flood ddos attacks on apache 2 and iis 10.0 web servers," in *2018 International Conference on Advanced Science and Engineering (ICOASE)*, IEEE, 2018, pp. 156–161.

[14] A. Parashar and P. Kakkar, "Defense against syn flooding attacks based on swarm intelligent ant colony optimization,"

[15] K. Hussain, S. J. Hussain, V. Dillshad, M. Nafees, and M. A. Azeem, "An adaptive syn flooding attack mitigation in ddos environment," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 7, p. 27, 2016.

[16] M. Bogdanoski, A. Toshevski, D. Bogatinov, and M. Bogdanoski, "A novel approach for mitigating the effects of the tcp syn flood ddos attacks," *World Journal of Modelling and Simulation*, vol. 12, no. 3, pp. 217–230, 2016.

[17] S. Ghanti and G. Naik, "Defense techniques of syn flood attack characterization and comparisons," 2018.

[18] J. Villing, "Investigating tcp syn flood mitigation techniques in the wild," *Network*, vol. 67, 2019.

[19] B. S. Kumar, T Ch, R. S. P. Raju, M Ratnakar, S. D. Baba, and N Sudhakar, "Intrusion detection system-types and prevention," 2013.

[20] N. Hubballi and V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: A survey," *Computer Communications*, vol. 49, pp. 1–17, 2014.

[21] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[22] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences*, vol. 9, no. 20, p. 4396, 2019.

[23] M. V. Pawar and J Anuradha, "Network security and types of attacks in network," *Procedia Computer Science*, vol. 48, pp. 503–506, 2015.

[24] S. Samonas and D. Coss, "The cia strikes back: Redefining confidentiality, integrity and availability in security.," *Journal of Information System Security*, vol. 10, no. 3, 2014.

[25] M. Malik and Y. Singh, "A review: Dos and ddos attacks," *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 6, pp. 260–265, 2015.

[26] K. M. Prasad, A. R. M. Reddy, and K. V. Rao, "Dos and ddos attacks: Defense, detection and traceback mechanisms-a survey," *Global Journal of Computer Science and Technology*, 2014.

[27] S. Alzahrani and L. Hong, "Generation of ddos attack dataset for effective ids development and evaluation," *Journal of Information Security*, vol. 9, no. 04, p. 225, 2018.

[28] A. Sharma, D. Batra, R. Pandey, P. Narwal, and V. Kumar, "Distributed denial of service attack and its countermeasures.," *IJWA*, vol. 10, no. 3, pp. 91–99, 2018.

[29] H. Harshita, "Detection and prevention of icmp flood ddos attack," *International Journal of New Technology and Research*, vol. 3, no. 3, 2017.

[30] ——, "Detection and prevention of icmp flood ddos attack," *International Journal of New Technology and Research*, vol. 3, no. 3, 2017.

[31] A. Bijalwan, M. Wazid, E. S. Pilli, and R. C. Joshi, "Forensics of random-udp flooding attacks," *Journal of Networks*, vol. 10, no. 5, p. 287, 2015.

[32] M. J. H. Mughal, "Data mining: Web data mining techniques, tools and algorithms: An overview," *Information Retrieval*, vol. 9, no. 6, 2018.

[33] A. Dey, "Machine learning algorithms: A review," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, pp. 1174–1179, 2016.

[34] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, 2016, pp. 1310–1315.

[35] A. J. Wyner, M. Olson, J. Bleich, and D. Mease, "Explaining the success of adaboost and random forests as interpolating classifiers," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 1558–1590, 2017.

[36] N. A. Zaidi, J. Cerquides, M. J. Carman, and G. I. Webb, "Alleviating naive bayes attribute independence assumption by attribute weighting," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1947–1988, 2013.

[37] U. Bashir and M. Chachoo, "Performance evaluation of j48 and bayes algorithms for intrusion detection system," *Int. J. Netw. Secur. Its Appl*, 2017.

[38] M. Alkasassbeh and S. Al-Daleen, "Classification of malware based on file content and characteristics," *arXiv preprint arXiv:1810.07252*, 2018.

[39] M. A. Ahmadi, M. Ebadi, A. Shokrollahi, and S. M. J. Majidi, "Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir," *Applied Soft Computing*, vol. 13, no. 2, pp. 1085–1098, 2013.

[40] K. M. Almhdi, P. Valigi, V. Gulbinas, R. Westphal, and R. Reuter, "Classification with artificial neural networks and support vector machines: Application to oil fluorescence spectra," *EARSeL eProceedings*, vol. 6, no. 2, pp. 115–129, 2007.

[41] M. Usama, J. Qadir, A. Raza, H. Arif, K.-L. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *IEEE Access*, vol. 7, pp. 65 579–65 615, 2019.

[42] Y. Y. Aung and M. M. Min, "An analysis of k-means algorithm based network intrusion detection system," *Advances in Science. Technology and Engineering Systems Journal*, vol. 3, no. 1, pp. 496–501, 2018.

[43] A. Kak, "Expectation-maximization algorithm for clustering multidimensional numerical data," in *An RVL Tutorial Presentation at Purdue University*, 2014.

[44] P. V. Tran, "Semi-supervised learning with self-supervised networks," *arXiv preprint arXiv:1906.10343*, 2019.

[45] P. Čisar, S. M. Cisar, and I. Fürstner, "Security assessment with kali linux," *Bánki Közlemények (Bánki Reports)*, vol. 1, no. 1, pp. 49–52, 2018.

[46] C. Gandhi, G. Suri, R. P. Golyan, P. Saxena, and B. K. Saxena, "Packet sniffer–a comparative study," *International Journal of Computer Networks and Communications Security*, vol. 2, no. 5, pp. 179–187, 2014.

[47] I. Russell and Z. Markov, "An introduction to the weka data mining system," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, ACM, 2017, pp. 742–742.

[48] C. P. Schultz and B. Perciaccante, *Kali Linux Cookbook*. Packt Publishing Ltd, 2017.

[49] B. Ratner, *Statistical and Machine-Learning Data Mining:: Techniques for Better Predictive Modeling and Analysis of Big Data*. Chapman and Hall/CRC, 2017.

[50] C Saranya and G Manikandan, "A study on normalization techniques for privacy preserving data mining," *International Journal of Engineering and Technology (IJET)*, vol. 5, no. 3, pp. 2701–2704, 2013.

[51] C. Shang, M. Li, S. Feng, Q. Jiang, and J. Fan, "Feature selection via maximizing global information gain for text classification," *Knowledge-Based Systems*, vol. 54, pp. 298–309, 2013.

[52] H.-s. Chae and S. H. Choi, "Feature selection for efficient intrusion detection using attribute ratio," *International Journal of Computers and Communications*, vol. 8, 2014.

[53] S. K. Biswas, "Intrusion detection using machine learning: A comparison study," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 19, pp. 101–114, 2018.

[54] G. Kumar, "Evaluation metrics for intrusion detection systems-a study," *Evaluation*, vol. 2, no. 11, 2014.