SDN Network DDOS Detection Using ML

[1]A. Bindu (Assistant Professor)
Dept. of Computer Science Engineering GVPCEW (JNTUK)
Visakhapatnam, India

[2]Ambati Venkata Sai Harika
Dept. of Computer Science Engineering GVPCEW (JNTUK)
Visakhapatnam, India

[3]Dandamudi Swetha
Dept. of Computer Science Engineering GVPCEW (JNTUK)
Visakhapatnam, India

[4]Malli Sahithi
Dept. of Computer Science Engineering GVPCEW (JNTUK)
Visakhapatnam, India

Abstract:- This paper describes a technique that uses the Ryu Controller and Mininet to identify and mitigate Distributed Denial of Service (DDoS) threats in Software Defined Networks (SDN). Using Mininet, the suggested method entails building a virtual network topology with connected switches and hosts. The Ryu Controller gathers traffic data while Mininet simulates several DDoS attack types, such as ICMP flood, land assault, TCP SYN flood, and UDP attacks. The Ryu Controller collects both benign and DDoS traffic into a dataset that is used to build a machine learning (ML) model that can detect DDoS attacks in real time.

Keywords:- *Software-Defined Networking, DDoS , Ryu Controller, Mininet Simulation, Machine Learning , ICMP Flooding, Land Attack Simulation, TCP SYN Flooding, UDP Flooding, Network Security Enhancement, Anomaly Detection, Traffic Classification, Real-time Monitoring, Network Topology Emulation.*

I. INTRODUCTION

The Distributed Denial of Service (DDoS) attacks are a serious threat to network performance and availability in conventional networking systems. These attacks usually entail sending a large amount of malicious traffic over network resources, blocking access to services for authorized users. SDN, or software-defined networking, has become a prominent paradigm for improving network security.SDN enables dynamic reconfiguration of network devices and centralized management by severing the control plane from the data plane.

There are many benefits to using SDN for DDoS detection and mitigation, such as improved visibility, scalability, and flexibility in response to changing threats. In this work, we present an approach that uses widely-used tools like Ryu Controller and Mininet to identify and mitigate DDoS attacks by utilizing the capabilities of SDN along with an Intrusion Detection System using Machine Learning.

II. METHODOLOGY

We built a virtual network environment utilizing Mininet, a well-known network emulator, and Ryu Controller for network administration in order to assess the efficacy of our suggested method. The network structure allowed for the simulation of several attack scenarios because it consisted of switches and hosts connected by virtual links. ICMP flood, TCP SYN flood, UDP flood, and LAND attack were created by mimicking actual threat scenarios with the aid of tools such as hping. The Ryu Controller collected and processed traffic data, including malicious and benign flows, and then put the information into a structured dataset for additional study.

➢ *Ryu Controller*

Ryu Controller:is an open-source python based programmable controller, which is used to define the rules and logic for the switches to follow in the methodology.

➢ *Mininet*

Mininet is a network simulator and creates a virtual network topology with controller, switches and hosts, in this work a single openVswitch with 10 and 25 hosts are created for multiple tests.

➢ *Hping3*

Hping3 is a packet generator which generates TCP/IP traffic in the network, it is mostly used to test network security. Normal and attack traffic scripts are written to generate traffic automatically using this tool .

➢ *IDS*

IDS**:** intrusion detection systems (IDS) provide proactive defense against hostile activity by analyzing system behavior and network traffic patterns to identify anomalies. ML-based intrusion detection systems are able to adapt and detect previously undiscovered threats with increased accuracy and efficiency because they are constantly learning from changing threat environments.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1031

A. Abbreviations and Acronyms

SDN: Software Defined Network, ML:Machine Learning IDS: Intrusion Detection System , DDOS: Distributed Deniel Of Service.ICMP: Internet Control Message Protocol,UDP: User Datagram Protocol,LAND: Local Area Network Denial,TCP-SYN: Transmission Control Protocol - SYNchronize

B. Simulation Platform Setup

This Section involves the steps and procedure to install all the packages and software required to implement the project. The Platform setup is done on Ubuntu 20.04.1 LTS operating system. The project demonstration and simulation was using the following tools:-

- Openflow Protocol For SDN
- Ryu Controller
- Mininet
- Hping3.

- *Openflow protocol*

Openflow protocol For SDN or OpenVswitch has to be installed as it is the standard protocol for software defined network- Switch (2009).1 Open Terminal and type in the following command:- -sudo apt-get install openvswitch-switch Give yes(Y) where ever it is asked, and to check the version and confirm the installation type in**:- -ovs-vsctl –version**.

To view the stages of detection and mitigation**.**

- *Ryu Controller*

Ryu Controller must be installed. Since Ryu is a Python-based controller, installing it requires installing PIP for Python in order to install Python packages. Enter the following commands into the terminal to install PIP and the Ryu controller.

- *Mininet*

Mininet is a network simulator and creates virtual network topology for software defined networks.

- Sudo apt-get install mininet -mn –version.

- *Hping3*

Hping3 is a network packet generator and traffic generator for TCP/IP protocol, mostly used for network testing . Iperf is network traffic performance tool to generate traffic and monitor it.

- Sudo apt-get install iperf -sudo apt-get install hping3

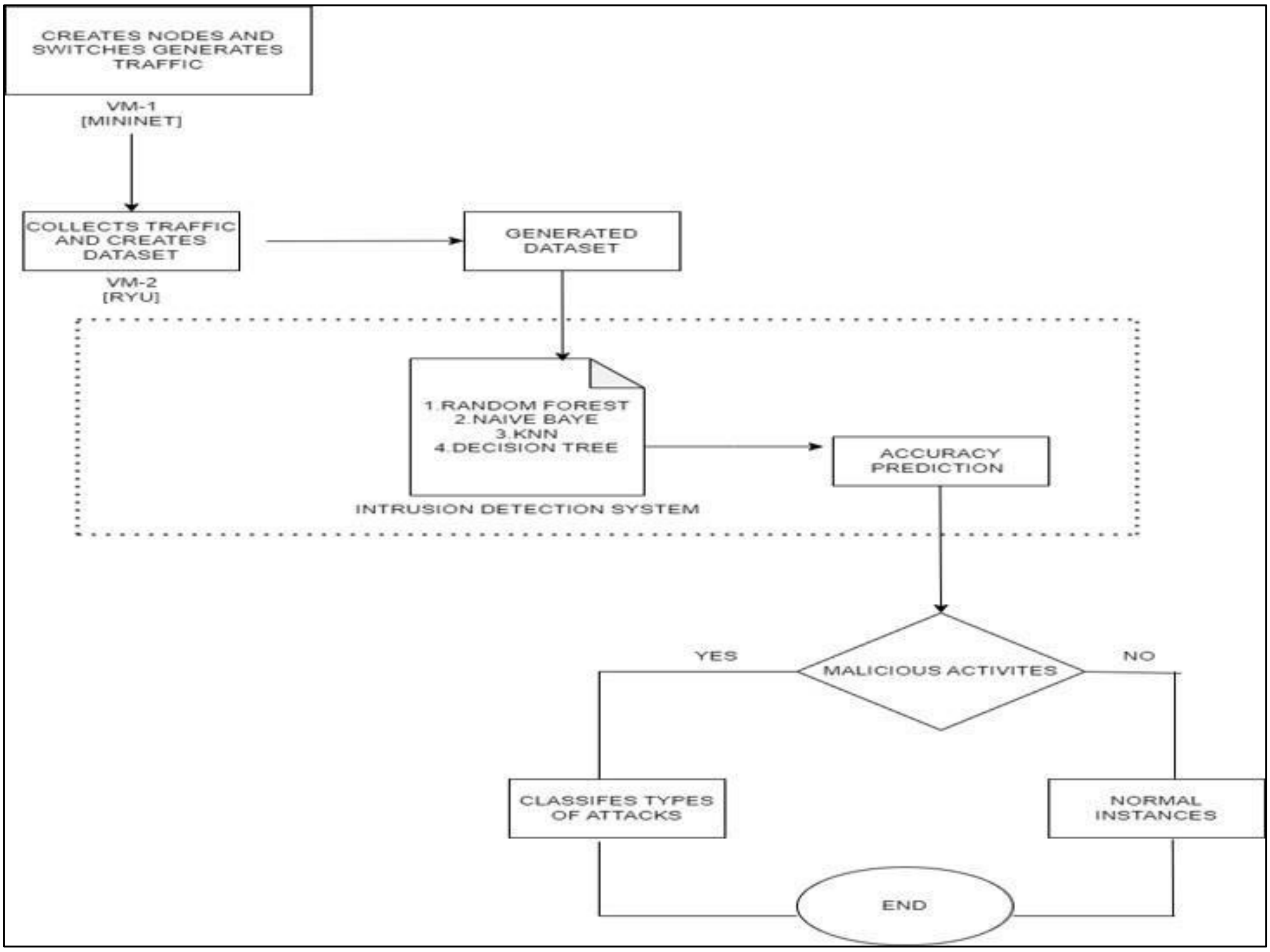


Fig 1 Architecture of the Project

C. Traffic Data Collection

In order for the learning algorithms of Random Forest, KNN, Naives Bayes, and linear regression to analyse and forecast attack traffic, traffic data must be gathered and saved in a CSV file.The objective is to demonstrate that if a hacker executes a DDoS attack on a system, traffic will be gathered and screened by an intrusion detection system before being collected or halted, depending on whether the traffic is authentic.

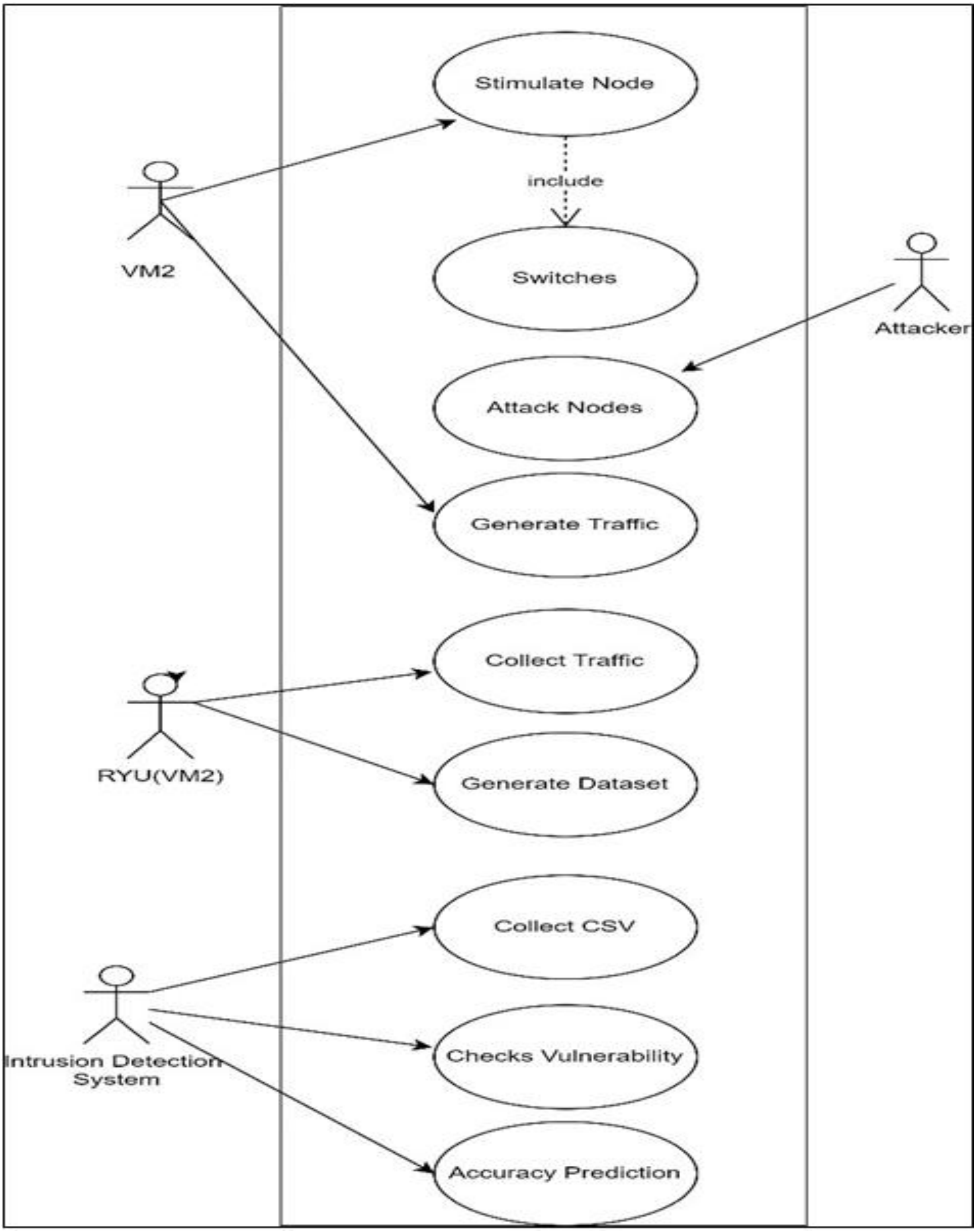


Fig 2 Use Case Diagram

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1031

D. Experimental Design

- *Network Topology:*

Design a simulated network topology using Mininet comprising switches, hosts, and the Ryu Controller. The topology should replicate a real-world network environment with interconnected devices.

- *Attack Scenarios:*

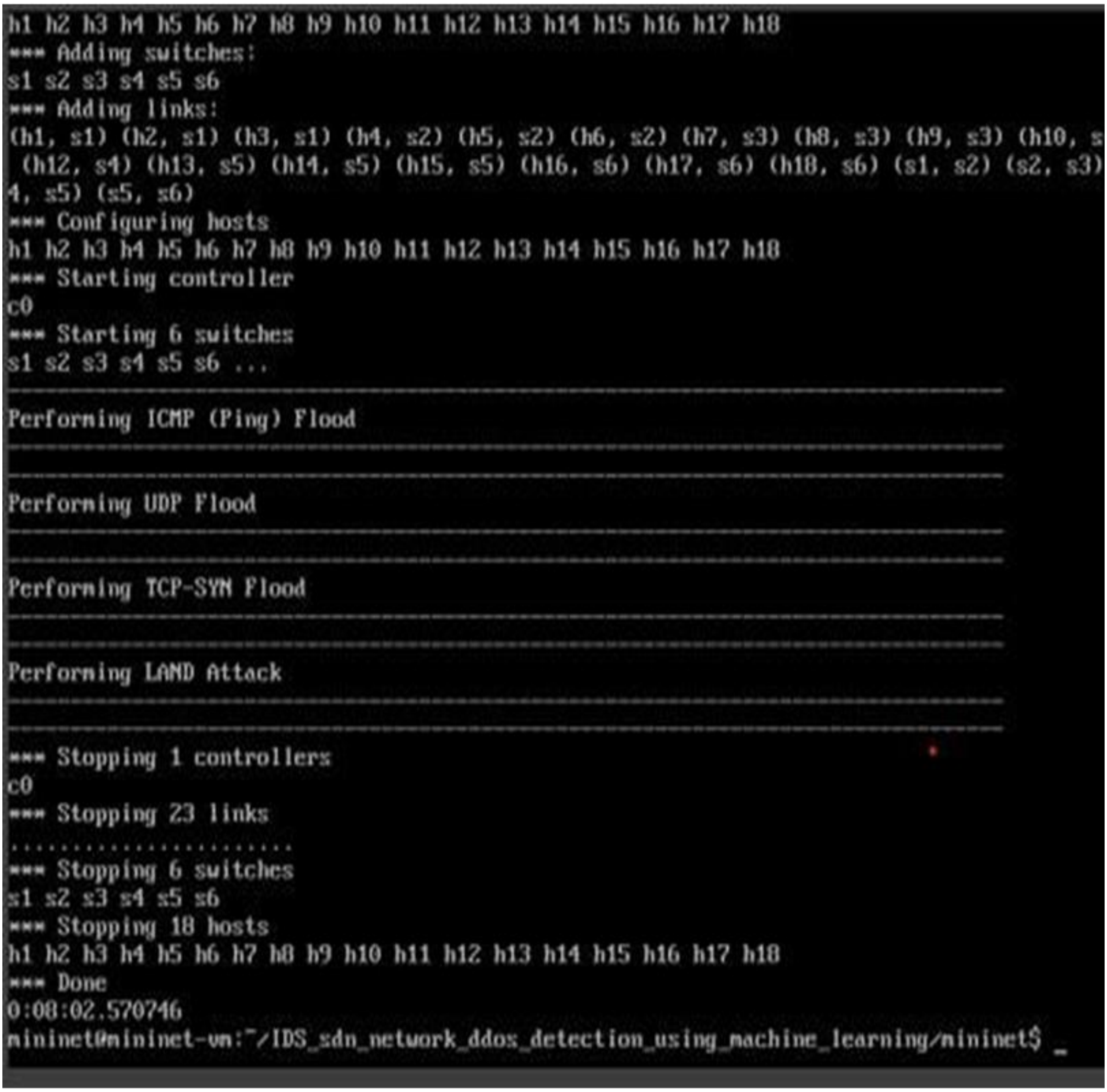
Define a set of DDoS attack scenarios to be simulated during the experiment, including ICMP flood, TCP SYN flood, UDP flood, and others. Specify the characteristics and intensity of each attack, such as packet rate, duration, and target IP addresses.

- *Traffic Generation:*

Utilize traffic generation tools, such as hping, to simulate both benign and malicious traffic flows within the network. Generate traffic patterns representative of normal network behavior as well as DDoS attack patterns for evaluation.

- *Ryu Controller Configuration:*

Configure the Ryu Controller to monitor network traffic, analyze packet headers, and identify anomalous patterns indicative of DDoS attacks. Implement detection algorithms and mitigation strategies within the Ryu application to respond to detected threats.



```
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
*** Adding switches:
s1 s2 s3 s4 s5 s6
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s2) (h5, s2) (h6, s2) (h7, s3) (h8, s3) (h9, s3) (h10, s
 (h12, s4) (h13, s5) (h14, s5) (h15, s5) (h16, s6) (h17, s6) (h18, s6) (s1, s2) (s2, s3)
4, s5) (s5, s6)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
--------------------------------------------------------------------------------
Performing ICMP (Ping) Flood
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Performing UDP Flood
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Performing TCP-SYN Flood
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Performing LAND Attack
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
*** Stopping 1 controllers
c0
*** Stopping 23 links
.......................
*** Stopping 6 switches
s1 s2 s3 s4 s5 s6
*** Stopping 18 hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
*** Done
0:08:02.570746
mininet@mininet-vm:~/IDS_sdn_network_ddos_detection_using_machine_learning/mininet$ _
```

Fig 3 Traffic Generation

III. INTRUSION DETECTION SYSTEM

In addition to the SDN-based detection and mitigation mechanisms implemented using the Ryu Controller, the experimental design includes the integration of an Intrusion Detection System (IDS) to augment threat detection capabilities. The IDS serves as an additional layer of defense, complementing the SDN-based approach by providing advanced threat detection and anomaly analysis capabilities. essential data on pharmaceutical the batch. Incorporating IDS into the experimental setup enriches the research methodology by providing a comprehensive evaluation of hybrid security approaches that combine both traditional and SDN-based security measures.

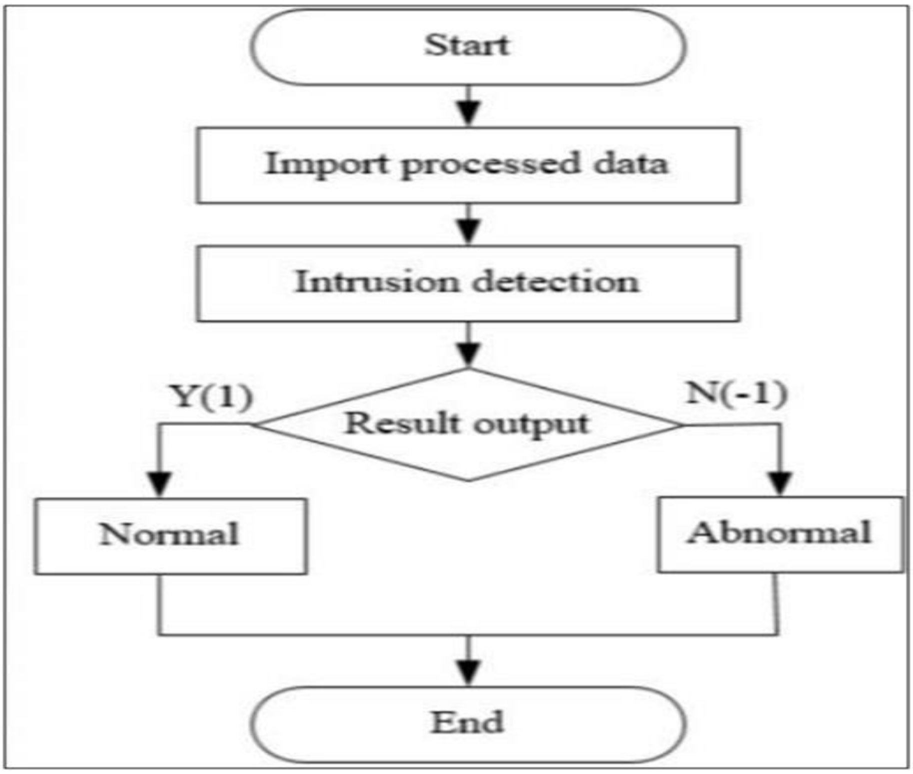


Fig 4 Flow Diagram

IV. DATASET

An essential tool for assessing the efficacy of the SDN-based DDoS detection and mitigation strategy is the data set created throughout the experiment. It includes a wide variety of network traffic patterns, such as malicious traffic produced during DDoS attack simulations and benign traffic under typical operational conditions. The data collection makes it possible to thoroughly analyse and validate mitigation plans, performance measures, and detection algorithms.

Without having to connect to the main Ethereum network.

A. *Machine Learning Model Training and Evaluation:*

➢ *Algorithm Selection:*

- *Random Forest:*

Utilize the Random Forest algorithm for its ability to handle high-dimensional data, handle non-linear relationships, and mitigate overfitting. Random Forest builds multiple decision trees and combines their predictions to improve accuracy and robustness.

- *Decision Trees:*

Employ Decision Trees for their interpretability and simplicity in modeling complex decision boundaries. Decision Trees partition the feature space based on attribute values to classify instances, making them suitable for DDoS detection tasks.

- *K-Nearest Neighbors (KNN):*

Apply the KNN algorithm for its simplicity and effectiveness in classification tasks. KNN classifies instances based on the majority class of their nearest neighbors in feature space, making it suitable for identifying patterns in unlabeled data.

- *Linear Regression:*

Although primarily used for regression tasks, Linear Regression can be adapted for binary classification by thresholding predicted probabilities. While simpler compared to other algorithms, Linear Regression provides insights into the linear relationships between features and target variables.

B. Frontend Setup using Flask Framework:

- *Purpose:*

The frontend setup using the Flask framework facilitates user interaction and real-time DDoS detection by providing a user-friendly interface to input instances and receive predictions from trained machine learning models.

- *Implementation:*

- *Flask Application Structure:*

Developed a Flask web application with a modular structure consisting of routes, templates, static files, and backend logic. Organize application components to ensure scalability, maintainability, and extensibility.

- *Input Interface:*

Design an input interface where users can input features representing network traffic instances for DDoS detection. Provide form fields or input boxes for entering relevant features such as packet rates, packet sizes, and protocol types.

- *Prediction Integration:*

Integrate the trained machine learning models into the Flask application to enable real-time prediction of DDoS attacks. Utilize Flask's routing mechanisms to handle incoming requests, preprocess input data, and invoke model predictions.

- *Output Display:*

Display the prediction results on the frontend interface to inform users whether the input instance is classified as a DDoS attack or benign traffic. Provide clear and intuitive visualizations or messages to convey the prediction outcome effectively.

The integration of machine learning algorithms and the setup of a frontend using the Flask framework enhance the experimental design by enabling real-time DDoS detection and user interaction. By leveraging trained models and intuitive interfaces, the experiment aims to demonstrate the feasibility and practicality of deploying SDN-based DDoS detection solutions in operational environments.

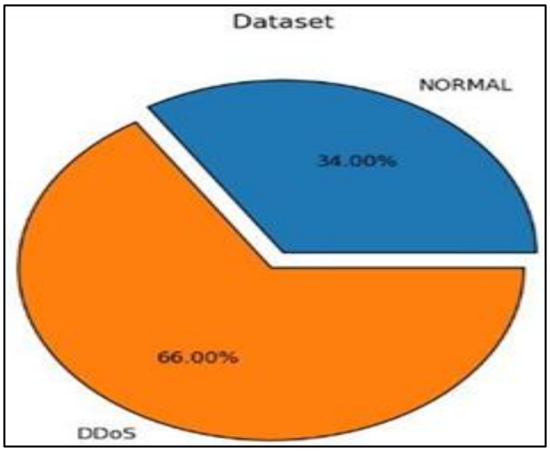


Fig 5 DDOS and Normal Instances in the Dataset

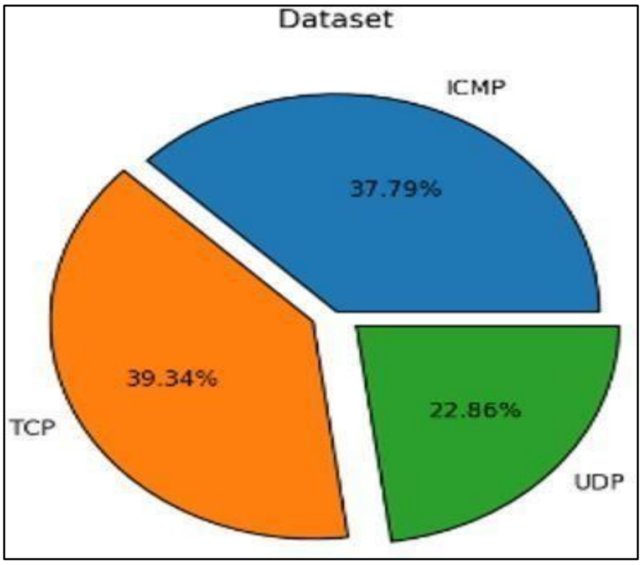


Fig 6 Types of Attack Performed in the Generated Dataset

V. CONCLUSION

In conclusion, our project underscores the potential of SDN and machine learning in fortifying network defenses against DDoS attacks. By combining the agility of SDN with the intelligence of machine learning, we present a proactive and adaptive approach to DDoS detection and mitigation. The integration of real-time detection capabilities with user-friendly interfaces paves the way for effective collaboration between network administrators and automated security systems, fostering a resilient and secure network infrastructure. Furthermore, our project emphasizes the importance of collaborative defense mechanisms and community engagement in combating cyber threats. By sharing insights, best practices, and threat intelligence within the cybersecurity community, we can collectively enhance our defenses and adapt to emerging threats more effectively.

Open-source tools such as the Ryu Controller and machine learning libraries foster collaboration and knowledge sharing, enabling researchers and practitioners to collaborate on innovative solutions for network security it is imperative to consider the ethical implications and responsible deployment of advanced security technologies in real-world settings. While SDN and machine learning offer powerful capabilities for DDoS detection and mitigation, their deployment must be guided by ethical principles, privacy concerns, and regulatory compliance. Ensuring transparency, accountability, and fairness in algorithmic decision- making processes is essential to maintain trust and integrity in network security operations.

REFERENCES

[1]. Nadeau T, Gray K. *SDN: Software Defined Networks*. O'Reilly Media; 2013.

[2]. Jarschel M, Zinner T, Hossfeld T, Tran-Gia P, Kellerer W. Interfaces, attributes, and use cases: a compass for SDN. *IEEE Commun Mag*. 2014; **52**(6): 210-217. doi:10.1109/MCOM.2014.6829966

[3]. Khondoker R, Zaalouk A, Marx R, Bayarou K. Feature-based comparison and selection of software defined networking (SDN) controllers. *In*. 2014; 1-7.

[4]. Correa Chica JC, Imbachi JC, Botero Vega JF. Security in SDN: a comprehensive survey. *J Netw Comput Appl*. 2020; **159**:102595. doi:10.1016/j.jnca.2020.102595

[5]. Einy S, Oz C, Navaei YD. The anomaly- and signature-based IDS for network security using hybrid inference systems. *Math Probl Eng*. 2021. doi:10.1155/2021/6639714

[6]. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*. 2019; **2**(1). doi:10.1186/s42400-019-0038-7

[7]. Rischke J, Sossalla P. Ch. 16 - Machine learning for routing. In: FH Fitzek, F Granelli, P Seeling, eds. *Computing in Communication Networks*. Academic Press; 2020: 289-296.

[8]. Geurts P, Khayat IE, Leduc G. A machine learning approach to improve congestion control over wireless computer networks; 2004; IEEE.

[9]. Park G, Lee W, Joe I. Network resource optimization with reinforcement learning for low power wide area networks. *EURASIP J Wirel Commun Netw*. 2020; **2020**(1). doi:10.1186/s13638-020-01783-5

[10]. Ali MHE. Deep learning-based pilot-assisted channel state estimator for OFDM systems. *IET Commun*. 2020; **15**(2): 257-264.doi:10.1049/cmu2.12051

[11]. Ajaeiya GA, Adalian N, Elhajj IH, Kayssi A, Chehab A. Flow-based intrusion detection system for SDN; 2017: 787-793.

[12]. Ye J, Cheng X, Zhu J, Feng L, Song L. A DDoS attack detection method based on SVM in software defined network. *Secur Commun Netw*. 2018; **2018**:9804061. doi:10.1155/2018/9804061

[13]. Myint Oo M, Kamolphiwong S, Kamolphiwong T, Vasupongayya S. Advanced support vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN). *J Comput Netw Commun*. 2019; **2019**:8012568. doi:10.1155/2019/8012568

[14]. Prakash A, Priyadarshini R. An intelligent software defined network controller for preventing distributed denial of service attack; 2018: 585-589.

[15]. Polat H, Polat O, Cetin A. Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustain For*. 2020; **12**(3): 1-16.

[16]. Elsayed MS, Le-Khac NA, Jurcut AD. InSDN: a novel SDN intrusion dataset. *IEEE Access*. 2020; **8**: 165263-165284. doi:10.1109/ACCESS.2020.3022633

[17]. Meti N, Narayan DG, Baligar VP. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks; 2017: 1366-1371.

[18]. Kaur K, Singh J, Ghumman N. Mininet as software defined networking testing platform; 2014.

[19]. Asadollahi S, Goswami B, Sameer M. Ryu controller's scalability experiment on software defined networks; 2018:1-5.