Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
Information Technology
Department Information Networks

A PROPOSED DDOS DETECTION APPROACH BASED ON ENTROPY AND MACHINE LEARNING IN SDN ENVIRONMENT

A Thesis Submitted

to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology / Information Networks.

By

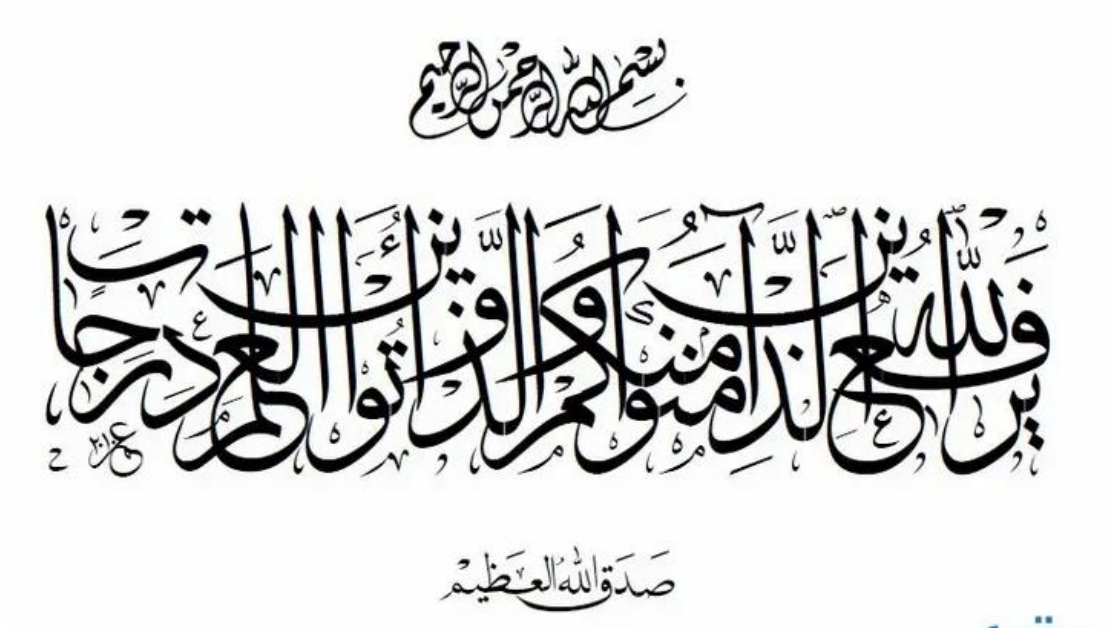
Abbas Jasem Abood Kalef

Supervised by

Asst. Prof. Dr. Aladdin Abbas Abdulhassan Hamza

Asst. Prof. Dr. Nawfal Turki Obeis

2022 A.D. **1444 A.H**



سورة المجادلة، الآية 11

Supervisor Certification

I certify that the thesis entitled (**A PROPOSED DDOS DETECTION APPROACH BASED ON ENTROPY AND MACHINE LEARNING IN SDN ENVIRONMENT**) was prepared under my supervision at the department of Information Networks/ College of Information Technology / University of Babylon as partial fulfillment of the requirements of the degree of Master in Information Technology-Information Networks.

Signature:

Supervisor Name: **Asst. Prof. Dr. Aladdin Abbas Abdulhassan Hamza**

Date: / /2022

Signature:

Supervisor Name: **Asst. Prof. Dr Nawfal Turki Obeis** Date: / /2022

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "**A PROPOSED DDOS DETECTION APPROACH BASED ON ENTROPY AND MACHINE LEARNING IN SDN ENVIRONMENT"** for debate by the examination committee.

Signature:

Prof. Dr. **Saad Talib Hasson Aljebori**

Head of Information Networks Department

Date: / /2022

Declaration

I hereby declare that this thesis, submitted to the University of Babylon in partial fulfillment of requirement for the degree of Master of Information Technology-Information Networks has not been submitted as an exercise for a similar degree at any other university. I also certify that this work described here is entirely my own except for experts and summaries whose sources are appropriately cited in the references.

Dedication

I dedicate this thesis

To my father

To the fountain of patience my mother

To my supervisors

To my family

To my friends

Acknowledgement

In the name of God, Most Gracious, Most Merciful, At first, Praise be to God and thanks to God and the satisfaction of parents and conciliation only from God greatest praise is to **Allah** for His assistance in facing the difficulty that I met in my study, and for always helping me to achieve my aims, also for His great graces and boons all the time.

I would like to express my deepest thanks to my supervisor **Dr. Aladdin Abbas Abdulhassan Alsharify, and Dr. Nawfal Turki Obeis** for their valuable advice, motivation, guidance, and for so many fruitful discussions throughout the preparation of this thesis.

I would like to extend my respect and deepest gratitude to the College of Information Technology.

Sincere appreciation and love go to my family, my father's soul who he was always be with me in my heart and my dear mother that whatever I did to her will not reward her they provide me with optimism and pure affection and they give me great hope, encouragement and they have stood with me in every step in this research. I dedicate this work and give special thanks to those who encouraged me to continue my scientific career, my wife, and my children.

Finally, Sincere thanks and appreciation to all friends, colleagues and loved ones

Abstract

Software-Defined Networking (SDN) is a networking paradigm that has redefined the term network by making the network devices programmable. SDN helps network engineers to monitor the network expedite, control the network from a central point, and identify malicious traffic and link failure in an easy and efficient manner. On other side, such flexibility provided by SDN, it is also vulnerable to attacks such as DDoS which can halt the complete network. To mitigate this attack, it is important to build a secure system to classify data traffic as normal and abnormal with machine learning techniques contributing to the rapid detection of these attacks. There are many researchers were interested to identify and mitigate different types of attacks on SDN especially DDoS , by using machine learning, intrusion detection system (IDS), and entropy in this side to classify traffic and decrease DDoS attack.

The proposed system makes use of entropy and machine learning (M.L)algorithms as (Neural Network (NN), Decision Tree (DT), and Support Vector Machine(SVM)) to monitor, and protecting the SDN controller from DDoS attacks through monitoring network behavior of normal state and recognizing the DDos attack state. It is implemented with two main stages, the first is Off-Line stage, and the second stage is Real-Time data traffic for testing incoming request and classify it as normal or abnormal and evaluated with main evaluation metrics as accuracy, F-Measure, Precision, and Recall.

The proposed system implemented in Python and achieved high accuracy were the best results of M.L were 99.9979 % accuracy when use DT of the off-line traffic phase with CICDDoS2019 dataset , and DT was 99.90 % of real-time accuracy beside, the accuracy of Off-line phase of NN was 99.8587 %, and SVM algorithm was 99.6 %.

Table of Contents

Table of Contents

Table of Tables

Table of Figures

Table of Algorithms

List of Abbreviations

Abbreviation	Description
API	Application Programming Interface
ANN	Artificial neural network
BT	Bagging Tree
CIC	Canadian Institute of Cybersecurity
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSV	Comma-Separated Values
DAD	DDoS Attacks Detection
DA	Discriminant Analysis
DDoS	Distributed Denial of Service
EDDSC	Efficient Detect DDoS attacks against the SDN controller
E-PCA	Entropy and Principal Component Analysis
FNN	Feedforward Neural Network
FN	False Negatives
FP	False Positive
GLM	Generalised Linear Model (GLM)
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IPS	Intrusion Protection Detection System
K-NN	K-Nearest Neighbors
ML	Machine Learning
MLP	Multi-Layer Perceptron
MTU	Maximum Transmission Units
ONOS	Open Network Operating System
OVS	Open Virtual Switch
PCA	Principal Component Analysis
RAM	Random Access Memory
RF	Random Forest
SAE	Stacked Auto Encoder
SDN	Software defined networking
SGS	Safe-guard Scheme
SL	Supervised Learning
SVM	Support Vector Machine
SVC-RF	Support Vector classifier with Random Forest

TLS	Transport Layer Security
TN	True Negative
TP	True Positive
XGBoost	Extreme Gradient Boosting

List of Thesis Related Publications

Paper 1st in Conference: International Conference on Innovations in Science, Hybrid Materials and Vibration Analysis (IC-ISHVA 2022).
Paper 2nd published in : Bulletin of Electrical Engineering and Informatics.

- **1st Paper Title: Improving the security of SDN controller using machine learning techniques.**
- **2nd Paper Title: DDoS attack detection in SDN controller using Machine Learning Techniques.**

- **Authors:**
  - Abbas Jasem Abood
  - Lect.Dr. Aladdin Abbas Alshrifi
  - Asst.Prof.Dr. Nawfal Turkey Obies
  - Information Network Department, Information Technology College, Babylon University.

Email: abbas.j.altamemi@gmail.com

Email: aladdin.alsharifi@uobabylon.edu.iq

Email: nawfal.aljumaili@uobabylon.edu.iq

IC-ISHVA 2022

International Conference on Innovations in Science, Hybrid Materials and Vibration Analysis

Online

Pune, Maharashtra, India, July 16-17, 2022

Acceptance Letter

30 June 2022

Paper Title: Improving SDN Controller Security using Machine Learning Techniques.

Paper ID: 172

Authors: Abbas Jasem Altamemi Aladdin Abdulhassan Nawfal Turki Obeis

Publication: AIP Conference Proceedings

Indexed in: Scopus, Web of Science (WOS), ISSN: 0094-243X (print); 1551-7616 (web)

Dear Authors,

Congratulations! We are pleased to inform you that your paper has been ACCEPTED for oral presentation at International Conference on Innovations in Science, Hybrid Materials and Vibration Analysis (IC-ISHVA-2022) to be held by Research Association of Masters of Engineering, Pune, Maharashtra, India during July 16-17, 2022.

Registration:

Confirmation of your presentation on the final schedule is contingent upon receipt of the presenting author's registration and full payment of the registration charges. Presenters must attend and present the paper in online mode in order to be included in the program and proceedings.

The fees for the conference registration and payment instructions details can be found at https://ishva.rame.org.in/D22/registration/

The author should share his/her presentation through the link given below up to 15th June 2022.

https://forms.gle/quUy7jd9G7rVdCgs5

Presentation Guidelines

The presentation guidelines and PowerPoint presentation format are available on the conference website. Click on the link below to download.

https://ishva.rame.org.in/D22/

Conference Program

The program and presentation schedule will be published on the conference website very soon. Kindly check the conference website for the latest updates.

For further query, kindly contact us at conference@rame.org.in

Thank you for your contribution.

Best regards,

Conference Chair, IC-ISHVA 2022

Research Association of Masters of Engineering, India

conference@rame.org.in

Intelektual Pustaka Media Utama

iaes Institute of Advanced Engineering and Science

Scopus indexed
CiteScoreTracker 2022: 2.3 | h-index: 16 (Last updated on 10 June, 2022)

SJR Q3 Computer Networks and Communications
0.357 0.730 2.4

CERTIFICATE

No. 4155/BEEI/A/08/2022

Bulletin of Electrical Engineering and Informatics (BEEI)

is hereby awarding this certificate to

Abbas Jasem Altamemi, Aladdin Abdulhassan, Nawfal Turki Obeis

in recognition of his/her contribution in this scientific journal

as *Authors* for paper entitled:

DDoS attack detection in software defined networking controller using machine learning techniques

in Vol. 11, No. 5, October 2022

ISSN 2089-3191
http://beei.org

Yogyakarta, August 19, 2022

Tole Sutikno
Managing Director, Journals

DDoS attack detection in software defined networking controller using machine learning techniques

Abbas Jasem Altamemi, Aladdin Abdulhassan, Nawfal Turki Obeis

Abstract

The term software defined networking (SDN) is a network model that contributes to redefining the network characteristics by making the components of this network programmable, monitoring the network faster and larger, operating with the networks from a central location, as well as the possibility of detecting fraudulent traffic and detecting special malfunctions in a simple and effective way. In addition, it is the land of many security threats that lead to the complete suspension of this network. To mitigate this attack this paper based on the use of machine learning techniques contribute to the rapid detection of these attacks and methods were evaluated detecting DDoS attacks and choosing the optimum accuracy for classifying these types within the SDN, the results showed that the proposed system provides the better results of accuracy to detect the DDos attack in SDN network as 99.90% accuracy of Decision Tree (DT) algorithm.

Chapter One

General Introduction

1.1 Introduction

Because of its scalability and adaptability, software defined network (SDN) has gained extensive adoption in the networking academic community and the networking business. With SDN, all of the resources in a network can be managed from one place. There has been considerable worry about security with SDN. Most modern Software security solutions are deployed at centralized controllers, with a primary emphasis on expanding SDN's receptiveness to new kinds of control. Spoofing, tampering, information leaking, and DoS are just some of the forms of attack that may be launched against an SDN distributed denial of service (DDoS). Distributed denial of service (DDoS) attack are particularly harmful because they may drastically reduce the efficiency of an SDN's services and because they are so easy to launch [1].

While SDN design provides certain benefits over the standard network, it is nevertheless vulnerable to network attacks and threats. The cost of licensing security devices and obtaining a global view is a direct reflection of the network dangers to SDN. By passing packets as according flow rules, physically security devices lose their authority to make decisions and pre-deployment attacks become possible. The controller acts as the hub of the network, from which other data may be accessed. The attacker may get a comprehensive understanding of the network by manipulating the controller [2]. For example, attacks on one layer of an SDN will seem quite different than attacks on another. Because the controller oversees the whole network, an assault on the control plane may have far-reaching consequences. In recent years, attack have been focused on controllers [3].

Data centers are one of the most promising new deployment scenarios for software-defined networking (SDN), and the key to their scalability is its centralized control plane. When it comes to the whole SDN, the controller is in charge. They provide orders to the switches and carry out network tasks using frame relay and routing SDN software. It's like the network's central processing unit (CPU), regulating the flow of data and keeping everything moving smoothly. Since the controller is the only node that can provide a bird's-eye view of the whole network, it is the top priority for any attacker. Several statistics and movement rules for network nodes make up the global perspective[4].

Degradation of SDN performance may be brought on by increasing latency, discarding genuine packets, and, extremely large losses. Since the switches in SDN deployments are dumber, malicious traffic cannot be halted. This makes it harder to defend against distributed denial of service attacks. The controller has no way of knowing if the arriving packets are from an attacker or a burst flow. Dropping malicious packets, restricting suspicious traffic, giving priority to scheduling, etc. are all examples of common approaches used in previous solutions. It's possible that implementing these fixes may need purchasing supplementary hardware, sending an extra control message, etc. They are inefficient and expensive. When it comes to classifying network traffic, a vast variety of network capacity and characteristics play an essential role, and Algorithms and SDN play a key part [5]. Numerous DDoS attacks are directed against each of SDN's three layers of abstraction. A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. The

communication channels between both the controller and the switches, as shown in Figure (1.1), enable DDoS attack to target planes of SDN [6].

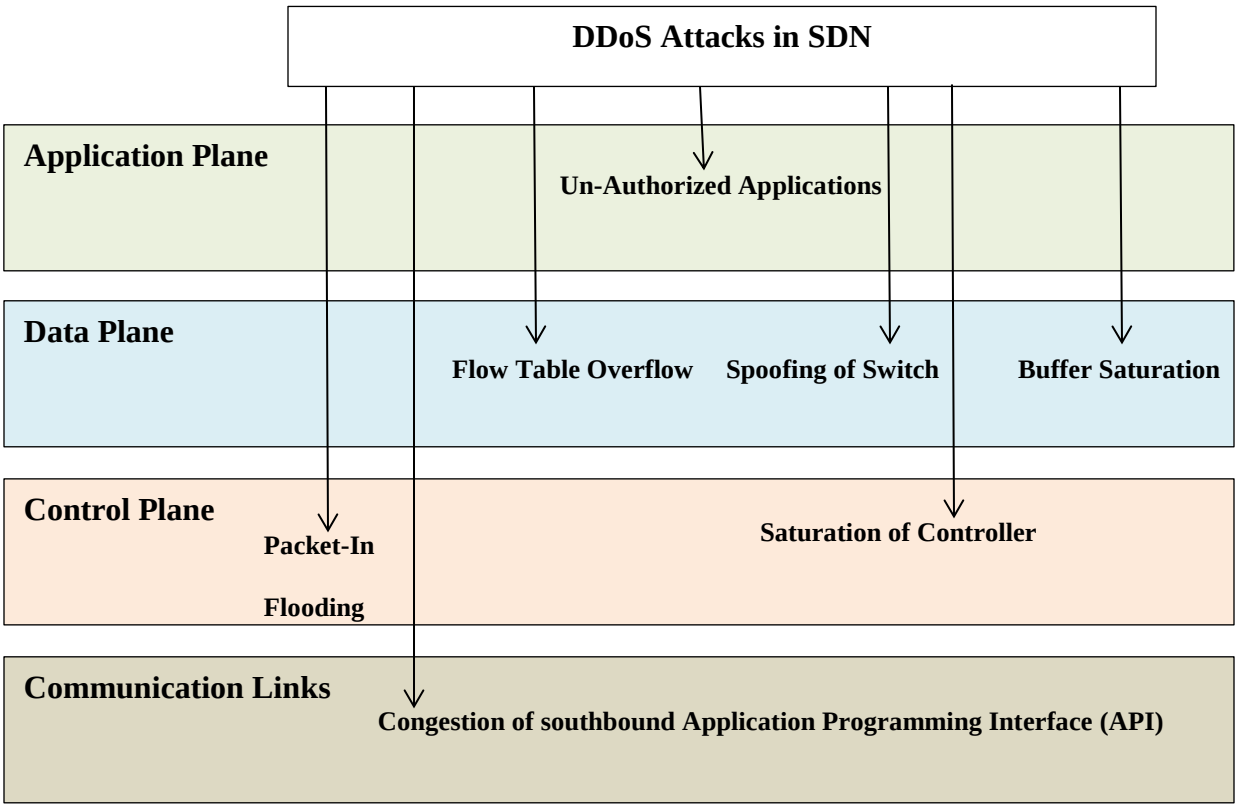


Figure 1.1: DDoS attack at the various SDN tiers[6].

1.2 Related Work

In this section, the most related works in term of improving the security of SDN controller using entropy and machine learning techniques have been discussed and overviewed as follows:

In (Rahman et al., 2019) to identify and prevent DDoS attacks in SDN networks, they compared many machine learning methods, including J48, Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Relatives (K-NN). Training and choosing the optimal model for the planned network and implementing it in a management and mitigation script to identify and mitigate threats were part of the evaluation process. Training and

testing times were much shorter for J48 compared to the other algorithms that were considered [7].

To counter distributed denial-of-service (DDoS) attack, a safeguard mechanism (SGS) was presented in (Wang et al., 2019), The effectiveness of SGS has been shown via simulation. DDoS attacks have definitely shortened the time required for both setup and controller reaction [8].

An examination of the difficulty of categorizing distributed denial-of-service (DDoS) attack over a storage area network (SDN) is presented in (Santos et al., 2020), along with a proposal to use four distinct machine (Support vector machines (SVMs), Multilayer Perceptron (MLP), Decision Tree (DT), and Random Forest (RF). Using the Scapy tool and a database of known good IP addresses to mimic the (Mininet 2.2.2) network, they found that the Random Forests achieved the highest accuracy while the Decision Tree technique achieved the fastest processing time. In addition, the three types of DDoS attack (controller attack, stream attack, and capacity attack) are explained, and the most significant elements for classifying DDoS attacks are shown [9].

In (AlMomin et al., 2020) they suggested a method to detect a DDoS attack that targeting one or multiple victims concurrently by combining two algorithms of Machine Learning (ML), which is entropy and Principal Component Analysis (PCA). Also, they examined the efficiency of our schema through a Mininet emulator and a pox controller and using open vSwitch as a switch. They have obtained high detection accuracy to detect DDoS attacks. [10].

In order to guarantee precise attack detection and effective usage of network resources, (Alamri et al., 2020) proposes a DDoS mitigation system for SDN. They used the Extreme Xgboost (XGBoost) Algorithm and a bandwidth management mechanism to provide a protocol for mitigating the effects of DDoS attack on SDNs [11].

In (Perez-Diaz et al., 2020) describe a highly flexible architecture that facilitates the diagnosis and mitigation of SDN attacks in SDN contexts. Based on training the IDS in the chosen architecture with a variety of ML models, including the J48, Random Tree, REP Tree, Random Forest, Multi-Layer Perceptron (MLP), and Support Vector Machines (SVM), also they compared their results on the Canadian Institute of Cybersecurity (CIC) DoS dataset Denial of Service (DoS) dataset. Despite the difficulties of identifying LR-DoS attack, the findings demonstrate a detection performance of 95%. To do this, they use an intrusion protection detection system (IPS) that uses the open network operating system (ONOS) controllers running on a Mininet virtual machine [12].

The faster and more flexible methodology in (Ramprasath et al., 2021) is used for earlier detection of the unusual traffic flow for sensing the DDoS attacks as well as the ways to mitigate in SDN will mitigate the effects of the DDoS, and show the capability of the proposed technique in improving network performance in comparison to the other methods. Mininet simulations are used to demonstrate the effectiveness of the suggested approach in terms of availability, adaptability, processing waste, cost, and performance [13].

Machine Learning for protecting SDN controllers from DDoS attack is analyzed in (Alamri et al., 2021). It contained on an adaptive bandwidth method, and adaptive threshold approach. The suggested technique leverages the best ML as a strategy for discovering security solutions that increase the security of a SDN controllers and network performance. This strategy employs ML techniques like XGBoost to improve the reliability of security solutions and the efficacy of the network as a whole [14].

A quick and efficient perturbation theory DDoS in SDN detection method was suggested in (Ujjan et al., 2021). Using a generalized entropy calculation based on a combination of Shannon and Renyi entropy, they were able to identify dispersed features of DDoS traffic; this not only helped the SDN controller deal with high volumes of malicious traffic, but also allowed them to collect and analyze data-plane traffic using Convolutional Neural Network (CNN). Additionally, the accuracy vs. false-positive rate tradeoff of Stacked Auto Encoder (SAE) vs. CNN classifiers was explored. In comparison to the CNN classifier's average accuracy of 93.9%, quantitative findings showed that SAE obtained substantially better detection accuracy of 94.9% with just 6.9% of false-positive alerts[15].

In (Yu et al., 2021) a collaborative DDoS assault detection technique based on entropy and ensembles learning has been suggested. The entropy of the edge switch is the foundation on which it rests. At the same time, an ensemble learning-based method is used to properly identify aberrant traffic, and a perfectly alright precise attack module is created in the controller. Simulation findings of two typical DDoS attack techniques, ICMP and SYN, indicate that the system can efficiently identify DDoS attack and

considerably minimize the southbound communication cost and the load of the controller and the detection latency of the attacks [16].

In (Aladaileh et al., 2021) suggested an entropy-based technique to identify high- and low DDoS attacks on the SDN controller. The suggested technique extended the Rényi joint probability for assessing the net traffic flow to identify DDoS assault traffic flow of various speeds. Using two incoming packets characteristics and extended Rényi combined entropy, the suggested technique produced a superior detection accuracy than the efficient detect DDoS attacks against the SDN controller (EDDSC) method that employs entropy metrics [17].

In (Ahuja et al., 2021) they used machine learning to distinguish between normal web traffic and malicious DDoS attacks. The contribution is the discovery of new characteristics for detecting distributed denial of service (DDoS) attacks. Machine learning algorithms are taught using the SDN dataset that is built by logging novel characteristics into a Comma-separated values (CSV) file. The traffic is best classified by the hybrid model combining Support Vector classifier with Random Forest (SVC-RF), which achieves an impressive 98.8% accuracy in testing with a negligible false alarm rate [18].

In (Tayfour et al., 2021) the technique has been suggested based on three parts: a classifier component, a mitigation component, and a collaboration component. InSDN2020, CICIDS2017, NSL-KDD, and UNSW-NB15 datasets were used to verify the proposed classifier's efficacy. Their classifier was also tested on real-world traffic on a simulated SDN. It was showed by the results, it is able to identify DDoS attack with high accuracy by using an ensemble classifier, which is more effective than

individual classifiers. DDoS attack detected and mitigated across several controller domains with little controller effort, and the false positive rate has been considerably decreased [19]

An anti-DDoS method based on machine learning has been utilized in (Sudar et al., 2021). Distributed denial of service attacks include numerous systems working together to overwhelm a single server. The devices in the infrastructure layer are managed by software, and the SDN's control layer acts as a central connecting point between the application and infrastructure layers. Also, they proposed a combination of the machine learning methods Decision Tree and Support Vector Machine (SVM) to identify malicious traffic. The results of our tests demonstrate that the Decision Tree and Support Vector Machine (SVM) method achieves the highest levels of accuracy and detection [20].

In (Abbas et al., 2021) they identify distributed denial-of-service (DDOS) network attacks based on three stages: encoding to transform the original nominal packets into quantitative characteristics, logarithmic method was used to accomplish the data standardization, and principal component analysis (PCA) is conducted eight times for various characteristics. The Random Forest (RF) algorithm is used to extract patterns from data during classification of the types of given features during the training step, and the Naive Bayes (NB) algorithm was also used to classify the data and compare its results to those of the classifier (RF). The best results for detection, with an accuracy of 99.9764% on the MIX dataset, a detection rate of 100%, a false alarm rate of 0%, and an F-measure of 99.9% when PCA = 25 [21].

In (Song et al., 2022) investigated and compared the effectiveness, using various ML approaches, to identify DDoS attack in SDN, in which both experimental datasets or ego traffic data are assessed. In addition, they suggested a simple supervised learning (SL) model for monitoring flow fluctuations as a means of detecting DDoS attack directed at the SDN controller. They used simulations and measurements on an actual testbed to confirm the result. According to the findings, SL can identify DDoS attack using only a single attribute. Both the size of the training set and the parameters utilized affect performance of the evaluated SL algorithms. Prediction accuracy using the same SL models might vary greatly depending on the dataset used for training[22].

To identify DDoS attack in SDN using Entropy, a fusion entropy approach was developed in (Cong Fan et al., 2022). The quickness with which attacks may be detected and the noticeable drop in entropy value are both benefits of this approach. The complementary nature of information volatility and log fuel entropy is used here. Based on experimental findings, the output value of attack scenarios is 91.25 percent lower than normal situations, which has larger benefits and relevance than previous attack detection approaches [23].

In (Musumeci et al., 2022), they compared two DDoS Attacks Detection (DAD) structures, standalone and associated DAD, built inside a single entity to investigate the use of AI and Ml techniques to execute autonomous car Ddos in SDN, with a focus on Transport - Layer Flooding attacks attack such as SDN controller). they tested an accuracy, precision, recall, and F1-score of above 98%[24].

Table 1.1 illustrates the aims of previous researchers and the main methods which are used to achieved the works.

Table 1.1: The literature survey summary

Ref, Year	Methods	Aims	Results
[7], 2019	J48, RF, SVM, K-NN	Detecting and block the DDoS attack in an SDN network	J48 performs better than the other evaluated algorithms
[8], 2019	SGS	Protecting control plane against DDoS attacks	setup time and controller response time have been reduced
[9], 2019	SVM, MLP, DT, and RF	Classifying DDoS attacks in an SDN	Detecting the three kinds of DDoS attacks discussed as (controller attack, flow-table attack, and bandwidth attack)
[10], 2020	Entropy and Principal Component Analysis (E-PCA)	Identifying DDoS attacks	high percentage of accuracy to detect the attack
[11], 2020	XGBoost	DDoS mitigation scheme for SDN	99.9% accuracy in detecting DDoS attacks with a low false-positive rate of 0.0002% in SDN.
[12],2020	J48, RT, REP Tree, RF, MLP, SVM, and IDS	Mitigating of DDoS attacks in SDN	ML detection rate of 95%, IDS detection system mitigates all attacks
[13],2021	Multinomial Regression	Detecting the DDoS attacks	Enhancing reliability, flexibility, processing overhead, cost, and throughput
[14],2021	XGBoost	Securing the SDN controller targeted by DDoS attacks	Enhancing the accuracy of the security solutions and improve the overall network performance.
[15],2021	Shannon and Renyi entropy, SAE, and CNN	DDoS in SDN detection	SAE achieved accuracy of 94% with only 6% of false-positive alerts, whereas the

			CNN classifier achieved an average accuracy of 93%
[16],2021	Entropy	DDoS attack detection	Reducing the southbound communication overhead
[17], 2021	Rényi joint entropy	Detecting low-rate and high-rate DDoS attacks against the SDN controller	Better detection rate than Shannon entropy metrics
[18],2021	Support Vector classifier with Random Forest (SVC-RF)	Classifying the benign traffic from DDoS attack traffic	Highest testing accuracy of 98.8% with a very low false alarm rate
[19],2021	V-NKDE (Voting - Naive Bayes, K Nearest Neighbors, Decision Tree, and Extra Trees)	Detecting DDoS attacks accurately	Detecting DDoS attacks with high accuracy using an ensemble classifier.
[20],2021	Decision Tree (DT), and Support vector machine (SVM)	Preventing the DDoS attack	The best results are: Decision Tree(DT 78 % Support vector machine(SVM) 85 %
[21],2021	Random Forest (RF), and Naïve Bayes (NB)	Detecting (DDOS) network attack	The best results are : Random Forest (RF) 99.9764 %, and Naïve Bayes (NB) 95.5469 %
[22], 2022	SVM, GLM, NB, DA, FNN, DT, KNN, BT	Building a new model to define DDoS attacks in a flexible way	BT has an accuracy of 99.46%, which is the best among the eight SL techniques they considered.
[23],2022	Fusion Entropy	Detecting DDoS attacks in SDN	entropy value of the attack scenarios 91.25% lower than normal scenarios
[24], 2022	KNN, SVM, RF, and ANN	Automated DDoS Attacks Detection (DAD) in SDN	Accuracy, precision, recall and F1-score are above 98% in most cases

1.3 Problem Definition

The proposed system is focused on the main problems, which they are explained as follow:

- Solving high data traffic from southbound malicious nodes attack that effect on the controller in the SDN.
- Congestion at the southbound interface. For example, if a traffic monitoring application listening to packet in events arriving from switch towards the controller suddenly bombards the network at a high rate, then this leads to the occupation of bandwidth of southbound channel caused by a new-flow based DDoS attack[25].
- Loss of confidence in controller-to-southbound application communication and an accompanying rise in the rate at which data is lost as a result of malicious attack.

1.4 Aims of the Study

The main aims of the study are :

- Control data traffic though training the system about maximum data rate for normal data traffic and discard malicious packets.
- Preventing unauthorized access to the network resources through matching randomness for real data traffic(real-time detection of a DoS attack) of incoming requests directly in the controller to classify into normal and abnormal traffic.
- Monitor, and protecting the SDN controller from DDoS attacks congestion through monitoring network behavior of normal state and recognizing the DDos attack state with the proposed machine learning algorithms.

- Reducing the timeout (delay) values of the flow table entries in the southbound layer to prevent them from being flooded by blocking malicious attack traffic in the network.

1.5 Thesis Contribution

The major contributions are represented by :

- Monitoring incoming packets by SDN controller, and detecting malicious packets of DDoS attack by keeping an information list of malicious packet types for future identification, and then mitigation DDoS attacks when discover new rule of flow table data traffic which it is not well-defined, the controller updated the traffic table row to block these malicious traffics from being calculated again.

1.6 Thesis Outline

Furthermore, this thesis contains four chapters in addition to chapter one:

Chapter Two: Theoretical background of the proposed system with the Software Defined Network (SDN), SDN Security, DDoS, Application layer- Control layer- Infrastructure layer of DDoS attack, and the main DDoS attack detection techniques have been explained such as the detection techniques based on entropy, and machine learning algorithms. In addition, the data analysis, taxonomy of machine learning algorithms as supervised Learning/Predictive Models, Unsupervised Learning, and Semi-supervised Learning also sorted. Furthermore, in this chapter the used Dataset, Evaluation metric, and the used implementation tools are showed as Mininet, Virtual Box, Azure cloud classic, Wireshark, Iperf, and ASP.NET.

Chapter Three: The proposed system approach with the proposed system steps for implementation with two phases are the off-line phase, and the On-line (Real-Time Traffic) phase, in addition, the used methods of On-line phase are entropy, and the trained model with DT.

Chapter Four: It describes the results and evaluates the used system.

Chapter Five: It presents the results conclusion, and, it described the future works suggestions.

Chapter Two

Theoretical and Technical Background

2.1 Introductory

The widespread use of Internet technology has entered a period of control, convenience, and security. With the rapid rise of new applications like e-health, e-commerce, and unmanned aircraft, etc., which need complex network regulations and challenging networking tasks, there has been a rising demand for a new and better approach to managing communication networks[25].

Most of the issues that have plagued communications systems, such as their static nature and the complexity of their management, have been resolved with the emergence of Software Defined Networking (SDN). There may be less labor involved in managing networks using SDN, as the underlying network architecture may be handled with greater dependence on software rather than equipment. By separating the data and control planes, the system's flexibility is increased [26].

The SDN is centralized network control using logically centralized control platforms , the operation of the entire network is dependent on those controllers. Despite the benefits, the isolation of the planes makes it easier to fingerprint the controllers and target them for Denial of Service (DoS) or Distributed DoS (DDoS) attacks. DDoS is a large network attack. It is continually expanding, with new ways to attack the system. As the Internet grows in popularity, more hosts become exposed to these attacks [27].

2.2. Software Defined Networks (SDN)

In Software Defined Networks (SDN), network management is made very simple by decoupling the control plane, and data plane form network elements, where the control plane is centralized as a directly

programmable controller and the network elements take care of packet forwarding operations. An administrator may improve the network's flexibility by implementing a set of dynamic rules and policies. According to Figure (2.1) showed the SDN Architecture. SDN facilitates the application to program the network operations through the open northbound API, OpenFlow protocol acts as the southbound API between the controller and forwarding elements [28].

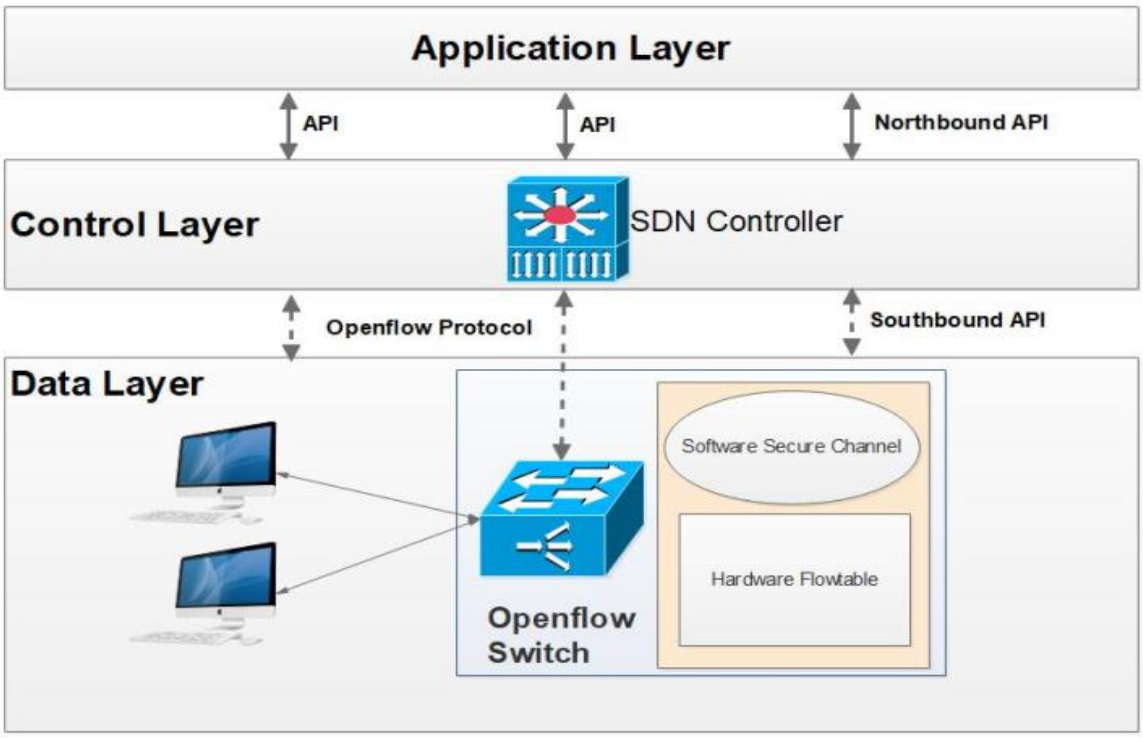


Figure 2.1: The architecture of SDN [28].

It is possible for a contemporary SDN controller to interact with an older network by using SDN adapters. The key elements of an SDN controller and switch interaction are the open virtual switch (OVS) switch kernel components, the OVS-switched daemon that works over the modules and the OpenFlow methods used as the southbound API. The forwarding component's control agent uses the southbound API of the OpenFlow protocol to carry out

all control orders from the controller [28]. In a switch, the forwarding table is managed and kept up to date by the OVS kernel module. The controller in an SDN sets up the forwarding table, which follows a set of operational and administrative guidelines. OpenFlow is a protocol that allows for asynchronous communication and coordination in a software-defined networking setting. Using software-based switches rather than dedicated hardware for SDN helps bring down infrastructure expenses [29].

2.3 Software Defined Networks (SDN) Security

The SDN's centralized control method paves the way for future gains in network infrastructure efficiency. To put it another way, it enhances a network system's usefulness and its programmability. The platform has also brought new security challenges. There are several issues that concern the SDN system [29].

- An attack on an access point or switch might cause a network to malfunction briefly or perhaps completely .
- Since SDN is logically centralized, an attack on the control plane has the potential to bring down the whole network infrastructure. Control plane attacks are a well-known issue in the SDN security.
- Communication security flaws the system administrator has the option of disabling the OpenFlow protocol, despite the fact that it employs a security mechanism like Transport Layer Security (TLS) to regulate data transport. It opens the door for "man in the middle" attacks. It is a major problem because an attacker may generate phony traffic streams to flood the network channels and create disruption. Overloading the

system's resources and causing damage is the goal of both DoS and DDoS attacks.

- Attacks on application programmable interface APIs provided a great advantage to the platform, but they also posed new challenges in terms of network security.

The use of certain protocols is required for managing this issue. Any weaknesses in the security measures for SDN architecture would leave the network vulnerable from known threats and attacks [30]

Therefore, it is critical to ensure the safety of each component of an SDN by taking the necessary precautions. The safety of the SDN controller is crucial since it manages the whole network. The entire network might fail if this does not take place. Security solutions that may be dynamically updated are necessary to defend the network infrastructure against the ever-evolving threats and attacks [30].

2.4 Distributed Denial of Service (DDoS) attack

DDoS attacks have been plaguing many kinds of networked applications and systems for quite time. The primary objective of a DDos attack is to overload the targeted service with requests from several sources. An attacker may, for instance, send a victim millions of packets in an effort to overload their accessible bandwidth with malicious content, making the victim's internet providers inaccessible [31].

A kind of DoS attack, a distributed DoS attack targets several hosts simultaneously. DoS attacks often involve the hacker using a single network connection to flood the target with bogus requests or attempt to exploit a security flaw. DoS attacks are more extensive in scope. To do this, it employs

a network of hundreds (or possibly millions) of interconnected nodes. DDoS attacks are notoriously difficult to counter because of the sheer number of connected devices. DDoS attacks are difficult to detect since their signs are common. Many of the indicators are familiar to anybody who regularly uses technology: poor upload or download speeds, the site becoming impossible to access, a failed internet connection, strange video and material, or an abundance of spam [32].

2.4.1 Application layer DDoS attacks

Application-layer malware attempts to consume all accessible system resources and block additional requests from being handled. This makes it harder to detect attacks on the network, while there is no clear indication of malicious vs benign data. Since application and resource separation are not entirely addressed in SDN, DDoS attacks on one application may have ripple effects on other applications, for example about application layer DDoS attacks are low-and-slow attacks, GET/POST floods, attacks that target Apache, Windows or OpenBSD vulnerabilities [33].

2.4.2 Control layer DDoS attacks

SDN controllers and the connectivity they use are vulnerable to a variety of attacks. Dangers that may cause significant harm include attacks on the control and data planes and connections between the controller and other parts of the network, such as the northbound API, the southbound API, the west/eastbound API. Since the controller is both a focal point and a scaling constraint, there is also the risk of performance concerns and control plane unavailability. Examples about control layer DDoS attacks are Mac spoofing, Mac flooding, ARP spoofing and poisoning [34].

2.4.3 Infrastructure layer DDoS attacks

The foundational level of infrastructure DDoS attacks may enter a system via switches or the southbound API. For example, in a DoS threat, the attacker generates several new flows and floods the node with traffic. tools required for construction, so an infrastructure attack is a DDoS attack that overloads the network infrastructure by consuming large amounts of bandwidth, for example by making excessive connection requests without responding to confirm the connection, as in the case of a SYN flood. A proxy server can protect against these kinds of attacks by using cryptographic hashtags and SYN cookies. For example about infrastructure DDoS attacks are TCP Syn Attack, TCP ACK Attacks, Tear drop attack (TCP Fragment-Error in Fragmentation Reassembly),Ping of Death Attack, ICMP Attack, Smurf Attack [35].

2.5 DDoS attack detection techniques

Numerous studies have shown the typical characteristics of DDoS attacks. SDN is a promising new architecture, but it is still early in its development, hence studies on it are still in their infancy. Additionally, SDN networks provide unique detection algorithms for various DDoS attacks. Methods based on entropy, machine learning, traffic pattern recognition, connectivity rate, and systems that combine the intrusion detection system and open flow are all examples [36].

2.5.1 Detection Techniques based on Entropy

One of the best ways to measure how disordered a system is via the usage of entropy. Useful definitions and interpretations may be found in the works of both C.E. Shannon (1948) and N. Wiener (1961). Since entropy-based methods can quantify unpredictable data in incoming packets, they hold great promise for DDoS detection. Entropy is proportional to the measure of randomness and rises as uncertainty grows. DDoS attacks can be identified when the entropy of flow counts at each instant deviates less than the threshold value, which is adjusted adaptively depending on traffic pattern situation to increase the detection accuracy. The managers of a SDN system might potentially identify network anomalies based on predefined criteria [37]. Spatially variable of other network characteristics, such as the Source IP, the Destination IP, and the port numbers, are also used into the entropy calculation.

As shown in equation (2.1) [37], the entropy of the probabilities associated with the random process X is denoted by the natural logarithm (pi,..., pn).

$$E(x) = -\sum_{i=1}^{n} P(i) \quad log_2(P\ (i)) \quad \ldots\ldots. (2.1).$$

Example 2.1 showed the Entropy explanation :

- The number of events : n = 6
- probability of one event: p = 1.0 /n
- Calculate Entropy
- Entropy = -sum([p * log2(p) for _ in range(n)])

Entropy= 2.585 bits.

2.5.2 Detection Techniques based on Machine learning algorithms

To spot anomalies in a network setting, it may use models and machine learning techniques, both supervised and unsupervised algorithms. Such algorithms take into account the overall state of the network and the characteristics of the traffic in order to spot anomalies[38].

Any system built to identify network anomalies must first collect the traffic on the network and extract some form of useful data from it [39]. Machine learning is a method that attempts to discriminate between normal and abnormal data transfer without explicitly understanding the pattern.

2.6 Data Analysis

The goal of this strategy is to unearth previously unknown patterns. Once these tendencies have been recognized, they may be used to inform strategic choices about the development of their enterprises. Massive amounts of raw data surround us, yet they cannot be processed promptly by humans or manually controlled software[40].

The amount and appropriateness of such data is crucial for the achievement of the provided essential by a data mining approach in any framework, in addition to the design and function of the method. This is due to the fact that the functionality and design of a data mining method determine how well the information it extracts performs in a given context. The Knowledge Discovery in Databases process showed in Figure (2.2) [41].

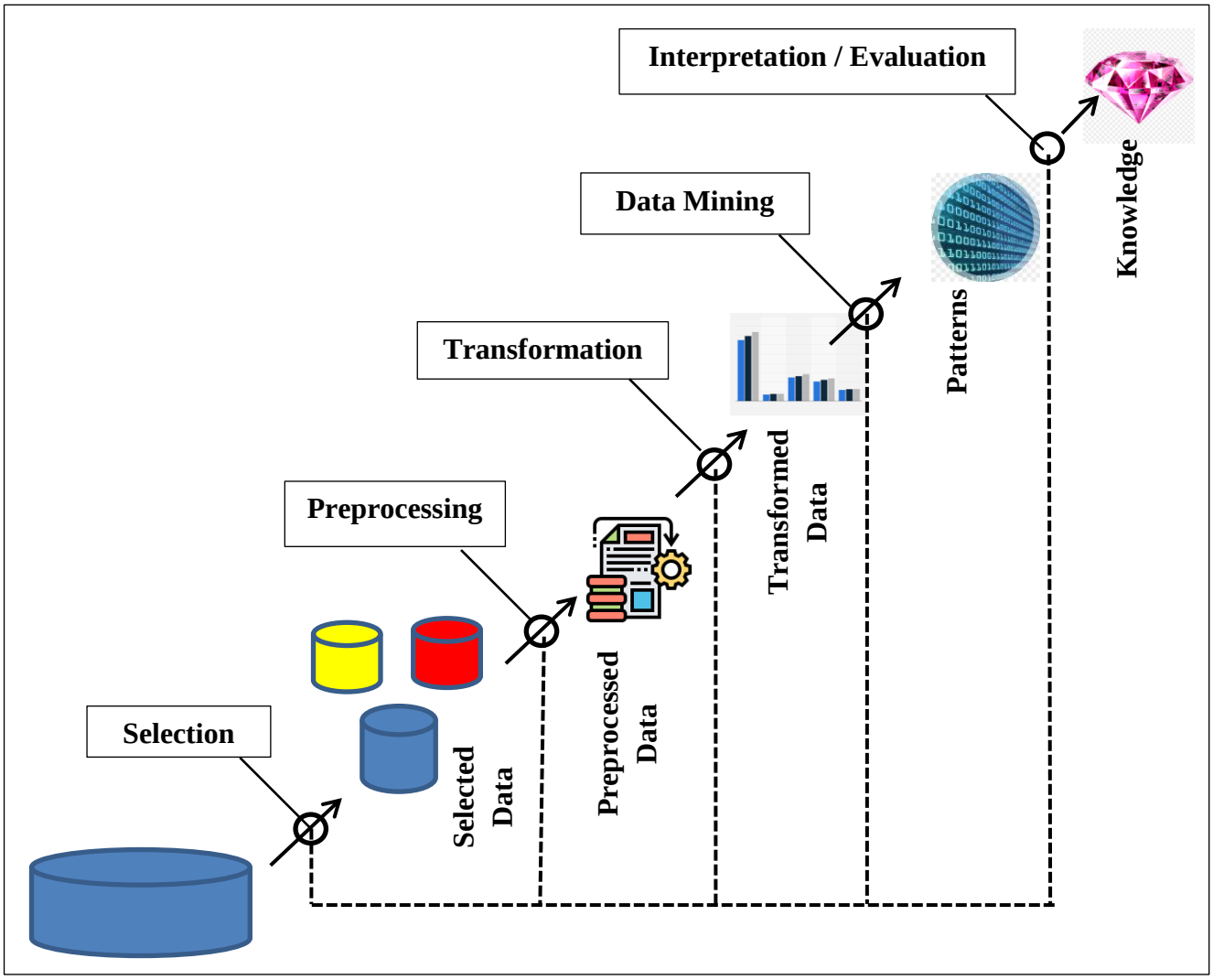


Figure 2.2 : The Knowledge Discovery in Databases (KDD) Process[41].

2.6.1 Data preprocessing

It describes the assortment of methods utilized before deploying a data mining approach, and is often recognized as one of the most pertinent problems within the well-known as extracting information from Data process. There are typically seven steps that must be accomplished before the information retrieval process can be regarded complete [42].

1- Data Integration: It involves compiling data from various sources.

2- Selecting useful consistent data

3- Missing values, and other blunders in the data as the data preparation.

4- Transforming Data: Normalization, Smoothing, and other data Mining-relevant activities.
5- Mining Data: Using extraction methods to uncover hidden pattern.
6- Display patterns, which includes both the presenting of patterns visually and the removal of duplicates.
7- Preprocessing operations to make decisions, the preprocessing tasks showed in Figure (2.3) [43].

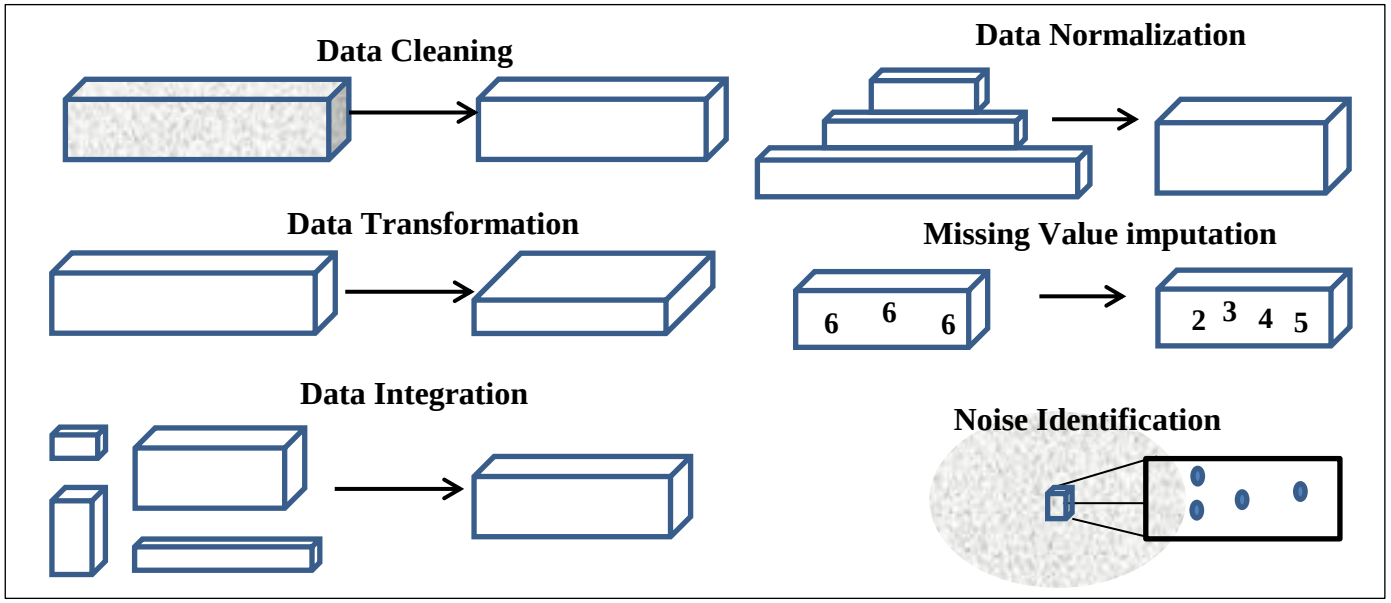


Figure 2.3.: Data Preprocessing Tasks[43].

Methods of data preprocessing include the following examples:

A- Imperfect Data

The majority of data mining techniques need a data set that has been idealized and is devoid of noise. On the other hand, the data that is obtained in the real world is often neither perfect nor exhaustive. During the process of data preparation, several methods are often employed to either reduce noise or impute (or fill in) missing information[44].

B- Missing Values

Many techniques of data mining begin with the assumption that the set of data is complete. Unfortunately, during the whole of the acquisition procedures, there are often instances of missing data. Knowledge that was not collected due to oversights in the sample technique, budgetary restrictions, or other hurdles to data collection are examples of what are known as sampling errors [45].

C- Feature Indexers and Encoders

The transformation of characteristics from one form to another may be accomplished via these procedures by using indexing or encoding algorithms [46].

- **StringIndexer** is one example of such a tool, and it used to convert a column containing strings into a column containing numerical indexes. The order of the indices may be determined based on the frequency of the labels[47].
- **OneHotEncoder** is responsible for transforming a string column into a binary vector column. This encoding provides a better description of category qualities than the previous methodology did since it does away with the number ordering that was enforced by that method[48].
- **VectorIndexer** indexes categorical features inside of a Vector. It decides which features are categorical and converts them to category indices [49].

2.7 Taxonomy of Machine Learning Algorithms

In order to select an acceptable algorithm for machine learning, it is important to compare the characteristics of the data that will be learned with

those of the techniques that are already in existence. There are basically three distinct categories of machine learning algorithms are as follows[50]:

2.7.1 The Supervised Learning/Predictive Models

In most cases, supervised learning methods are used during the construction of prediction models. Using the other values in the dataset, a modelling approach is used in order to derive a missing value from the dataset as a whole[51]. Supervised learning is an instance of a learning algorithm. It takes in data and provides an output, then utilizes the input and output to build a model that can reliably predict how a trained algorithm will respond to a new dataset. Instances of supervised learning include decision trees, SVM, and many more types of learning systems [51].

A- Support Vector Machine(SVM)

It is a kind of controlled machine learning technology often used to classification problems. The main objective of the support vector machine is to locate the optimal dividing hyperplane in order to get a result that maximizes the margin of the training data [52]. The decision boundary should classify all points correctly. Both inequalities can be summarized in a more compact form by Equation (2.2) [52]:

$$y(wT.xi + b) \geq 1, i = 1,2, \dots, N \quad (2.2)$$

B- Decision Tree (DT) algorithm

It is made up of internal nodes of the tree, which are labeled with the term, branches that branch out from them and are labeled with a particular sample on the weight, and leaf nodes that indicate the class labels that correspond to those labels[53]. A decision tree may classify a text document beginning at the root and proceeding through the query architecture until it achieves a certain leaf, which indicates the goal for document classification.

Because it requires the constant exchange of training tuples, the majority of the training data cannot be used in the development of the decision tree memory [54]. This is because it is inefficient. In order to design a decision tree, it was essential to calculate two distinct types of entropy using frequency analysis. The entropy of a single feature may be calculated based on its frequency table as it showed in Equation (2.3)[54]:

$$E(S) = \sum_{i=1}^{c} -P_i \, log_2 \, P_i \qquad (2.3)$$

Besides, Algorithm (2.1) showed the main DT algorithm steps.

Algorithm (2.1): The used Decision Tree (DT) Algorithm

Input: Training set S, features F, Class Y, number of trees attributes B, the Weight H.

Output: DT classifier

Begin

Function Decision Tree(S,F)

H=0

For i = 0 to B

- si = sample subset of S
- hi = DecisionTreeLearned (si,F)
- H = H + hi

End for

Return H

End function.

Function Decision Treeearned (S, F)

- At each Node
- f = small subset of F
- spilt on best feature of f

return learned DT classifier

End function

End

End of Algorithm

It is a development of the decision tree concept. A collection of choice Directed Acyclic Graphs (DAGs)is what makes up a decision tree. A decision DAG often has greater generalization capability than a single tree because it allows large trees to merge into one another. However, this comes at the expense of a somewhat longer amount of time-spent training the model. DAGs offer the benefit of being able to do combined feature selection and classification, and they are durable when dealing with noisy features[56].

Figure (2.4) presents an illustration of a DAG's organizational structure. It makes better use of available memory due to the fact that it does away with the need of duplicating leaf nodes and enables branches to combine, but on the other hand, it requires greater processing time [57].

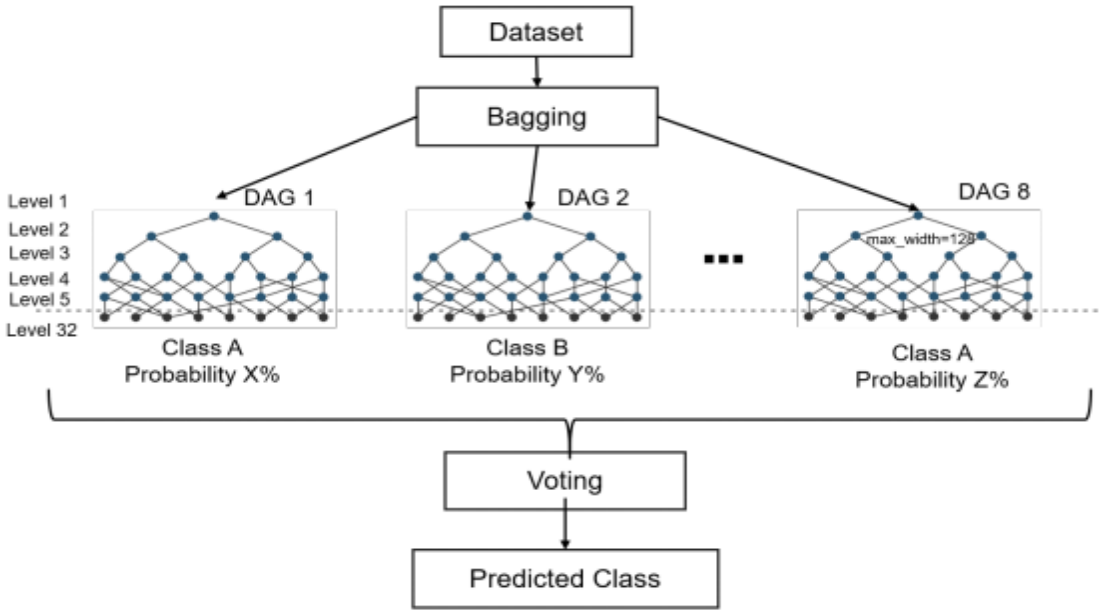


Figure 2.4: Parameterization of a DAG [57].

Algorithm mathematical model shown in Equation (2.4)[57].

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (fi - yi)^2 \qquad (2.4)$$

N is the total number of observations, f(i) is the result from the model, and y(i) is the true value at observation i. The formula determines which path through the forest is the best option by calculating the distance between the actual value and the anticipated one at each node. The decision tree's output, denoted by fi, is based on the value of the input testing point located at node I which is denoted by yi. The Algorithm(2.2) for a Decision Tree (DT) [58].

Algorithm (2.2): The standard Decision Tree (DT)
Input : dataset = pd.read_csv(path, names = headernames) **Output :**Subset of features
1- Initial variables - X = dataset.iloc[:, :-1].values - y = dataset.iloc[:, 4].values 2- Building dataset.head() 3- Call main class sklearn.ensemble import DT main class 4- Building the classifier 5- classifier = DecisionTreeClassifier(n_estimators = 50) classifier.fit(X_train, y_train) from sklearn.metrics 6- Define metrics - multi-classification_report, confusion_matrix, accuracy_score 7- Evaluating and prints results - result = confusion_matrix(y_test, y_pred) - print("Confusion Matrix:") - print(result) - result1 = multi-classification_report(y_test, y_pred) - print("multi-Classification Report:",) - print (result1) - Output: result2 = accuracy_score(y_test,y_pred) - print("Accuracy:",result2) **End Algorithm**

C- Neural Network (NN)

Rather than dependent on a single neuron, which uses comparable output data when there is a binary outcome that can contain a large number of binary receptors and thus help in the process of number classes used, this solution offers a natural enhanced version to the challenge that relates to

multiple layers. It does this by providing a logical extension to the problem that corresponds to multiple layers. The equation (2.5) used for entropy that is derived from the frequency distribution table of two characteristics [55]:

$$E(T, X) = \sum_{c\,x} P\,(c)E(c) \qquad (2.5)$$

2.7.2 Unsupervised Learning

The unsupervised learning approaches are used to create descriptive models. The model's output is a unknown, but it know the inputs. In most cases, transactional data is employed while using unsupervised learning. Some examples of clustering algorithms are the k-Means clustered and k-Medians clustering [59].

2.7.3 Semi-supervised Learning

The training dataset for a semi-supervised learning approach may include both unlabeled examples. Semi-supervised learning encompasses a variety of methods, including classification and regression. Examples of regression methods include logistic regression and linear regression [60].

2.8 The used Dataset

CICDDoS2019 includes safe and up-to-date typical DDoS attacks, simulating actual data from the real world (PCAPs). Network traffic analysis findings from CICFlowMeter-V3 are also included, together with labeled flows based on timestamps, IP addresses, ports, protocols, and attacks (CSV files) [61].

There are a total of 50,063,112 records in the CIC-DDoS2019 dataset, consisting of 50,006,249 rows related to DDoS attacks and 56,863

rows related to normal traffic. Each column consists of 86 characteristics. Table (2.1) provides an overview of the attack statistics in both the training and testing phases of the dataset[62].

The training dataset includes 12 DDoS attacks, such as those against Network Time Protocol (NTP), Domain Name System (DNS), Lightweight Directory Access Protocol (LDAP), Microsoft SQL Server (MSSQL), NETwork Basic Input Output System (NetBIOS), Simple Network Management Protocol (SNMP), Simple Service Discovery Protocol (SSDP), User Datagram Protocol (UDP), UDP-Lag, WebDDoS, SYN and TFTP; the test dataset includes 7 attacks, such as those against Microsoft SQL Server Resolution (MSSQL), Network Basic Input/Output System (NetBIOS), PortScan, Lightweight Directory Access Protocol (LDAP), User Datagram Protocol (UDP), UDP-Lag, and SYN in testing day[63]

- **NTP-based attack** : a reflection-based DDoS attack is one that takes use of the functionality of Network Time Protocol (NTP) servers to bombard the victim client-server or even other network with an overwhelming volume of User Datagram Protocol (UDP) data traffic. The target and its supporting network infrastructure may become unreachable to legal traffic as a result of this kind of attack [64].
- **DNS-based attack** : they are a kind of consequence DDoS attack in which a Botnet has been used to produce a flood of resolves requests to a specified IP address [65].
- **LDAP-based attack** : reflection-based DDoS attacks begin when an attacker can send queries to such a publicly accessible vulnerable LDAP server. It exploits security loopholes caused by un-sanitized user input data. LDAP injections create malformed queries to gain access in order to

potentially change data in a directory. LDAP queries contain special characters such as asterisks, brackets, ampersands and quotes [66].

- **MSSQL-based attack**: reflection-based Distributed Denial-of-Service (DDoS) attacks, or MSSQL-based attacks, occur when an attacker uses the Microsoft SQL Server Authorities Impose (MC-SQLR) by launching programmed queries from a spoofed IP address, making them seem to originate from the targeted server[67].
- **NetBIOS-based attack**: an attacker using a reflection-based DDoS attack on NetBIOS might cause a vulnerable computer to reject all network traffic by sending faked "Name Release" or "Name Conflict" notifications[68].
- **SNMP-based attack** : one such volumetric DDoS attack is based on the Simple Network Management Protocol (SNMP), and it may create attack volumes in the hundreds of gigabits per second needed to choke the target's network pipes[69].
- **SSDP-based attack** : an SSDP-based attack is a reflection-based Distributed Denial-of-Service (DDoS) attack in which the attacker transmits a mirrored copy of traffic to the victim Universal Plug and Play (UPnP) networking protocols, resulting in a flood of data for the victim to process[70].
- **UDP-Lag-based attack**: it use IP packets carrying UDP datagrams in an effort to saturate the network connection of the victim host and disrupt its operation[71].
- **WebDDoS-based attack** : the Web DDoS attack is a kind of attack that compromises a Web server or application by using seemingly innocuous HTTP GET or POST requests[72].

- **SYN-based attack**: the SYN-based attack makes advantage of the regular TCP three-way handshake (that is, sending SYN, transmitting SYN-ACK, and returning with an ACK) in order to deplete the resources available of the victim's server computer and deliver it inoperable[73].
- **TFTP-based attack**: as its name suggests, an attack based on the TFTP uses online TFTP servers to get access to sensitive information. In this case, an attacker makes a default request for a document, and the victims TFTP server returns the data to the attacker's target host[74].
- **PortScan-based attack**: a port scan may be performed on a single computer or on a whole network as part of a port scan-based attack. During the scanning process, questions are asked of the remote server to learn what services it offers [75].

Table 2.1: Attack types in CICDDoS2019 dataset.

Attack Type	**Flow Count**
Benign	56,863
DDoS_DNS	5,071,011
DDoS_LDAP	2,179,930
DDoS_MSSQL	4,522,492
DDoS_NetBIOS	4,093,279
DDoS_NTP	1,202,642
DDoS_SNMP	5,159,870
DDoS_SSDP	2,610,611
DDoS_SYN	1,582,289
DDoS_TFTP	20,082,580
DDoS_UDP	3,134,645
DDoS_UDP-Lag	366,461
DDoS_WebDDoS	43

2.9 Evaluation metric

The evaluation metrics are based on the confusion matrix as a technique for summarizing the performance of a classification algorithm, and classification is a process of categorizing a given set of data into classes.

The evaluation metrics were defined based on the confusion matrix, as shown in Equations (2.6) to (2.9).

A- Precision

It is the total number of true positive (TP) divided by total of all TP and False positive P. Exactness may be calculated using the equation (2.6) [76].

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (2.6)$$

B- Accuracy

It's calculated by dividing the number of accurate forecasts by the overall number of forecasts. Accuracy can be calculated using the following equation (2.7) [77].

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TP+TN+FP + FN}} \quad (2.7)$$

C- Recall

It is calculated by dividing the sum of all TP by the sum of all FN. equation (2.8) provides a formula for calculating this measure [78].

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (2.8) .$$

D- F1-score

It is the result of 2*((precision*recall)/ (precision + recall)). The f1-score or f1-measure are two more names for this concept. In order to get the formula of this measure, it used equation (2.9) [79].

$$\mathrm{F1-score} = \frac{(2*\mathrm{TP})}{(2*\mathrm{TP}+\mathrm{FN}+\mathrm{FP})} \quad (2.9)$$

2.10 Confusion Matrix

For model evaluation, the confusion matrix is utilized to determine metrics including accuracy, precision, recall, and f-measure. The results of a model are summarized in a confusion matrix. Confusion matrix describes the performance of a model on set of test data. It gives two types of correct predictions and two types of incorrect predictions for the classifier. The resulting confusion matrix is shown in Table (2.2). The TP prediction, the TN prediction, the FP prediction, and the FN prediction all describe different outcomes. Here are several definitions[80] for accuracy, precision, recall, and f-measure (f-score)[80]:

- Different metrics may be used to assess the effectiveness of the categorization method. The confusion matrix was selected as the primary tool for measuring performance. A classification method's or "classifier's" performance on test data was previously described using this term [81].
- **True Positive (TP)** is a case that has been correctly labeled as a positive example.
- **False Negative (FN)** is an uncorrected positive instances

- **False Positive (FP)** is used to describe the incorrectly predicted and categorized negative occurrences.
- **True Negative (TN)** refers to the negative examples that the categorization model correctly predicted[82].

Table 2.2: Confusion Matrixes.

Confusion Matrix		**Predicated Class**	
		Positive +	**Negative -**
Actual Class	**Positive +**	**TP**	**FN**
	Negative -	**FP**	**TN**

2.11 The utilized Tools and Softwares

The proposed system is implemented, nd tested with two phases as on-line and off-line phases with the following tools:

2.11.1 Mininet

It is a software-defined-network (SDN) simulation program that allows for the creation of a network topology comprised of many virtual hosts, switches, controllers, and connections. Mininet's primary benefit is that it facilitates the rapid deployment and evaluation of SDN and console application protocols [83]. Prototyping a huge virtualized network infrastructure may now be done quickly with only a single standard computer in a simulated setting. Using a basic virtual operating system, it provides different prototypes of scalable networks based on software like OpenFlow. This raw material facilitates the rapid development, interaction, and customization of SDN prototypes. To name a few of Mininet's [83] features,

it easy and inexpensive to put development networks based on the OpenFlow API model, the features are sorted as follow:

- To facilitate several researchers' independent exploration of a shared network architecture.
- Making it possible to try out complex topologies before deploying them in the physical network.
- Features network debugging and test running utilities.
- Many topologies are supported, and a core set of topologies is provided[84].

2.11.2 Virtual Box

VirtualBox is a free and open-source virtualization application compatible with Windows, Linux, Mac OS, and other platforms. It allows users to construct virtual disc units in which a guest operating system may be installed and used in the same manner as if it were truly installed on the machine[85].

The system's virtual machine is highly adaptable, it provided a tools to modify its CPU, RAM, and storage to suit your specific demands. It managed to run and interact with both the hosting and guest operating systems [85]. The software works with Windows 10, macOS Yosemite, and the most recent releases of Ubuntu and other Linux distributions. One apparent advantage of virtualization is the ability to run platform-specific applications; further advantages include the portability of virtual machines and the ease with which backups may be created [86].

2.11.3 Azure cloud classic

One kind of Microsoft's Azure Cloud Services (PaaS). This solution, like Azure App Service, is meant to back up scalable, dependable, and low-cost apps. Similarly, to how App Service runs on VMs, Azure Cloud Services uses this technology to power its infrastructure. But with virtual machines, it provided more command in advanced mode. Virtual machines hosted by Azure Cloud Services allow for the installation of user-provided software and remote access [87].

Microsoft has successfully thwarted a huge number of distributed denial of service (DDoS) attacks. It offers extensive defenses against distributed denial of service (DDoS) attacks on layers three (L3) and four (L4), which include TCP SYN, new connections, and UDP/ICMP/TCP floods. Microsoft DDoS Protection takes use of Azure's worldwide deployment size, is decentralized, and provides 60Tbps of global attack mitigation capacity [88].

In this thesis Azure used to analysis traffic in real time. The DDoS attack on SDN with DDoS protection standard is detected and mitigated by monitoring from metrics to find public IP is under DDoS attack (Detect DDoS attack) through:

- Azure Portal, Resource Group, VM-SDN, Metrics, Select specific Public IP in resource option: Under DDoS attack or not in metrics filter.

In addition, monitoring SDN controller from metrics to find public IP inbound packets status (Detect DDoS attack).

- Azure Portal, Resource Group, VM-SDN, Metrics, Select options from metrics filter: inbound packets DDoS, inbound packets dropped DDoS, and inbound packets forwarded DDoS

2.11.4 Wireshark

Wireshark is a free and open source packet analyzer. The packets in this tool can be captured directly from the network and displayed in a human-readable format [89]. It is used for network troubleshooting, analysis, software development, communication protocols, education, filtering packets based on special criteria, generating statistics, etc. The project was called in Originally called Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues [90].

2.11.5 Iperf

It is a tool for network performance measurement and tuning [91]. It can be used for measuring the throughput between the two ends in one or both directions, also it is used for other measurements like jitter, and packet losses. It is important to mention that this tool works within the application layer. This means that the measured throughput is not limited to the performance of the links, but also to the packet process through the TCP / IP model. Additionally, Iperf can specify the package type (UDP, TCP) of different Maximum Transmission Units (MTUs) and at a special rate and interval of transmission [92].

2.11.6 ASP.NET Framework

ASP.NET Core is a popular framework. Key benefits include features like cross-platform execution, high-performance, built-in

Dependency Injection, and modular HTTP request pipeline. It used to monitor traffic and recognize normal and abnormal traffics, ASP.Net Core web apps, it can prevent DDoS by adding one annotation to the controller [92].

Chapter Three

The Proposed Approach

3.1 Overview

This chapter describes the proposed system implementation, setup, and installation of the proposed system, as well as the machine learning algorithms Support vector machine (SVM), Decision tree(DT), and Neural Network (NN). The used machine learning algorithms provided the classification traffic in SDN environments as the normal traffic and abnormal traffic depending on the trained classifier with offline dataset and with Entropy matching for online data traffic.

3.2 The Proposed System Steps

To implement the proposed strategy in SDN, a module that detects DDoS attacks must first be created in the RYU controller. Prior to the controlling and networking installation and configuration required for launching SDN network, the network setup must be complete. In addition, the proposed system is implemented in Python programming language, virtual environment. It is based on two phases; the first one is the off-line state to classify data traffic into normal and abnormal from CIC-DDoS2019 dataset to analyze the efficiency of machine learning strategy for multi-class classification. The second phase is the real-time data traffic to analysis incoming requests and tested to recognize traffic then classify traffic with block malicious source of requests.

Malicious activity should be detected immediately upon occurrence. Therefore, detection speed, precision, and dependability are essential to guarantee that the device executes its tasks without negatively impacting network components. The suggested system offers an effective DDoS mitigation and detection solution based on the Entropy concept and

machine learning methods in an SDN context. Figure 3.1 showed the main techniques of the proposed system.

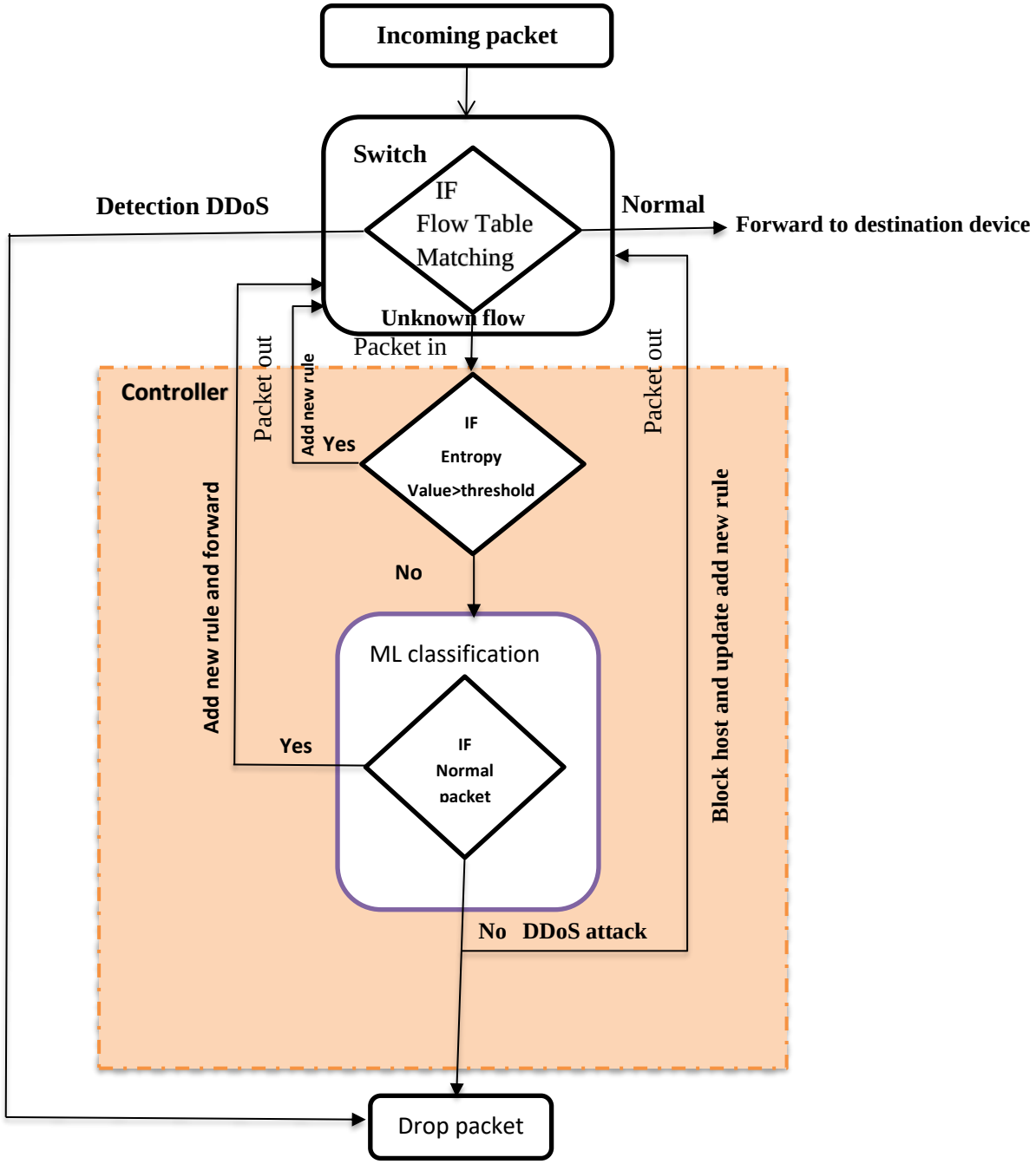


Figure 3.1: The main steps of the proposed system.

3.2.1 The off-line phase

There are three primary stages to the proposed system for protecting the SDN controller against DDoS attacks; these phases are based on three algorithms for machine learning that are used throughout the offline phase of the system's operation:

The first stage: Data preprocessing on the whole data collection to change the row data into a format that is usable and efficient.

The second stage: Machine learning classifiers are used to categorize data.

The third stage: Evaluate the trained model using multiple evaluation metrics. In addition, the proposed system steps based on the use machine learning, data analyzing and system evaluation showed in Figure 3.2.

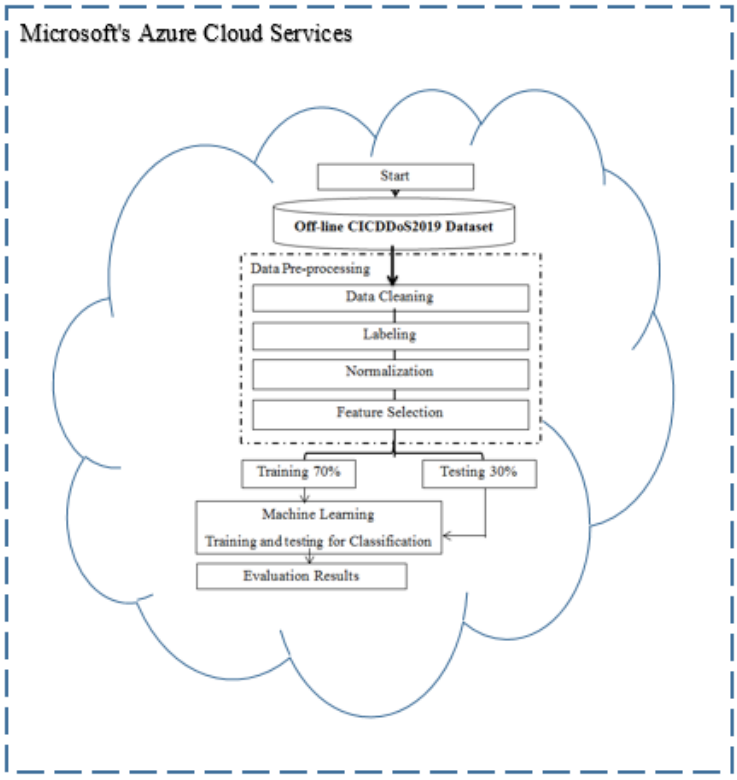


Figure 3.2: The proposed Offline phase with Machine Learning Model.

The basic steps in brief of machine learning approach for Off-Line phase can be summarize as follows.

A. Collect the raw data. (CICDDoS2019 Dataset)
B. Preprocess the dataset to insert missing values and feature extractions.
- Data cleaning : it used to fill of missing values in CICDDoS2019 Dataset, smoothing or removing noisy data and outliers along with resolving inconsistencies.
- Labeling : It used to assign labels to dataset attributes.
- Normalization : It used as the process of scaling individual samples to have unit norm. It applied to columns containing numeric values.
- Feature selection : It used to identify the most important features in CICDDoS2019 dataset.

C. Create a sub-dataset with the most relevant features.
D. Train the three algorithms classifier.
E. Calculate the accuracy of the model.
F. Test with unclassified (unstructured) data.
G. Evaluate the results

Besides, the proposed explanation for the Off-Line data traffic approach steps as follows:

Step1: **Input the CICDoS2019 dataset**

The used Off-line dataset provides the most up-to-date and benign frequent DDoS attacks, which closely reflects the real-time data (PCAPs). The data traffic analyzed with CICFlowMeter-V3 according to the labeled state for time stamp, addresses of Source IP /Destination IP, Ports, Protocols, and

security issues methods are also included (CSV files). It is picked for a variety of reasons including :

1. It is designed for use with a TCP/IP communication stack.
2. The nature of DDoS attacks features.
3. DDoS attack required accurate detection system to verify for instance there are 50,063,112 CICDDoS2019 occurrences, comprising 56863 benign ones and 50,006,249 DDoS attacks.

There are two new TCP/UDP-based application layer CICDDoS2019 attack variants, the first of which is characterized as: Reflection-based. It may be difficult for victims to tell genuine users from attackers in this attack ,the attackers disguise themselves and utilize legitimate equipment, such as internet routers, to deliver traffic to the victim (such as HTTP requests). Both the TCP and UDP protocols, as well as a mix of the two, are being used to launch these attacks. DDoS attacks are being exploited. Both TCP and UDP attacks may include SYN floods, but only TCP attacks have SYN floods. For the DDOS attack detection example, a sample dataset of 3,000,000 data points with 88 attributes has been put together.

Step 2 : The pre-processing step

Preprocessing is a crucial stage in machine learning since it improves the data's quality and facilitates the extraction of meaningful findings from the data. Data preprocessing in the context of Machine Learning refers to the activity of preparation (cleanup and sorting) the raw data to make it suitable for the training and creation of models for machine learning. Pre - processing stage in M L is a technique to data mining that transforms raw information into a form that can be comprehended and processed.

Preprocessing is very important steps which is done to get a better quality input. The main advantage of preprocessing is cleaning and arranging the text to be classified.

The pre-processing datasets in this stage is applied due to reasons:

1) Reducing the dataset size to get hold of the best efficient data analysis,
2) Making dataset adaptable to provide a better analysis selection method.

The developed M.L module the data preparation is the first phase that signifies the beginning of the process itself. Actual real-world data is often imprecise, inconsistent, erroneous (including outliers/mistakes), and lacking in attribute features and trends. As a result, data preparation is introduced into the phase, it assists in the cleaning, formatting, and organization of raw data, preparing it for use by Machine Learning models. There are four different sub-steps into the pre-processing:

A- Data Cleaning

It used in the proposed system to provide dataset consistency. It is performed to prepare the CICDDoS2019 dataset for usage in machine learning mode to eliminate unnecessary columns from the dataset, data preparation involves looking for missing values in order to remove them. So, the original data contains a large amount of missing (nan) and infinity values it removes all these values from the data.

The main processes of data cleaning sorted as follow:

1- Missing Data: In machine learning, handling missed data is crucial, since it may lead to inaccurate model predictions. Consequently, null values are removed by propagating the latest valid observation along the column axis forward.

2- Undefined Data The removal of null values may produce undefined data. Since there are no columns to offer a value, a null field with out any cells to its left remains NaN after propagation. Therefore, these values are encoded as 0

B- Encoding

It used by the trained model of the proposed system for binary classification to classify the input traffic into normal or malicious. Therefore, it consider all DDoS classes as attack category, besides the normal traffic. Then, it encoding the string value for normal and attack label to binary value of 0 and 1, respectively. The proposed system is based on two primary encoding modalities: label encoding and One-Hot encoding.

In One-Hot Encoding (OHE) processing, a new entry is created for each category in the database, with such a value for one given if the recording is a match for that category and zero otherwise. This encoding is required whenever activation function-equipped output devices are set up for use with Softmax. Softmax converts a column of k actual values to a matrix of k actual values that sum to unity. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always remain between 0 and 1. Softmax function shown using equation (3.1).

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad (3.1)$$

σ = softmax

z= input vector

- All the zi values are the elements of the input vector to the softmax function, and they can take any real value, positive, zero or negative. For example a neural network could have output a vector such as (-0.62, 8.12, 2.53), which is not a valid probability distribution, hence why the softmax would be necessary.

e^{z_i} = standard exponential function for input vector.

- The standard exponential function is applied to each element of the input vector. This gives a positive value above 0, which will be very small if the input was negative, and very large if the input was large. However, it is still not fixed in the range (0, 1) which is what is required of a probability.

K = number of classes in the class classifier.

z_j= standard exponential function for output vector.

where all the zi values are the elements of the input vector and can take any real value. The term on the bottom of the formula is the normalization term which ensures that all the output values of the function will sum to 1, thus constituting a valid probability distribution.

C- Normalization

In the used CICDDoS2019 dataset the features data have different numerical values. Training the model directly with the original data can cause classification error, and then the model takes much time during its training. It normalized the data where the minimum value is zero, and the maximum is one.

The process of normalization guarantees that no one variable has an out-of-control impact on the model's output because of the magnitude of its effect. A number of methods exist for achieving normalization.

It is customary to use the Z-score and the min-max normalization procedures when analyzing data. To reduce large numbers to tiny ones, it used min-max normalization in this study. The following equation utilized for normalization:

$$X_{scaled} = \frac{x - min(x)}{max(x) - min(x)} \quad (3.2)$$

Example 2.2 showed the normalization explanation :

Min-max normalization is used to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1.

If the minimum value of a feature was 20, and the maximum value was 40, then 30 would be transformed to about 0.5 since it is halfway between 20 and 40.

$$(30)_{scaled} = \frac{30-20}{40-20} = \frac{10}{20} = 0.5$$

D- Feature selection

It's a method for streamlining a model's inputs in order to improve its forecast accuracy. There is some evidence that reducing the number of variables may boost the performance of the models while simultaneously reducing the lot of time and cost needed to run the analysis. Correlation and extra-tree classifiers are two of the most used approaches for selecting features. To perform feature selection, the proposed system applied additional tree classifier techniques in this thesis. Also, the goal of feature selection step is to decrease the dimensions of the dataset by omitting characteristics which are not related to the categorization [103].

$$weight_{x,y} = \begin{cases} log(tf_{x,y} + 1)\, log\frac{c}{z_x}\ if\ tf_{x,y} \geq 1 \\ 0 \quad ,otherwise \end{cases} \quad (3.3)$$

Where, $tf_{x,y}$ is the occurrence of term x in document y, c is the count of attributes in dataset and z_x is the count of features where term x manifests, as features showed in Table 3.1. The labels and features are explained in (Section 2.8) with the used dataset with different attacks attributes.

Table 3.1 The used feature selection for labeling.

Label	Feature
UDP-Lag	**ACK Flag Count, Init Win bytes forward, min seg size forward, Fwd IAT Mean, Fwd IAT Max**
TFTP	**Fwd IAT Mean, min seg size forward, Fwd IAT Max, Flow IAT Max, Flow IAT Mean**
WebDDoS	**ACK Flag Count, Init Win bytes forward, Fwd Packet Length Std, Packet Length Std, min seg size forward**
DNS	**Max Packet Length, Fwd Packet Length Max, Fwd Packet Length Min, Average Packet Size, Min Packet Length**
Benign	**ACK Flag Count, Flow IAT Min, Init Win bytes forward, Fwd Packet Length Std, Packet Length Std**
MSSQL	**Fwd Packets/s, Protocol**
LDAP	**Max Packet Length, Fwd Packet Length Max, Fwd Packet Length Min, Average Packet Size, Min Packet Length**
NetBIOS	**Fwd Packets/s, min seg size forward, Protocol, Fwd Header Length, Fwd Header Length.1**
NTP	**Subflow Fwd Bytes, Length of Fwd Packets, Fwd Packet Length Std, min seg size forward, Flow IAT Min**
SSDP	**Destination Port, Fwd Packet Length Std, Packet Length Std, Protocol, min seg size forward**
SNMP	**Max Packet Length, Fwd Packet Length Max, Fwd Packet Length Min, Average Packet Size, Min Packet Length**
Syn	**ACK Flag Count, Init Win bytes forward, min seg size forward, Fwd IAT Total, Flow Duration**
UDP	**Destination Port, Fwd Packet Length Std, Packet Length Std, min seg size forward, Protocol**

Step 3: Splitting Data

In order to build a ML model, each dataset must be split into two distinct parts: train data and testing data. The quantity of used data has had a greater impact on how evenly it has split data than any other factor.

No split will provide enough variation for cross-validation if the data is too little, in matter of fact if the large data is generated, it doesn't important to know whether it splits. In this experiment, data has been separated as 70% training data and 30% testing data.

Step 3: Machine learning algorithms

The proposed system in the Off-line phase built with the three main machine learning multiclass classifiers to classify attributes features in the dataset into normal and abnormal to evaluate the accuracy of tested system for DDoS attack detection and trained the model to be flexible for such type of attacks and related attacks. The used algorithms are SVM, NN, and DT for Off-Line phase.

A. Support Vector Machine (SVM) Algorithm

Non-probabilistically, a SVM is a linear classifier for binary data. Its main strength is that it is not based on probability. That's in contrast to Naive Bayes and other probabilistic classifiers. A support vector machine (SVM) classifies records along a plane (decision boundary) defined by a small sample of the records (feature vectors). The support vectors are the subset of data that helps to justify the decision boundary. Where the decision border is drawn in the subspace is entirely independent of the other feature vectors in the dataset.

To identify DDoS assaults, SVM is employed as a classifier. Low false-positive rates and margin maximization through hyperplane discovery

are two of SVM's many strengths. It provides an accurate categorization and is quite reliable. In order to categorize two categories of data, it is helpful for DDoS attack analysis to extract the flow information and updates (normal and malicious).

B. Neural Network (NN) Algorithm

Class categorization in the situation of distributed denial of service attacks is the goal of the method. A method of encoding is required for this purpose. The dataset labels must be transformed for this method so the model can read them. One Hot Encoding was employed in the proposed system. Assigning a 1 if the item is part of that category and a 0 otherwise, OHE processing creates a new row for each label in the dataset. The addition of an output nodes necessitates the use of this encoding scheme. Since the used NN may also output the sort of attack existing in the flow, it provides a better solution at the phase whenever a malicious flow is recognized.

C. Decision Tree (DT) Algorithm

As was seen in subsection (2.11.3) for Azure's conventional cloud environment, the Machine Learning classifier, DT, is a collection of decision trees. Ensemble models, as opposed to individual decision trees, often provide more coverage and precision. In terms of computational and memory efficiency, it performs well in the context of training and prediction. Together, feature selection and categorization are carried out by it. It uses a collection of directed acyclic graphs (DAGs) to make decisions, where it trains these graphs by measuring the decision at a level based on the characteristics of the branching structure of the nodes by reducing the objective function that is based on the forecasts provided by the sub-nodes whose nodes have been recognized Its divided attributes.

The proposed decision tree is a more memory-efficient classifier that may be employed in situations when memory consumption is very important (e.g. embedded systems). Not only did their ideas drastically cut memory use, but they also regularly improved generalization performance. This makes decision tree useful not just for memory-intensive classification problems, but also for all classification tasks. Algorithm (3.1) showed the used DT algorithm with Entropy calculation in the proposed system.

Algorithm (3.1): The used Decision Tree (DT) Algorithm with Entropy calculation
Input : The data-set D is fed Output: Decision tree
Begin: 1. if D is "pure" OR other stopping criteria met then 2. terminate 3. end if 4. for all attribute ϵ D do 5. Compute information gain and entropy as shown in the following **Entropy (Term)=$-\sum_{j=1}^{n} Pjlog2pj$,Gain(age) = Entropy(s) – Infoage(s)** 6. abest =Best attribute according to above computed information gain 7. Tree=Create a decision node that tests abest in the root 8. Dv= Induced sub-datasets from D based on abest 9. For all Dv do 10. Treev=j48 (Dv) 11. Attach Treev to the corresponding branch of Tree 12. End for 13. Return Tree **End**
End of algorithm

3.2.2 The proposed On-line (Real-Time Traffic) Phase

The proposed real-time data traffic transmission is implemented using a mix of entropy and machine learning algorithms, which are used to assess system performance and identify occurrences that deviate from typical network behavior.

The Entropy model utilized for the discovery of DDoS attacks in communication networks detects irregular motions, especially in networking systems with a high data density. Entropy may determine the unpredictability of incoming network communication. It increased due to the randomness of the traffic load increased.

3.2.2.1 Entropy

Initial stage in the proposed system, using the entropy method to compute the randomness of flow during a given time of network packets, the destination IP address, and other details about the packet features of these attributes, revealed other details about the packet features of these attributes. The SDN controller has rules for flowing into each switch in the network by matching them in the flow table.

The correct course of action and application of the flow table rules. For example, if the controller receives a peak flow in a huge number of unknown flows, it will calculate the randomness and take the appropriate decision for packets if the randomness of these packets is greater than the threshold specified in the network or not.

If the randomness in the time is less than the threshold, the flow will be sent to the machine learning model for the purpose of checking it

accurately and taking the appropriate action for it. The rules are updated and new rules are added to the flow table, as it showed in Figure 3.3.

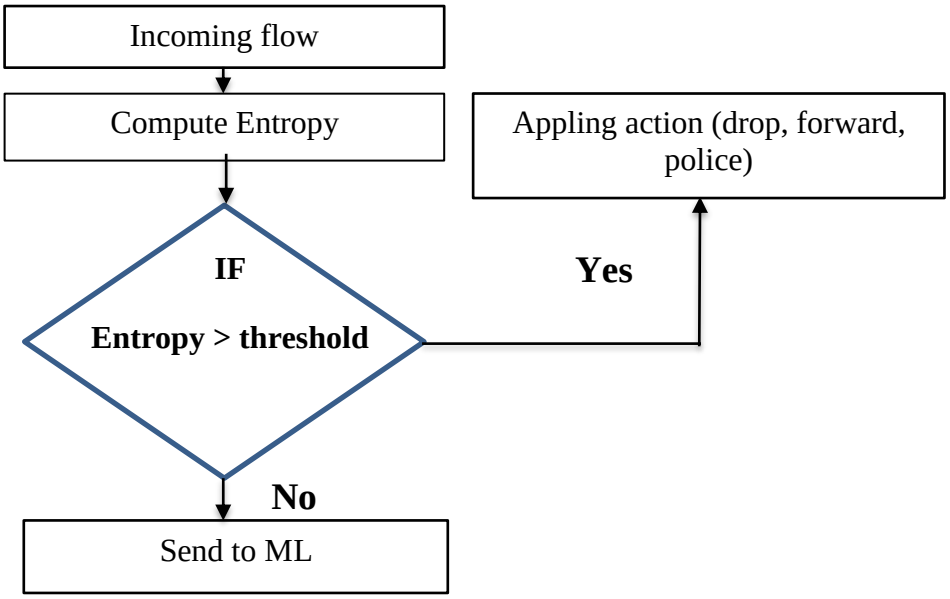


Figure 3.3: compute entropy for each incoming packet unknown.

The incoming flow from network elements devices and switches redirected to the controller which contains on the rule management to monitor network traffic with Entropy value. If the calculated value of entropy is greater than threshold it applied stored action as drop, forward, and police. Otherwise, if incoming request calculated entropy value is less than the threshold, it sent to the machine learning to the trained classifier to classify incoming requests as normal and abnormal.

In addition, Figure 3.4 showed the structure of matching technique in real time model. When start initializing incoming traffic to the switches from hosts switches redirect requests to the controller which used feature matching techniques to test if matched or not then add new flow table as DDoS or normal traffic by trained classifier as real-time machine learning model and evaluated the results to the evaluation phase.

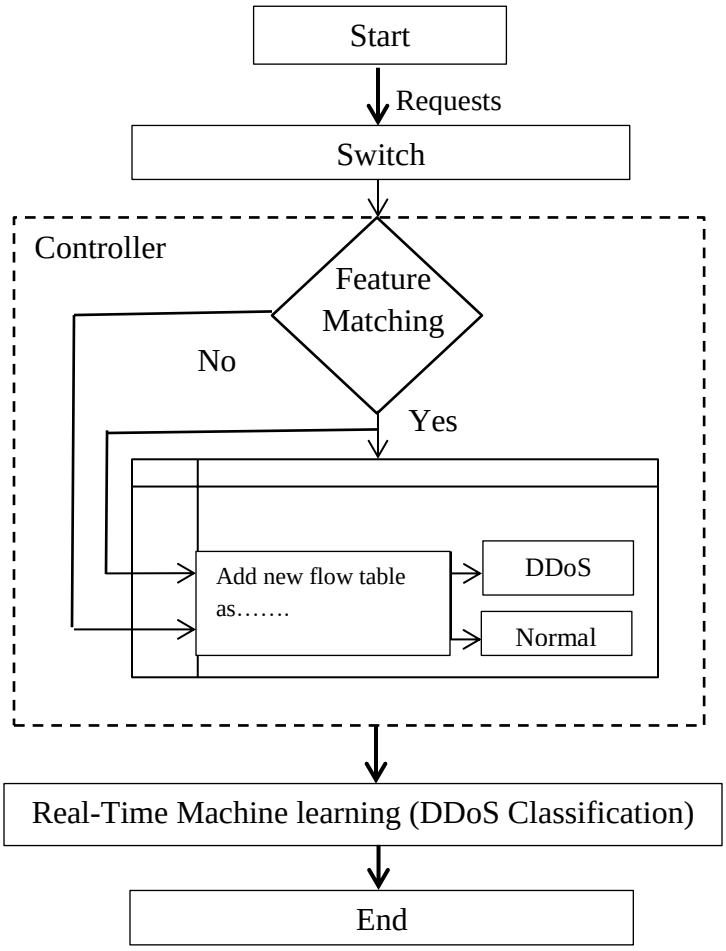


Figure 3.4: The proposed Real-Time data traffic phase.

Additionally, network traffic is comprised of both benign and malicious data. To prevent policy breaches and strengthen defenses, this kind of communication has to be analyzed and managed using a customized model. IP spoofing means that just blocking the purported attacker's IP won't stop the attack; instead, you need to focus on the attack's characteristics to counter them. DDoS attacks are difficult to detect because they look and act like normal network traffic. However, some patterns in DDoS attacks have been uncovered.

A DDoS attack aims to overload a network with packets destined for various locations, as was already explained. Various techniques, such TCP, UDP, ICMP, Random IP, and others, can be used in Network threats to flood the victim network. DDoS attacks in a SDN setting provide the basis of the suggested method.

The proposed machine learning focus on the development and investigation of capable of learning from data without even being explicitly programmed to do so. The used algorithms distinguish between DDoS attacks and regular traffic, hence mitigating DDoS attacks. The characteristics used by the trained classifier model parameters that may be used to detect a distributed denial of service attack include the total data packets, the packet size, the total amount of bytes, the total number of packets the packets and bit speeds, etc.

The proposed matching approach for incoming data comes after the Entropy process model and is based on the corresponding the incoming real-time from the nodes and compared it with the trained classification model stored behavior and attitude as (SDN-specific dataset generated by Mininet analyzer and used for packet classification using machine learning).

The results of this comparison with the main DDoS attributes are shown in Table 3.2. If an incoming request's values match those in the features database, the request is considered legitimate and processed normally; otherwise, it is labeled as attack traffic and processed in the opposite manner. With the DT algorithm as its foundation, the suggested system can handle Real-Time data flow.

Table 3.2 : The used features for DDos attack detection model.

Flow Duration	Flow duration
Total Fwd Packet	Total packets in the forward direction
Total Bwd packets	Total packets in the backward direction
Total Length of Fwd Packet	The total size of packets in the forward direction
Fwd Packet Length Max	Maximum size of packets in the forward direction
Fwd Packet Length Min	The minimum size of packets in the forward direction
Fwd Packet Length Mean	The average size of packets in the forward direction
Fwd Packet Length Std	Standard deviation size of packets in the forward direction
Bwd Packet Length Max	Maximum size of packets in the backward direction
Bwd Packet Length Min	The minimum size of packets in the backward di rection
Bwd Packet Length Mean	Mean size of packets in the backward direction
Bwd Packet Length Std	Standard deviation size of packets in the backward direction
Flow Bytes/s	flow byte rate that is the number of packets trans ferred per second
Flow Packets/s	flow packets rate that is the number of packets transferred per second
Fwd Packets/s	Number of forwarding packets per second
Bwd Packets/s	Number of backward packets per second
Packet Length Min	Minimum length of a flow
Packet Length Max	The maximum length of a flow
Packet Length Mean	Mean length of a flow
Packet Length Std	Standard deviation length of a flow
Packet Length Variance	Minimum inter-arrival time of packet
Average Packet Size	The average size of packets

The algorithm (3.2) showed the used Real-Time data traffic phase. The input value is incoming packets tested for all flow table to match and forward to the destination device or it discarded as unknown packets. Entropy

calculated and matched with stored threshold to add new flow table if matched otherwise redirect to the machine learning classifier as normal and abnormal packets and then showed flow table matching to end procedure.

Figure 3.5 shows the testing phase in the machine-learning model

Algorithm (3.2): The used Real-Time data traffic Algorithm

```
Input Incoming Packets
Output Packets classified into normal and abnormal
1 Begin
2   for Input = 1 to end of input do match inputs for every run
3   If flow-table matching then
                Forward to destination device
        Else
                Unknown-packets
4     for in-packets = 1 to all-packets do Increase flow-table
5       If Entropy (in-packets) > threshold then
6         Add new flow table (FT)
7       Else
8         Machine Learning classification
9           If (in-packets) = normal packets then
10            Add new row and forward to flow-table
11          Else
            DDos attack then Drop packets
12          End if
13      End if
14    End for
15  End if
16  End for
17  return flow-table matching
18 end procedure
```

End of Algorithm

of the proposed system. Get the input from the controller every 1 second and

pass it as the input to the machine learning classifier model built in training phase. Check if attacked is detected raise the alarm and record the attack to a log file. Otherwise, the processing the incoming packets rolling back.

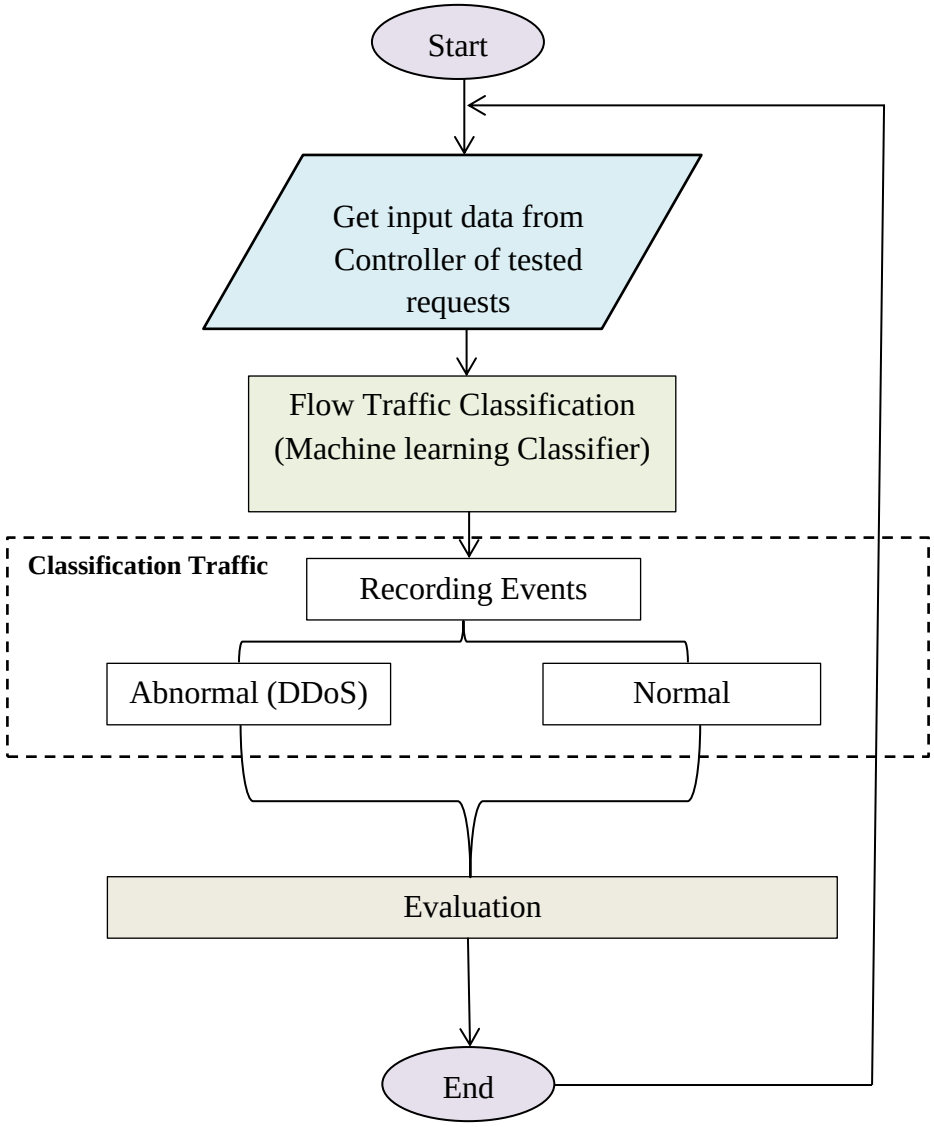


Figure 3.5: Flowchart for machine learning classifier.

3.2.2.2 The trained Model with Decision Tree (DT)

Producing classification schemes from a large number of variables or developing predictive model for a target attribute falls under this category. The inverted tree is constructed using this method by slicing a populations into segments that resemble tree branches, then adding a tree trunk, processing components, and leaf nodes. The algorithm is non-parametric and can efficiently deal with large, complicated datasets without imposing a complicated parametric structure.

The trained model was applying in the system after implementing the above algorithms from the proposed system, according to the findings, it is the efficient algorithm based on binary classification of data traffic with two classes as the normal class, and DDoS attack abnormal class.

This methodology yields a trained model with real data classification for incoming traffic from data plane layer. Both classes can handled by trained model and the applied system provided accurate results depending on the compared behavior for DDoS attack in the RYU-SDN environment. Algorithm (3.3) showed the used DT algorithm.

Algorithm (3.3): The used trained model (Decision Tree)

Input : The data-set D is fed

Output: Decision tree

1 Begin

2 **if** D is "pure" OR other stopping criteria met then

Terminate

3 end if

4 **For** all attribute ϵ D do

Compute information gain and entropy as shown in the following

Entropy (Term)=$-\sum_{j=1}^{n} Pjlog2pj$**,Gain(age) = Entropy(s) – Infoage(s)**

abest =Best attribute according to above computed information gain

Tree=Create a decision node that tests abest in the root

Dv= Induced sub-datasets from D based on abest

5 **For** all Dv do

6 Treev=tree (Dv)

Attach Treev to the corresponding branch of Tree

7 **End for**

8 End for

8 Return Tree

End

End of algorithm

3.3 Summary

This chapter has shown the method used to simulate the Software-Defined Networks (SDN) which is enable the network administrator to build, configure, monitor and troubleshoot various networking devices and hence it has proved to be time-saving and prevents the tedious manual configuration of the setup through implementing real-time data traffic, thus it is proves to be the advanced networks settings. Also, it shows the configuration of the used system to mitigate DDoS attack in such environment. In addition to, the used machine learning classification algorithms to classify normal and abnormal attack are explained. Finally, the chapter shows how to use this system architecture for implement host request with trained classifier for SDN RYU controller.

Chapter Four

Results and Discussion

4.1 Introductory

In this chapter the proposed systems results has been described, which are talked about in chapter three. The proposed system is implemented in Mininet, VirtualBox, RYU controller with Python programming language, and the results are tested with network tools such as Hping, Wireshark and it has been evaluated with three machine learning algorithms including: DT, SVM, and NN. The results of the proposed model can classify network traffic with high classification accuracy and network performance into normal, and abnormal as DDoS attack.

4.2 System Implementation

The proposed system has been built using an environment that is dependent on the environment requirements that are showed in Table 4.1. In addition, the programming language Python was used to write the code for the proposed system.

Table 4.1: System environment requirements.

Operating Systems	Windows 10
CPU	Core (TM) I5-3630
RAM	16.00 GB
Implementation Tools	Python, Cloud Azure

The proposed system is based on a set of tools, frameworks, and functions in the Virtual Mininet emulator. This application manipulates packets interactively. It can decode packets of several protocols, broadcast

them on the network, simulate attacks, collect packets, match request and answers, showed in Table (4.2).

Table 4.2: Details the components of the network Tools for DDoS detection.

Implementation Tools	
Software	**Purpose**
Hping3	is a command-line oriented TCP/IP packet assembler/analyzer. It mainly used as a security tool for Firewall testing, Advanced port scanning, Network testing, using different protocols and Advanced traceroute, under all the supported protocols.
Wireshark	Emulates, generates normal traffic and DDoS attack traffic using UDP packets. It analyses the packets sent and received for the source and destination addresses.
Python	It used to implement network topology, setting, and execute commands.
Cloud Azure	It used to store and make configuration Dataset dealt with DDoS attack.
VirtualBox	It used to make configuration and setting the environment with Linux, and Mininet simulator.
Mininet simulator	It used to build and create network architecture and manage traffic with DDoS attack.
RYU SDN-Controller	The DDoS detection model is implemented within the controller to facilitate centralized monitoring of network.

The proposed network architecture has been based on the data plane for host devices (as shown in Figure 4.1), the switch as the access switch, and the SDN RYU controller for the controller plane, while the application plane is explained by system statistics from the lower plane layers. The proposed system based on the three layers:

1- Application layer: it represented by system statistics from the lower plane layers.
2- Controller Layer (RYU controller) : it used to manage and monitor network for both normal and abnormal network traffic
3- Data Plane Layer: it applied as the Host, Switch, and server to pass network traffic among them.

The proposed system implemented in data plane and control plane layers. The described state of the network topology based on the OpenFlow protocol, which builds flow tables. The controller will make settings based on the version of the switch, meaning the controller will change the operation depending on the version of the switch, as it showed in Figure 4.1.

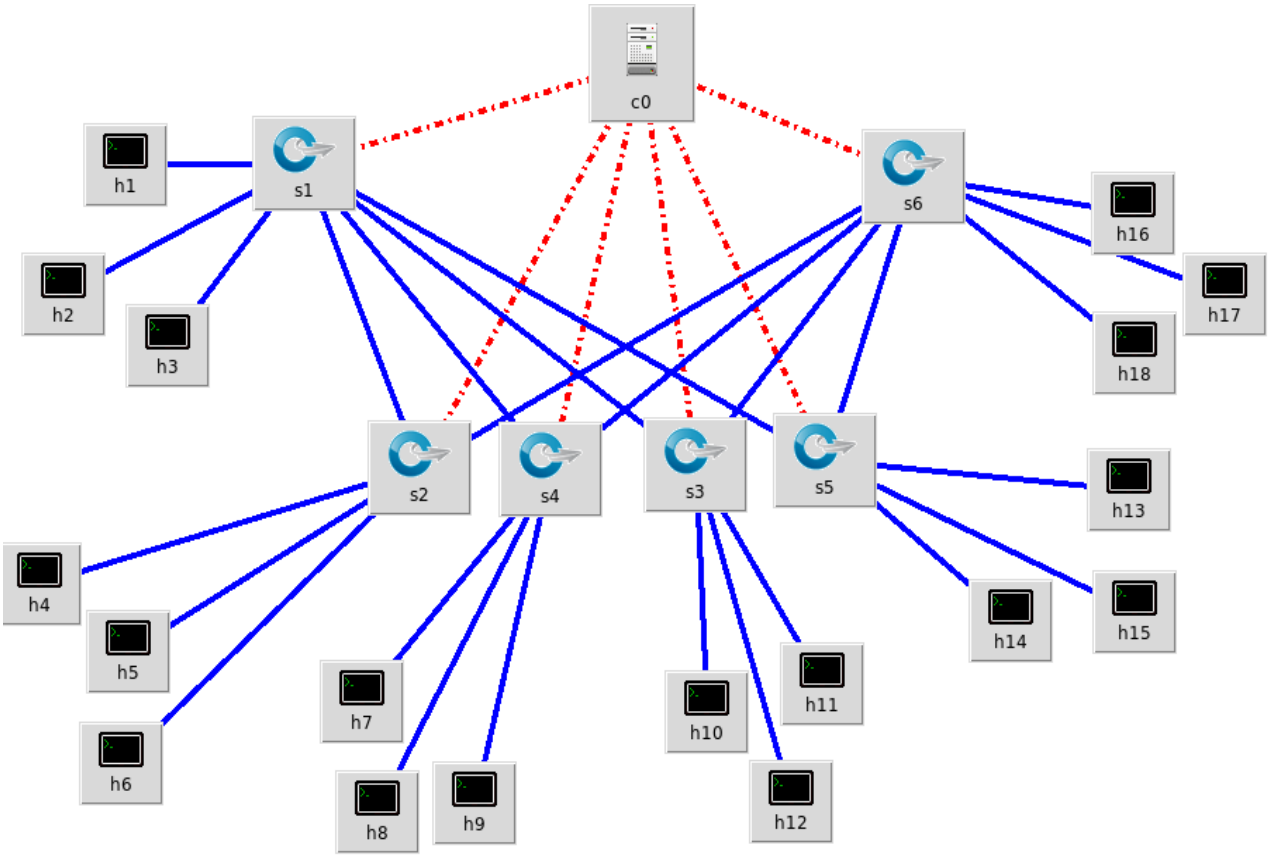


Figure 4.1: The proposed Network Topology.

Because the controller has the main view of the network, it will build the network setting during the initialize state. In addition, each switch will redirect the traffic based on the flow table that it has. In some cases of new hosts, requests that are not found in the flow table of the connected switch, the request will redirect into controller through southbound API interface. The proposed system has a reconfigurable feature, which it summarizes as the controller will reconfigurable itself after a specific period of time to update the flow table to add the new entry (all network parameters of the host collected from the host request).

Besides, the connection between hosts in the data plane is summarized as the data traffic state in Figure 4.2.

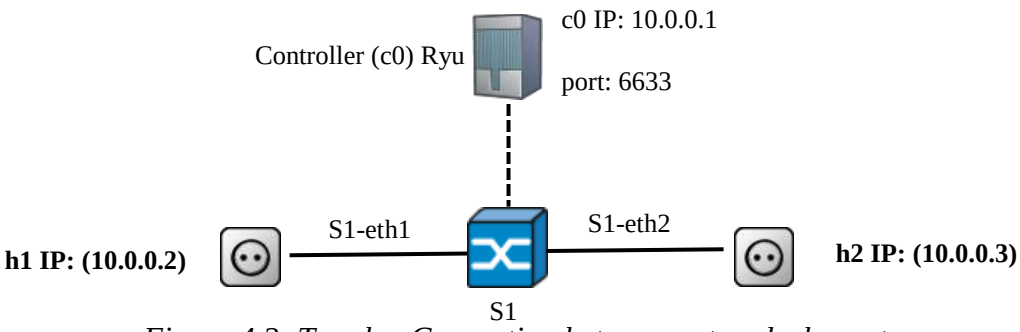


Figure 4.2: Topolgy Generation between network elements.

Besides, the main connection steps in the python running interface:

- net.addHost('h1')
- net.addHost('h2')
- net.addSwitch('s1')
- net.addController('c0')
- net.addLink(h1, s1) (h2, s1)
- net.start()
- CLI (net)

4.3 The first Phase (Offline State)

The proposed system based on three machine learning algorithms including (DT, SVM, and NN) to detect DDoS attack in SDN networks as shown in Figure 4.3.

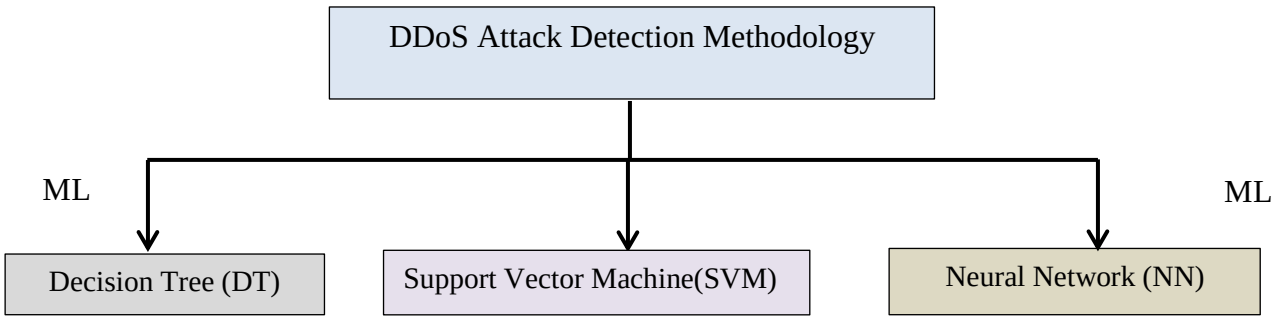


Figure 4.3: The used Machine Learning Algorithms.

The offline Phase of the proposed system for data analysis based on an offline dataset using Neural Network (NN) algorithm, Support Vector Machine (SVM) algorithm, and Decision Tree (DT) algorithm have all been used to test and assess its performance, as explained in the following subsections:

4.3.1 Neural Network (NN) algorithm

Neural Network (NN) has been implemented to provide a natural extension to the problem that relates to multiple layers rather than relying on a single neuron, which uses identical outputs when there is a binary output that can contain a number of binary neurons and thus contribute to the process of multi-classification of the classes used, as it is shown in Table 4.3 and Table 4.4 respectively.

Table 4.3: The evaluation of NN in offline phase.

Metrics	Value
Accuracy	99.8587%
Precision	99.8587%
Recall	99.8587%

Table 4.4: The prdicted and actual classes of the NN in offline phase.

Predicted Class			
Actual Class	BENIGN		DDoS-NTP
	BENIGN	97.1%	2.9%
	DDoS-NTP	0.1%	99.9%

4.3.2 Support Vector Machine(SVM) algorithm

SVM is has been used as the non-probability side, as this aspect is the source of power for this algorithm, and it contrasts with many algorithms that are based on probabilistic classifiers, as they include a set of feature vectors, and the set of data that is used from them is a subset of data based on decision limits appropriately for vector support. The goal of SVM is to determine the ideal dividing hyper plane in order to maximize the margin of the training data. As it is shown in Table 4.5, the classification algorithm means it is used to predict if there is something that belongs to a particular category.

Table 4.5: The results of SVM in offline phase.

Metrics	Value
Accuracy	99.6%
Precision	99.8%
Recall	99.8%

4.3.3 Decision Tree(DT) algorithm

Decision Tree (DT) has been relied on a set of decision-guided loop graphs called DAGs, where it trains these graphs by measuring the decision at a level based on the characteristics of the branching structure of the nodes by reducing the objective function that is based on the predictions made by the sub-nodes resulting from the nodes after recognition. Its split features, as shown in Figure 4.6 and Figure 4.7.

Table 4.6: The evaluation metrics of the DT case study.

Metrics	Value
accuracy	99.9979%
precision	98.7006%
recall	99.9979%

Table 4.7: The prdicted and actual classes of the 3rd case study.

Predicted Class			
Actual Class	**BENIGN**		**DDoS-NTP**
	BENIGN	95.5%	4.5%
	DDoS-NTP	0.0%	100.0%

The proposed system has been evaluated with regard to a number of case studies in addition to existing publications in this field. The system comparison between the various proposedmachine learning methods is shown in Table (4.8) and Figure (4.4) respectively.

The proposed system demonstrated that the DT algorithm provides higher accuracy results of identifying DDoS attacks in the network while it is in the off-line state as a decision that accurately categorizes a suspicious attacks as being malicious. This is possible as a result of the system's ability to build a strong forecasting model classifying not only specific elements, but

also about their cross correlations. It is, in essence, the simplest tree possible, with just one node and main branch, which results in a much improved capacity for predictive modeling.

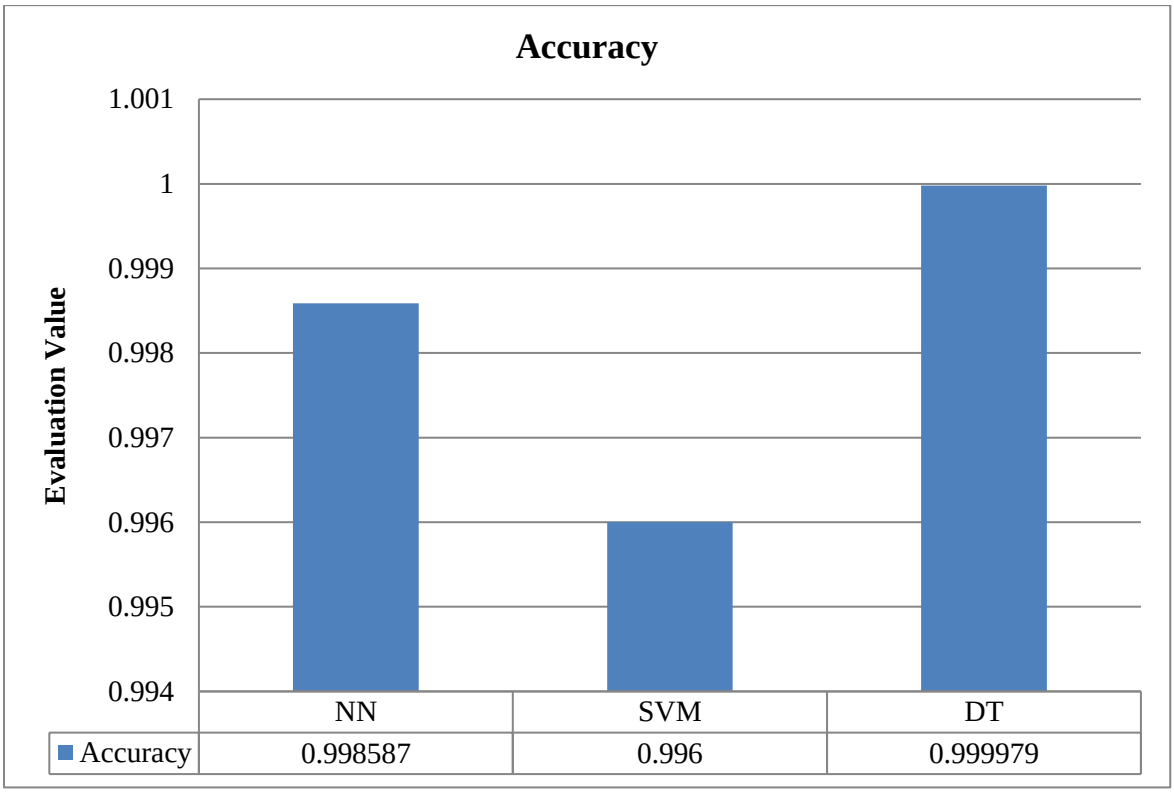


Figure 4.4 : The proposed system accuracy comparison.

4.4 The second Online State(Real-data Traffic)

Using the trained model DT classifier, the proposed system has been built in an online condition to test and assess incoming requests instantly following the various processes of calculating entropy and identifying tcp connections as DDoS attack traffic or regular data traffic. The used topology is showed above in Figure (4.1).

Table (4.8) showed the network elements characteristics for the implemented environment.

Table 4.8 showed the network elements characteristics.

Node Type	Connected Device	Network interface
Switch 1	Host 1, Host 2, Host 3	eth1-eth3
Switch 2	Host 4, Host 5, Host 6	eth1-eth3
Switch 3	Host 7, Host 8, Host 9	eth1-eth3
Switch 4	Host 10, Host 11, Host 12	eth1-eth3
Switch 5	Host 13, Host 14, Host 15	eth1-eth3
Switch 6	Host 16, Host 17, Host 18	eth1-eth3
Switch 1	Switch 2	eth4
Switch 2	Switch 3	eth5-eth4
Switch 3	Switch 4	eth5-eth4
Switch 5	Switch 6	eth5-eth4
Controller	Switch 1, Switch 2, Switch 3, Switch 4, Switch 5, Switch 6	eth0

Both Mininet-Wi-Fi and RYU will be installed using the script, and also, as of the time of writing, the Ubuntu 19.10 official repository includes the stable releases for the software. Table (4.9) showed the system tools specifications based on tool name and the particular version.

Table 4.9: The used Tools name and their Version.

Tool Name	Version
Openvswitch	1.3
Wireshark	2.6.6
Mininet -WiFi	2.2.2
IPERF	2.0.5
RYU Controller	4.23
VirtualBox	6.1.10 Build 138449 X64
Linux Ubuntu	19.10-desktop-amd64
ASP.NET C#	2010

4.4.1 Building Model to RYU controller

The initialization state of the proposed model based on the call the main class model to match the incoming unknown request features with the stored trained features in local dataset.

4.4.2 The generated Normal Traffic

The main steps of the generated normal traffic are explained as follow :

1- Creating network topology in Mininet.

- With the Python programming language from terminal and code behind to build the network with host and switches and controller.

2- Generating network traffic and managed with IPerf tool.

- Using network traffic tool to build packets among hosts and redirection packets from source node to destination node.

3- Calculating Entropy of incoming traffic.

- Measuring Entropy based on incoming traffic to the controller and matched the calculated value with stored threshold.

4- Matching the calculated Entropy with the stored Threshold value.

5- The results showed that the nature of traffic is normal.

Figure (4.5) shows the normal flooding state for each second of time in the network after packets are generated in the network. It was from host to host or from host to server, and the entropy value was more than the specified threshold according to the state of the network. In addition, it showed the overall generated packets in pluses and error packets.

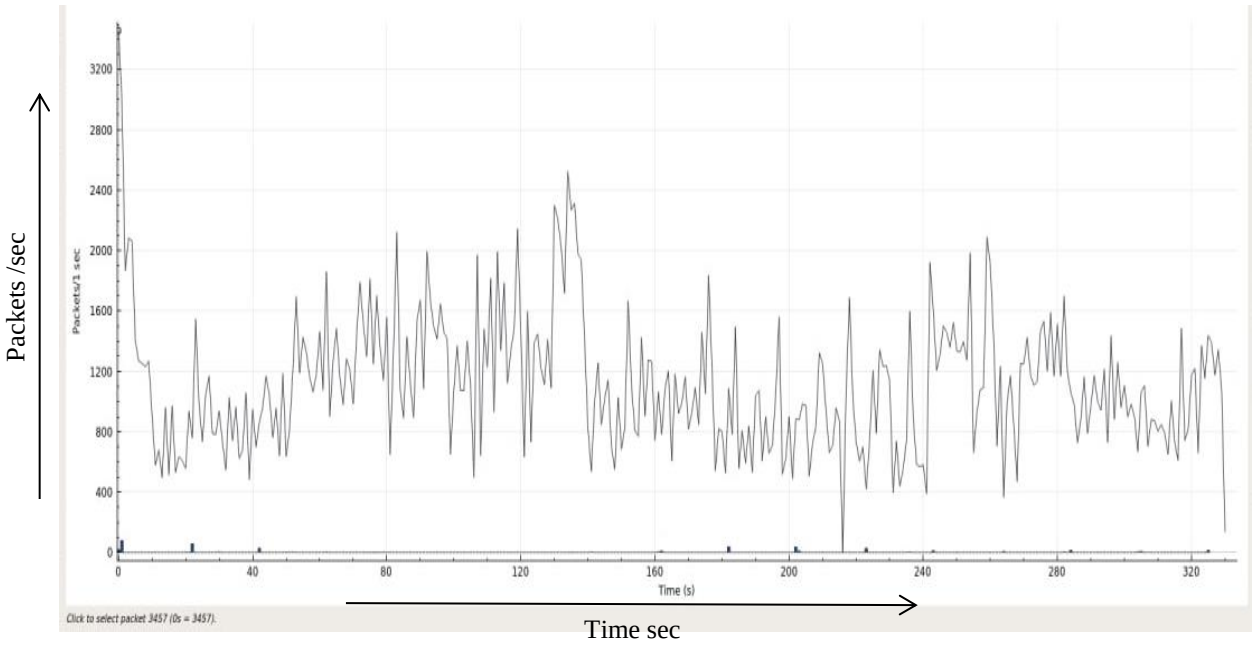


Figure 4.5: The overall generated packets and error packets

Figure (4.6) showed the amount of normal throughput during the 320 seconds in the normal traffic packets.

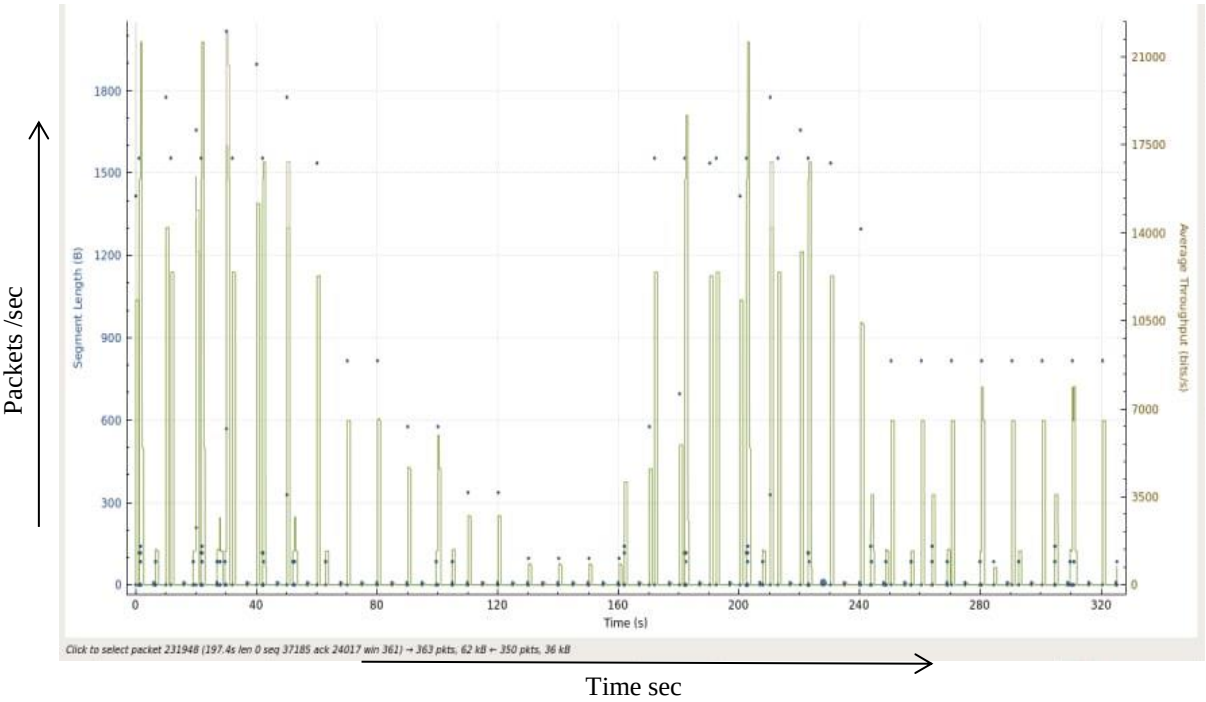


Figure 4.6: Normal packet traffic throughput.

Figure (4.7) showed the generated ICMP, TCP/UDP packets between hosts with uploading and downloading traffic files as index, test files for normal traffics.

Figure 4.7: the generated traffic files within normal traffic.

4.4.3 The generated Abnormal Traffic

The abnormal traffic is has been generated with the different types of network tools such as Hping, Iperf to flood network with the huge amount of traffic to provide DDoS attack in the network elements and especially controller. These attack tools have been created to prevent legitimate users from accessing network resources or to significantly slow down their access to those resources.

Figure (4.8) represented the increase in the transmission rate of packets in the network, and this has been indicated that the network is in a

state of stable or not. After reviewing the entropy value, it found that it is less than the threshold and dimension, so the packets are sent to machine learning to classify the scattering, whether it is normal or abnormal. A decision has been made. In the event of an attack, the controller updates the flow table for the purpose of blocking the attack and protecting the controller. In addition it showed number of error packets during the 320 seconds period.

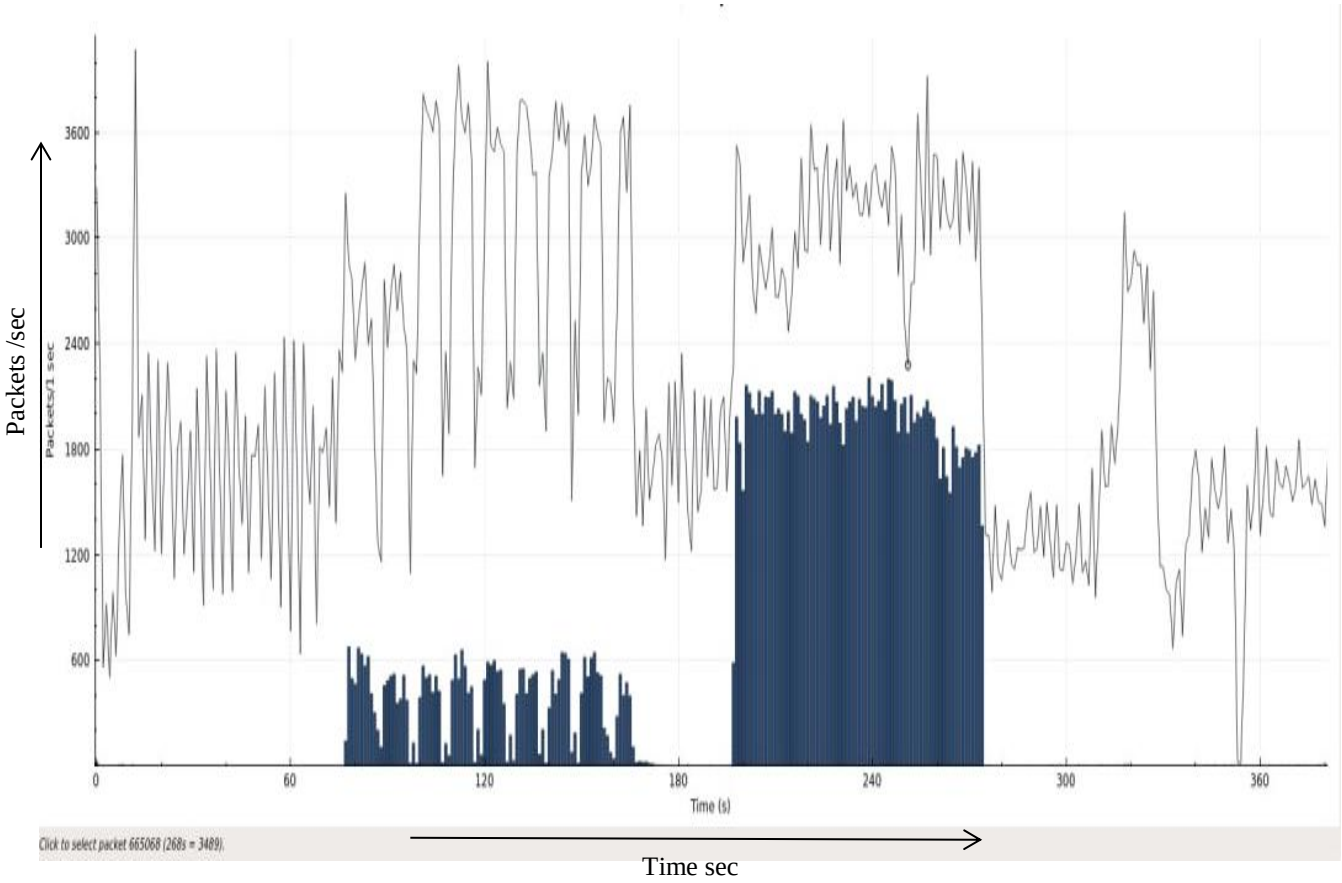


Figure 4.8: showed abnormal DDos attack traffic.

Figure (4.9) showed the throughput and sequence length and the data is congested and flooded in the network segment.

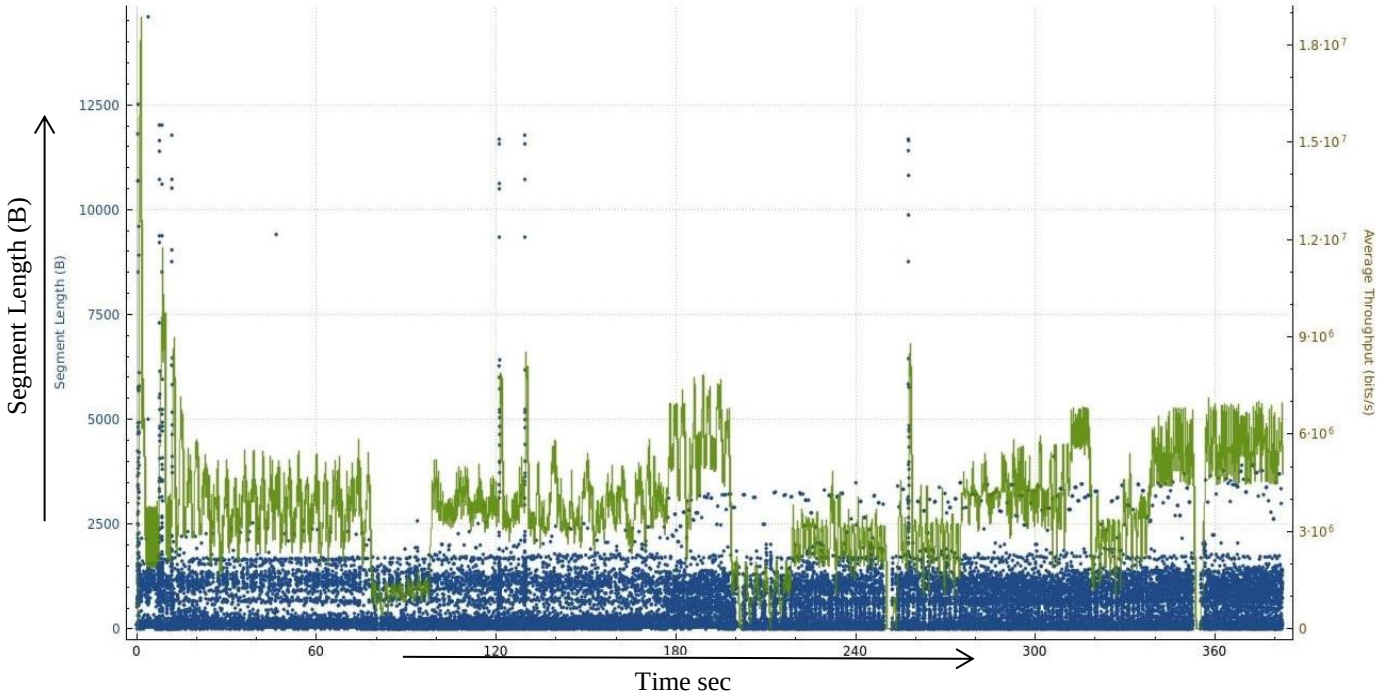


Figure 4.9 : Throughput and flooded packets of abnormal DDoS traffic.

Figure (4.10) showed the flooding data traffic form different methods as the ICMP, TCP, and UDP to make DDoS attack in the network.

Real Time Traffic

Create New DeleteAll

Entrpy	State	Date	Ips	
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:32 PM	2470	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:31 PM	2586	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:29 PM	2756	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:28 PM	2723	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:27 PM	3745	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:26 PM	9274	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:22 PM	2363	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:21 PM	2171	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:20 PM	2143	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:19 PM	3458	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:17 PM	1090	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:16 PM	2425	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:15 PM	2434	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:14 PM	5446	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:13 PM	2871	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:12 PM	5445	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:09 PM	19087	Edit | Details | Delete
-0	DDoS traffic victim is host: h14	10/16/2022 11:20:02 PM	6000	Edit | Details | Delete

Figure 4.10 : the performing ICMP, UDP, TCP-Syn flooding packets of the abnormal DDoS attack.

4.4.4 The Trained Decision Tree (DT) Model

The used DDos attack has been implemented with six hosts and tested system with an Entropy value, and the system has been evaluated to classify traffic as normal or abnormal. In addition, the machine learning algorithm provides traffic classification with normal and DDos attacks. The used model based on the 22 features which they are matched with the attributes of the incoming traffic.

Attacks in SDN networks may be automatically detected and mitigated with the help of a trained DT algorithm by creating classification schemes based on numerous variables or by designing classification algorithm for a target attribute. An inversion tree with such a parent node, intermediate nodes, and number of nodes is built using this approach by categorizing a population by branch-like segments. As can be shown in Table (4.10), the method is non-parametric and is capable of effectively handling huge, complex datasets while imposing a sophisticated parametric framework.

Table 4.10: DT DDoS attack detection algorithm results.

Method Name	**Accuracy**	**Confusion Matrix**		**Time**
		False Positive Rate	**False Negative Rate**	
Decision Tree (DT)	99.90 %	0	0.1	11.919 sec

In addition, there are other analysis measures that are shown in Table (4.11).

Table 4.11 : Machine learning details for SDN DDoS attack detection.

Evaluation Parameters	Machine Learning Algorithms
	DT
Accuracy	99.90

In SDN environment the proposed system based on the RYU to deal with DDoS attack due to the results show that the RYU appears to be the most resilient controller, with the ability to handle a much higher attack rate before it completely loses the ability to handle traffic. In addition to the ability of the RYU to work in distributed environment. The system is compared to complementary works in Table 4.12. In the instance of DT, the accuracy of the proposedsystem was demonstrated to be 99.90% of the time during the actual data traffic phase, which was the highest of all the case studies. In the instance of DT, the off-line phase accuracy of the proposedsystem is proven to be 99.999%, the highest of all of the phases.

Table 4.12 : The proposedsystem's results with SDN DDoS attacks.

Ref.No	Year	Dataset	Method Name	Accuracy
[18]	2021	Real-Time dataset using RYU API-Mininet	Support-Vector-classifier with Random-Forest (SVC-RF)	98.8 %
			Logistic Regression (LR)	83.69%
[19]	2021	CICIDS2017 dataset	V-NKDE (Voting -Naive Bayes, K Nearest Neighbors,	99.67 %
		KDD dataset		99.77
		UNSW-NB15 dataset		98.09

			Decision Tree, and Extra Trees)	
[20]	2021	KDD99 dataset	Decision Tree (DT)	78 %
			Support vector machine (SVM)	85 %
[21]	2021	(Portmap+LDAP) =Mix Dataset from CICDDOS2019	Random Forest (RF)	99.9764 %
			Naïve Bayes (NB)	95.5469 %
Proposed- system		**Off-Line CICDoS2019**	**Neural Network (NN)**	99.8587 %
			Support Vector Machine(SVM)	99.6 %
			Decision Tree(DT)	99.9979 %
Proposed- system		**Real-Time DDoS attack Classification**	**Decision Tree (DT)**	99.90 %

4.5 Summary

DDoS attacks are becoming increasingly complicated and able to overcome many standard defense mechanisms, which may cause to network damage if not handled effectively. SDN uses Machine Learning to improve network security. NN, SVM, and (DT) algorithms are employed to generate explicit or implicit models from supplied data to develop systems that really

can learn from information without being trained, which helps identify hidden information and leads to improved insights. Machine learning may help improve the network's effectiveness by mitigating DDoS attacks intelligently. DT has the highest off-line accuracy of 99.9979%. Compared to other algorithms, Decision Tree (DT) has 99.90% Real-Time accuracy.

Chapter Five

Conclusions and Suggestions for Future Works

5.1 Conclusions

This chapter explains the proposed system conclusions and the main suggestions for future works, the main conclusions can be summarized as :

1- The proposed system prevents DDoS attacks on the SDN controller by detecting abnormal malicious packets from legitimate packets using a training model to analyze internet traffic in a normal state and determining the DDoS attack state using the proposed machine learning techniques and entropy. The study is aimed to provide a robust system to detect either normal and abnormal network data traffic

2- The implemented system is based on two main case phases are :

A- The 1st phase is the (Offline State) with CICDoS2019 dataset

B- The 2nd phase is the Online State(Real-data Traffic)

3- The performance evaluation showed that best accuracy of the offline case study as Decision Tree(DT) is 99.9979 % accuracy for DDos attack detection with CICDDoS2019 dataset , and DT was 99.90 % of real-time accuracy beside, the accuracy of Off-line phase of NN was 99.8587 %, and SVM algorithm was 99.6 %.

4- The results showed that the DT is the better from SVM, and NN due to the ability to select the best parameters as well as the correlation of the features used from the dataset.

5- There are some of limitation faced in the proposed system are Mininet and Controller installation, training in the cloud Azure system, and connecting monitor with controller.

6- The better accuracy performance of the Real-Time data traffic model was a Decision Tree (DT) of DDos attack classification.

7- Entropy methods helps to reduce time by checking the randomness of the flow and comparing it with the threshold before sending the flow to the machine learning classifier.

5.2 Suggestions for Future Works

Using the following suggestions, several considerations might be realized for future extension of current research as follow:

1- Other datasets, and deep learning techniques can be used depending on the proposed system to classify and mitigate DDos attack.
2- Detecting DDoS attacks in structures that connect or communicate between multiple SDN controllers.
3- Building a model for the detection of low-rate DDoS assaults, which are more challenging to spot than other types of DDoS attacks since it might be difficult to tell the difference between regular network traffic and reduced DDoS attacks when they occur on a network.
4- Exploring the feasibility of detecting additional types of DDoS attacks, such as TCP sync, ICMP, and HTTP flooding attacks
5- Building the different dynamic Entropy thresholds for securing the emerging 5G, and 6G networks for specific quality of service requirements.

REFERENCES

[1] Mehr, S. Y., & Ramamurthy, B. (2019, December). An SVM based DDoS attack detection method for Ryu SDN controller. In *Proceedings of the 15th international conference on emerging networking experiments and technologies* (pp. 72-73).

[2] Haji, S. H., Zeebaree, S. R., Saeed, R. H., Ameen, S. Y., Shukur, H. M., Omar, N., ... & Yasin, H. M. (2021). Comparison of software defined networking with traditional networking. *Asian Journal of Research in Computer Science*, *9*(2), 1-18.

[3] Mishra, P., Puthal, D., Tiwary, M., & Mohanty, S. P. (2019). Software defined IoT systems: Properties, state of the art, and future research. *IEEE Wireless Communications*, *26*(6), 64-71.

[4] Kaur, K., Garg, S., Kaddoum, G., Gagnon, F., Kumar, N., & Ahmed, S. H. (2019, April). An energy-driven network function virtualization for multi-domain software defined networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 121-126). IEEE.

[5] Swami, R., Dave, M., & Ranga, V. (2019). Software-defined networking-based DDoS defense mechanisms. *ACM Computing Surveys (CSUR)*, *52*(2), 1-36.

[6] Aladaileh, M. A., Anbar, M., Hasbullah, I. H., Chong, Y. W., & Sanjalawe, Y. K. (2020). Detection techniques of distributed denial of service attacks on software-defined networking controller–a review. *IEEE Access*, *8*, 143985-143995.

[7] Rahman, O., Quraishi, M. A. G., & Lung, C. H. (2019, July). DDoS attacks detection and mitigation in SDN using machine learning. In *2019 IEEE world congress on services (SERVICES)* (Vol. 2642, pp. 184-189). IEEE.

[8] Wang, Y., Hu, T., Tang, G., Xie, J., & Lu, J. (2019). SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking. *IEEE Access*, *7*, 34699-34710.

[9] Santos, R., Souza, D., Santo, W., Ribeiro, A., & Moreno, E. (2020). Machine learning algorithms to detect DDoS attacks in SDN. *Concurrency and Computation: Practice and Experience*, *32*(16), e5402.

[10] AlMomin, H., & Ibrahim, A. A. (2020, June). Detection of distributed denial of service attacks through a combination of machine learning algorithms over software defined network environment. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-4). IEEE.

[11] Alamri, H. A., Thayananthan, V., & Yazdani, J. (2021). Machine Learning for Securing SDN based 5G network. *Int. J. Comput. Appl*, *174*(14), 9-16.

[12] Perez-Diaz, J. A., Valdovinos, I. A., Choo, K. K. R., & Zhu, D. (2020). A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. *IEEE Access*, *8*, 155859-155872.

[13] Ramprasath, J., & Seethalakshmi, V. (2021). Improved network monitoring using software-defined networking for DDoS detection and mitigation evaluation. *Wireless Personal Communications*, *116*(3), 2743-2757.

[14] Alamri, H. A., & Thayananthan, V. (2020). Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks. *IEEE Access*, *8*, 194269-194288.

[15] Ujjan, R. M. A., Pervez, Z., Dahal, K., Khan, W. A., Khattak, A. M., & Hayat, B. (2021). Entropy based features distribution for anti-ddos model in sdn. *Sustainability*, *13*(3), 1522.

[16] Yu, S., Zhang, J., Liu, J., Zhang, X., Li, Y., & Xu, T. (2021). A cooperative DDoS attack detection scheme based on entropy and ensemble learning in SDN. *EURASIP Journal on Wireless Communications and Networking*, *2021*(1), 1-21.

[17] Aladaileh, M. A., Anbar, M., Hasbullah, I. H., & Sanjalawe, Y. K. (2021). Information theory-based approaches to detect DDoS attacks on software-defined networking controller a review. *International Journal of Education and Information Technologies*, *15*, 83-94.

[18] Ahuja, N., Singal, G., Mukhopadhyay, D., & Kumar, N. (2021). Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications*, *187*, 103108.

[19] Tayfour, O. E., & Marsono, M. N. (2021). Collaborative detection and mitigation of DDoS in software-defined networks. *The Journal of Supercomputing*, *77*(11), 13166-13190.

[20] Sudar, K. M., Beulah, M., Deepalakshmi, P., Nagaraj, P., & Chinnasamy, P. (2021, January). Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE.

[21] Abbas, S. A., & Almhanna, M. S. (2021, February). Distributed denial of service attacks detection system by machine learning based on dimensionality reduction. In *Journal of Physics: Conference Series* (Vol. 1804, No. 1, p. 012136). IOP Publishing.

[22] Song. Wang, Balarezo, J. F., Chavez, K. G., Al-Hourani, A., Kandeepan, S., Asghar, M. R., & Russello, G. (2022). Detecting flooding DDoS attacks in software defined networks using supervised learning techniques. *Engineering Science and Technology, an International Journal*, 101176.

[23] Fan, C., Kaliyamurthy, N. M., Chen, S., Jiang, H., Zhou, Y., & Campbell, C. (2021). Detection of DDoS attacks in software defined networking using entropy. *Applied Sciences*, *12*(1), 370

[24] Musumeci, F., Fidanci, A. C., Paolucci, F., Cugini, F., & Tornatore, M. (2022). Machine-Learning-enabled DDoS attacks detection in P4 programmable networks. *Journal of Network and Systems Management*, *30*(1), 1-27.

[25] Bernardos, C. J., De La Oliva, A., Serrano, P., Banchs, A., Contreras, L. M., Jin, H., & Zúñiga, J. C. (2014). An architecture for software defined wireless networking. *IEEE wireless communications*, *21*(3), 52-61.

[26] Haji, S. H., Zeebaree, S. R., Saeed, R. H., Ameen, S. Y., Shukur, H. M., Omar, N., ... & Yasin, H. M. (2021). Comparison of software defined networking with traditional networking. *Asian Journal of Research in Computer Science*, *9*(2), 1-18.

[27] Ahmad, S., & Mir, A. H. (2021). Scalability, consistency, reliability and security in SDN controllers: a survey of diverse SDN controllers. *Journal of Network and Systems Management*, *29*(1), 1-59.

[28] Javeed, D., Gao, T., Khan, M. T., & Ahmad, I. (2021). A hybrid deep learning-driven SDN enabled mechanism for secure communication in Internet of Things (IoT). *Sensors*, *21*(14), 4884.

[29] Yurekten, O., & Demirci, M. (2021). SDN-based cyber defense: A survey. *Future Generation Computer Systems*, *115*, 126-149.

[30] Arshi, M., Nasreen, M. D., & Madhavi, K. (2020). A survey of DDoS attacks using machine learning techniques. In *E3S Web of Conferences* (Vol. 184, p. 01052). EDP Sciences.

[31] Abhishta, A., van Heeswijk, W., Junger, M., Nieuwenhuis, L. J., & Joosten, R. (2020). Why would we get attacked? An analysis of attacker's aims behind DDoS attacks. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *11*(2), 3-22.

[32] Tan, L., Pan, Y., Wu, J., Zhou, J., Jiang, H., & Deng, Y. (2020). A new framework for DDoS attack detection and defense in SDN environment. *IEEE Access*, *8*, 161908-161919.

[33] Bhardwaj, A., Mangat, V., & Vig, R. (2020). Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of DDoS attacks in cloud. *IEEE Access*, *8*, 181916-181929.

[34] Zhang, M., Li, G., Wang, S., Liu, C., Chen, A., Hu, H., ... & Wu, J. (2020, February). Poseidon: Mitigating volumetric ddos attacks with programmable switches. In *the 27th Network and Distributed System Security Symposium (NDSS 2020)*.

[35] Zhang, M., Li, G., Wang, S., Liu, C., Chen, A., Hu, H., ... & Wu, J. (2020, February). Poseidon: Mitigating volumetric ddos attacks with programmable switches. In *the 27th Network and Distributed System Security Symposium (NDSS 2020)*.

[36] Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)* (pp. 1-8). IEEE.

[37] Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic distributed denial of service (DDoS) attack dataset and

taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)* (pp. 1-8). IEEE.

[38] Pei, J., Chen, Y., & Ji, W. (2019, June). A DDoS attack detection method based on machine learning. In *Journal of Physics: Conference Series* (Vol. 1237, No. 3, p. 032040). IOP Publishing.

[39] Pande, S., Khamparia, A., Gupta, D., & Thanh, D. N. (2021). DDOS detection using machine learning technique. In *Recent Studies on Computational Intelligence* (pp. 59-68). Springer, Singapore.

[40] Novikov, A. V. (2019). PyClustering: Data mining library. *Journal of Open Source Software*, *4*(36), 1230.

[41] Viloria, A., Acuña, G. C., Franco, D. J. A., Hernández-Palma, H., Fuentes, J. P., & Rambal, E. P. (2019). Integration of data mining techniques to PostgreSQL database manager system. *Procedia Computer Science*, *155*, 575-580.

[42] Alexandropoulos, S. A. N., Kotsiantis, S. B., & Vrahatis, M. N. (2019). Data preprocessing in predictive data mining. *The Knowledge Engineering Review*, *34*.

[43] Jane, V. A. (2021). Survey on iot data preprocessing. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(9), 238-244.

[44] Bock, F. E., Aydin, R. C., Cyron, C. J., Huber, N., Kalidindi, S. R., & Klusemann, B. (2019). A review of the application of machine learning and data mining approaches in continuum materials mechanics. *Frontiers in Materials*, *6*, 110.

[45] Hernández-Blanco, A., Herrera-Flores, B., Tomás, D., & Navarro-Colorado, B. (2019). A systematic review of deep learning approaches to educational data mining. *Complexity*, *2019*.

[46] Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., ... & Hluchý, L. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, *52*(1), 77-124.

[47] Ren, J., Guo, J., Qian, W., Yuan, H., Hao, X., & Jingjing, H. (2019). Building an effective intrusion detection system by using hybrid data optimization

based on machine learning algorithms. *Security and communication networks*, *2019*.

[48] Rani, P., Kumar, R., Jain, A., & Lamba, R. (2020). Taxonomy of machine learning algorithms and its applications. *Journal of Computational and Theoretical Nanoscience*, *17*(6), 2508-2513.

[49] Dimitropoulos, X., Krioukov, D., & Riley, G. (2006). Revealing the autonomous system taxonomy: The machine learning approach. *arXiv preprint cs/0604015*.

[50] Singh, A., Bhatia, R., & Singhrova, A. (2018). Taxonomy of machine learning algorithms in software fault prediction using object oriented metrics. *Procedia computer science*, *132*, 993-1001.

[51] Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. *Machine learning*, 3-23.

[52] Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20-28.

[53] Pal, M., & Mather, P. M. (2003). An assessment of the effectiveness of decision tree methods for land cover classification. *Remote sensing of environment*, *86*(4), 554-565.

[54] Liu, Y., Zhang, D., & Lu, G. (2008). Region-based image retrieval with high-level semantics using decision tree learning. *Pattern Recognition*, *41*(8), 2554-2570.

[55] Wang, S., Cao, Y., Huang, T., Chen, Y., & Wen, S. (2020). Event-triggered distributed control for synchronization of multiple memristive neural networks under cyber-physical attacks. *Information Sciences*, *518*, 361-375.

[56] Zhang, Y., & Bao, Y. (2020). Event-triggered hybrid impulsive control for synchronization of memristive neural networks. *Science China Information Sciences*, *63*(5), 1-12.

[57] Guo, Y., Ling, Y., & Chen, H. (2020). A neighbor-guided memory-based neural network for session-aware recommendation. *IEEE Access*, *8*, 120668-120678.

[58] Shakiba, F. M., & Zhou, M. (2020). Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks. *IEEE Transactions on Industrial Electronics*, *68*(11), 10856-10867.

[59] Gao, J., Zhong, C., Chen, X., Lin, H., & Zhang, Z. (2020). Unsupervised learning for passive beamforming. *ieee communications letters*, *24*(5), 1052-1056.

[60] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, *33*, 9912-9924.

[61] Zhang, M., & Piggott, M. D. (2020, June). Unsupervised learning of particle image velocimetry. In *International Conference on High Performance Computing* (pp. 102-115). Springer, Cham.

[62] Im, W., Kim, T. K., & Yoon, S. E. (2020, August). Unsupervised learning of optical flow with deep feature similarity. In *European Conference on Computer Vision* (pp. 172-188). Springer, Cham.

[63] Ibrahim, Z. A., & Mohammed, I. J. (2022). Analysis of Features Selection Effects on Different Classification Algorithms with Performance Metrics Improvement based on PortScan-attack of CICDDoS2019 Dataset. *JOURNAL OF ALGEBRAIC STATISTICS*, *13*(3), 1712-1723.

[64] Sindian, S., & Samer, S. (2020). An enhanced deep autoencoder-based approach for DDoS attack detection. *Wseas Trans. Syst. Control*, *15*, 716-725.

[65] Novaes, M. P., Carvalho, L. F., Lloret, J., & Proença, M. L. (2020). Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment. *IEEE Access*, *8*, 83765-83781.

[66] Salahuddin, M. A., Bari, M. F., Alameddine, H. A., Pourahmadi, V., & Boutaba, R. (2020, November). Time-based anomaly detection using autoencoder. In *2020 16th International Conference on Network and Service Management (CNSM)* (pp. 1-9). IEEE.

[67] Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)* (pp. 1-8). IEEE.

[68] Gayathri, R., & Neelanarayanan, V. (2019). Identification of Regression function and distribution model for Denial of Service attack in Second Life online community using Simple Network Management Protocol. *International Journal of Web Based Communities*, *15*(3), 225-237.

[69] Colella, A., & Colombini, C. M. (2014, September). Amplification DDoS attacks: Emerging threats and defense strategies. In *International Conference on Availability, Reliability, and Security* (pp. 298-310). Springer, Cham.

[70] Boyar, O., Özen, M. E., & Metin, B. (2018, June). Detection of denial-of-service attacks with SNMP/RMON. In *2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES)* (pp. 000437-000440). IEEE.

[71] Lopez Giron, A. J. (2021). Analysis of Machine Learning Techniques to Secure 5G Networks.

[72] Lucky, G. A. (2021). *Distributed Network Monitoring for Distributed Denial of Service Attacks Detection and Prevention* (Doctoral dissertation, The University of Liverpool (United Kingdom)).

[73] Khan, I. Q. (2020). *Anomaly Detection with Machine Learning in IoT Cellular Networks* (Master's thesis).

[74] Qiu, S. B., Yuan, B., & Zhang, K. L. (2008, October). Building TFTP server on embedded system. In *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing* (pp. 1-4). IEEE.

[75] Sieklik, B., Macfarlane, R., & Buchanan, W. J. (2016). Evaluation of TFTP DDoS amplification attack. *computers & security*, *57*, 67-92.

[76] Bzdok, D., & Meyer-Lindenberg, A. (2018). Machine learning for precision psychiatry: opportunities and challenges. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, *3*(3), 223-230.

[77] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240).

[78] MacEachern, S. J., & Forkert, N. D. (2021). Machine learning for precision medicine. *Genome*, *64*(4), 416-425.

[79] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 1-13.

[80] To, Q. G., To, K. G., Huynh, V. A. N., Nguyen, N. T., Ngo, D. T., Alley, S. J., ... & Vandelanotte, C. (2021). Applying machine learning to identify anti-vaccination tweets during the COVID-19 pandemic. *International journal of environmental research and public health*, *18*(8), 4069.

[81] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *sensors*, *16*(8), 1222.

[82] Dijkstra, K., Loosdrecht, J., Schomaker, L. R., & Wiering, M. A. (2018, September). Centroidnet: A deep neural network for joint object localization and counting. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 585-601). Springer, Cham.

[83] Prabhu, N. (2020). Network Virtualization and Emulation using Docker, OpenvSwitch and Mininet-based Link Emulation.

[84] Makori, D. O. (2018). *Machine learning based ddos attack detection for software-defined networks: Yazılım tanımlı ağlar için makine öğrenme esaslı ddos attack algılama* (Master's thesis, Sakarya Üniversitesi).

[85] Brugnoli, I., Bon, M. F., Mazzuco, D., Sena, G. G., Rattaro, C., & Bentancur, S. (2021, March). Tunnelless SDN overlay architecture for flow based QoS management. In *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)* (pp. 62-69). IEEE.

[86] Persico, V., Marchetta, P., Botta, A., & Pescapé, A. (2015, December). On network throughput variability in microsoft azure cloud. In *2015 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-6). IEEE.

[87] Gunarathne, T., Wu, T. L., Qiu, J., & Fox, G. (2010, June). Cloud computing paradigms for pleasingly parallel biomedical applications. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (pp. 460-469).

[88] Jawaharan, R., Mohan, P. M., Das, T., & Gurusamy, M. (2018, July). Empirical evaluation of sdn controllers using mininet/wireshark and comparison with

cbench. In *2018 27th international conference on computer communication and networks (icccn)* (pp. 1-2). IEEE.

[89] Kavana, H. M., Kavya, V. B., Madhura, B., & Kamat, N. (2018). Load balancing using SDN methodology. *Int. J. Eng. Res. Technol*, *7*(5), 206-208.

[90] Tarasov, V., & Malakhov, S. (2015). Statistical data handling program of Wireshark analyzer and incoming traffic research. In *Proceedings of the Spring/Summer Young Researchers' Colloquium on Software Engineering* (Vol. 27, No. 3, pp. 303-314). Федеральное государственное бюджетное учреждение науки Институт системного программирования Российской академии наук.

[91] Abdullah, T. (2014). Testing and analysis SDN technology. *ScienceRise*, *3*(2), 57-62.

[92] Preamthaisong, P., Auyporntrakool, A., Aimtongkham, P., Sriwuttisap, T., & So-In, C. (2019, July). Enhanced DDoS detection using hybrid genetic algorithm and decision tree for SDN. In *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 152-157). IEEE.

الخلاصة

الشبكات المعرفة بالبرمجيات (SDN) هي نموذج للشبكات التي تعيد تعريف مصطلح الشبكة بجعل أجهزة الشبكة قابلة للبرمجة. يساعد SDN مهندسي الشبكات على مراقبة تسريع الشبكة ، والتحكم في الشبكة من نقطة مركزية ، وتحديد حركة المرور الضارة وفشل الارتباط بطريقة سهلة وفعالة. إلى جانب ذلك ، هذه المرونة التي توفرها SDN ، فهي أيضًا عرضة لهجمات مثل DDoS التي يمكن أن توقف الشبكة بالكامل. وللتخفيف من هذا الهجوم ، من المهم بناء نظام آمن لتصنيف حركة البيانات على أنها طبيعية وغير طبيعية مع تقنيات التعلم الآلي التي تساهم في الكشف السريع عن هذه الهجمات. هنالك الكثير من الباحثين المهتمين بتحديد كشف وتقليل انواع مختلفة من الهجمات على بيئة الشبكات المعرفة بالبرمجيات وخصوصا هجوم DDoS وذلك باستخدام خوارزميات التعلم الالي ونظام كشف المتسللين IDS والانتروبي في هذا المجال وذلك لتصنيف وتقليل من تأثيرات الهجوم على هذه الشبكة.

يستند النظام المقترح على خوارزميات التعلم الالي والمتمثلة (Neural Network (((NN), Decision Tree (DT), and Support Vector Machine(SVM ودالة ال Entropy لمراقبة وحماية وحدة تحكم SDN من ازدحام هجمات DDoS من خلال مراقبة سلوك الشبكة للحالة الطبيعية والتعرف على حالة هجوم DDos باستخدام تلك الخوارزميات المقترحة. يتم تنفيذ النظام المقترح مع حالتين رئيسيتين ، الأولى هي دراسة السلوك غير اللحظي ، والثانية هي حركة مرور البيانات في الوقت الفعلي اللحظي لاختبار الطلب الوارد وتصنيفها على أنها عادية أو غير طبيعية وتقييمها بمقاييس التقييم الرئيسية مثل الدقة ، F-Measureمقياس ، الدقة ، والاستدعاء.

تم تطبيق النظام المقترح بلغة البايثون وانجز دقة عالية حيث اوضح ان أفضل نتائج لخوارزمية التعلم الآلي هي DT بدقة 99.9979٪ لحركة المرور غير اللحظية المعتمدة على قاعدة البيانات CICDDoS2019 ، و DT بنسبة 99.90٪ من الدقة في الوقت الفعلي مقارنة بالخوارزميات الأخرى المستخدمة ، حيث وصلت دقة دراسة الحالة غير اللحظية لـ NN بنسبة 99.8587٪ ، وخوارزمية SVM بنسبة 99.6٪.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
تكنولوجيا المعلومات

نهج مقترح لاكتشاف هجوم الحرمان من الخدمة الموزع بالاعتماد على الانتروبي وتعلم الالة في بيئة الشبكات المعرفة برمجيا

رسالة مقدمة

إلى مجلس كلية تكنولوجيا المعلومات في جامعة بابل والتي هي جزء من متطلبات

الحصول على درجة الماجستير في تكنولوجيا المعلومات / شبكات المعلومات

من قبل الطالب

عباس جاسم عبود كلف

باشراف

أ.م.د علاء الدين عباس عبدالحسن حمزة

أ.م.د نوفل تركي عبيس

2022 م 1444 هـ