DETECTION OF DISTRIBUTED DENIAL-OF-SERVICE (DDoS) ATTACK WITH HYPERPARAMETER TUNING BASED ON MACHINE LEARNING APRROACH

CHOO YONG HAN

BACHELOR OF COMPUTER SCIENCE (COMPUTER SYSTEMS & NETWORKING) WITH HONOURS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : CHOO YONG HAN

Date of Birth

Title : DETECTION OF DISTRIBUTED DENIAL-OF-SERVICE (DDoS) ATTACK WITH HYPERPARAMETER TUNING BASED ON MACHINE LEARNING APPROACH

Academic Session : SEMESTER 1 ACADEMIC SESSION 2022/2023

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

_____
(Student's Signature)

_____
New IC/Passport Number

Date: 20/1/2023

_____
(Supervisor's Signature)

WAN NURULSAFAWATI BINTI WAN MANAN
LECTURER
FACULTY OF COMPUTING
COLLEGE OF COMPUTING & APPLIED SCIENCE
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG DARUL MAKMUR
TEL: 09-4241279, FAX: 09-4244666

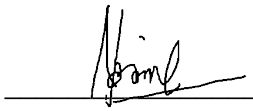
Puan Wan Nurulsafawati Binti Wan Manan

Date: 21/01/2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis/project* and in my opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Computer Systems & Networking) with Honours.



(Supervisor's Signature)

Full Name : Puan Wan Nurulsafawati Binti Wan Manan

Position : Lecturer

Date : 21/01/2023

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : CHOO YONG HAN

ID Number : CA19088

Date : 20 January 2023

DETECTION OF DISTRIBUTED DENIAL-OF-SERVICE (DDoS) ATTACK WITH HYPERPARAMETER TUNING BASED ON MACHINE LEARNING APPROACH

CHOO YONG HAN

Thesis submitted in fulfillment of the requirements
for the award of the degree of Bachelor of Computer Science
(Computer Systems & Networking) with Honours

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG

JANUARY 2023

ACKNOWLEDGEMENTS

First and foremost, I want to express my gratitude to God for giving me the opportunity and blessing of finishing my Final Year Project (FYP) on schedule.

My supervisor, Puan Wan Nurulsafawati Binti Wan Manan, has been a huge help to me during my FYP, thus I want to convey my sincere gratitude for her excitement, patience, insightful remarks and information and never-ending suggestions. My successful completion of the FYP is due to her vast knowledge, extensive experience and professional expertise in Data & Network Security. My FYP would not have been possible without her excellent help and direction.

Moreover, I would also like to express my gratitude to Dr Ku Muhammad Na'im Ku Khalif and Dr Chuan Zun Liang for providing the necessary knowledge and skills information regarding the FYP.

Besides, I would like to express my appreciation to my friends and colleagues who have always willingly to show their support and encouragement throughout the time of project period.

Last but not least, I want to extend my heartiest thank to my parents Choo Ching Leong and Chong Lai Yong for always being a source of moral support for me.

ABSTRAK

Serangan Penafian-Perkhidmatan Teragih merupakan salah satu serangan siber yang dilancarkan di seluruh dunia untuk menganggu trafik rangkaian seseorang sasaran dengan melancarkan banjir rangkain Internet untuk mengalahkan seseorang sasaran. Serangan Penafian-Perkhidmatan Teragih menjadi semakin kritikal kerana teknik Serangan Penafian-Perkhidmatan Teragih menjadi semakin canggih dari masa ke masa, kesukaran dalam membezakan trafik normal dan trafik serangan apabila trafik rangkaian menjadi berat kerana rangkaian trafik yang berat menyebabkan tugas untuk penapisan diganggu dan batasan yang dihadapi oleh teknik pembelajaran mesin yang menyebabkan kesilapan dalam klasifikasi serangan dengan baik. Lima teknik pembelajaran mesin dipilih iaitu DNN, KNN, SVM, NB dan DT untuk mengesan Serangan Penafian-Perkhidmatan Teragih dan mencadangkan teknik pembelajaran mesin yang terbaik dengan mengambil kira *accuracy, precision, recall, F1-Score, ROC-AUC Curve Area* dan *Confusion Matrix*. Satu set data standard, iaitu *DDoS Attack SDN Dataset* telah diaplikasikan untuk menjalani kajian ini. *EDA* dan *Data Preprocessing* dilaksanakan untuk menghasilkan satu set data yang bersih untuk mendapatkan keputusan prestasi pengesanan yang lebih tepat. Dalam kalangan lima model ini, DNN merupakan model yang terbaik kerana model ini menunjukkan 99.84% *accuracy,* 100.00% *precision,* 100.00% *recall,* 100.00% *F1-Score* dan 99.86% *ROC AUC Curve Area* untuk mengesan Serangan Penafian-Perkhidmatan Teragih.

ABSTRACT

Distributed Denial-of-Service (DDoS) attack is one of the common cyber threats that launched around the world to disrupt the traffic of a target by performing a flood of Internet traffic to overwhelm the target. DDoS attack becomes critical as it is hard to detect DDoS attack as becoming sophisticated from time to time in terms of attack techniques, hard in differentiating the normal traffic and attack traffic when the network traffic becomes heavy as filtering task will be disturbed during facing the heavy network traffic and limitations of machine learning techniques that cause misclassification. There are five selected machine learning techniques are identified such as DNN, KNN, SVM, NB and DT to detect the DDoS attack and proposed the best machine learning model in terms of accuracy, precision, recall, F1-Score, ROC-AUC Curve Area and Confusion Matrix. To conduct the study, a standard benchmark dataset DDoS Attack SDN Dataset is applied. EDA and Data Preprocessing are performed to ensure a clean dataset is produced for obtaining an accurate and meaningful detection performance results. Among the five models, DNN is the best model as it has shown 99.84% accuracy, 100.00% precision, 100.00% recall, 100.00% F1-Score and 99.86% ROC AUC Curve Area to detect DDoS attack.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

e	Euler's constant
|	Or
√a	Square root
%	Percentage

LIST OF ABBREVIATIONS

DoS	Denial-of-Service
DDoS	Distributed Denial-of-Service
Q4	Fourth Quarter
VNI	Visual Networking Index
TCP	Transmission Control Protocol
IP	Internet Protocol
SYN	Synchronize
ICMP	Internet Control Message Protocol
SVM	Support Vector Machine
DNN	Deep Neural Network
NB	Naive Bayes
DT	Decision Tree
KNN	K-Nearest Neighbor
KDD	Knowledge Discovery in Databases
DARPA	Defense Advanced Research Projects Agency
IDS	Intrusion Detection System
SDN	Software-Defined Networking
LR	Logistic Regression
UNB	University of New Brunswick
FP	False Positives
FN	False Negatives
RF	Random Forest
LR	Logistic Regression
SVC	Support Vector Classifier
ANN	Artificial Neural Network
EDA	Exploratory Data Analysis
FFNNS	Feed Forward Neural Networks
TP	True Positives
TN	True Negatives
PDF	Portable Document Format
HTML	Hypertext Markup Language
SOC	Security Operations Center
ROC	Receiver Operating Characteristic Curve
AUC	Area Under the ROC Curve

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Cyber security is a way to secure systems, networks and programs from digital threats or cyber-attacks. The cyber-attacks are becoming critical as they aimed at attempting malicious activities such as blackmail, erasing sensitive data, disrupting services and related malicious activities. Cyber-attacks become more advanced and critical than previous as the security risks and vulnerabilities will be leveraged to launch attack. Among the cyber-attacks, Distributed-Denial-of-Service attack or known as DDoS attack is one of the common cyber threats that launched around the world.

DDoS attack is a malicious attack to disrupt the traffic of a target by performing a flood of Internet traffic to overwhelm the target. These attack targets encompass the network bandwidth, system resources, servers and another resources. DDoS attack is a dangerous destructive network attack because it can cause the system unable to work properly and ruin the available network services and threatening the network. The network resources and services will be interrupted and jammed due to consumption of the network resources that caused by huge malicious data packets that generated by DDoS attack(Fan et al., 2022) . Consequently, the target will be compelled slacken off or completely shut down and denying its service to legitimate users or systems. DDoS attack can be performed from distributed and multiple or more than one device to flood the system. The targets to flood are the devices and protocols that connect to the network.

To launch the DDoS attack, firstly the attacker will recruit the botnet, which means an army of bots. The attackers will develop the sophisticated and specialized malware and attempt to spread the malware as many as possible to the target. The

malware can be spread through the compromised websites, electronic mail(e-mail) or organization's network. These infected devices will unintentionally become into a bot. After that, these devices will connect to the attacker's control circuit and ready to receive and accept the order from attacker's machine. The order includes the directions to launch a DDoS attack from bots to a target using chosen attack methods. The infected device will follow the instruction or order from attacker to launch a coordinated, well-timed distributed attack after the attacker send a message to their botnet's control server(Altomare F, 2021).

There are some reports show DDoS attacks are on the rise from year to year, which means it becomes critical. According to a report that conducted by Kaspersky, Kaspersky observed that there is a high spike in the number of DDoS attack from October to the end of December 2021, which is the fourth quarter(Q4) in 2021. Besides, based on Cisco Annual Internet Report (2018-2023) White Paper, it shows that the number of DDoS attack increases from year to year and Cisco estimate that the number of DDoS attack will double to 15.4 million by 2023, which is next year. Furthermore, according to Cisco Visual Networking Index (VNI), DDoS attacks will increase to 14.5 million by 2022, accounting for up to 25% of a country's total Internet traffic. These reports have shown how serious the DDoS attacks are and require more attention to solve such issue.

Figure 1. 1 *High spike in the number of DDoS attack in Q4 2021 that conducted by Kaspersky.*

Source: (Kaspersky, 2022)

DDoS attacks happen around the globe. For instance, a DDoS attack was unleashed on Ukrainian government sites and banking sector on 15 February 2022. This caused these targeted sites unable to work properly and paralyzed. A senior central bank official stated that the services at Ukraine bank PrivatBank has succeeded by hackers to disrupt. Besides, a DDoS attack was launched against Liberian telecom Lonestar and the attack was powerful that they knocked out the Internet connectivity across the country, causing Lonestar to lose millions of dollars(Zinets N, 2022).

To combat the DDoS attack, it is important to detect the DDoS attack because it is hard to be detected in detecting and differentiating the abnormality form traffic. Hence, researchers work hard to find the suitable techniques to detect the DDoS attack effectively and accurately to mitigate the attack so these abnormal malicious traffic packets blocked from reaching the destination and secure the system from affecting by DDoS attack. Besides, it is a demand to analyze and investigate the observed patterns at a granular level to reduce the false positive rates which means detecting DDoS attack inaccurately. To do so, machine learning techniques are explored and applied in earlier papers by adopting different techniques and algorithms. This is because machine learning can figure out the generalizable predictive patterns by applying data mining techniques to analyze the unknown patterns in a large volume of data and use these patterns to build a model to detect the anomalies in the future. Moreover, machine learning can minimize the detection error by applying their different learning approaches.

Hence, to overcome this issue, machine learning algorithm has been applied to detect the new incoming data (network traffic). The attributes or features of these data will be used in detecting the status of network class such as normal traffic or attack traffic. Machine learning techniques are powerful in making the detection because most DDoS attack have the same average packet size(Saini et al., 2020). In this study, the machine learning techniques will be explored and discussed to detect the DDoS attack in the network traffic.

1.2 PROBLEM STATEMENT

Firstly, it is hard to detect DDoS attack as becoming sophisticated from time to time in terms of techniques. Since DDoS attacks have evolved over time, these attacks shift from heavy-handed hardware overloads to application-layer attacks that can pass as genuine normal network traffic(Schakenbach J, 2013). It is difficult to detect as these attacks can pretend as normal traffic with Transmission Control Protocol (TCP) connections at Application Level and abide by the protocol rules(Srikanth K Ballal et al., 2018). Hence, it makes more difficult to detect the threat. Besides, nowadays, the attacker has been applied the sophisticated techniques that are hard to detect and cause this attack rise quickly

Next, the detection system is hard in differentiating the normal traffic and attack traffic when the network traffic becomes heavy. This is because the filtering task will be disturbed during facing the heavy network traffic. It will only work properly after the attack is ceased. Moreover, most detection mechanism showing the limited success in detection of attack as the attack will always use genuine requests to flood the target and making the detection system difficult to discern between legitimate traffic and DDoS attack traffic and big amount of data occurred and operated in the network and caused the system unable to detect accurately(Suresh & Anitha, 2011)

Furthermore, the existing machine learning techniques have the limitations that cause sometimes will cause misclassification of detection. The techniques have the limitations in terms of optimal feature selection that cause the detection become low accuracy and efficiency(Nadeem et al., 2022).

Hence, it is demanded to develop a DDoS detection system that able to detect accurately.

1.3 OBJECTIVE

- To study the performance metrics of different machine learning models on DDoS attack detection.
- To improve the detection performance metrics of DDoS attack.
- To determine the best machine learning technique in DDoS detection performance.

1.4 SCOPE

Research Scope:

1. A static dataset will be chosen and downloaded from public dataset website.
2. The dataset will undergo Exploratory Data Analysis (EDA) to discover the patterns of data.
3. The dataset will undergo data preprocessing to enhance the performance metrics and reliability of machine learning models.
4. The dataset will undergo data splitting where divided into 70:30 ratio where 70% of dataset is applied for training purpose and 30% of dataset is applied for testing purpose. The reason of choosing this ratio is it follows the Hold-out method. This ratio is generally 70:30 to be applied in Hold-out method and this ratio is well suited if wish to compare accuracy of different machine learning techniques and choose the best technique to be applied in this study(Ajitesh Kumar, 2022).
5. Several machine learning techniques will be determined and chosen.

Development Scope:

1. Python language is chosen in this study.
2. Training the machine learning models.
3. Evaluate their performance metrics of several machine learning techniques in terms of accuracy, precision, recall, F1-Score, ROC-AUC Curve and Confusion Matrix.
4. Performing hyperparameter tuning to improve the detection performance of every machine learning technique.

5. Evaluate their performance metrics again.
6. Compare the performance metrics before and after hyperparameter tuning.
7. Determine the best machine learning model in showing highest detection performance after hyperparameter tuning.

1.5 SIGNIFICANXCE OF PROJECT

i. Corporate company

The operation of corporate company will not be disrupted by DDoS attack especially dealing with online activities such as online conference, ordering and so on. The corporate company also can be prevented form lost productivity since the staffs can continue to perform their task without compelled to cease the tasks for temporarily and bring the inconvenience to complete the tasks before deadline. Besides, they can attract the new clients as well since no disruption preventing them to promote their services.

ii. Bank sector

Bank sector can be prevented from losing the profit by preventing to be suffered in DDoS attack by keeping the customers to trust the bank and not change to another bank options. Bank sector can operate their management and workflow as well through online. Bank sector also can prevent from suffering the brand damage that may cause customer to consider the security vulnerabilities of the bank to use the bank's service.

iii. Government website

Government website can prevent from suffering the brand damage that may give the bad reputation to public to consider the security vulnerabilities that exist in the government website. Furthermore, government website can spread and publish the important news and information to public without any disruption of DDoS attack. Government also can be prevented from suffering of losing the profit as the public will be unable to perform transaction for certain activities such as paying taxes, paying fines and so on if DDoS unleashed on the website.

1.6 REPORT ORGANIZATION

This report consists of five chapters.

Chapter 1 explains about the overview of project. This overview encompasses the Introduction, Problem Statements, Objective, Scope, Significance of Project and Report Organization.

Chapter 2 explains the literature review on existing different machine learning techniques that applied in DDoS attack detection.

Chapter 3 explains the methodology applied in this study. This methodology encompasses Research Framework, Project Requirement, Flowchart, Dataset description, Evidence of Early Work, Testing Plan and Potential Use of Proposed Solution.

Chapter 4 explains the implementation process and testing and result discussion.

Chapter 5 briefly summarize the whole study and figure out the research constraints and future work.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter covers the explanation on Distributed Denial-Of-Service (DDoS), Types of DDoS Attack, three previous research works description, summary of review pervious works and proposed work.

2.2 DISTRIBUTED DENIAL-OF-SERVICE(DDoS)

DDoS attack is a form of DoS (Denial-of-Service) malicious network attack to cease the system partially or completely by using a request flood using Internet or Intranet and collapse the network or server so the service is unavailable for the legitimate users. The first DDoS attack was happened in 1997 to shutdown the whole Internet access for a few hours in Vegas Strip during a hacker's conference event in Las Vegas by Khan C. Smith. After that, it led to online attack on some corporations after a release to a DDoS sample code during the event(Sagar Joshi, 2022) and make DDoS attack has been existed until now.

To launch DDoS attack, Figure 2.1 illustrates how DDoS attack works. Firstly, a malicious software will be spread by an attacker to the victim's computers via infected emails and attachments to create a network of infected machines to become zombie agents. After that, the agents are ready to send dummy malicious requests to the victim at the attacker's command. The attacker will send the command to command and control servers (CnC) and this CnC signals will transfer to the zombie to launch the DDoS attack. For instance, Mirai Botnet which comprised 380,000 bots(Beek, 2017)

applied by attacker to shut down the Internet access for about one million Deutshe Telekom customers for two days in 2016(Mercer C, 2017).

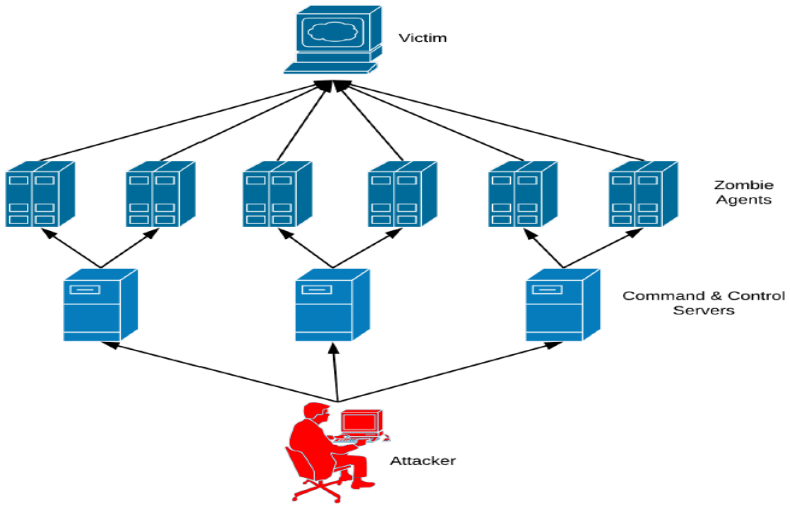


Figure 2. 1 *DDoS Attack Scenario*

Source: (Brooks & Özçelik, 2020)

Usually, the attackers are motivated by several aims. Firstly, they are affected by competitive benefits. Some organizations offer some rewards or benefits to such communities to stagger their rival's resources to hit the reputation of the rival because being a victim of DDoS attack indicates their weak security vulnerabilities. Besides, DDoS attack is launched to voice their opinion or stance out. Certain communities will leverage DDoS attack to show their stance on certain issue. Usually, this attack is focused on ethical dilemmas, protest an issue or online communities. For example, Telegram was announced they suffered DDoS attack with phony requests to disrupt the connection to Internet and Pavel Durov as one of the founders of Telegram claimed the attacking IP addresses were came from China and this attack had coincided with the protests the latest extradition bill issue in Hong Kong. Furthermore, it is used to obtain ransom. Some attackers plan to blackmail to compel the target organization to pay a ransom to them to stop the attack, otherwise they will continue to disrupt the service.

On the other hand, DDoS attack displays the growing menace by showing its high spiking in numbers of launching the DDoS attack from year to year as shown on

below figures. This is considered a major critical threat because it will cause lose revenue for service providers.

It is claimed that it will cost $30 million if the large e-commerce company face 24-hour connection.

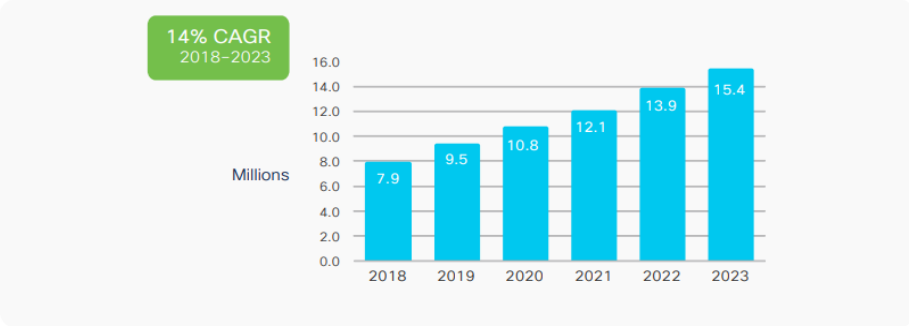


Figure 2. 2 *The number of DDoS attacks increase from year to year and will double to 15.4 million by 2023.*

Source: (Cisco, 2020)

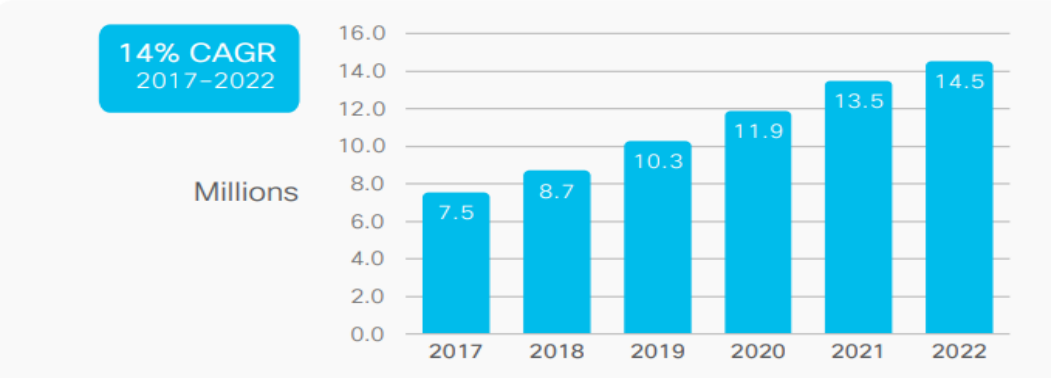


Figure 2. 3 *DDoS attacks increase to 14.5 million by 2022.*

Source: (Cisco, 2019)

2.3 TYPE OF DISTRIBUTED DENIAL-OF-SERVICE(DDoS) ATTACK

There are four types of attack as shown as below.

Table 2. 1: Four Types of DDoS Attack

NO	TYPE OF ATTACK	DESCRIPTION
1	Volumetric Attacks	• It is a network-layer attack to overload the available networking resources, bandwidth and services until unable to handle the increased traffic volumes by leveraging the flood of data packet. • Overloading the targeted service with high volumes of traffic congestion to cease the service within a few minutes. • Examples: TCP SYN attack, ICMP attack, Smurf attack.
2	Hit and Run Attacks	• It is specifically designed to avoid from detecting by slow-reacting DDoS defense solutions. • This attack will last for 20 until 60 minutes. This attack will exist again after another 12 until 48 hours after causing collateral damage to the target. • This attack will compel the anti-DDoS defense solution to be active all the time so it is easier to elude such preventions mechanism such as DNS rerouting and tunneling.

3	Browser-based Bot Attacks	• This bot is sneakily installed on a target system once visiting a malicious website and become active during a legitimate web browsing session. • Emulate a legitimate user browsing behavior to avoid the DDoS defense solutions. • Application layer is targeted to crash down the system with about 50 until 100 requests per seconds. • This attack is hard to be detected.
4	Shared Botnets	• Available public botnets are used either on rent or on sharing basis to launch attack. • It is not hard to be identified as applying advanced volumetric attacks using unique traffic patterns but tricky in elude the existing DDoS defense system.

Source: (Bhatia et al., 2018)

2.4 PREVIOUS RESEARCH WORK DESCRIPTION

2.4.1 Research 1: Detection of DDoS Attacks in Software Defined Networks

The authors(Karan B. V et al., 2019) choose to use Support Vector Machine (SVM) and Deep Neural Network (DNN) to create a trained model in Software-Defined Networking (SDN) environment based on their chosen dataset and compare and evaluate their performance in terms of accuracy, precision and recall detecting DDoS attack in SDN. The dataset they chosen was KDD Cup'99 dataset. This dataset is prepared by Stolfo et al. and generated based on DARPA'98 IDS evaluation dataset. It has been widely used for evaluation of anomaly detection techniques since 1999. It consists of about 4,900,000 data with 41 features and is labeled as either normal or an attack as shown in Table 2.3. However, they only chosen seven features which are **duration**, **src bytes**, **protocol type**, **count**, **dst bytes**, **srv count** and **service** after performing the data preprocessing and feature selection to ensure the efficient evaluation result. To verify these techniques, they evaluate in SDN environment by using Mininet emulator with Ryu controller. After conducting their study, they concluded that DNN is suitable for classification of traffic types either normal or attack as it scored better than SVM in terms of accuracy, precision and recall as shown as below:

Table 2. 2 Comparison of DNN and SVM for these three performance metrics

TECHNIQUES	ACCURACY (%)	PRECISION (%)	RECALL (%)
DNN	92.30	90.00	92.30
SVM	74.30	70.00	74.30

Based on Table 2.2, accuracy is meant by how correctly the techniques determine either positive class or negative class. They claimed that DNN scored better than SVM in terms of accuracy because DNN consists of its weights and number of hidden layers to give the great advantages to classify the features accurately especially when the dataset consists of many complex features compared to SVM just determine the new requests based on built hyperplane that consists of given values. Besides, precision is meant by how precise the technique in identifying the request as normal with low false positives rate. They claimed DNN can classify correctly almost every data as normal because DNN has internal layers that only exists in neural network and able to provide back propagation. Furthermore, recall is meant by how many normal instances are correctly classified with low false negative rate. DNN is claimed scored better in recall because it displayed low false negative compared to SVM.

Table 2. 3 KDD Cup'99 Dataset Feature Description

NO	**FEATURE NAME**	**DESCRIPTION**
1.	count	Number of connections to the same host using the ongoing connection within last two seconds
2.	destination bytes	Number of bytes sent from destination to source
3.	diff srv rate	Percentage of connections to various services
4.	dst host count	Number of connections which have the same destination hosts
5.	dst host diff srv rate	Percentage of various services on the ongoing host

6.	dst host rerror rate	Percentage of connections to ongoing host that consists of RST error
7.	dst host same src port rate	Percentage of connections to ongoing host that consists of same src port
8.	dst host same srv rate	Percentage of connections having the similar destination host and using the same service
9.	dst host serror rate	Percentage of connections to the ongoing host which have an S0 error
10.	dst host srv count	Number of connections having the similar destination host and using the similar service
11.	dst host srv diff host rate	Percentage of connections to the similar service from various hosts
12.	dst host srv rerror rate	Percentage of connections to the ongoing host and specified service that have an RST error
13.	dst host srv serror rate	Percentage of connections to the ongoing host and specified service that have an S0 error
14.	duration	Duration of the active connection
15.	flag	Condition of connection
16.	hot	Number of “hot”

17.	is guest login	Showing two outputs where: 1: Guest login 0: Not Guest Login
18.	is host login	Showing two outputs where: 1: Host login 0: Not Host Login
19.	land	Showing two outputs where: 1: Connection is from/to the similar host/port 0: Connection is not from/to the similar host/port
20.	logged in	Showing two outputs where: 1: Successfully logged in 0: Fail to log in
21.	num access files	Number of operations exist on the access control files
22.	num comprised	Number of compromised circumstances

23.	num failed logins	Number of fail to login the system
24.	num file creations	Number of file creation
25.	num outbound cmds	Number of outbound commands in an FTP session
26.	num root	Number of “root” accesses
27.	num shells	Number of shell prompts
28.	protocol type	Type of connection protocol
29.	rerror rate	Percentage of connection that showing “REJ” errors
30.	root shell	Showing two outputs where: 1: Root shell is obtained 0: Root shell is failed to obtain
31.	same srv rate	Percentage of connection to the similar service
32.	serror rate	Percentage of connection that showing “SYN” error
33.	service	Type of destination service
34.	src bytes	Number of bytes sent from source to destination

35.	srv count	Number of connections to the similar service as the ongoing connection within last two seconds
36.	srv diff host rate	Percentage of connections to various hosts
37.	srv rerror rate	Percentage of connections that display “REJ” errors
38.	srv serror rate	Percentage of connections that display “SYN” errors
39.	su attempted	Showing two outputs where: 1: try “su root” command 0: never try “su root” command
40.	urgent	Number of urgent packets
41.	wrong fragment	Number of wrong fragments

Source: (Kaskar et al., 2014)

2.4.2 Research 2: Detection of Distributed Denial of Service Attacks Based on Machine Learning Algorithms

The author(Rahman, 2020) used various machine learning techniques, which are Support Vector Machines (SVM), Decision Tree(DT) and Logistic Regression (LR) to perform the comparison of performance in detecting DDoS attack to detect DDoS attack in securing the servers. To conduct the study, the dataset of Canadian Institute of Cybersecurity as shown in Table 2.5 has been taken to be examined for DDoS attack detection. Canadian Institute of Cybersecurity is an innovative hub for research, industry collaboration and training in cybersecurity realm and located at University of New Brunswick (UNB) in Canada. This dataset encompasses many types of network and sessions (attack and non-attack phase) with two outcomes, which are benign and DDoS attack. This dataset is widely used to examine the effectiveness of machine learning based network intrusion detection model. Before proceeded to comparison of result with various techniques, the author has gone through Data Pre-procesing, Data Training and Dataset Testing to ensure obtain accurate and effective result. In the end, the author claimed that SVM scored great performance in terms of accuracy, precision, F1-score, Sensitivity, False Positive (FP) and False Negative (FN) among the techniques as shown below:

Table 2. 4 Comparison Table of Performance Metrics of Different Machine Learning Algorithms

TECHNIQUES	ACCURACY	PRECISION	F1-SCORE	SENSITIVITY	FP	FN
SVM	0.971	0.980	0.971	0.962	0.021	0.037
DT	0.827	0.865	0.833	0.803	0.159	0.196
LR	0.593	0.647	0.616	0.589	0.409	0.410

The author observed that the imperative part was SVM scored high detection accuracy of DDoS attack, which able to combat the DDoS attack well. However, the author suggested that additional research should be conducted since some work limitations exist.

Table 2. 5 Dataset of Canadian Institute of Cybersecurity Feature Description

NO	FEATURE NAME	DESCRIPTION
1.	destination	Number of bytes sent from destination
2.	flow duration	Duration of a flow (ms)
3.	total fwd pkts	Indicates the total packets in forward direction
4.	total bwd pkts	Indicates the total packets in backward direction
5.	total length of fwd pkts	Total size of packets in forward direction
6.	total length of bwd pkts	Total size of packets in backward direction
7.	initial window bytes fwd	Total number of bytes sent in initial window in forward direction
8.	initial window bytes bwd	Total number of bytes sent in initial window in backward direction
9.	label	Showing output whether benign or attack

Source: (CIC UNB, 2018)

2.4.3 Research 3: Automated DDOS Attack Detection in Software Defined Networking

The authors(Ahuja et al., 2021) work on DDoS detection in Software Defined Networking (SDN) by using machine learning techniques and identify the contributing features in DDoS detection. To figure out the best machine learning technique to detect DDoS attack well, the authors applied several machine learning techniques, which are Logistic Regression (LR), SVC, KNN, Random Forest (RF), Ensemble Classifier, ANN and SVC-RF to compare their detection performance. To conduct the study, they generated a dataset of DDoS attack in SDN environment in mininet emulator and posted in Mendeley Data entitled “DDoS attack SDN dataset”. The dataset encompasses of 1,04,345 data with 23 features. However, they chosen 17 features including output feature after performed data preprocessing. In the end, the authors proposed SVM-RF which is a hybrid model of SVM and RF model as it scores highest accuracy, precision, sensitivity and true positives because RF can filter the misclassification results performed by SVM. However, the authors suggested to apply deep learning models in the future as these models can provide encouraging detection performance.

Table 2. 6 Results of Performance Metrics of Research 3

TECHNIQUES	ACCURAC-Y (%)	DETECTIO-N RATE (%)	FAR	SPECIFICITY (%)	PRECISION (%)	F1-SCORE (%)
LR	83.69	82.46%	0.175	83.97	83.31	82.26
SVC	85.83	87.46%	0.125	84.04	85.79	86.61
KNN	95.22	94.37%	0.056	92.34	96.83	95.58
RF	97.20%	95.45%	0.045	94.56	96.56	96.23
ENSEMBLE CLASSIFIER	97.50%	96.43%	0.036	95.32	96.43	96.72
ANN	98.20%	97.84%	0.022	97.43	97.43	97.12
SVC-RF	98.80%	97.91%	0.02%	98.18	98.27	97.65

Table 2. 7 Dataset of DDoS Attack SDN Feature Description

NO	FEATURE NAME	DESCRIPTION
1.	src	Source IP
2.	dst	Destination IP
3.	pktcount	Number of Packets Counted
4.	bytecount	Number of Bytes Counted
5.	dur	Duration of a flow (in seconds)
6.	dur_nsec	Duration of a flow (in nanoseconds)
7.	tot_dur	Total Duration of a Flow (Sum of dur)
8.	flows	Number of packets per flow
9.	packetins	Number of Packet_Ins messages
10.	pktperflow	Packet count during a single flow
11.	byteperflow	Byte count during a single flow
12.	pairflow	Total flow entries in switch
13.	port_no	Port Number

14.	tx_bytes	Number of bytes transferred from switch port
15.	rx_bytes	Number of bytes received from switch port
16.	tot_kbps	Port Bandwidth (sum of tx_kbps and rx_kbps)
17.	label	Class label either benign or attack where: 0: Benign 1: DDoS Attack

Source: (Ahuja et al., 2021)

2.5 SUMMARY OF COMPARISIONS OF THREE PREVIOUS RESEARCH WORKS

Based on the review in Section 2.4, Table 2.8 shows a summary of the comparison of three previous research works. There are 12 elements used for comparison of three previous research works such as research and author(s), objective, proposed technique, best performance of technique in comparison, type of best performance of technique in comparison, data, number of original features from data, do author(s) pick some selected features, number of features that author(s) chosen, tool(s) to conduct, advantage(s) of the technique which shows best performance, disadvantage(s) of the technique which shows best performance and limitation(s) of the technique which shows best performance.

Table 2. 8 Comparison Table of Three Research Works

ELEMENTS	**RESEARCH 1**	**RESEARCH 2**	**RESEARCH 3**
Research and Author(s)	Detection of DDoS Attacks in Software Defined Networks (Karan B. V et al., 2019)	Detection of Distributed Denial of Service Attacks based on Machine Learning Algorithms (Rahman, 2020)	Automated DDOS Attack Detection in Software Defined Networking (Ahuja et al., 2021)
Objective	Use two different machine learning techniques to detect DDoS attack in SDN.	Use different machine learning approaches to detect DDoS attack in securing the servers.	Use different machine learning approaches to detect DDoS attack in SDN environment and identify the contributing

			features in DDoS detection.
Proposed Techniques	• Support Vector Machine (SVM) • Deep Neural Network (DNN)	• Support Vector Machines (SVM) • Decision Tree (DT) • Logistic Regression (LR)	• Logistic Regression (LR) • Support Vector Classifier (SVC) • K-Nearest Neighbor (KNN) • Random Forest (RF) • Ensemble Classifier • Artificial Neural Network (ANN) • Support Vector Classifier – Random Forest (SVC-RF)

Best Performance of Technique in Comparison	Deep Neural Network (DNN)	Support Vector Machines (SVM)	Support Vector Classifier – Random Forest (SVC-RF)
Type of Best Performance of Technique in Comparison	Supervised Learning	Supervised Learning	Supervised Learning
Data	KDD Cup'99 Dataset	Dataset of Canadian Institute of Cybersecurity	DDoS Attack SDN Dataset
Number of Original Features from Data	41	9	23
Do Author(s) Pick Some Selected Features?	Yes	No	Yes
Number of Features That Author(s) Chosen	7	-	17
Tool(s) to conduct	Mininet emulator with Ryu controller	Not mentioned	Ryu Controller

Advantage(s) of the technique which shows best performance	DNN 1. Able to show accurate result especially dealing with dataset with complex features. 2. Consists of internal layers that able to provide back propagation to improve the precision. 3. Its parallel and distributed algorithm allow to train faster.	SVM 1. Able to obtain effective classification result without require huge training data. 2. Capable in regularization because having L2 Regularization feature to prevent from over-fitting. 3. Effective in high-dimensional spaces especially dealing with a circumstance where the number of dimensions is more than number of samples.	SVM-RF 1. Useful in figuring out the separating hyperplane to classify the classes accurately 2. Able to process large dataset efficiently. 3. Good computational complexity to normalize the data.
Disadvantage(s) of the technique which shows best performance	DNN 1. Unable to provide accurate data when dealing with approximate statistics.	SVM 1. Not suitable for large data. 2. Unable to perform well when	SVM-RF 1. Unable to perform well when the dataset has noises.

	2.Increase computational cost.	the dataset has noises. 3. Require many memory to store whole support vectors in the memory and this cause to increases abruptly with training dataset size.	2. Time consuming to train large dataset. 3. Has possibility to overfit for data which easily predicts inaccurately.
Limitation(s) of the technique which shows best performance	DNN 1. Require advanced optimization techniques to get effective output.	SVM 1. Underperform if the number of features is greater than the number of samples. 2. Consumes more training time when dealing with large data.	SVM-RF 1. Difficult to choose a suitable kernel function.

2.6 SUMMARY OF REVIEW PREVIOUS RESEARCH WORKS

Based on Table 2.8 shows the comparison of three different previous research works.

In terms of dataset, it is found that researcher used different dataset. Research 1 use KDD Cup'99 Dataset which is a mostly and commonly applied dataset around globe for evaluation of intrusion as it consists of direct and derived available features. Research 2 use dataset of Canadian Institute of Cybersecurity. This is probably because of Research 2 prefer Canadian Institute of Cybersecurity offered many types of latest security dataset to ensure an effective evaluation work can be conducted by institution or organization compared to KDD Cup' 99 dataset which is just an only dataset for computer intrusion system detection among KDD Dataset. Research 3 use DDoS Attack SDN Dataset. This is because Research 3 is aimed to evaluate the accurate detection performance result over the latest DDoS dataset in SDN environment and figure out the imperative and contributing features which are useful in DDoS detection.

In terms of features selected and used, Research 1 and Research 3 do perform the feature selection. Research 1 chosen seven imperative features while Research 3 chosen 17 features. This is because feature selection can reduce the computational cost and improve the detection performance by removing the unnecessary features.

Moreover, it is found that different research proposes unique different machine learning approach. It is noticed that all research proposed to use several machine learning techniques to make comparison and choose the machine learning technique which shows the best performance in detection. For instance, Research 1 found that Deep Neural Network (DNN) shows the better performance than Support Vector Machine (SVM) while Research 2 found SVM shows the best performance among the techniques such as Decision Tree and Logistic Regression. Research 3 proposed that SVM-RF hybrid model is the best machine learning model in DDoS detection. However, it is noticed that all research works agreed and suggested to have an additional work or research in the future to get a detailed and accurate result in evaluation of DDoS detection in the future.

It is found that the technique which shows best performance of three research works use supervised machine learning algorithm, which are DNN, SVM and SVM-RF. Supervised machine learning algorithm is an algorithm or technique to train the algorithm to classify data or predict the result accurately using labelled dataset.

The techniques which show best performance has their advantages. For DNN, it can show accurate result especially dealing with dataset with complex features, consists of internal layers that able to provide back propagation to improve the precision and allow to train faster using its parallel and distributed algorithm. For SVM, it can obtain effective classification result without require huge training data, capable in regularization because having L2 Regularization feature to prevent from over-fitting and effective in high-dimensional spaces especially dealing with a circumstance where the number of dimensions is more than number of samples. For SVM-RF, it can figure out the separating hyperplane to classify the classes accurately, able to process large dataset efficiently and it provides a good computational complexity to normalize the data

However, every coin has two sides. These techniques have their disadvantages. For DNN, it is unable to provide accurate data when dealing with approximate statistics. For SVM, it is not suitable for large data because its training complexity is high, unable to perform well when the dataset has noises and require many memory to store whole support vectors in the memory and this cause to increases abruptly with training dataset size. For SVM-RF, it is unable to perform well when the dataset has noises. time consuming to train large dataset and has possibility to overfit for data which easily predicts inaccurately.

Besides, these techniques have their limitations. For DNN, it requires huge data sets with many features to train and requires advanced optimization techniques to get effective output. For SVM, it will underperform if the number of features is greater than the number of samples and require more training time when dealing with large data. For SVM-RF, it is difficult to choose a suitable kernel function.

2.7 PROPOSED WORK

In this study, the proposed work is applying several machine learning algorithms to compare their accuracy in DDoS detection and choose the technique which shows the best performance to verify its performance in system model. A static dataset will be used to verify their performance and train and test the model.

The dataset chosen is DDoS attack SDN Dataset which found from Mendeley Data which is a website to provide an open and free research data resource. This dataset is a SDN specific dataset that have been generated by Mininet emulator and can be applied for traffic classification using machine learning and deep learning algorithm. This dataset is prepared and contributed by Nisha Ahuja et al. from Bennett University in India in September 2020. Besides, this dataset consists of 1,04,345 data with 23 features. Its output will show numeric value, which 0 indicates benign traffic and 1 indicates malicious traffic. This dataset is chosen because it is a latest and up-to-date dataset from SDN network platform and it is believed that it can be used for evaluation of machine learning techniques with accurately.

In this study, several supervised learning algorithms will be proposed to use in classifying the types of traffic. DNN is proposed to use and another four machine learning technique will be used to compare their performance with DNN. They are KNN, SVM, Decision Tree (DT) and Naïve Bayes (NB). These 5 techniques are proposed as there are some journals state their benefits in detecting DDoS attack. For instance, DNN is proposed to use because this technique is believed to perform quickly with high accuracy(Cil et al., 2021), KNN is proposed to use as it has better classification results and detection performance(Alharbi et al., 2021), SVM is proposed to use as its kernel function enhances the processing of sample data with good accuracy(Ye et al., 2018), DT is proposed to use as it gives accurate results in terms of performance and accuracy(Fatima et al., 2018) and NB is proposed as it is faster to detect the DDoS attack(Aslam et al., 2022). Hyperparameter tuning will be conducted to improve their detection performance. Comparison between before hyperparameter tuning and after hyperparameter tuning will be discussed and analyze the detection performance of selected machine learning techniques. The machine learning technique which shows the best detection performance will be proposed.

To conduct this study, Jupyter Notebook will be used. This is because Jupyter Notebook is suitable to deal with the computational output and visualizations and it supports Python programming. Python programming will be used for importing and extracting data, data visualization, data preprocessing and comparison of accuracy of machine learning technique.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter covers Research Framework and Project Requirement, Dataset Description, Evidence of Early Work, Testing Plan and Potential Use of Proposed Solution.

3.2 RESEARCH FRAMEWORK

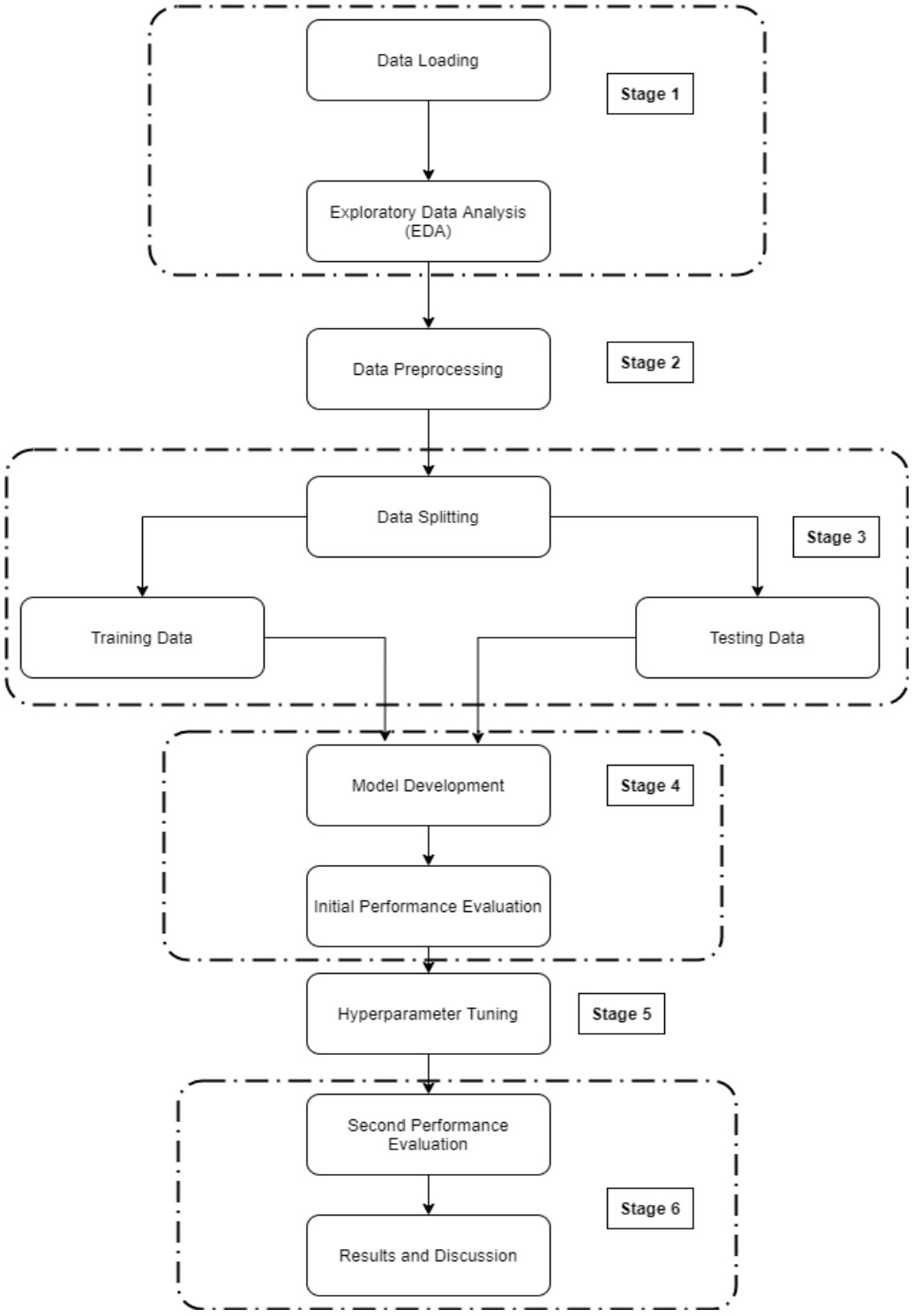


Figure 3. 1 *Research Framework*

There are six stages in this research framework.

3.2.1 Stage 1

3.2.1.1 Data Loading

Since DDoS detection is a long-term study, the datasets usually provided by the organization publicly such as in Mendeley Data and Kaggle for research and analysis purposes. The dataset can be found in the public dataset collection website and download the dataset after determining that such dataset is suitable to be used in study. The dataset will be loaded in Jupyter Notebook to conduct the study.

3.2.1.2 Exploratory Data Analaysis (EDA)

It is impossible or hard to understand the characteristics of data from just merely reading the whole larger dataset one by one by using spreadsheet software to open. Hence, EDA is applied to describe the characteristics of data by applying statistical and visualization form that is easily read and understand by user, especially the non-technical users. EDA is used to check the missing values and duplicate values and display the visualization form such as how many normal traffic and DDoS attack in bar chart.

3.2.2 Stage 2

3.2.2.1 DATA PREPROCESSING

Before ready to use the dataset for analyzing or evaluating the performance, it is required to ensure the data is clean so an effective and accurate result can be presented in front of users. Data Preprocessing offer many techniques such as Data Cleaning, Data Integration, Data Reduction and Data Transformation to produce a clean and usable dataset.

3.2.3 Stage 3

3.2.3.1 Data Splitting

It is ready to use the usable clean data. Hence, a data splitting will be performed to get training data and testing data as two different categories. To conduct data splitting, a suitable ratio is required to choose and justify producing a better accurate result in evaluation work such as 80:20, 70:30 and so on. In this framework, 70:30 ratio is applied. The 70% training data will be kept in model of machine learning techniques to analyze the data patterns and the 30% testing data will be used to test their performance.

3.2.4 Stage 4

3.2.4.1 Model Development

Several machine learning approaches will be selected. In this study, there are five approaches selected, which are Deep Neural Network (DNN), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT) and Naïve Bayes (NB). These approaches will be built and trained.

3.2.4.2 Initial Performance Evaluation

The first detection performance of every selected approach will be evaluated using different performance metrics, which are accuracy, precision, recall, F1-Score, ROC-AUC Curve and Confusion Matrix.

3.2.5 Stage 5

3.2.5.1 Hyperparameter Tuning

Hyperparameter tuning is a process to improve the detection performance of machine learning approaches by searching a set of optimal parameters from a list of parameters given. To perform hyperparameter tuning, GridSearch CV is chosen because

it is a suitable method to find out the suitable optimal parameters from every combination.

3.2.6 Stage 6

3.2.6.1 Second Performance Evaluation

The second performance of every selected approach will be evaluated using different performance metrics, which are accuracy, precision, recall, F1-Score, ROC-AUC Curve and Confusion Matrix after undergoing hyperparameter tuning. Both first and second evaluation results will be compared to verify the effectiveness of hyperparameter tuning.

3.2.6.2 Results and Discussion

The second performance evaluation will be taken as the final performance metrics of every approaches. Their detection performance will be discussed further and choose the best machine learning approach in DDoS detection.

3.3 PROJECT REQUIREMENT

3.3.1 Input

The input is a chosen static dataset, named DDoS attack SDN Dataset (Nisha Ahuja et al., 2020) which found from Mendeley Data. It is used because it is different from other datasets such as KDD Cup'99, NSL-KDD and CICIDS2017 dataset which created by applying traditional network. The input is in csv file. Since this study is required to use latest dataset that found from SDN network platforms, hence DDoS attack SDN Dataset is chosen. This dataset consists of 1,04,345 data with 23 features and output will only show value 0 or 1 which means 0 indicates normal traffic and 1 indicates malicious traffic. They generate this dataset by using Mininet emulator and make it to be accessed publicly in Mendeley Data website. This dataset consists of extracted feature which means the data is obtained directly from a platform and calculated feature which means the data has generated after performing some computational calculation. Below shows the table to explain the features of DDoS attack SDN Dataset. The details will be discussed in Chapter 4.

3.3.2 Output

There are six outputs will be provided which are:

(a) Accuracy

(b) Recall

(c) F1-Score

(d) ROC-AUC Area

(e) Confusion Matrix which encompasses True Positives, True Negatives, False Positives and False Negatives.

(f) A comparison table to compare the original correct output (actual output) and result of selected highest performance of machine learning (predicted output). Below shows the example of the table:

Table 3. 1 Example of Comparison Table

	Actual Output	Predicted Output
1150	1	0
22280	1	1
664	0	0
…	…	…

3.3.3 Process Description

Firstly, DDoS Attack SDN Dataset which found from Mendeley Data website will be downloaded as .csv file. After that, Jupyter Notebook will be used to import the dataset to perform Exploratory Data Analysis (EDA). After that, it will be undergoing data preprocessing which encompasses data cleaning, data transformation and data standardization in this study. After that, the data will be begun to split into 70:30 ratio where 70% is training data and 30% is testing data for model training and testing purpose. After that, an initial detection performance evaluation of every selected machine learning approach which are DNN, KNN, SVM, DT and NB will be recorded using different performance metrics such as accuracy, precision, recall, F1-Score, ROC-AUC Curve and Confusion Matrix. To improve their detection performance, a hyperparameter tuning process will be conducted to find out a set of optimal parameters. A second detection performance evaluation will be taken to verify the effectiveness of hyperparameter tuning. After that, the second detection performance evaluation result will be taken as final results and perform comparison among the machine learning technique to do the discussion and analysis. A best machine learning model which shows the highest detection performance result will be proposed. A detailed explanation will be given in **3.3.5 Flowchart**.

3.3.4 Constraints and Limitations

However, there are some constrains in this research. This study is just focused on several machine learning techniques. It becomes a constrain to claim which is the best technique from whole machine learning techniques as other techniques are not included and proposed in this study. Next, the implementation of DDoS detection algorithm in this study is unable to work in real-time platform since real-time platform have dynamic DDoS attack data, compared to this study which just use the static data for evaluation of performance. Thirdly, it is given limited time to evaluate and study all machine learning techniques since these techniques are vary. It is impossible to study, analyze and evaluate all the machine learning techniques in this study.

Besides, there are some limitations. Firstly, the proposed machine learning techniques is not always accurate sometimes since it may show different performance in different dataset. Secondly, since the dataset is static. the performance result that showed by different machine learning techniques might not be suitable to represent in real-time detection circumstance as it might produce different performance result that will cause confusion or ambiguous. Thirdly, the dataset is updated to 2020 only. Probably there is some new DDoS attack techniques have not covered and updated in the dataset. This might miss the opportunity to test the detection over the new DDoS attack techniques but it gives the message that this study should be conducted by time to time using updated dataset to ensure accurate detection can be maintained in securing the system resources.

3.3.5 Flowchart

The flowchart will be separated into 2 pages due to limitation of pages.

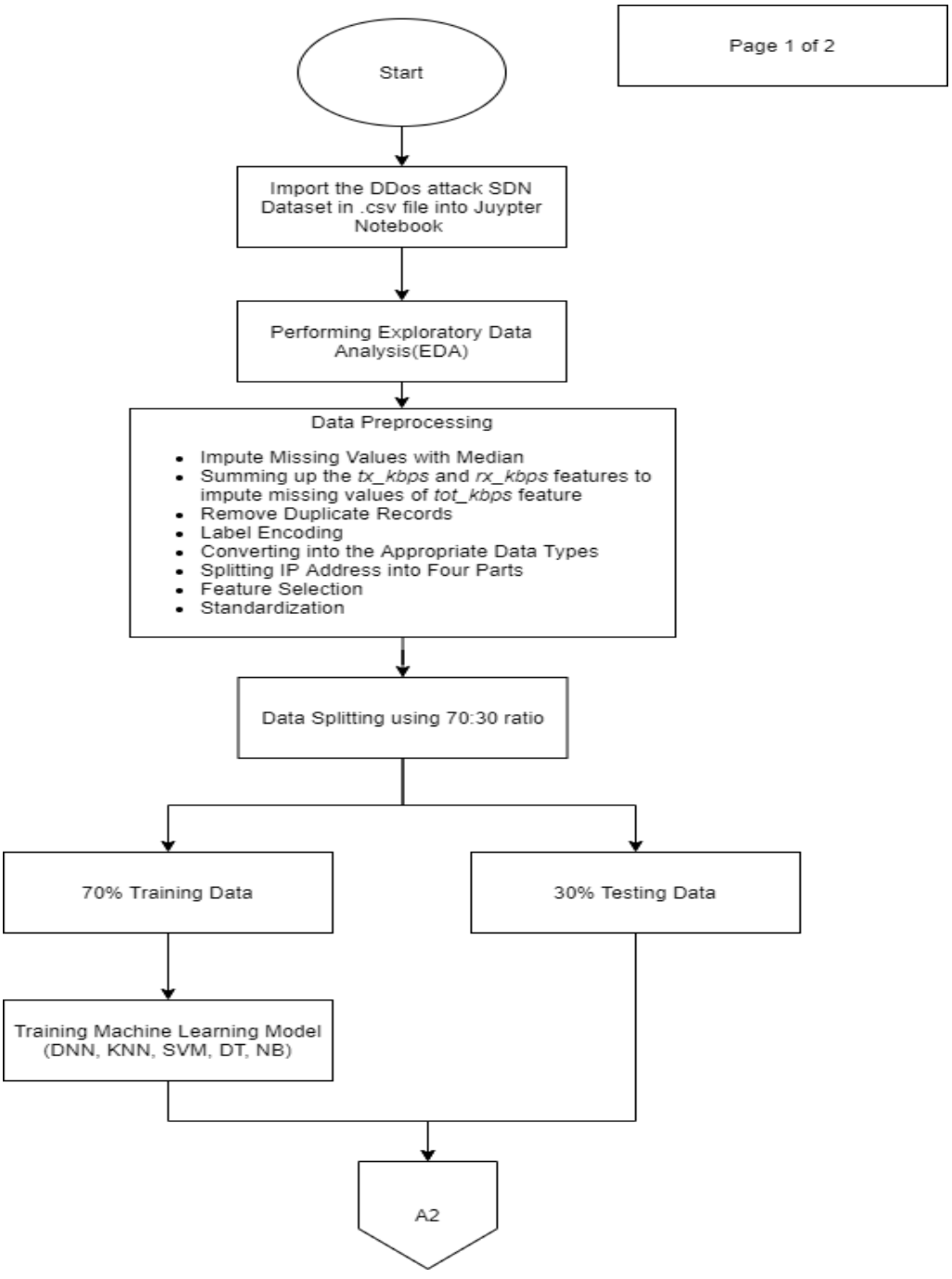


Figure 3. 2 *Flowchart of Process Description in Page 1*

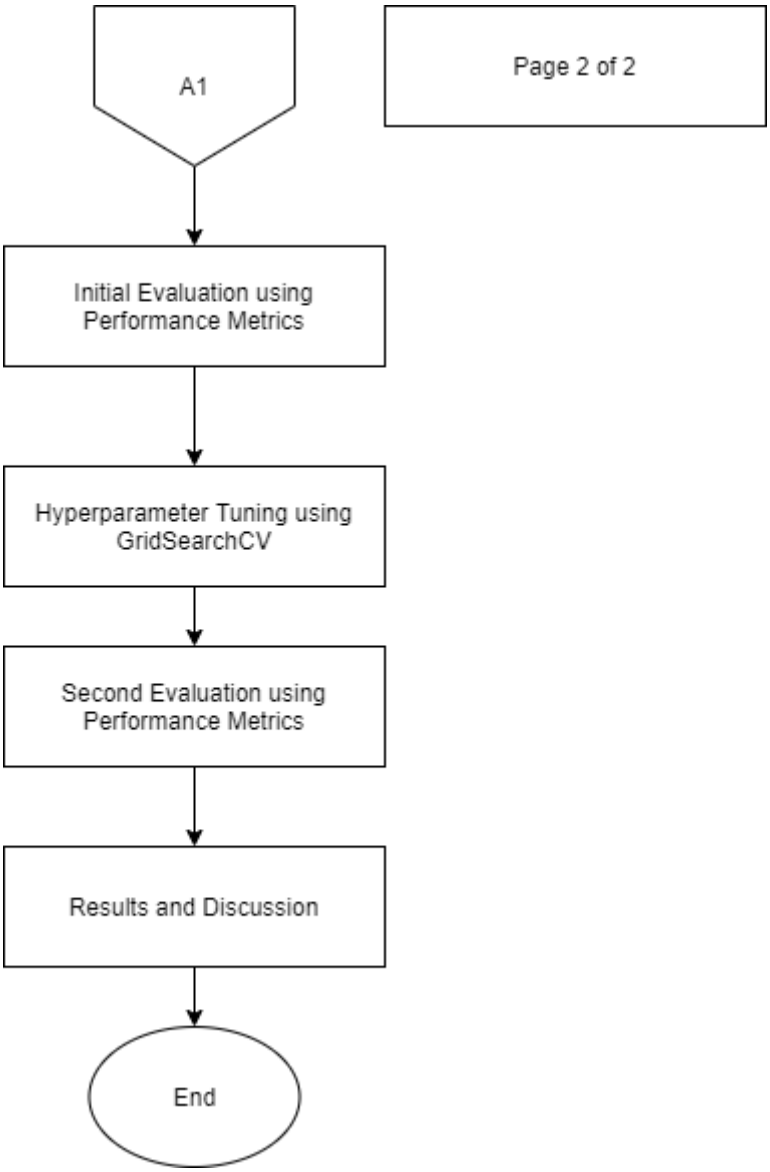


Figure 3. 3 *Flowchart of Process Description in Page 2*

Based on Figure 3.2 and Figure 3.3, since this study is evaluating the performance of proposed machine learning techniques and verifying the highest accuracy that showed by the selected technique by developing a DDoS system model, a DDoS dataset is required to find for training and testing purposes. The chosen DDoS dataset is DDoS Attack SDN Dataset which is an SDN specific DDoS attack dataset.

After that, download the dataset as .csv file to ensure Jupyter Notebook can open the dataset even in different programs or environments.

After that, activate the Jupyter Notebook and import the dataset in Jupyter Notebook. Now it is ready to extract the data. Python language is used in whole study.

It is required to understand the characteristics of this dataset, hence it will go through Exploratory Data Analysis (EDA). In this stage, the dataset will be described in statistical and statistical and visualization form such as visualizing the numbers of protocol in pie chart form, reading how many attributes are in a generated table and so on. EDA will be performed to check the existence of missing values and duplicate values and performing the visualization form such as bar chart to show how many normal traffic and DDoS attack in dataset.

After understanding the data, it will undergo data preprocessing. In this study, data preprocessing includes data cleaning, data transformation and data normalization. Firstly, data cleaning is conducted to treat the errors and multiple duplicate data to produce clean effective and productive data. Not only that, conversion to appropriate data types will be conducted and the IP addresses will be split into four parts. Secondly, data transformation will be conducted to convert the categorical values into numeric values to make the dataset well-organized and readable by machine learning techniques to improve and ensure the accuracy result. Through this part, Label Encoding will be applied to convert the categorical values into numerical values. Thirdly, data standardization will be performed to rescale the dataset where the mean is 0 and standard deviation is 1 to ensure all features have the same weights that no single bigger number variable will steer the machine learning algorithm performance and affect the performance result. After

undergoing the data preprocessing, it is believed that the data has been cleaned effective and is ready to be explored.

Next, it will be undergoing data splitting using 70:30 ratio where 70% is training data and 30% is testing data. 70% training data will be used to train the model and 30% testing data will be used to evaluate the detection performance of DDoS attack.

Several machine learning techniques will be chosen, which are DNN, KNN, SVM, DT and NB to build and train the models. An initial detection performance evaluation will be conducted to record and evaluate their detection performance. After that, hyperparameter tuning process will be conducted to improve their detection performance by selecting a set of optimal features.

Second detection performance evaluation will be taken again to evaluate the detection performance after undergoing hyperparameter tuning. A discussion will be conducted to discuss and analyze the results.

Finally, a best machine learning which shows the best detection performance will be proposed.

3.3.6 Software Equipment

Table 3.2 shows the description of software equipment.

Table 3. 2 Software Equipment

SOFTWARE	**VERSION**	**PURPOSE**
Microsoft Word 365	Version 2211	Used for research documentation.
Microsoft Power Point 365	Version 2211	Preparing slide for presentation.
Microsoft Excel 365	Version 2211	Open the .csv file and create graphs.
Google Chrome	Version 108.0.5359.125	Search and exploring the information in this research.
Draw .io	Version 14.5.1	Draw flowchart.
Anaconda	Version 1.10.0	Used to activate Jupyter Notebook in this research. Jupyter Notebook is used for EDA, Data Preprocessing, Data Splitting, Machine Learning Techniques and Performance Evaluation of Selected Techniques.

3.3.7 Hardware Equipment

The hardware used should be capable of Python programming and storing the large dataset.

Table 3. 3 Hardware Equipment

HARDWARE	**SPECIFICATION**	**PURPOSE**
Laptop	CPU Processor: Intel® Core™ i5-8265U CPU @ 1.60GHz 1.80GHz RAM: 8.00GB System type: 64-bit operating system, x64-based processor OS Build: 19044.2364 Edition: Windows 10 Home Single Language 21H2 Hard Drive: HP SSD S700 500GB	Used for research documentation, presentation and development.

3.3.8 Machine Learning Techniques

There are five machine learning techniques are selected, which are Deep Neural Network (DNN), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT) and Naïve Bayes (NB).

3.3.8.1 Deep Neural Network (DNN)

DNN is an Artificial Neural Network (ANN) that consists of multiple layers between input and output layers. It is a technique where the neurons obtain as input the neuron activation from previous layer and perform a simple computation(Montavon et al., 2018). DNN is a supervised learning as the data used to train is labelled and the output is known(Georgevici & Terblanche, 2019). It is known as DNN once the neural network has at least two hidden layers. Besides, it is also known as Feed Forward Neural Networks (FFNNS). The data will only be flowing in the forward direction in this network. This network can handle unstructured data, unlabeled data and non-linearity data. This network consists of hierarchical organization of neurons which mimic human brain. A layer consists of many neurons. The sigmoid activation function is applied for each hidden layers to assists the network learn the complex patterns in the data(Jain V, 2019). A gateway will be formed to pass the signal to the next connected neuron. Besides, weights initially assigned randomly and getting optimized when the network trained iteratively to ensure making a correct prediction. The signal will be passed by neurons to the next neurons depending on the input received and the output will be passed until reach to output layer to provides the prediction or detection if the signal value is greater than the threshold value else being ignored.

In DDoS detection, it is used in study to detect DDoS attack on the sample of packets obtained from network traffic. Some studies claim that DNN can perform well with high accuracy even in small dataset as DNN has feature extraction and classification processes in its network structure and consists of layers that will update itself within the training process(Cil et al., 2021).

There is a formula to calculate sigmoid activation function:

$$\text{sigmoid}\ (x) = \frac{1}{e^{-x}+1} \qquad 3.\ 1$$

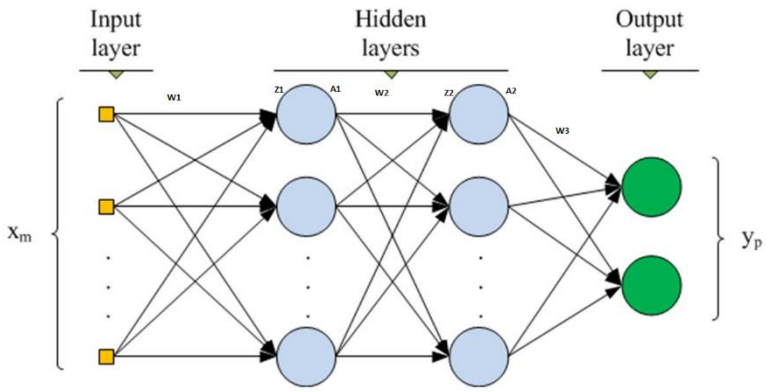


Figure 3. 4 *Structure of Deep Neural Network (DNN)*
Source: (Uday Paila, 2018)

3.3.8.2 K-Nearest Neighbor (KNN)

KNN is a supervised learning algorithm in regression and classification. KNN is non-parametric as it does not make any assumptions for underlying data assumptions. It is a distance-based algorithm and known as lazy algorithm because it does not learn during the training phase but learn during testing phase. It will categorize the data in various classes based on labelled training data. It just stores the data points for learning process during testing phase. To perform this KNN, A *k* value will be chosen. Usually, *k* value is odd number or using the formula where *k = sqrt(n)* to calculate. After that, the distance between the training points and testing points will be calculated and sort the computed distance in ascending order. The first *K* distances from the sorted list will be chosen and finally a mode or mean which associated with the distances will be taken. To calculate the distance, Euclidean Distance is a most common formula to be used(Danades et al., 2017):

$$\sqrt{(a_1 + b_1)^2 + (a_2 + b_2)^2 + \cdots + (a_n + b_n)^2}$$

Figure 3. 5 *Euclidean Distance Formula*
Source: (Danades et al., 2017)

In DDoS detection, KNN will figure out the *k* nearest neighbor of the traffic profile *S*, which is our dataset and utilize their classifications to vote for the label for *S*(Feng & Li, n.d.). The *k* nearest neighbors of incoming data is placed and these k

neighbors will indicate the classification of the incoming data such as benign or attack(Shieh et al., 2021).

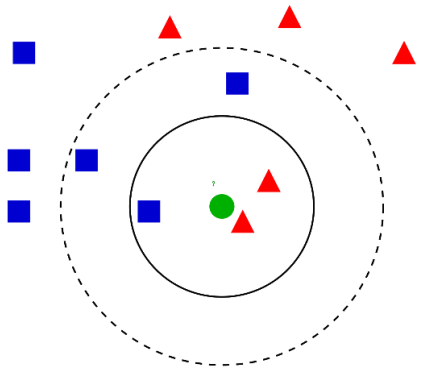


Figure 3. 6 *Structure of K-Nearest Neighbor (KNN)*

Source: (Jakub Adamczyk, 2020)

3.3.8.3 Support Vector Machine (SVM)

SVM is a supervised machine learning technique that used for classification, regression and outliers detection. Yet, it is commonly used in classification issues. SVM is a technique which the classification using hypothesis space in form of linear functions in a feature space high dimension, trained with learning algorithm using optimization method by leveraging the learning bias from statistical learning theory(Danades et al., 2017).

In this technique, the data item is plotted as a point in n-dimensional space, where n refers to a number of features in dataset. Next, the data points will be taken to output the hyperplane which separates the two classes of data points. Hyperplanes will classify the data point because the data points falling on either side of hyperplane will lead to different classes.

In DDoS detection, SVM will utilize hyperplane to classify the traffic such as normal or attack to identify the attacks flows rapidly and precisely especially when attacking traffic is hidden among the huge volume of normal flows(Cheng et al., 2009).

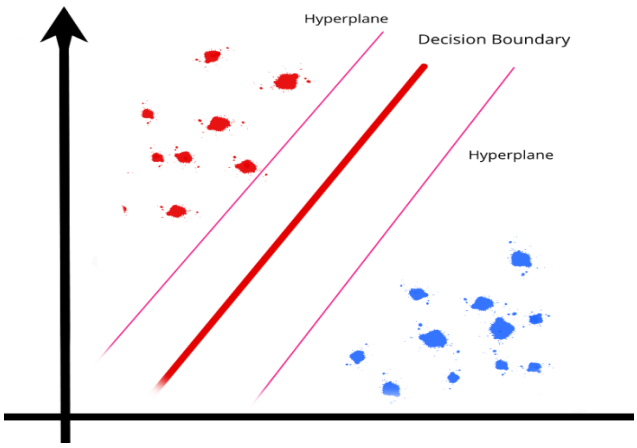


Figure 3. 7 *Structure of Support Vector Machine (SVM)*

Source: (Satyam Mishra, 2020)

3.3.8.4 Decision Tree (DT)

DT is a non-parametric supervised learning technique that used for classification and regression. DT is a flowchart-like structure in which every internal node denotes to a test on a feature, every leaf node denotes a class label, branches denote conjunctions of features that bring to the class labels and the paths from root to lead denotes classification rules. It begins with a root node and ends with a decision made by leaves.

In DT, it will begin from the root node of the three and compare the values of root attribute with the dataset features and follows the branch and move to next node after referring to the comparison. For next node, the features value will be compared with another sub-nodes and move further. It will continue the process until it ceases at the leaf node of the tree. Besides, hyperplane is applied to divide the feature space into classification.

In DDoS detection, the construction of decision tree is based on training data meanwhile the classifier is based on new data. To perform the construction, the attribute with the largest gain ratio will be selected and a branch for each possible value will be generated for selected attribute. After that, the features of the training data will be divided into subsets and repeat until cease at leaves. Finally, the incoming traffic is classified within the process in DT(Wu et al., 2011).

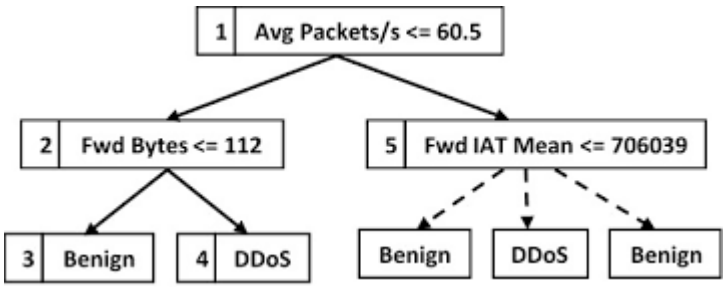


Figure 3. 8 *Structure of Decision Tree (DT)*

Source: (Lucky et al., 2020)

3.3.8.5 Naïve Bayes (NB)

NB is a supervised classification learning technique which is based on Bayes theorem which is a formula to calculate conditional probabilities:

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Figure 3. 9 *Bayes Theorem formula*

Source: (Jinde Shubham, 2018)

Referring to the above formula, B refers to evidence and A refers to hypothesis. The assumption made is that features are independent which does not affect the other features. Besides, the detection is quickly because it is a probabilistic model.

To perform NB, the dataset will be converted into a frequency table to create a Likelihood table by finding the probabilities. After that, Bayes Theorem will be used to calculate the posterior probability for each class. The outcome of prediction or detection is the class with the highest probability.

In DDoS detection, NB will classify DDoS attack using Performance Parameters. Firstly, NB will classify such variables(Bista et al., 2017):

Input(D): Dataset

C: Set of classes such as benign and attack

X: Data record to be classified

H: Hypothesis which X is classified into C

The probablity will be counted and used in Performance Metrics which are Accuracy(Ac), Detection Rate(DR) and False Positive Rate (FPR):

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

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

$$\text{FalsePositiveRate (FPR)} = (\text{FP}) / (\text{FP} + \text{TN})$$

Figure 3. 10 *Performance Metrics*

Source: (Bista et al., 2017)

where True Positive (TP) refers to number of malicious packets correctly classified as malicious, False Positive (FP) refers to number of normal traffic falsely classified a malicious, False Negative (FN) refers to malicious traffic classified as normal traffic and True Negative (TN) refers to number of benign packets correctly classified as benign(Fadlil et al., 2017).

3.4 DATASET DESCRIPTION

There are only 16 features including output label selected. However, there are 23 features after conducting data preprocessing.

Below table shows the description of every feature:

Table 3. 4 Processed DDoS attack SDN Dataset Feature Description

NO	FEATURE NAME	DESCRIPTION
1	pktcount	Number of Packets Counted
2	bytecount	Number of Bytes Counted
3	tot_dur	Total Duration of a Flow (Sum of dur)
4	flows	Number of packets per flow
5	packetins	Number of Packet_Ins messages
6	pktperflow	Packet count during a single flow
7	byteperflow	Byte count during a single flow
8	pktrate	Number of packets send per second
9	pairflow	Total flow entries in switch
10	protocol	Types of protocol (0 stands for UDP, 1 stand for TCP and 2 stands for ICMP)
11	port_no	Port Number
12	tx_bytes	Number of bytes transferred from switch port
13	rx_bytes	Number of bytes received from switch port
14	tot_kbps	Port Bandwidth (sum of tx_kbps and rx_kbps)
15	label	Class label either benign or attack
16	Source Oct1	First Octet/Part of Source IP Address
17	Source Oct2	Second Octet/Part of Source IP Address
18	Source Oct3	Third Octet/Part of Source IP Address

19	Source Oct4	Fourth Octet/Part of Source IP Address
20	Dst Oct1	First Octet/Part of Destination IP Address
21	Dst Oct2	Second Octet/Part of Destination IP Address
22	Dst Oct3	Third Octet/Part of Destination IP Address
23	Dst Oct4	Fourth Octet/Part of Destination IP Address

3.5 EVIDENCE OF EARLY WORK

A typical framework of DDoS detection using machine learning techniques is used in previous research works as shown below:

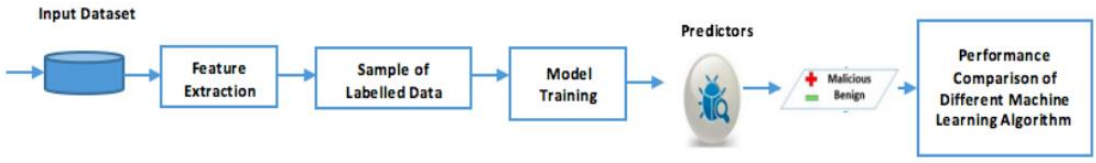


Figure 3. 11 *Typical Framework of DDoS Detection using Machine Learning Techniques*

Source: (Rahman, 2020)

This framework encompasses of several components that applied in early works. Below shows how the previous research works use this typical framework in conducting their study to detect DDoS attack.

Table 3. 5 Several Components in Typical Framework

Components	**Descriptions**
Feature selection	Reducing the number of input variables to improve the performance of model and reduce the computational cost when developing a predictive model.
Data Preprocessing	To produce a high-quality readable data where the data is understandable and consistent.
Data Analysis	It is used in determining the label of output for detection such as benign and attack.
Model Training	Machine learning algorithms will be trained by experienced through the DDoS dataset which consists of two classes where benign and attack.
Model Testing	A new dataset will be used to test the performance of a model in detecting DDoS attack. A detection result will be coming out whether benign or attack.
Model Evaluation	Different evaluation metrics such as precision, accuracy and so on will be applied to understand the machine learning model's performance.

Source: (Rahman, 2020)

3.5.1 Research 1: Detection of DDoS Attacks in Software Defined Networks

The authors(Karan B. V et al., 2019) chose to use Support Vector Machine (SVM) and Deep Neural Network (DNN) to create a trained model. The dataset they chosen was KDD Cup'99 dataset which consists of about 4,900,000 data with 41 features and is labeled as either normal or an attack. The authors compare these techniques in term of accuracy, precision, and recall. Below shows the results of comparison of DNN and SVM. The authors concluded that DNN is suitable for classification of traffic either normal or attack because it shows better result than SVM as shown on below table.

Table 3. 6 Results of Research 1

TECHNIQUES	**ACCURACY (%)**	**PRECISION (%)**	**RECALL (%)**
DNN	**92.30**	**90.00**	**92.30**
SVM	74.30	70.00	74.30

3.5.2 Research 2: Detection of Distributed Denial of Service Attacks Based on Machine Learning Algorithms

The author (Rahman, 2020) chose to use Support Vector Machine (SVM), Decision Tress (DT) and Logistic Regression (LR) to perform the comparison of performance in detecting DDoS attack. The dataset they chosen was Canadian Institute of Cybersecurity dataset which consists of 9 features. The authors compare these techniques in term of accuracy, precision, and F1-score, Sensitivity, False Positive (FP) and False Negative (FN). Below shows the results of comparison of SVM, DT and LR. The author claimed that SVM scored great performance in terms of accuracy, precision, F1-score, Sensitivity, False Positive (FP) and False Negative (FN) among the techniques as shown below.

Table 3. 7 Results of Research 2

TECHNIQUES	ACCURACY	PRECISION	F1-SCORE	SENSITIVITY	FP	FN
SVM	**0.971**	**0.980**	**0.971**	**0.962**	**0.021**	**0.037**
DT	0.827	0.865	0.833	0.803	0.159	0.196
LR	0.593	0.647	0.616	0.589	0.409	0.410

3.5.3 Research 3: Automated DDoS Attack Detection in Software Defined Networking

The authors(Ahuja et al., 2021) selected Logistic Regression (LR), Support Vector Classifier (SVC), K-Nearest Neighbor (KNN), Random Forest (RF), Ensemble Classifier, Artificial Neural Network (ANN) and Support Vector Classifier-Random Forest (SVM-RF). They used DDoS Attack SDN Dataset from Mendeley Data and selected 17 features including output feature. The authors evaluate the detection performance using accuracy, detection rate, False Acceptance Rate (FAR), specificity, precision and F1-Score. Below shows the detection performance results. The authors proposed SVM-RF hybrid model as it scores highest accuracy, detection rate, specificity, precision and F1-Score and lowest FAR.

Table 3. 8 Results of Research 3

TECHNIQ-UES	ACCURAC-Y (%)	DETECTIO-N RATE (%)	FAR	SPECIFIC-ITY (%)	PRECISI-ON (%)	F1-SCOR-E (%)
LR	83.69	82.46%	0.175	83.97	83.31	82.26
SVC	85.83	87.46%	0.125	84.04	85.79	86.61
KNN	95.22	94.37%	0.056	92.34	96.83	95.58
RF	97.20%	95.45%	0.045	94.56	96.56	96.23
ENSEMBL-E CLASSIFI-ER	97.50%	96.43%	0.036	95.32	96.43	96.72
ANN	98.20%	97.84%	0.022	97.43	97.43	97.12
SVC-RF	**98.80%**	**97.91%**	**0.02 %**	**98.18**	**98.27**	**97.65**

3.6 TESTING PLAN

To conduct the project, Jupyter Notebook is chosen. Jupyter Notebook is virtual lab notebook which is open-source, browser-based that used in workflows, code, data and visualizations. It is a human-readable software(Randles et al., 2017). Jupyter Notebook is useful in data preprocessing, statistical modelling, training machine learning models and data visualizations. Jupyter Notebook brings convenience in conducting the study such as view the results of code-in-line without depending on other parts of code, supporting visualizations such as graphic and charts by installing and importing modules such as Matplotlib, Pandas, Seaborn and so on and provides an interactive computational environment for developing Python based Data Science applications. Besides, its output can be exported as a PDF or HTML file.

To conduct testing plan, firstly DDoS attack SDN Dataset will be downloaded as .csv file. After that, importing the csv dataset file into the Jupyter Notebook. Before begin the study, the related modules such as Tensorflow(Keras), Pandas, Matplotlib, Seaborn are installed and ready to be imported, otherwise it will shows error and unable to run the program. After that, the data will be splited into 70% training data to learn and predict the desired outcome and 30% testing data to predict the outcome of data. The initial performance metrics results of every machine learning technique will be recorded and undergoing hyperparameter tuning. After that, the second performance metrics will be evaluated and do results and discussion. The best machine learning approach in DDoS detection will be chosen. These will be conducted by using Python language.

Figure 3. 12 *Jupyter Notebook Logo*

Source: (QuantStack, 2017)

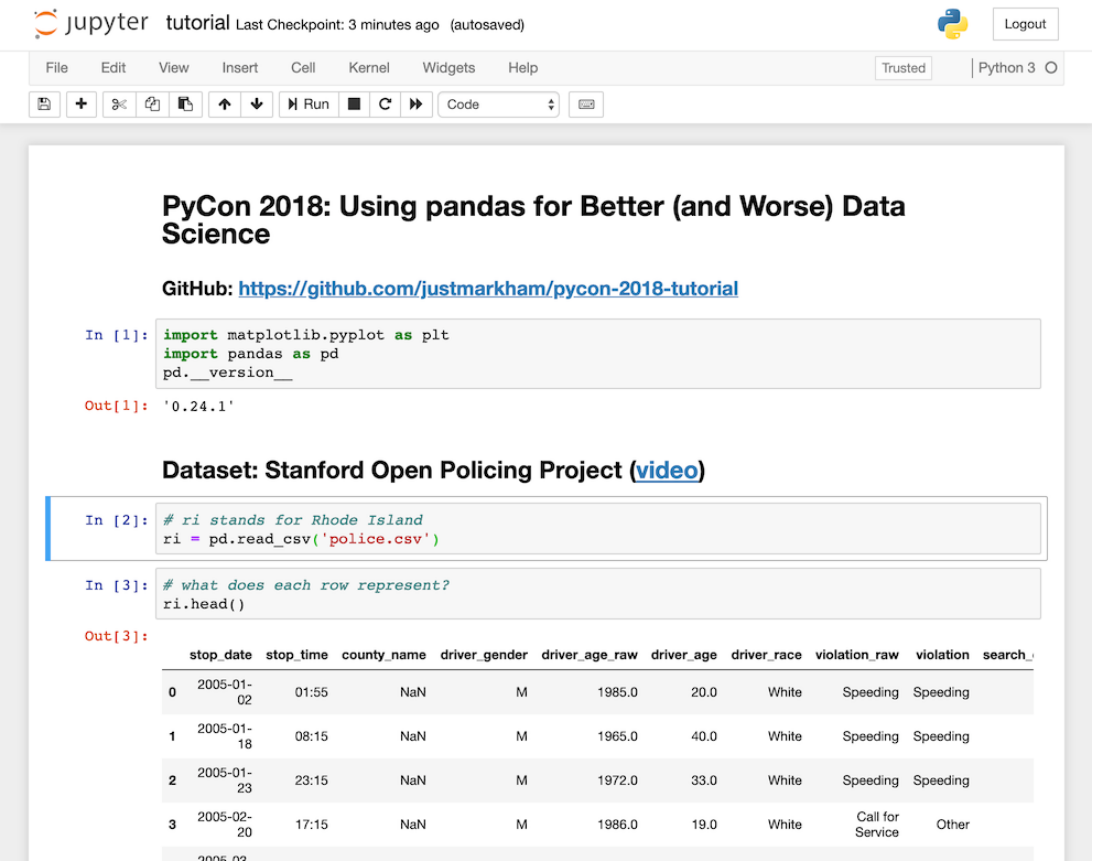


Figure 3. 13 *Jupyter Notebook Interface*

Source: (Kevin Markham, 2019)

3.7 POTENTIAL USE OF PROPOSED SOLUTION

3.7.1 Intrusion Detection System (IDS)

IDS is a monitoring system to detect suspicious activities and triggers alerts when the suspicious activities are detected. IDS can be used to perform scanning process to detect the DDoS pattern by analyzing the data collected from a local host machine or from network(Saghezchi et al., 2022). The machine learning techniques that deployed into IDS will assist IDS to detect the traffic anomaly and issues an alert if DDoS attack is detected. This is especially useful to Security Operations Center (SOC) to monitor and alert with the cyber attack.

3.7.2 DDoS Protection Tool

This tool can be used to detect and mitigate the web and infrastructure DDoS attack such as Cloudflare. This tool will leverage the machine learning techniques to detect DDoS attack and online monitoring on network traffic. For instance, Cloudflare claims that they deploy machine learning to detect every hit performed on the website with a big in-memory pattern database to decide the user should be accessed or not in combating DDoS attack. Once the DDoS attack is detected, it will issue an alert to be triggered and block the network visitors with abnormal high request rates.

3.7.3 Detecting Zero-Day DDoS Attack

Zero-Day refers to a cyber-attack targeting a software vulnerability which is unknown to the software vendor or to antivirus vendor. Sometimes, there are new attack exists or generated by DDoS attack but the protection tool unable to detect due to have not received the attack information. The attackers will leverage this opportunity to explore vulnerabilities that have not been noticed by the developer team. Hence, by leveraging machine learning techniques, these techniques can address the detection to reduce the complications and issues that associated with the new attacks(Hindy et al., 2020).

CHAPTER 4

IMPLEMENTATION, RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter covers Implementation Process and Testing and Result Discussion.

4.2 IMPLEMENTATION PROCESS

To conduct this study, a dataset named DDOS attack SDN dataset consists of 1,04,345 rows and 23 columns is used for DDoS detection using machine learning techniques which are DNN, SVM, KNN, Naïve Bayes and Decision Tree. Before begin classification process, several processes will be conducted to ensure an accurate result is produced.

4.2.1 Exploratory Data Analysis (EDA)

EDA implemented in this research encompasses of checking the existence of null values and duplicate values, check the data types of each attribute and determine the correlation of output label with features. Below shows the flowchart of EDA implemented.

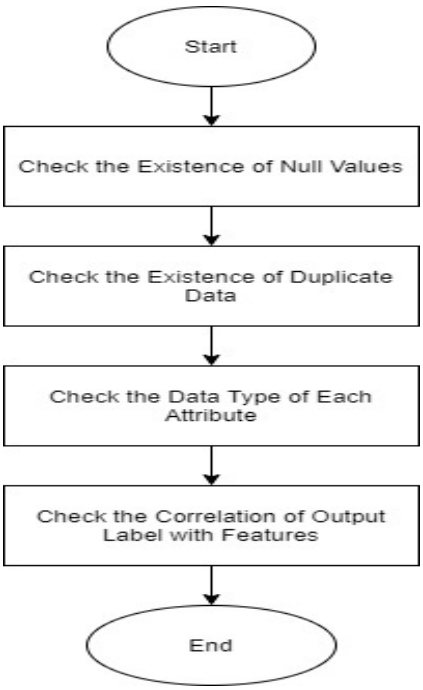


Figure 4. 1 *Flowchart of EDA implementation*

4.2.1.1 Check the Existence of Null Values

Null values are the blank values and will affect the performance to DDoS detection of machine learning models. In this section, there are 506 null values in *rx_kbps* and *tot_kbps* features. A treatment such as imputation of missing value will be conducted later.

```
In [7]: ddos.isnull().sum()
Out[7]: dt                0
        switch            0
        src               0
        dst               0
        pktcount          0
        bytecount         0
        dur               0
        dur_nsec          0
        tot_dur           0
        flows             0
        packetins         0
        pktperflow        0
        byteperflow       0
        pktrate           0
        Pairflow          0
        Protocol          0
        port_no           0
        tx_bytes          0
        rx_bytes          0
        tx_kbps           0
        rx_kbps         506
        tot_kbps        506
        label             0
        dtype: int64
```

Figure 4. 2 *506 Missing Values Detected*

4.2.1.2 Check the Existence of Duplicated Data

Duplicate values are the value where appearing more than once and will affect the performance to become bias and inaccurate. In this section, it is found that there are 5091 duplicate records. Hence, it is necessary to remove them.

```
In [8]: ddos.duplicated().sum()
Out[8]: 5091
```

Figure 4. 3 *5091 duplicate records found*

4.2.1.3 Check the Data Type of Each Attribute

In this section, there are 23 features in this data. It is observed that there are three features showing incorrect data types as shown below. The data types of *tot_dur*, *rx_kbps*, *tot_kbps* features are incorrect, which are float64. This is because the *tot_dur* feature is the sum of *dur* and *dur_nsec* features, it is incorrect to get the sum in float data type from integers. Meanwhile, *rx_kbps*, *tot_kbps* should be an integer. Hence, a correct conversion of data types is needed to perform.

```
In [6]: ddos.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 104345 entries, 0 to 104344
Data columns (total 23 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   dt           104345 non-null  int64
 1   switch       104345 non-null  int64
 2   src          104345 non-null  object
 3   dst          104345 non-null  object
 4   pktcount     104345 non-null  int64
 5   bytecount    104345 non-null  int64
 6   dur          104345 non-null  int64
 7   dur_nsec     104345 non-null  int64
 8   tot_dur      104345 non-null  float64
 9   flows        104345 non-null  int64
 10  packetins    104345 non-null  int64
 11  pktperflow   104345 non-null  int64
 12  byteperflow  104345 non-null  int64
 13  pktrate      104345 non-null  int64
 14  Pairflow     104345 non-null  int64
 15  Protocol     104345 non-null  object
 16  port_no      104345 non-null  int64
 17  tx_bytes     104345 non-null  int64
 18  rx_bytes     104345 non-null  int64
 19  tx_kbps      104345 non-null  int64
 20  rx_kbps      103839 non-null  float64
 21  tot_kbps     103839 non-null  float64
 22  label        104345 non-null  int64
dtypes: float64(3), int64(17), object(3)
```

Figure 4. 4 *Incorrect data types*

4.2.1.4 Correlation of Output Label with Features

To understand the strength of relationship between numeric variables with output variable, which is *label*, a Pearson Correlation heatmap or matrix is used to visualize the strength of relationship. This heatmap will calculate Pearson Correlation Coefficient to measure the strength of relationship. The correlation ranges from -1 to 1, where 1 indicates negative correlation, 0 means no correlation and 1 refers to positive correlation. Below shows the extraction of strength of *label* feature with another numeric variables from Pearson Correlation heatmap.

Based on the Pearson Correlation matrix, it shows that *label* feature has a weak positive correlation with *pktcount*(0.41) and *bytecount*(0.28). Range of weak positive correlation is from 0 to 0.5. This indicates that these features will contribute to DDoS detection.

Figure 4. 5 *Correlation of label feature with another numeric variables*

4.2.2 Data Preprocessing

In this section, data preprocessing encompasses several methods which are handling null values and duplicate values, converting into correct data type, label encoding, splitting IP address into four parts, feature selection, preparing a new clean dataset and standardization. Below shows the flowchart of data preprocessing:

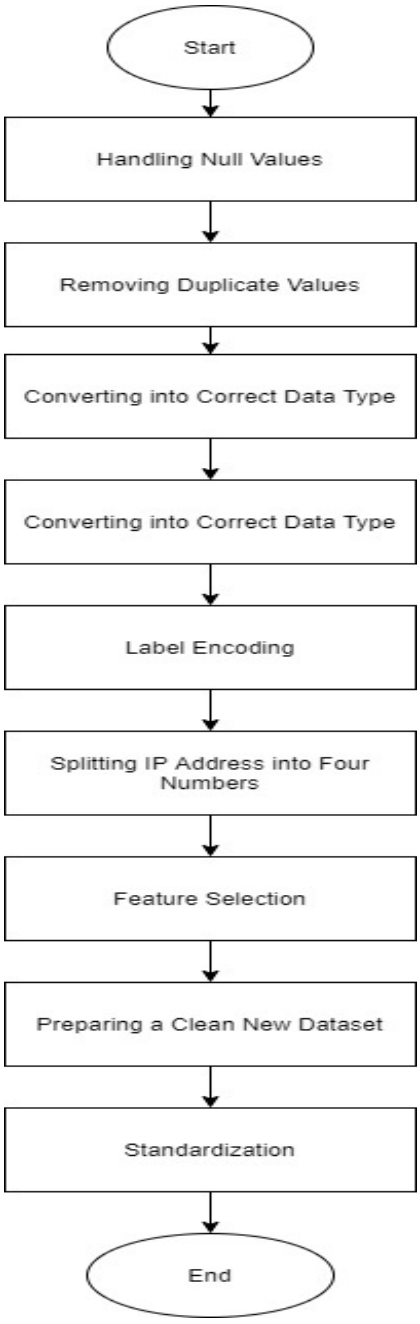


Figure 4. 6 *Flowchart of Data Preprocessing implementation*

4.2.2.1 Handling Null Values

Null values are unable to be processed by machine learning technique to read and identify. To deal with missing values of *rx_kbps* feature, imputation using their median of their features is implemented as shown below because this method can avoid hampering the dataset which consists of outliers that could affect the mean and standard deviation of data.

```
In [11]: ddos['rx_kbps'] = ddos['rx_kbps'].fillna(ddos['rx_kbps'].median())
```

Figure 4. 7 *Impute rx_kbps feature with median*

To deal with missing value of *tot_kbps* feature, tot_kbps is the sum of *tx_kbps* and *rx_kbps* features. Hence, *tx_kbps* and *rx_kbps* features are summed up to obtain the latest accurate data of *tot_kbps* feature.

```
In [12]: columns_names = ['tx_kbps', 'rx_kbps']

         ddos['tot_kbps'] = ddos[columns_names].sum(axis=1)
```

Figure 4. 8 *Summing Up tx_kbps and rx_kbps Features*

After treating the null values, it is found that there are no null values exist, especially on those two features as shown below.

```
In [13]: ddos.isnull().sum()
Out[13]: dt              0
         switch          0
         src             0
         dst             0
         pktcount        0
         bytecount       0
         dur             0
         dur_nsec        0
         tot_dur         0
         flows           0
         packetins       0
         pktperflow      0
         byteperflow     0
         pktrate         0
         Pairflow        0
         Protocol        0
         port_no         0
         tx_bytes        0
         rx_bytes        0
         tx_kbps         0
         rx_kbps         0
         tot_kbps        0
         label           0
         dtype: int64
```

Figure 4. 9 *No null values exist*

4.2.2.2 Removing duplicate values

Below code shows the process to remove the duplicate records. It will identify the repeated data and remove the related data. It is crucial to ensure the data quality is maintained.

```
In [15]: ddos.drop_duplicates(keep='first', inplace=True)
```

Figure 4. 10 *Removing duplicate records*

4.2.2.3 Converting into Correct Data Types

Incorrect data types will lead machine learning techniques to a poor detection performance since the incorrect data types are ambiguous to these techniques and affect their accuracy of detection performance. Below code shows the process to convert into an appropriate and correct data type. The data types of *tot_dur*, *rx_kbps*, *tot_kbps* features have been converted from float to integer. It shows that the conversion is successful.

```
In [18]: ddos['tot_dur'] = ddos['tot_dur'].astype(np.int64)

In [19]: ddos['rx_kbps'] = ddos['rx_kbps'].astype(int)

         ddos['tot_kbps'] = ddos['tot_kbps'].astype(int)
```

Figure 4. 11 *Conversion to Correct Data Type*

```
In [20]: ddos.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 99254 entries, 0 to 104344
Data columns (total 23 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   dt           99254 non-null  int64
 1   switch       99254 non-null  int64
 2   src          99254 non-null  object
 3   dst          99254 non-null  object
 4   pktcount     99254 non-null  int64
 5   bytecount    99254 non-null  int64
 6   dur          99254 non-null  int64
 7   dur_nsec     99254 non-null  int64
 8   tot_dur      99254 non-null  int64
 9   flows        99254 non-null  int64
 10  packetins    99254 non-null  int64
 11  pktperflow   99254 non-null  int64
 12  byteperflow  99254 non-null  int64
 13  pktrate      99254 non-null  int64
 14  Pairflow     99254 non-null  int64
 15  Protocol     99254 non-null  object
 16  port_no      99254 non-null  int64
 17  tx_bytes     99254 non-null  int64
 18  rx_bytes     99254 non-null  int64
 19  tx_kbps      99254 non-null  int64
 20  rx_kbps      99254 non-null  int32
 21  tot_kbps     99254 non-null  int32
 22  label        99254 non-null  int64
dtypes: int32(2), int64(18), object(3)
memory usage: 17.4+ MB
```

Figure 4. 12 *The conversion is successful*

4.2.2.4 Label Encoding

Since most of machine learning algorithms are unable to read categorical variables, a technique is required to convert these categorical variables into numerical variables or machine-readable format, which is label encoding. Label encoding is used to convert into numerical variables and ensuring to not increasing the dimension of dataset(Hasan et al., 2019). In this section, label encoding is used to convert *Protocol* feature into a numeric data type. Unique numeric value will be assigned to each data of this feature where '0' refers to UDP, '1' refers to TCP and '2' refers to ICMP. These values are assigned according to the descending of the features.

```
In [9]: prot = ddosclean.groupby(['Protocol'])['label'].size().sort_values().index
        prot
Out[9]: Index(['UDP', 'TCP', 'ICMP'], dtype='object', name='Protocol')

In [10]: dict1={key:index for index, key in enumerate(prot,0)}
         dict1
Out[10]: {'UDP': 0, 'TCP': 1, 'ICMP': 2}

In [11]: ddosclean['Protocol'] = ddosclean['Protocol'].map(dict1)
```

Figure 4. 13 *Performing Label Encoding*

Label Encoding is proven to be successful to be conducted after verifying the data as shown below.

pktperflow	byteperflow	pktrate	Pairflow	Protocol	port_no
13535	14428310	451	0	0	3
13531	14424046	451	0	0	4
13534	14427244	451	0	0	1
13534	14427244	451	0	0	2
13534	14427244	451	0	0	3

Figure 4. 14 *Protocol feature has become a numeric format*

4.2.2.5 Splitting IP Address into Four Numbers

Since IP Addresses is an object data format in the dataset, which indicates that machine learning models are unable to read the data. Hence, it is necessary to convert the IP addresses into a machine-readable data. Among techniques to handle IP addresses, splitting IP addresses into four numbers is an effective method to handle such data. This method can save computational costs and time to process the data and increase the accuracy of models in DDoS detection as it will not increase the dimensions of dataset to affect badly the result of detection(Shao, 2019). Below shows how I split the IP addresses into four numbers for *src* and *dst* features.

```
In [6]: ddosclean.loc[:, 'Src Oct1'] = ddosclean['src'].apply(lambda x: x.split(".")[0])
        ddosclean.loc[:, 'Src Oct2'] = ddosclean['src'].apply(lambda x: x.split(".")[1])
        ddosclean.loc[:, 'Src Oct3'] = ddosclean['src'].apply(lambda x: x.split(".")[2])
        ddosclean.loc[:, 'Src Oct4'] = ddosclean['src'].apply(lambda x: x.split(".")[3])
```

Figure 4. 15 *Split src feature into four numbers*

```
In [7]: ddosclean.loc[:, 'Dst Oct1'] = ddosclean['dst'].apply(lambda x: x.split(".")[0])
        ddosclean.loc[:, 'Dst Oct2'] = ddosclean['dst'].apply(lambda x: x.split(".")[1])
        ddosclean.loc[:, 'Dst Oct3'] = ddosclean['dst'].apply(lambda x: x.split(".")[2])
        ddosclean.loc[:, 'Dst Oct4'] = ddosclean['dst'].apply(lambda x: x.split(".")[3])
```

Figure 4. 16 *Split dst feature into four numbers*

Below shows the splitting of IP addresses for *src* and *dst* features has been successful.

Src Oct1	Src Oct2	Src Oct3	Src Oct4	Dst Oct1	Dst Oct2	Dst Oct3	Dst Oct4
10	0	0	1	10	0	0	8
10	0	0	1	10	0	0	8
10	0	0	2	10	0	0	8
10	0	0	2	10	0	0	8
10	0	0	2	10	0	0	8

Figure 4. 17 *Successfully to split IP addresses into four numbers*

4.2.2.6 Feature Selection

Feature Selection is an effective and useful strategy to prepare a clean data, Feature selection can improve the detection performance and save computational costs by only selecting the useful and necessary features. Selecting the necessary features will improve the accuracy of detection performance because the machine learning techniques could perform a well detection by identifying the significant necessary features. To perform this, unnecessary columns will be dropped after referring to some journals. Below shows the removed unnecessary columns which are *src*, *dst*, *dt*, *switch*, *dur*, *dur_nsec*, *tx_kbps* and *rx_kbps* to perform feature selection. *src* and *dst* features are being removed as they are unreadable by machine learning models and duplicate to the split data.

```
ddosclean.drop(columns=['src', 'dst'],axis=1, inplace=True )
```

Figure 4. 18 *Remove src and dst features*

```
In [6]: ddosclean.drop(columns=['dt','switch', 'dur', 'dur_nsec', 'tx_kbps', 'rx_kbps'],axis=1, inplace=True )
```

Figure 4. 19 *Remove Unnecessary Features*

This indicates the final dataset would have 99254 data with 23 features.

```
In [20]: ddosclean.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99254 entries, 0 to 99253
Data columns (total 23 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   pktcount     99254 non-null  int64
 1   bytecount    99254 non-null  int64
 2   tot_dur      99254 non-null  int64
 3   flows        99254 non-null  int64
 4   packetins    99254 non-null  int64
 5   pktperflow   99254 non-null  int64
 6   byteperflow  99254 non-null  int64
 7   pktrate      99254 non-null  int64
 8   Pairflow     99254 non-null  int64
 9   Protocol     99254 non-null  int64
 10  port_no      99254 non-null  int64
 11  tx_bytes     99254 non-null  int64
 12  rx_bytes     99254 non-null  int64
 13  tot_kbps     99254 non-null  int64
 14  label        99254 non-null  int64
 15  Src Oct1     99254 non-null  int64
 16  Src Oct2     99254 non-null  int64
 17  Src Oct3     99254 non-null  int64
 18  Src Oct4     99254 non-null  int64
 19  Dst Oct1     99254 non-null  int64
 20  Dst Oct2     99254 non-null  int64
 21  Dst Oct3     99254 non-null  int64
 22  Dst Oct4     99254 non-null  int64
dtypes: int64(23)
```

Figure 4. 20 *Latest Clean Data*

4.2.2.7 Preparing Clean New Dataset

After performing data preprocessing, it is believed that the latest dataset is clean and ready to be used for next stages in machine learning models. This latest dataset will be created automatically in .csv file named *ddosafter.csv*.

```
In [24]: ddosclean.to_csv('ddosafter.csv', index=False)

In [25]: ddosafter=pd.read_csv('ddosafter.csv')

         ddosafter.head(20)
```

Out[25]:

	pktcount	bytecount	tot_dur	flows	packetins	pktperflow	byteperflow	pktrate	Pairflow	Protocol	port_no	tx_bytes	rx_bytes	tot_kbps	label
0	45304	48294064	101000000000	3	1943	13535	14428310	451	0	0	3	143928631	3917	0	0
1	126395	134737070	281000000000	2	1943	13531	14424046	451	0	0	4	3842	3520	0	0
2	90333	96294978	201000000000	3	1943	13534	14427244	451	0	0	1	3795	1242	0	0
3	90333	96294978	201000000000	3	1943	13534	14427244	451	0	0	2	3688	1492	0	0
4	90333	96294978	201000000000	3	1943	13534	14427244	451	0	0	3	3413	3665	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15	45304	48294064	101000000000	3	1943	13535	14428310	451	0	0	2	3795	1492	0	0
16	45304	48294064	101000000000	3	1943	13535	14428310	451	0	0	2	4047	193291210	6307	0

Figure 4. 21 *Latest Dataset has been Created and Able to be Opened*

4.2.2.8 Standardization

Standardizing is a process to re-scaling the values to ensure the mean of data is 0 and the standard deviation is 1 to decrease the ambiguity by preventing the features with large ranges affect the performance metric in detection(Raju et al., 2020). To perform this, standardization process has been conducted as below for DNN, KNN, SVM and Naïve Bayes, except Decision Tree as it will not be influenced by magnitude of features using the new clean dataset.

```
In [6]: scaler = StandardScaler()
        x = scaler.fit_transform(x)

In [7]: x

Out[7]: array([[-0.12833641,  0.24572408, -0.79168404, ...,  0.
                 0.        ,  0.2615767 ],
                [ 1.41922972,  2.03234746, -0.16581867, ...,  0.
                 0.        ,  0.2615767 ],
                [ 0.73101118,  1.2378177 , -0.44398106, ...,  0.
                 0.        ,  0.2615767 ],
                ...,
                [-0.99234057, -0.75236565, -1.03227711, ...,  0.
                 0.        , -0.53570863],
                [-0.99234057, -0.75236565, -1.03227711, ...,  0.
                 0.        , -0.53570863],
                [-0.99234057, -0.75236565, -1.03227711, ...,  0.
                 0.        , -0.53570863]])
```

Figure 4. 22 *Performing Standardization Process*

4.3 TESTING AND RESULT DISCUSSION

4.3.1 Training and Testing Data Ratio

To perform training and testing phase, 70:30 ratio is used where 70% of data is for training and 30% data is for testing. It indicates that there are 69477 data is used for 70% training data whereas 29777 data is used for 30% testing data.

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_size=0.30)

print("Number of Original Data:", ddosafter.shape[0])
print("Number of Original Features of Data:", ddosafter.shape[1])
print("\n")
print('Training Data:', x_train.shape[0])
print('Features of Training Data :', x_train.shape[1])
print("\n")
print('Testing Data:', x_test.shape[0])
print('Features of Testing Data:', x_test.shape[1])
```

```
Number of Original Data: 99254
Number of Original Features of Data: 23


Training Data: 69477
Features of Training Data : 22


Testing Data: 29777
Features of Testing Data: 22
```

Figure 4. 23 *Training and Testing Data Size*

4.3.2 Building and Training Machine Learning Models

Below shows training machine learning model phase.

4.3.2.1 Deep Neural Network (DNN)

This model is imported from *TensorFlow keras* module. The coding is shown as below. The model and coding implemented are used to build and train the DNN model.

```
from keras.models import Sequential

from keras.layers import Dense

from keras.callbacks import EarlyStopping

from numpy.random import seed

import tensorflow

tensorflow.random.set_seed(19088)
```

```
In [8]:
model = Sequential()

model.add(Dense(200, activation='relu', input_shape=(22,), name="1st_Hidden_Layer"))
model.add(Dense(200, activation='relu', name="2nd_Hidden_Layer"))
model.add(Dense(1, activation='sigmoid', name="Output_Layer"))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 1st_Hidden_Layer (Dense)    (None, 200)               4600

 2nd_Hidden_Layer (Dense)    (None, 200)               40200

 Output_Layer (Dense)        (None, 1)                 201

=================================================================
Total params: 45,001
Trainable params: 45,001
Non-trainable params: 0
```

Figure 4. 24 *Defining and Compiling DNN Model*

```
md = model.fit(x_train, y_train, validation_split=0.3, epochs=150, callbacks=[early_stopping_monitor], batch_size=10, validation_

_, accuracy = model.evaluate(x_train, y_train)
```

Figure 4. 25 *Training DNN Model*

4.3.2.2 Decision Tree (DT)

This model is imported from *sklearn DecisionTreeClassifier* module. The coding is shown as below. The model and coding implemented are used to build and train the DT model.

```
from sklearn.tree import DecisionTreeClassifier
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_size=0.30)
dt = DecisionTreeClassifier(criterion="entropy", max_depth=3, min_samples_leaf=5, random_state=42)
dt = dt.fit(x_train,y_train)
y_pred = dt.predict(x_test)
```

Figure 4. 26 *Training DT Model*

4.3.2.3 K-Nearest Neighbors (KNN)

This model is imported from *sklearn KNeighborsClassifier* module. The coding is shown as below. The model and coding implemented are used to build and train the KNN model.

```
from sklearn.tree import KNeighborsClassifier
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_size=0.30)
knn = KNeighborsClassifier()
knn.fit(x_train, y_train)
y_pred = knn.predict(x_test)
```

Figure 4. 27 *Training KNN Model*

4.3.2.4 Naïve Bayes (NB)

This model is imported from *sklearn GaussianNB* module. The coding is shown as below. The model and coding implemented are used to build and train the NB model.

```
from sklearn.naive_bayes import GaussianNB
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_size=0.30)
gnb = GaussianNB()
gnb = gnb.fit(x_train,y_train)
y_pred = gnb.predict(x_test)
```

Figure 4. 28 *Training NB Model*

4.3.2.5 Support Vector Machine (SVM)

This model is imported from *sklearn SVC* module. The coding is shown as below. The model and coding implemented are used to build and train the SVM model.

```
from sklearn.svm import SVC
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_size=0.30)
svm = SVC(kernel='rbf')
svm.fit(x_train, y_train)
y_pred = svm.predict(x_test)
```

Figure 4. 29 *Training SVM Model*

4.3.3 Performance Metrics

Performance metrics are necessary to evaluate the DDoS detection of every selected machine learning algorithm. Thus, there are some metrics will be used.

4.3.3.1 Confusion Matrix

This metric is a 2x2 matrix summary of predicted result and the actual result in DDoS detection using True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) to evaluate the detection performance of machine learning models.

		Predicted Class	
		Negative (Normal)	Positive (Attack)
Actual Class	Negative (Normal)	True Negative (TN)	False Positive (FP)
	Positive (Attack)	False Negative (FN)	True Positive (TP)

Figure 4. 30 *Confusion Matrix*

Source: (Kok et al., 2019)

TN or True Negatives refers to number of normal traffic records that is correctly classified.

FP or False Positives refers to number of normal traffic records are mistakenly classified as DDoS traffic.

FN or False Negatives refers to number of DDoS attack records are mistakenly classified as normal traffic.

TP or True Positives refers to number of DDoS attack records that is correctly classified.

4.3.3.2 Accuracy

Accuracy is a performance metric to evaluate the percentage of correct detection of DDoS attack(Yaser et al., 2022).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad 4.1$$

4.3.3.3 Recall

Recall is a performance metric to calculate proportion of true positives from all data that are actually normal traffic to get a higher recall value(Mon Swe et al., 2021; Yaser et al., 2022).

$$Recall = \frac{TP}{TP+FN} \qquad 4.2$$

4.3.3.4 Precision

Precision is a performance metric to calculate proportion of true positives from all data that are predicted as normal traffic to get a higher precisions and lower false alarms (Mon Swe et al., 2021; Yaser et al., 2022).

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

4.3.3.5 F1-Score

This performance metric is a weighted mean of precision and recall for a better accuracy measure to get higher F1-scores (Mon Swe et al., 2021; Yaser et al., 2022).

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \qquad 4.4$$

4.3.3.6 ROC-AUC Area

Area Under the Curve(AUC) for the Receiver Operating Characteristic(ROC) Curve is a metric to evaluate DDoS classification(Corsetti et al., 2022). This metric will display how much the machine learning model can classify the classes. The accuracy of detection is told to be higher when the ROC Curve is getting closer to the upper left corner in graph(Lopez et al., 2019).

4.3.4 Comparison Results of Before Hyperparameter Tuning and After Hyperparameter Tuning of Performance of Proposed Machine Learning Models

Hyperparameter tuning is a process to find a set of optimal parameters for machine learning in dealing with the current data to improve a better detection. Grid Search CV method is chosen to conduct hyperparameter tuning to figure out the best-performing model with hyperparameter values in the grid to build and evaluate the model for every combination of hyperparameters.Below table displays the results of performance of machine learning models in terms of accuracy, recall, precision, F1-Score and ROC AUC.

Table 4. 1 Comparison Results of Before Hyperparameter Tuning and After Hyperparameter Tuning of Performance of Machine Learning Models

Without Hyperparameter Tuning	**Machine Learning Models**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC AUC Area (%)**
	DNN	98.87	97.00	100.00	99.00	99.07
	KNN	97.90	97.00	97.00	97.00	97.79
	SVM	97.32	96.00	97.00	97.00	97.34
	DT	93.58	86.00	100.00	92.00	94.81
	NB	66.34	56.00	57.00	56.00	64.48
After Hyperparameter Tuning	DNN	99.84	100.00	100.00	100.00	99.86
	KNN	99.00	99.00	99.00	99.00	98.94

	SVM	98.96	98.00	99.00	99.00	98.97
	DT	98.42	97.00	99.00	98.00	98.53
	NB	66.34	56.00	57.00	56.00	64.48

Below table shows results of Confusion Matrix of each selected machine learning models.

Table 4. 2 Results of Confusion Matrix of Before Hyperparameter Tuning and After Hyperparameter Tuning of Machine Learning Models

Without Hyperparameter Tuning	**Machine Learning Models**	**True Negatives (TN)**	**False Positives (FP)**	**False Negatives (FN)**	**True Positives (TP)**
	DNN	18108	329	8	11332
	KNN	18117	320	304	11036
	SVM	17933	504	293	11047
	DT	16525	1912	0	11340
	NB	13329	5108	4915	6425
	DNN	18395	42	6	11334

After Hyperparameter Tuning	KNN	18289	148	149	11191
	SVM	18241	196	114	11226
	DT	18085	352	118	11222
	NB	13329	5108	4915	6425

Based on Table 4.1 and Table 4.2, it shows the difference of results of before hyperparameter tuning and after parameter tuning of each proposed machine learning model in terms of performance metrics and confusion matrix. It is significantly to observe that hyperparameter tuning method has improved and increased most of the performance metrics and confusion matrix of DNN, KNN, SVM and DT except NB which shows the static result where no improvement of performance. Below shows the improvement of accuracy graph on before undergoing hyperparameter tuning and after doing hyperparameter tuning.

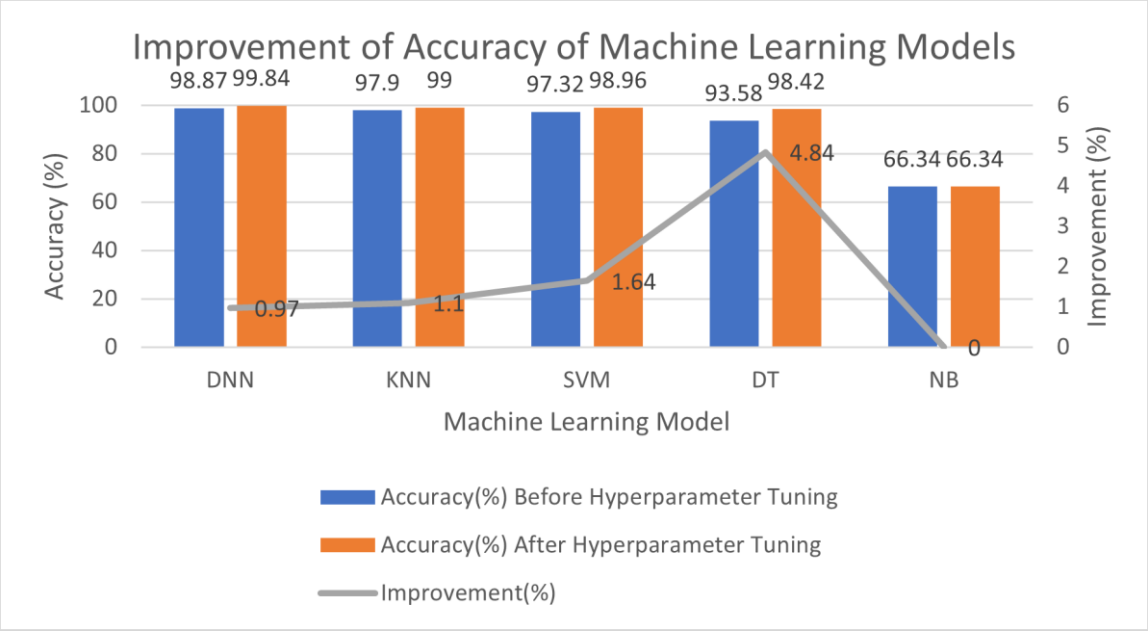


Figure 4. 31 *Improvement of Accuracy of Machine Learning Models*

Table 4. 3 Improvement of Accuracy after performing hyperparameter tuning

Machine Learning Models	Accuracy (%)		Improvement (%)
	Before Hyperparameter Tuning	After Hyperparameter Tuning	
DNN	98.87	99.84	0.97
KNN	97.90	99.00	1.10
SVM	97.32	98.96	1.64
DT	93.58	98.42	**4.84**
NB	66.34	66.34	0.00

Based on Figure 4.31 and Table 4.3 above, it shows a significant positive improvement of accuracy of DNN, KNN, SVM and DT where 0.97%, 1.10%, 1.64% and 4.84% respectively, except NB which shows no improvement, which is 0.00%. Among the machine learning models, DT shows the highest improvement of accuracy result compared to NB which shows nothing in improvement of accuracy result. The reason of Naïve Bayes does not show any improvement in hyperparameter tuning process is because of its unsuitable assumption that assuming whole features are independent and shows no improvement after implementing hyperparameter tuning.

4.3.5 Finalized Performance Result of Proposed Machine Learning Models

Below shows that finalized performance result of proposed machine learning models.

Table 4. 4 Results of Finalized Performance Metrics of Proposed Machine Learning Models

Machine Learning Models	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC AUC Area (%)
DNN	**99.84**	**100.00**	**100.00**	**100.00**	**99.86**
KNN	99.00	99.00	99.00	99.00	98.94
SVM	98.96	98.00	99.00	99.00	98.97
DT	98.42	97.00	99.00	98.00	98.53
NB	66.34	56.00	57.00	56.00	64.48

Based on Table 4.4, it shows that DNN scores the best scores of detection performance in terms of accuracy, precision, precision, F1-Score and ROC AUC which are 99.84%, 100.00%, 100.00%, 100.00% and 99.86% meanwhile Naïve Bayes scores the lowest scores of detection performance which are 66.34%, 56.00%, 57.00%, 56.00% and 64.48%. These results show that, DNN detect the 99.84% test data for accurate detection compared to Naïve Bayes which just can detect 66.34% test data for accurate detection. In addition, out of all data the model predicted is DDoS attack, DNN could detect 100.00% DDoS attack compared to NB which can detect 56.00% of DDoS attack in terms of precision. Besides, out of the actual DDoS attack data, DNN could

detect there are 100.00% of DDoS attack are predicted correct compared to NB which can detect 57.00% of DDoS attack in terms of recall. Moreover, DNN performs 100.00% well job in detecting DDoS attack compared to NB which can perform 56.00 % a moderate job in detecting DDoS attack. Besides, DNN can distinguish the classes between normal traffic and DDoS attack for 99.86% compared to NB which just able to distinguish the classes for 64.48% in terms of ROC AUC Area.

Table 4. 5 Results of Finalized Confusion Matrix Result of Proposed Machine Learning Models

Machine Learning Models	True Negatives (TN)	False Positives (FP)	False Negatives (FN)	True Positives (TP)
DNN	**18395**	**42**	**6**	**11334**
KNN	18289	148	149	11191
SVM	18241	196	114	11226
DT	18085	352	118	11222
NB	13329	5108	4915	6425

Based on Table 4.5, DNN shows the best result of Confusion Matrix among the machine learning models in terms of True Negatives (TN), False Positives (FP), False Negatives (FN) and True Positives (TP) which are 18395, 42, 6 and 11334 respectively compared Naïve Bayes shows the lowest result of Confusion Matrix among machine learning models in terms of True Negatives (TN), False Positives (FP), False Negatives (FN) and True Positives (TP) which are 13329, 5108, 4915 and 6425 respectively. This indicates that DNN can detect the normal traffic and DDoS attack more accurately int

terms of TN and TP and shows the less false detection in distinguishing the normal traffic and DDoS attack in terms of FP and FN compared to Naïve Bayes.

DNN scores better because DNN comprises of weights and number of hidden layers to classify the features accurately especially handling the dataset with many complex features. Meanwhile, Naïve Bayes assumes the probability of all features are independent however the correlation of features which is not independent and that leads Naïve Bayes shows the poor detection performance.

4.3.6 Information of Author who conduct DDoS Detection using DDoS Attack SDN Dataset

Table 4. 6 : Information of Author who conduct DDoS SDN Detection

No	Author(s) and Year	Title	Dataset	The Same Machine Learning Models Used	The Same Performance Metrics Used
1	(Tonkal et al., 2021)	Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking	DDoS Attack SDN Dataset	KNN DT SVM	Accuracy Recall/Sensitivity Precision F1-Score
2	(Ahuja et al., 2021)	Automated DDOS Attack Detection in Software Defined Networking	DDoS Attack SDN Dataset	SVM KNN	Accuracy Recall/Detection Rate Precision F1-Score

4.3.7 Comparison of Proposed Results with Another Authors

4.3.7.1 Proposed Model VS (Tonkal et al., 2021)

Table 4. 7 Comparison of Proposed Model and (Tonkal et al., 2021)

Machine Learning Models	**Author**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
KNN	**Proposed Model**	**99.00**	**99.00**	**99.00**	**99.00**
	(Tonkal et al., 2021)	97.75	97.76	97.44	98.17
DT	Proposed Model	98.42	97.00	99.00	98.00
	(Tonkal et al., 2021)	**99.18**	**99.18**	**99.88**	**99.32**
SVM	**Proposed Model**	**98.96**	**98.00**	**99.00**	**99.00**
	(Tonkal et al., 2021)	81.48	81.34	82.87	85.23

Based on Table 4.7, the detection performance using KNN and SVM of proposed model is higher than the author's model but proposed DT model performance is lower than author's model.

The accuracy, recall, precision and F1-Score of proposed KNN model are 99.00% compared to the accuracy, recall, precision and F1-Score of the author's model which are 97.75%, 97.76%, 97.44% and 98.17% respectively. The whole performance metrics of proposed KNN model is higher than author's model.

Besides, the accuracy, recall, precision and F1-Score of proposed DT model is 98.42%, 97.00%, 99.00% and 98.00% compared to the accuracy, recall, precision and F1-Score of the author's model which are 99.18%, 99.18%, 99.88% and 99.32% respectively. The whole performance metrics of proposed DT model is lower than author's model.

Next, the accuracy, recall, precision and F1-Score of proposed SVM model is 98.96%, 98.00%, 99.00% and 99.00% compared to the accuracy, recall, precision and F1-Score of the author's model which are 81.48%, 81.34%, 82.87% and 85.23% respectively. The whole performance metrics of proposed SVM model is higher than author's model.

It is noticed that among the proposed models in terms of highest accuracy, KNN shows the highest accuracy result. However, among the author's models in terms of higher accuracy, DT is suggested to have highest accuracy result.

4.3.7.2 Proposed Model VS (Ahuja et al., 2021)

Table 4. 8 Comparison of Proposed Model and (Ahuja et al., 2021)

Machine Learning Models	**Author**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
SVM	**Proposed Model**	**98.96**	**98.00**	**99.00**	**99.00**
	(Ahuja et al., 2021)	85.83	85.79	87.46	86.61
KNN	**Proposed Model**	**99.00**	**99.00**	**99.00**	**99.00**
	(Ahuja et al., 2021)	95.22	96.83	94.37	95.58

Based on Table 4.8, the detection performance using SVM and KNN of proposed model is higher than the author's model

The accuracy, recall, precision and F1-Score of proposed SVM model is 98.96%, 98.00, 99.00% and 99.00% compared to the accuracy, recall, precision and F1-Score of the author's model which are 85.83%, 85.79%, 87.46% and 86.61% respectively. The whole performance metrics of proposed SVM model is higher than author's model.

The accuracy, recall, precision and F1-Score of proposed KNN model are 99.00% compared to the accuracy, recall, precision and F1-Score of the author's model

which are 95.22%, 96.83%, 94.37% and 95.58% respectively. The whole performance metrics of proposed KNN model is higher than author's model.

It is noticed that among the proposed models and authors' model in terms of highest accuracy, KNN shows the highest accuracy result and SVM shows the lowest accuracy for both models. However, it is observed that, the accuracy of whole proposed models is higher than author's model.

4.3.8 Summary of Comparison between Proposed Model and Authors' Model

It is observed that most of proposed model shows higher detection performance result than authors' model.

This could be proven by comparison of proposed model and first author's model. Proposed KNN and SVM models show higher performance than author's model in terms of accuracy, precision, recall and F1-Score. However, proposed DT Model shows lower performance than author's model in terms of accuracy, precision, recall and F1-Score.

For the comparison of proposed and second author's model, our proposed KNN and SVM models show higher performance than author's model in terms of whole selected performance metrics such as accuracy, precision, recall and F1-Score.

The proposed models show higher performance is because of undergoing a proper data preprocessing process after conducting EDA to check the data quality issues and provide appropriate treatments. Besides, hyperparameter tuning method also contributing to improvement of detection performance to figure out the suitable optimal parameters for DDoS detection.

It is observed that these two authors do not apply DNN for DDoS detection study. DNN is proposed to be applied as it shows an impressive detection performance that evaluated by using performance metrics.

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

DDoS attack detection becomes a main concern in Software-Defined Networking (SDN) environment as SDN controller is vulnerable to DDoS attack. This research discusses the types of DDoS attacks and risks that brought by DDoS attack. Several research works have been reviewed to understand their methodology to apply various of machine learning techniques in DDoS detection. This research's contributions are evaluating the performance of machine learning techniques for the selected static dataset, studying and analyzing the features in detecting DDoS attack. Furthermore, the experiment has improved the detection performance results and analysis and discussion with related works was performed. This study proposed a DDoS attack detection using machine learning algorithm. To conduct this study, the DDoS Attack SDN Dataset which consists of 1,04,345 data with 23 features is used for performance evaluation. From the 23 features, there are 15 features has been as input features in this experiment. Several processes such as EDA, data preprocessing and hyperparameter tuning were conducted before performance evaluation to ensure a fair and accurate performance results are generated. Hyperparameter tuning has proven that it can improve the performance metrics of machine learning techniques. It is worth highlighting that these significant improvement for accuracy performance result of DNN, KNN, SVM and DT which are 0.97%, 1.10%, 1.64% and 4.84% respectively except DT which shows no improvement (0.00%) because of the unsuitable assumption to assume all features are independent that lead lowest detection performance. While, the DNN has shown significant improvement in detecting the DDoS attack as it achieved highest accuracy rate with 99.84%. Besides its 100.00% precision indicates DNN can detect out of all data the model predicted is DDoS attack, its 100.00% recall

indicates DNN is able to predict the DDoS attack data correctly, its 100.00% F1-Score indicates DNN performs a well job in detecting DDoS attack and its 99.86% ROC AUC Area Curve indicates DNN can distinguish the classes between normal traffic and DDoS attack perfectly. DNN also shows the highest TN and TF to detect the normal traffic and DDoS attack correctly and lowest FN and FP to minimize its fault to wrongly distinguish the normal traffic and DDoS attack. DNN can score better among the machine learning techniques because of its weights and number of multiple hidden layers are contributing to improving the detection performance. The results show that the DNN has a good detection performance for the current popular DDoS attack. The experimental analysis shows that, this research could perform well by using different types of datasets, correct features selection technique and criteria with correct identification of classifier. Besides, more deep learning techniques will be considered in the future study since there is a deep learning technique which is DNN only be used in this study.

5.2 RESEARCH CONSTRAINT

During conducting the study, there are several constraints are identified as the challenge to be faced in the study.

Firstly, there is time limitation to conduct this study. As the given time to accomplish the study is limited, there are another machine learning techniques are not able to be evaluated within the give time.

Besides, the performance evaluation result provided might be different with real-time situation. The performance evaluation result in this study is obtained by testing the static data. It might show the difference in real-time data as real-time DDoS attack scenario is dynamic and could be affected by several real-time factors.

Furthermore, the laptop specification used is ordinary and challenging to conduct the study. Since the hardware specification used is ordinary, it consumes more time to conduct this study especially in model development and hyperparameter tuning. A better computer specification could handle such complex and time-consuming study.

5.3 FUTURE WORK

In the future work, the researcher will apply DNN to evaluate detection performance in multiple another cyber-attacks to investigate and analyses its accuracy result. Besides, DNN will be used to detect the real-time DDoS attack for detection performance evaluation to evaluate and verify its performance result in real-time scenario.

REFERENCES

Ahuja, N., Singal, G., Mukhopadhyay, D., & Kumar, N. (2021). Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications, 187*. https://doi.org/10.1016/J.JNCA.2021.103108

Ajitesh Kumar. (2022, April 26). *Hold-out Method for Training Machine Learning Models*. Vitalflux.

Alharbi, Y., Alferaidi, A., Yadav, K., Dhiman, G., & Kautish, S. (2021). *Denial-of-Service Attack Detection over IPv6 Network Based on KNN Algorithm*. https://doi.org/10.1155/2021/8000869

Altomare F. (2021, April 21). *DDoS (Distributed Denial of Service) Explained*.

Aslam, M., Ye, D., Tariq, A., Asad, M., Hanif, M., Ndzi, D., Chelloug, S. A., Elaziz, M. A., Al-Qaness, M. A. A., & Jilani, S. F. (2022). Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT. *Sensors, 22*(7), 2697. https://doi.org/10.3390/s22072697

Beek, C. (2017). *McAfee Labs Threats Report: April 2017*. www.mcafee.com/us/mcafee-labs.aspx

Bhatia, S., Behal, S., & Ahmed, I. (2018). Distributed Denial of Service Attacks and Defense Mechanisms: Current Landscape and Future Directions. In *Advances in Information Security* (Vol. 72, pp. 55–97). Springer New York LLC. https://doi.org/10.1007/978-3-319-97643-3_3

Bista, S., Chitrakar, R., Bista, S., & Chitrakar, R. (2017). DDoS Attack Detection Using Heuristics Clustering Algorithm and Naïve Bayes Classification. *Journal of Information Security, 9*(1), 33–44. https://doi.org/10.4236/JIS.2018.91004

Brooks, R. R., & Özçelik, İ. (2020). Distributed Denial of Service Attacks. *Distributed Denial of Service Attacks*, 6–7. https://doi.org/10.1201/9781315213125

Cheng, J., Yin, J., Liu, Y., Cai, Z., & Wu, C. (2009). DDoS attack detection using IP address feature interaction. *International Conference on Intelligent Networking and Collaborative Systems, INCoS 2009*, 113–118. https://doi.org/10.1109/INCOS.2009.34

CIC UNB. (2018). *IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*. https://www.unb.ca/cic/datasets/ids-2018.html

Cil, A. E., Yildiz, K., & Buldu, A. (2021). Detection of DDoS attacks with feed forward based deep neural network model. *Expert Systems with Applications*, *169*. https://doi.org/10.1016/J.ESWA.2020.114520

Cisco. (2019). *Cisco Visual Networking Index: Forecast and Trends, 2017–2022*.

Cisco. (2020). *Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper - Cisco*. Cisco. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

Corsetti, S., Purkayastha, A., Hasandka, A., & Samon, M. (2022). *Automatic DDoS Attack Detection on SDNs: Preprint*. https://www.nrel.gov/docs/fy22osti/81041.pdf.

Danades, A., Pratama, D., Anggraini, D., & Anggriani, D. (2017). Comparison of accuracy level K-Nearest Neighbor algorithm and support vector machine algorithm in classification water quality status. *Proceedings of the 2016 6th International Conference on System Engineering and Technology, ICSET 2016*, 137–141. https://doi.org/10.1109/FIT.2016.7857553

Fadlil, A., Riadi, I., & Aji, S. (2017). Review of Detection DDOS Attack Detection Using Naive Bayes Classifier for Network Forensics. *Bulletin of Electrical Engineering and Informatics*, *6*(2), 140–148. https://doi.org/10.11591/EEI.V6I2.605

Fan, C., Kaliyamurthy, N. M., Chen, S., Jiang, H., Zhou, Y., & Campbell, C. (2022). Detection of DDoS Attacks in Software Defined Networking Using Entropy. *Applied Sciences (Switzerland)*, *12*(1). https://doi.org/10.3390/app12010370

Fatima, M., Rehman, O., & Rahman, I. (2018). Impact of Features Reduction on Machine Learning Based Intrusion Detection Systems. *ICST Transactions on Scalable Information Systems*, 447. https://doi.org/10.4108/eetsis.vi.447

Feng, Y., & Li, J. (n.d.). *Toward Explainable and Adaptable Detection and Classification of Distributed Denial-of-Service Attacks*. https://doi.org/10.1007/978-3-030-59621-7_6

Georgevici, A. I., & Terblanche, M. (2019). Neural networks and deep learning: a brief introduction. *Intensive Care Medicine*, *45*, 712–714. https://doi.org/10.1007/s00134-019-05537-w

Hasan, M., Milon Islam, M., Ishrak Islam Zarif, M., & Hashem, M. (2019). *Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches*. https://doi.org/10.1016/j.iot.2019.10

Hindy, H., Atkinson, R., Tachtatzis, C., Colin, J. N., Bayne, E., & Bellekens, X. (2020). Utilising deep learning techniques for effective zero-day attack detection. *Electronics (Switzerland)*, *9*(10), 1–16. https://doi.org/10.3390/electronics9101684

Jain V. (2019, December 30). *Everything you need to know about "Activation Functions" in Deep learning models*. Towards Data Science.

Jakub Adamczyk. (2020). *k nearest neighbors computational complexity | by Jakub Adamczyk | Towards Data Science*. Towards Data Science. https://towardsdatascience.com/k-nearest-neighbors-computational-complexity-502d2c440d5

Jinde Shubham. (2018, June 30). *Naive Bayes Theorem. Introduction | by Jinde Shubham | Becoming Human: Artificial Intelligence Magazine*. Medium. https://becominghuman.ai/naive-bayes-theorem-d8854a41ea08

Karan B. V, Narayan D. G, & P. S. Hiremath. (2019, July 25). Detection of DDoS Attacks in Software Defined Networks. *IEEE*.

Kaskar, J., Bhatt, R., & Shirsath, R. (2014). *A System for Detection of Distributed Denial of Service (DDoS) Attacks using KDD Cup Data Set*. www.ijcsit.com

Kaspersky. (2022). *DDoS attacks hit a record high in Q4 2021 | Kaspersky*. Kaspersky. https://www.kaspersky.com/about/press-releases/2022_ddos-attacks-hit-a-record-high-in-q4-2021

Kevin Markham. (2019, March 28). *Six easy ways to run your Jupyter Notebook in the cloud*. https://www.dataschool.io/cloud-services-for-jupyter-notebook/

Kok, S. H., Abdullah, A., Supramaniam, M., Pillai, T. R., Abaker, I., & Hashem, T. (2019). A Comparison of Various Machine Learning Algorithms in a Distributed Denial of Service Intrusion. *International Journal of Engineering Research and Technology*, *12*(1), 1–7. http://www.irphouse.com

Lopez, A. D., Mohan, A. P., Nair, S., Lopez, A., & Mohan, A. (2019). Network Traffic Behavioral Analytics for Detection of DDoS Attacks. In *SMU Data Science Review* (Vol. 2, Issue 1).

https://scholar.smu.edu/datasciencereviewhttp://digitalrepository.smu.edu.Availableat:https://scholar.smu.edu/datasciencereview/vol2/iss1/14

Lucky, G., Jjunju, F., & Marshall, A. (2020). *A Lightweight Decision-Tree Algorithm for detecting DDoS flooding attacks*. https://doi.org/10.1109/QRS-C51114.2020.00072

Mercer C. (2017, May 17). *How does a DDoS attack work?* Tech Advisor.

Mon Swe, Y., Pye Aung, P., & Su Hlaing, A. (2021). *A Slow DDoS Attack Detection Mechanism using Feature Weighing and Ranking*.

Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing: A Review Journal*, *73*, 1–15. https://doi.org/10.1016/J.DSP.2017.10.011

Nadeem, M. W., Goh, H. G., Ponnusamy, V., & Aun, Y. (2022). Ddos detection in sdn usingmachine learning techniques. *Computers, Materials and Continua*, *71*(1), 771–789. https://doi.org/10.32604/cmc.2022.021669

QuantStack. (2017, November 30). *Interactive Workflows for C++ with Jupyter | by QuantStack | Jupyter Blog*. Medium. https://blog.jupyter.org/interactive-workflows-for-c-with-jupyter-fe9b54227d92

Rahman, M. A. (2020). Detection of Distributed Denial of Service Attacks based on Machine Learning Algorithms. *International Journal of Smart Home*, *14*(2), 15–24. https://doi.org/10.21742/IJSH.2020.14.2.02

Raju, V. N. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., & Padma, V. (2020). Study the Influence of Normalization/Transformation process on the Accuracy of Supervised Classification. *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020*, 729–735. https://doi.org/10.1109/ICSSIT48917.2020.9214160

Randles, B. M., Golshan, M. S., Pasquetto, I. v, & Borgman, C. L. (2017). *Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study*. https://doi.org/10.1016/j.future.2011.08.004

Sagar Joshi. (2022, April 28). *What Is a DDoS Attack? How to Stop Malicious Traffic Floods*. G2.

Saghezchi, F. B., Mantas, G., Violas, M. A., de Oliveira Duarte, A. M., & Rodriguez, J. (2022). Machine Learning for DDoS Attack Detection in Industry 4.0 CPPSs. *Electronics (Switzerland)*, *11*(4). https://doi.org/10.3390/electronics11040602

Saini, P. S., Behal, S., & Bhatia, S. (2020). Detection of DDoS attacks using machine learning algorithms. *Proceedings of the 7th International Conference on Computing for Sustainable Global Development, INDIACom 2020*, 16–21. https://doi.org/10.23919/INDIACom49435.2020.9083716

Satyam Mishra. (2020, October 29). *Breaking Down the Support Vector Machine (SVM) Algorithm | by Satyam Mishra | Towards Data Science*. Towards Data Science. https://towardsdatascience.com/breaking-down-the-support-vector-machine-svm-algorithm-d2c030d58d42

Schakenbach J. (2013, January 29). *Report: DDoS attacks harder to detect and defeat*. Boston Business Journal.

Shao, E. (2019). *ENCODING IP ADDRESS AS A FEATURE FOR NETWORK INTRUSION DETECTION*.

Shieh, C. S., Lin, W. W., Nguyen, T. T., Chen, C. H., Horng, M. F., & Miu, D. (2021). Detection of unknown ddos attacks with deep learning and gaussian mixture model. *Applied Sciences (Switzerland)*, *11*(11). https://doi.org/10.3390/app11115213

Srikanth K Ballal, Lalitha S Prasad, Madhusudhan Rajappa, & Ashiq Khader. (2018, April 27). *Bumper to Bumper: Detecting and Mitigating DoS and DDoS Attacks on the Cloud, Part 1*. Security Intelligence.

Suresh, M., & Anitha, R. (2011). Evaluating machine learning algorithms for detecting DDoS attacks. *Communications in Computer and Information Science*, *196 CCIS*, 441–452. https://doi.org/10.1007/978-3-642-22540-6_42

Tonkal, Ö., Polat, H., Başaran, E., Cömert, Z., & Kocaoğlu, R. (2021). Machine learning approach equipped with neighbourhood component analysis for ddos attack detection in software-defined networking. *Electronics (Switzerland)*, *10*(11). https://doi.org/10.3390/electronics10111227

Uday Paila. (2018). *Deep Neural Network for Classification from scratch using Python | by Uday Paila | Medium*. Medium. https://medium.com/@udaybhaskarpaila/multilayered-neural-network-from-scratch-using-python-c0719a646855

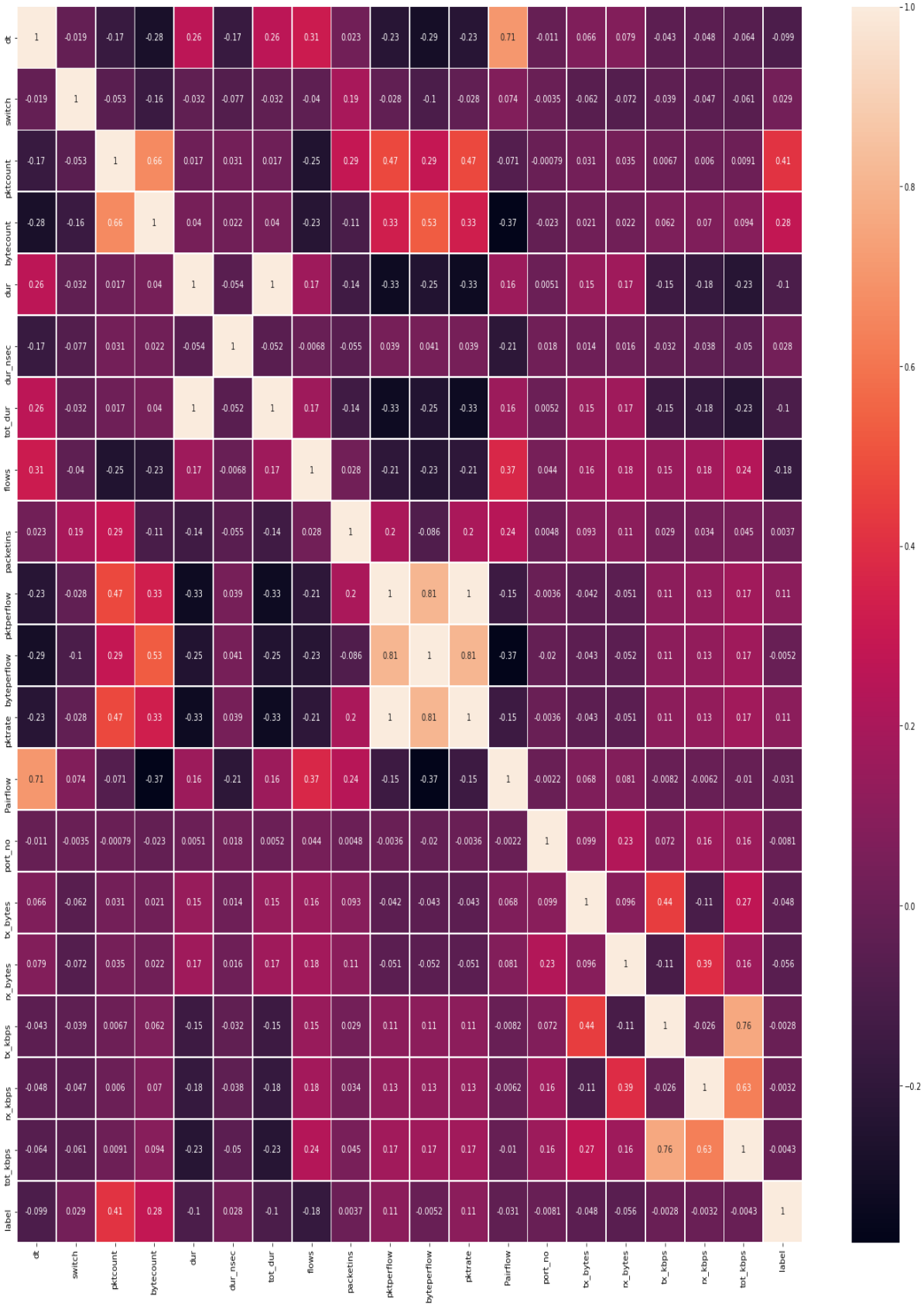
Wu, Y. C., Tseng, H. R., Yang, W., & Jan, R. H. (2011). Ddos detection and traceback with decision tree and grey relational analysis. *International Journal of Ad Hoc and Ubiquitous Computing*, *7*(2), 121–136. https://doi.org/10.1504/IJAHUC.2011.038998

Yaser, A. L., Mousa, H. M., & Hussein, M. (2022). Improved DDoS Detection Utilizing Deep Neural Networks and Feedforward Neural Networks as Autoencoder. *Future Internet*, *14*(8). https://doi.org/10.3390/fi14080240

Ye, J., Cheng, X., Zhu, J., Feng, L., & Song, L. (2018). *A DDoS Attack Detection Method Based on SVM in Software Defined Network*. https://doi.org/10.1155/2018/9804061

Zinets N. (2022, February 16). *Ukraine points finger of suspicion at Russia over massive cyberattack*. Reuters.

APPENDIX A
CORRELATION HEATMAP



APPENDIX B
GANTT CHART

ID	Name	Start Date	End Date	Duration
1	Construct Introduction Part	7/3/2022	17/3/2022	11 days
2	Construct Literature Review	28/3/2022	7/4/2022	11 days
3	Construct Methodology	18/4/2022	20/5/2022	33 days
4	FYP 1 Presentation	15/6/2022	15/6/2022	1 day
5	FYP 1 Report Submission	24/6/2022	24/6/2022	1 day
6	Data Loading	8/8/2022	8/82022	1 day
7	EDA and Data Preprocessing	9/8/2022	26/8/2022	18 days
8	Model Development and Initial Detection Performance Evaluation	5/9/2022	10/9/2022	6 days
9	Hyperparameter Tuning and Second Detection Performance Evaluation	12/9/2022	19/9/2022	8 days
10	Construct Chapter 4	17/10/2022	6/11/2022	31 days
11	Construct Chapter 5	27/12/2022	28/12/2022	2 days