SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Comparative Analysis of Machine Learning Algorithms for DDoS Detection in SDN Topology

Marwa Alsayed Rshad[1], Mahmoud Mohammed Alzalabani[1], El Said Ahmed Marzouk[1], Haitham Mahmoud Abdelghany[2]

[1]Electronics and Communication Engineering Department, Faculty of Engineering, Mansoura University, Egypt.
[2]Higher Technological Institute of Applied Health Sciences, Shrbin, Egypt.

Abstract
Software-Defined Networking (SDN) is network control centralized, rendering the controller susceptible to a number of attacks. The Distributed Denial of Service (DDoS) attack is one of the most critical risks and it overwhelms the target with false traffic denying any real user the right to use the services. A number of techniques, such as traffic filtering, anomaly-based systems, and machine learning (ML) algorithms are used to identify and prevent such attacks. This paper has gathered a network traffic data to perform training and testing of four machine learning algorithms: Support Vector Machine, Naive Bayes, Decision Tree, and Random Forest. Five constructed features were used: Speed of Flow Entry (SFE), Speed of Source IP (SSIP), Ratio of Flow Pair (RFIP), and the standard deviations of flow packets/bytes (SDFP/SDFB), across five different data split ratios (90:10, 80:20, 70:30, 60:40, and 50:50). The algorithms were evaluated using cross-validation, Accuracy, Precision, Recall/Detection Rate (DR), F1, False Detection Rate (FDR), and AUC metrics. The results demonstrated that all ML algorithms effectively detected DDoS attacks with high accuracy; however, the Random Forest algorithm achieved the best performance, reaching 99.99% accuracy at the 80:20 split ratio.

KEYWORDS: Software-Defined Networking (SDN), Random Forest, Flow-Level Feature Analysis, DDoS Attack Mitigation, Tree Topology.

1. Introduction

The aspect of modern communication networks has led to the paradigm of Software-Defined Networking (SDN) that decouples the control plane from data plane to offer centralized management control and better programmability (Mahar et al., 2024).Since SDN separates the control and data planes, it offers enhanced network scalability and programmability, but this architectural design also introduces vulnerabilities that attackers exploit to disrupt network services (Ashodia & Makadiya, 2022).

On the one hand, SDN creates flexibility in networks, it also brings security weaknesses, specifically, Distributed Denial of Service (DDoS) attacks that may cause the network failure due to the overloading of the SDN controller (Ahmed et al., 2023).

These attacks employ a botnet, which comprises a network of compromised computers controlled remotely by malicious software, unbeknownst to their owners. The purpose of utilizing a botnet in a DDoS attack is to inundate a server or another online target with a barrage of fraudulent requests (Al-Fayoumi & Abu Al-Haija, 2023).

Conventional security systems, such as firewalls and authentication-based intrusion detection systems fail to work properly in detection systems fail to work properly in SDN architectures because network traffic that the infrastructure of the network itself is not simple to scale and manage (Rakissaga et al., 2025). Furthermore, the traditional DDoS detection does not allow adopting new types of attacks or low-rate stealth cyberthreats prevalent in contemporary attacks (Sahosh et al., 2024) (Thwaini, 2022). Also, the traditional techniques for detecting and preventing Distributed Denial of Service (DDoS) attacks are often inadequate due to the unique characteristics of Software-Defined Networking (SDN). Techniques like IP spoofing pose a serious threat to methods that depend on IP address-based filtering because they make it more difficult to distinguish between malicious and legitimate traffic (Aliyu et al., 2020).

[1] Corresponding Author. Electronics and Communication Engineering Department, Faculty of Engineering, Mansoura University, Egypt. E-mail address: engmarwaalsayed@std.mans.edu.eg

In SDN environments, DDoS attacks targeting many objectives illustrate DDoS attack types as Vulnerability Attacks and Flooding Attacks (Sharafaldin et al., 2019). DDoS attacks are characterized as volumetric or protocol-based floods that exhaust network resources (Ali et al., 2023). The data plane Attacks can cause congestion and service degradation (Islam et al., 2020), whereas attacks targeting the control plane can have even more severe consequences, disrupting the entire network operation by compromising the centralized controller (Leqing, 2020) .

Moreover, DDoS attacks can target network interfaces, exploit vulnerabilities in network devices, and disrupt the communication between different components of the network, leading to service outages and compromised network integrity (Facchini et al., 2021).

The evolution of SDN security research has progressed from traditional signature-based detection to advanced machine learning-based approaches, reflecting the need for adaptive and automated defense mechanisms (Fakolujo & Qureshi, 2023) (Kadam, 2024).

Supervised machine learning, in particular, leverages the centralized visibility of SDN controllers to extract flow-level statistics and identify behavioral signatures of malicious traffic. Due to the asymmetry of network data, in which cases of attack are fewer than normal traffic, to make an effective detection, a multidimensional analysis is needed in terms of preciseness, recall, F1 measure, and the Area Under the Curve (AUC).

The proposed study will compare different supervised ML algorithms with the aim of finding the best feature sets that provide a good discriminative performance and, at the same time, have the necessary computational power that is needed to process data in real-time.

This study will offer a basis of lightweight, scalable security architectures that have the potential to secure SDN infrastructures without impacting its fundamental performance advantages through the analysis of these algorithms in a simulated SDN environment.

The structure of this paper is as follows: Section 2 provides an overview of the related work on ML approaches for DDoS detection. Section 3 Methodology for the study. In Section 4, the proposed model Performance Evaluation. The simulation setup and the results obtained are discussed in Section 5. Finally, Section 6 concludes the paper and outlines potential future research directions.

2. Related work

To deal with these threats, ML approaches have been adopted as powerful methods of performing real-time threats detection in SDN-based networks. ML algorithms can be trained on past traffic trends to recognize patterns and deviations and create a better detector that does not use specific rules (Hemanth Kumar et al., 2023).

When used in SDN, ML can increase capability of the system to distinguish between legitimate and malicious flows based on the flow-level characteristics. machine learning algorithms such as Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Decision Tree (DT), and Random Forest (RF) for Distributed Denial of Service (DDoS) detection in Software-Defined Networking (SDN) topology has emerged as a critical area of inquiry due to the increasing reliance on SDN for flexible and programmable network management and the growing threat of sophisticated DDoS attacks targeting its centralized control plane (Wabi et al., 2024) (Ali et al., 2023)

The target of DDoS attacks is not only the device in the data plane, but also the control plane in the SDN. Nowadays, it is more difficult to detect DDoS attacks than in a traditional network because of the complexity of traffic flow features. SDN has a decoupled architecture that allows us to build a new model that defines the DDoS attacks flexibly. Nonetheless, to make sure that the definition of attacks is correct, the dynamics of DDoS attacks within the SDN will be considered in the new contexts. Machine learning (ML) as a recently emerged trend in decision-making in most fields can provide a creative solution to SDN.. A real-time SL classifier forecasts the network state based on past training sets and notifies the controller (Wang et al., 2022)

The specific problem addressed is the effective detection and mitigation of DDoS attacks within SDN environments using machine learning classifiers, focusing on the comparative performance of SVM, GNB, DT, and RF algorithms (Alnatsheh et al., 2023) (Ferdiansyah et al., 2024).

Despite numerous studies, a knowledge gap persists regarding the optimal selection and integration of these algorithms in real-world SDN topologies, especially considering feature selection, dataset variability, and computational efficiency (Rizaldi et al., 2024) (Roopesh et al., 2024) (Wabi et al., 2024).

There are controversies as to what algorithm gives the optimal trade-off of accuracy, speed, and resource consumption; RF tends to be more accurate but more expensive, and GNB is cheaper but faster (Singh, 2021) (Isyaku et al., 2023).

The effects of such disconnect are possible delays in the detection of attacks and insufficient mitigation, which

threaten the loss of network availability and service failure (Hamarshe et al., 2023).

SDN The SDN conceptual framework describes network architecture which decouples control and data planes in order to provide a centralized control (Liu et al., 2023(.

Machine learning software can be used as a classifier to analyze the characteristics of traffic in a network and differentiate between normal and malicious traffic flows (Roopesh et al., 2024).

The correlation between these concepts supports the purpose of the research: to compare and contrast the effectiveness of SVM, GNB, DT, and RF to detect DDoS attacks in SDN based on the theoretical background of supervised learning (Ashwin et al., 2024) (Al Essa and Bhaya, 2023).

This systematic review is aimed at synthesising existing empirical evidence regarding the relative performance of SVM, GNB, DT, and RF algorithms in SDN topologies in detecting DDoS (Pydala et al., 2024). The value of this review is that it brings together different datasets, feature selection approaches, and evaluation measures to guide the considerations of best practices in implementing machine learning-based security solutions in SDN (Almaiah et al., 2024) (Sumukh & Sandhya, 2024).

The reviewed study uses a thorough methodology as it includes choosing peer-reviewed articles published between 2020 and 2025 that test the mentioned machine learning algorithms on SDN DDoS detection (Sumadi, 2020) (Haeruddin et al., 2025).

Inclusion criteria lay stress on empirical measures of performance like accuracy, precision, recall and computational efficiency. Comparative statistical analysis and thematic synthesis of feature engineering techniques are also categorized as analytical frameworks (Mohsin and Hamad, 2022).

This study describes how four supervised ML models, i.e., Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Decision Tree (DT), and Random Forest (RF), can be used in detection of DDoS attacks in SDNs. It identifies five key traffic characteristics, i.e., Speed of Flow Entry (SFE), Speed of Source IP (SSIP), Ratio of Flow Pair (RFIP), and standard deviations of flow bytes (SDFB) and packets (SDFP).

The aim was to compare the performance level of individual classifiers in varying learning ratios under the measures of accuracy, precision, recall, F1-score, and AUC. Its conclusions are supposed to define lightweight scalable solutions to the intelligent DDoS mitigation problem in SDN networks and lead to network security framework systems in the future (Musa et al., 2024).

Table 1: Related work

Study	Focus Area	Key Findings/Contributions
Hemanth Kumar et al. (2023)	ML for threat detection	ML approaches adopted for real-time threat detection in SDN-based networks
Wabi et al. (2024)	ML algorithms for DDoS	SVM, GNB, DT, and RF algorithms critical for DDoS detection in SDN
Ali et al. (2023)	Attack characterization	DDoS attacks characterized as volumetric or protocol-based floods that exhaust network resources
Wang et al. (2022)	Real-time classification	Real-time ML classifier forecasts network state based on past training sets
Alnatsheh et al. (2023)	ML classifier comparison	Comparative performance of SVM, GNB, DT, and RF algorithms for DDoS detection
Ferdiansyah et al. (2024)	Algorithm effectiveness	Focus on effective detection and mitigation using ML classifiers
Rizaldi et al. (2024)	Knowledge gaps	Knowledge gap regarding optimal algorithm selection in real-world SDN topologies
Roopesh et al. (2024)	Feature selection	Issues with feature selection, dataset variability, and computational efficiency
Singh (2021)	Algorithm trade-offs	RF achieves higher accuracy but at greater computational cost; GNB offers faster processing with lower accuracy
Isyaku et al. (2023)	Performance trade-offs	Controversies over best trade-off between accuracy, speed, and resource consumption

Hamarshe et al. (2023)	Detection consequences	Potential delays in attack detection risk network downtime and service degradation
Liu et al. (2023)	SDN conceptual framework	SDN defined as network architecture that decouples control and data planes
Ashwin et al. (2024)	Supervised learning foundations	Research guided by theoretical foundations in supervised learning
Al Essa & Bhaya (2023)	ML algorithm evaluation	Evaluation and comparison of SVM, GNB, DT, and RF efficacy
Pydala et al. (2024)	Systematic review	Synthesis of empirical findings on comparative algorithm performance
Almaiah et al. (2024)	Best practices	Consolidation of datasets, feature selection methods, and evaluation metrics
Sumukh & Sandhya (2024)	Security solutions	Best practices for deploying ML-based security solutions in SDN
Sumadi (2020)	Review methodology	Selection of peer-reviewed studies from 2020 to 2025
Haeruddin et al. (2025)	Recent evaluations	Evaluation of ML algorithms for DDoS detection in SDN
Mohsin & Hamad (2022)	Analytical frameworks	Comparative statistical analysis and thematic synthesis of feature engineering techniques
Musa et al. (2024)	Experimental approach	Two experiments on linear and tree topologies using Mininet with Ryu controller; identification of five key traffic characteristics

3. Methodology

In this study, a machine learning-based approach is implemented to detect the Distributed Denial of Service (DDoS) attacks in Software-Defined Networking (SDN) networks. The whole procedure includes modeling of an SDN network, creation of both benign and attack traffic, obtaining flow-level features, training of classification models, and evaluation of their performance in experimental environments.

Simulation of realistic SDN topologies required a choice of a lightweight Mininet emulator with native OpenFlow switch virtualization support. The network was controlled using the Ryu controller as the control plane software to handle flow rules and communications. The attack traffic was added via the hping3 tool to query non-attack traffic that corresponds to normal streaming and attacking traffic with SYN floods carried out by multiple hosts. The configuration enabled collecting traffic-recognizing traces in both good and DDoS circumstances (Setitra et al., 2023).

Based on the obtained traffic data collected from the network, they were used to derive five important features that are flow-based, and these features served to be the input of the machine learning classifiers (Yao & Guinko, 2025). The features chosen where Speed of Flow Entry (SFE) which reflects the rate at which new flows are initiated, Speed of Source IP (SSIP) which relates to the frequency of packets that are created by an individual IP (Babbar et al., 2024) and Ratio of Flow Pair (RFIP) which reflects the communication balance between the two end points, Standard Deviation of Flow Packets (SDFP) and Standard Deviation of Flow Bytes (SDFB) (Patel et al., 2024).

Reduce misclassification error in DDoS detection by selecting the most relevant features and performing parameter tuning of machine learning models. The features mentioned were arranged in the form of a structured CSV dataset and were used as input to machine learning classifiers to enhance detection accuracy and reduce misclassification errors in identifying DDoS attacks. This approach and methodology presented by (Estupiñán Cuesta et al., 2025).

Four supervised machine training were used to classify the work, including Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Decision Tree (DT), and Random Forest (RF). These models were trained using five training/testing ratios of 90:10, 80:20, 70:30, 60:40, and 50:50 to evaluate their performance in different situations where data is not fully available. The entire execution was done in Python, using the Scikit-learning

library. The classifiers were created in order to predict the state of a particular instance of the traffic flow later referred to as normal or as a part of a DDoS attack using the extracted features only (Sahosh et al., 2024).

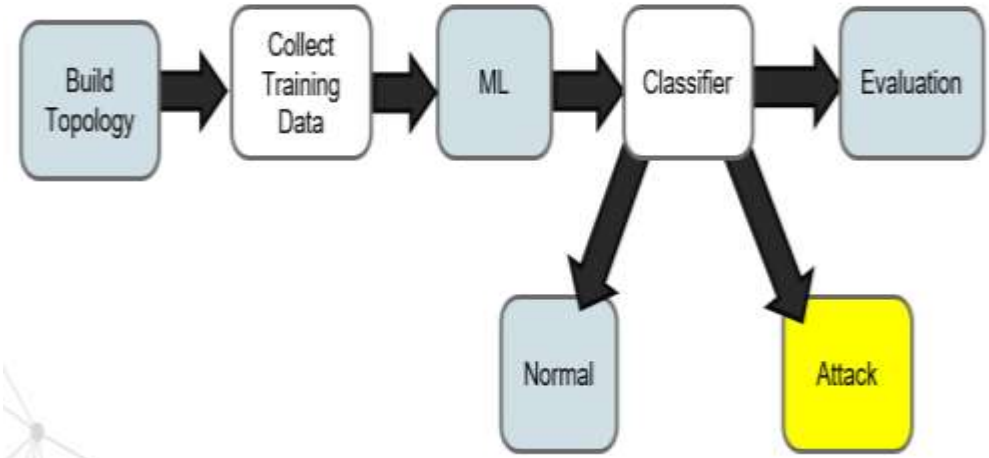


Figure 1. System design: DDoS Attack Detection Using Machine Learning

3.1 Simulation and Traffic Generation of the Network

The simulation was done in SDN environment using Mininet, which is an emulator that can support generation of virtual networks where OpenFlow-enabled switches can be applied. Path Programming: Flow rules were handled and forwarding decision was performed by adjusting the flow rules using the Ryu controller, which is an open-source SDN controller that operates in Python.

There were normal traffic flows and attack flows. It established normal traffic at the network host to host by using ping command to simulate normal traffic in the network. In the case of attack traffic, hping3 tool has been deployed where SYN flood attacks have taken place against a victim node. The resulting traffic was kept watch over and recorded, yielding a collection of move records fascinating both normal and adverse samples (Setitra et al., 2023)(Aliar et al., 2024).

3.2 Collecting Flow Statistics

Monitoring network traffic involves periodically recording specific flow statistics. The described parameters serve as crucial elements that help identify abnormal traffic patterns. The key parameters collected include:

- Speed of Flow Entry Speed (SFE) is a network monitoring tool that tracks to determine the rate of new flow entry additions. Where sudden spikes in entry speed may indicate a DDoS attack or unusual network activity.

$$SFE = \frac{N}{T} \qquad (1)$$

  - **N** (Number of New Flow Entries): The number of new flow entries that have been added to the flow table during the measurement period.
  - **T** (Time Interval): The time period during which the number of new flow entries is measured, typically expressed in seconds.

- The Speed of Source IP (SSIP) is the rate at which an IP address transmits packets over a network, which is vital in the analysis of network performance and in network security as well as the frequency of new connections made by a given source IP address. SSIP is the frequency with which an individual source IP produces traffic. Once this is suddenly high, it can be a sign of scanning, botnet, or a DDoS attack by the IP.

$$SSIP = \frac{\sum IPsrc}{T} \qquad (2)$$

  - **∑IPsrc** (Sum of Source IP Packet Counts): The summation of packets or new connections initiated by a particular source IP address throughout the measure time..
  - **T** (Time Interval): The duration over which the packets or new connections from the source IP are measured, typically expressed in seconds.

- Ratio of Flow Pair (RFIP) is a measure that is employed to assess the amount of two-way communication existing in a network. It is the number of flows that create reciprocal pairs to the total number of flows that are observed. An increase in the value of RFIP shows stronger patterns of two way communication characteristic of normal legitimate traffic. A decrease in RFIP can be an indication of abnormal activity like DDoS attacks in which the traffic is predominantly unidirectional.

$$RFIP = 2 * \frac{PairSum}{N} \quad (3)$$

  - **PairSum** (Sum of Flow Pairs): The number of bidirectional flow pairs that were happening in the network within the measurement time. A flow pair may be two flows between the same source and destination that proceed in opposite directions..
  - **N** (Total Number of Flows): The total number of all flow entries recorded in the network during the measurement period, including both unidirectional and bidirectional flows.
- • Standard Deviation of Flow Bytes (SDFB) identifies the variation of the packet sizes. Big variability indicated by a high SDFB may be an indicator of an unusual traffic pattern whereas small variability indicated by a low SDFB is an indicator of a stable size of packets that is typical of a stable network behavior.

$$SDFB = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(packet_i - MeanPackets)^2} \quad (4)$$

  - **N** (Total Number of Packets or Flows): The total count of packets or flow-byte measurements included in the calculation.
  - **packet$_i$** (Flow Byte Value of the i-th Packet): The size, in bytes, of the i-th packet or flow entry being analyzed.
  - **MeanPackets** (Average Packet Size): Mean of all the sizes of packets in the measurement set.
  - Standard Deviation of Flow Packets (SDFP) measure looks at the frequency changes in flow packets during the analysis process. It quantified the variation in the numbers of packets per flow. Big deviation is one of the indications of an irregular or suspicious traffic behavior whereas small deviation implies predictable and stable network behavior..

$$SDFP = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(bytes_i - MeanBytes)^2} \quad (5)$$

  - **N** (Total Number of Packets or Flow Records): The total number of packet-count observations or flow entries included in the analysis.
  - **bytes$_i$** (Packet Count of the i-th Flow): The number of bytes or packets recorded for the i-th flow during the measurement period.
  - **MeanBytes** (Average Packet Count): The arithmetic mean of all packet or byte counts across the observed flows.

The parameters collected through the experiment are saved in a result.csv file with a structured CSV format for use as training data by machine learning algorithms.

3.3 Feature Extraction

Flow-level features were found by processing recorded traffic to provide the possibility to detect DDoS attacks with necessary accuracy. The features chosen where Speed of Flow Entry (SFE) which reflects the rate at which new flows are initiated, Speed of Source IP (SSIP) which relates to the frequency of packets that are created by an individual IP and Ratio of Flow Pair (RFIP) which reflects the communication balance between the two end points, Standard Deviation of Flow Packets (SDFP) and Standard Deviation of Flow Bytes (SDFB. All these five features were selected due to their sensitivity to shortcomings of abnormal behavior and effectiveness in prior DDoS detection practices The extracted features were stored in a CSV file to be used to train and validate the model.

3.4 Training and Classification

Four supervised machine learning algorithms, including Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Decision Tree (DT), and Random Forest (RF), were used to conduct the classification task. The amount of features was extracted with all of them trained on the same features with five different train-test split settings: 90:10, 80:20, 70:30, 60:40, and 50:50. This was to be able to analyze how each of the models performed under different conditions of availability of the data available and to make a balanced comparison. It has been

implemented on the Python programming language with the help of the Scikit-learn library, and the classifiers were trained to determine whether a traffic example was a regular or an attack flow (Sahosh et al., 2024).

4. Performance Evaluation

The results of every classification model were evaluated against several common evaluation measures. These were the accuracy that is what gauge the general correctness of the predictions; the precision that indicates the ratio between the number of properly predicted attacks to each predicted attack; the recall (or detection rate), which reflects the percentage of actual attacks compared to those that were identified appropriately; and the F1-score, a function that balances between the accuracy and the recall level. Also, the Area Under the ROC Curve (AUC) was employed in assessing the capability of the classifier to make a distinction between classes.

A 5-fold cross-validation technique as shown in in Figure 2. was used to rule out overfitting, to assess models against other settings in the dataset. Confusion matrices were created to visualize values of true positive and false positive as shown in Figure 3, false negative, and true negative values to learn more about reliability of the model. The composite analysis was also useful in finding the optimum trade-off algorithm in terms of its detection performance using least amount of computation.

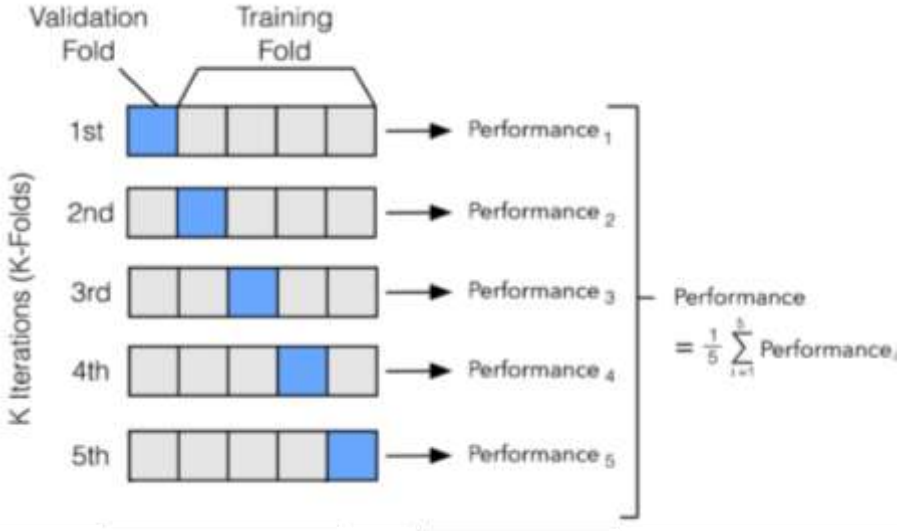


Figure 2. Cross-Validation with K-Fold =5

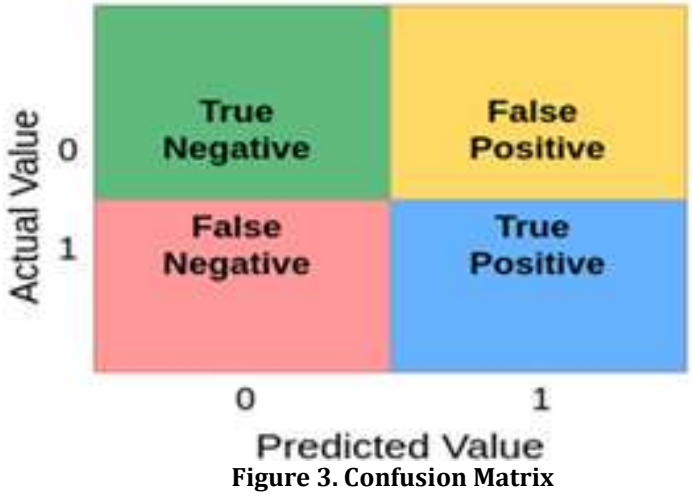


Figure 3. Confusion Matrix

- Accuracy: It measures the overall correctness of the model. It is the ratio of correct predictions to the total number of predictions. This is the most fundamental metric used to evaluate the model. The formula is given by

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (6)$$

- Precision: It shows the exact number of attacks detected among all predictions. Calculate the ratio of true positives to the sum of true and false positives. It basically analyses optimistic predictions.

$$Precision = \frac{TP}{(TP+FP)} \quad (7)$$

- Recall (sensitivity, true positive rate (TPR)): It measures a model's ability to identify real attacks. Recall is the ratio of true positives to the sum of true positives and false negatives. It basically analyses the number of correct positive samples.

$$Recall = \frac{TP}{(TP+FN)} \quad (8)$$

- F1 score: which calculates a weighted average between Precision and Recall providing a holistic model evaluation. The F1 score is the harmonic mean of precision and recall. It is observed that during the precision-recall trade-off, as Precision increases, Recall decreases, and vice versa. The goal of the F1 score is to combine precision and recall.

$$F1\ score = \frac{2Precision*Recall}{(Precision+Recall)} \quad (9)$$

- Cross-Validation: It is a machine learning method for valuating how well a model functions on unseen data. It reduces overfitting and provides a more accurate estimate of a model's performance.

5. Results and Discussion

In this section, a DDoS detection engine deployed inside a Software-Defined Networking (SDN) framework, based on machine learning, is introduced. The first one is to assess the capability of the system to differentiate genuine versus malignant traffic with strong precision. The performance of the system is measured by using key evaluation metrics like accuracy, precision, recall and F1-score. The influence of training data percentage and network topology also on detection performance are included in the analysis.

The network was simulated through Mininet, which enables one to create lightweight virtual network topologies based on virtual machines. Mininet is selected because of its flexibility and ease to contain realistic SDN scenarios. It allows an extensive variety of SDN controllers and OpenFlow-enabled devices. In the context of this paper, Ryu SDN controller is chosen because it is a modular architecture developed in Python, and it is also a popular research SDN controller in academia. Ryu provides extendibility, ability to see traffic in real time and compatibility with Mininet (Smida et al., 2020)

The hardware configuration comprised an HP laptop with Windows 11 installed, featuring an Intel Core i5-5200U processor and 16 GB of RAM. The exercise itself was implemented on a virtualized system with VirtualBox. Ryu version 4.30 was deployed as a management tool to handle SDN control logic. The simulation was a custom topology that was loaded through Python scripts in Mininet.

When working with the simulation, Mininet brings up the virtual network and links switches to the Ryu controller through the OpenFlow protocol. Discovery messages are sent by the controller to the switches, where the network topology is mapped and a graph representation is generated upon which decisions can be made. With traffic going between hosts, the controller fills flow tables on the switches by installing forwarding rules that are dependent on traffic patterns observed and logic predetermined. Flow tables begin as empty and packet-in messages are sent to the controller, where it takes a decision on what to do. The switches start forwarding packets automatically when flow rules are installed.

Data flows between hosts are generated by traffic simulation tools e.g. ping to simulate normal as well as attack conditions. Such an installed infrastructure can enable the proposed ML-based detection model to be tested in an SDN environment. This is aimed at enhancing more accurate detection and network reliability by means of sharing the decision making with adaptive traffic control, flow analysis and network reliability analysis.

DDoS Detection Using Machine Learning

The experiment that was conducted in this section marks the first of a series of experiments that would be run to assess the effectiveness of machine learning models in identifying the Distributed Denial-of-Service (DDoS) attacks within Software-Defined Networking (SDN) configurations. In Experiment, A one-switch and twenty-hosts SDN topology is emulated in the simulation environment with Mininet, an SDN flexible and easy-to-use simulator. This layout enables us to test real-time communication with SDN features (controllers, switches and hosts) and intervene and track network-based attacks.

The main aim of the present experiment is to test how different machine learning algorithms would perform on detecting DDoS attacks using labeled network traffic datasets. RYU controller is used to inspect traffic and

provide mitigation measures by dynamically updating flow tables as per the traffic pattern. The hping3 tool is used to simulate the attack and the controller configuration blocks the malicious traffic by disabling the attacker's port disabled until a specified period.

A certain real-life dataset of SDN traffic is employed in the experiment and consists of 302,003 flow records, including normal (159,974) and attack traffic (142,029). The data contains five main engineered features, which are Source Flow Entry (SFE), Speed of Source IP (SSIP), Ratio of Flow Pair (RFIP), Standard Deviation of Flow Packets (SDFP) and Standard Deviation of Flow Bytes (SDFB). These characteristics are used as input to four models of supervised learning models, which include Support Vector Machine (SVM), Gaussian Naive Bayes (GNB), Decision Tree (DT) and Random Forest (RF).

The models are all trained and tested on various learning rates (90%, 80%, 70%, 60% and 50%) in order to determine how they generalize, how precise they will prove to be and detection accuracy to various data availability conditions. Outcomes are then matched on a basis to include Detection Rate (DR), False Detection Rate (FDR), Accuracy, Precision, F1-score, Cross-Validation Score, and area under the ROC Curve (AUC). Such an arrangement enables us to study how the size of training affects model robustness and performance, and determine the best

learning rates and the best classifier DDoS in SDN topologies can be assured of being detected

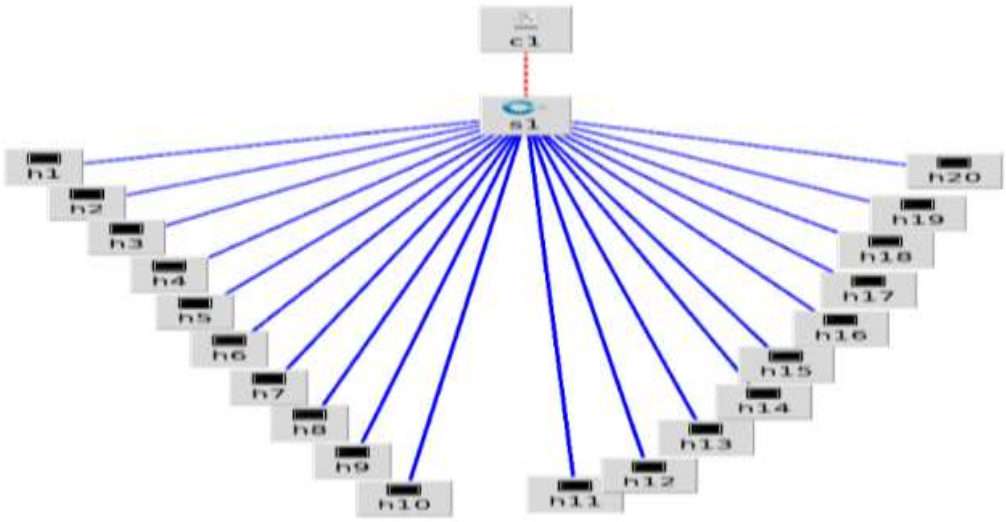


Figure 2. Network Architecture for the Experiment

The following figures show The Experimental stages.

The RYU controller starts, and the topology is configured using Mininet, then checking connectivity by (pingall)

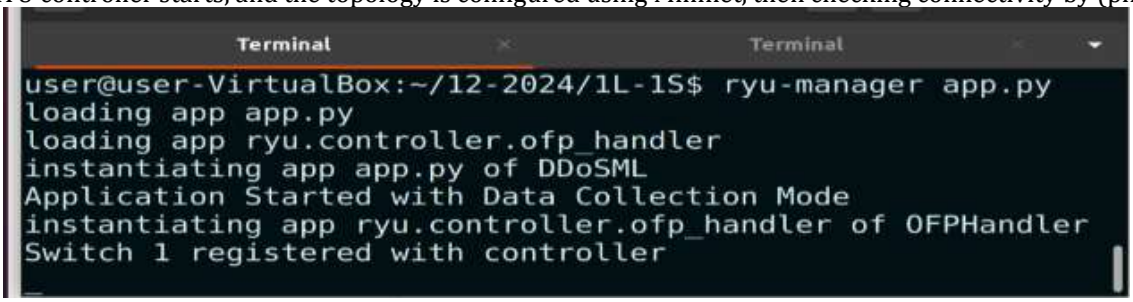


Figure 3. Start RYU for the Experiment

Figure 4. Start RYU for the Experiment

After trading the model, the controller's operation was tested by creating an attack from h1 to h20 using the hping3 tool.

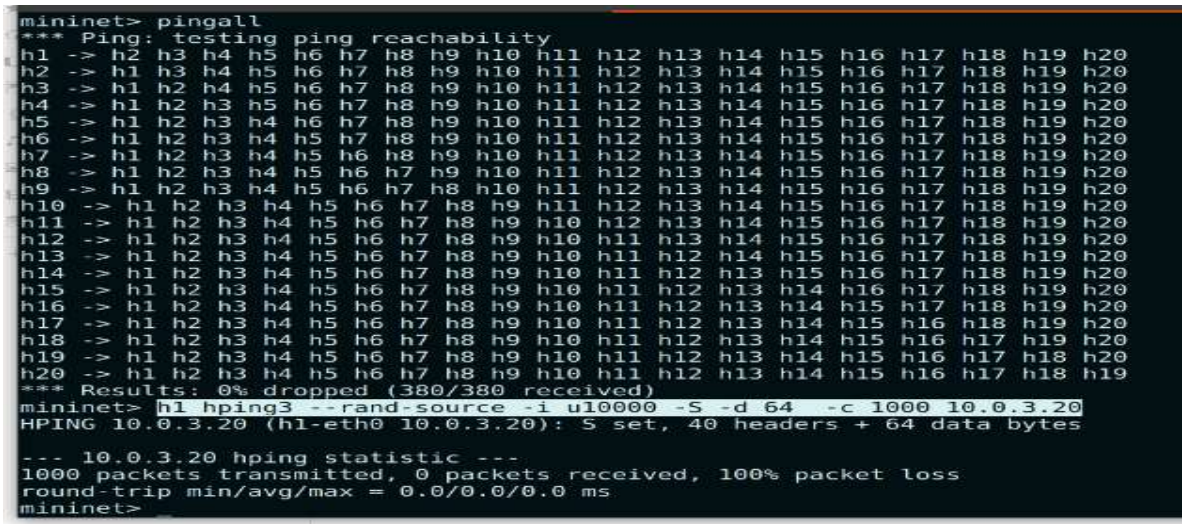


Figure 5. Creating an attack for Experiment

The controller detected the attack and blocked the source port on the switch for 60 seconds, as programmed.

Figure 6 Block port for Experiment

After that, an attempt was made to connect to the victim host, and it could not be reached until the port was unblocked by the controller.

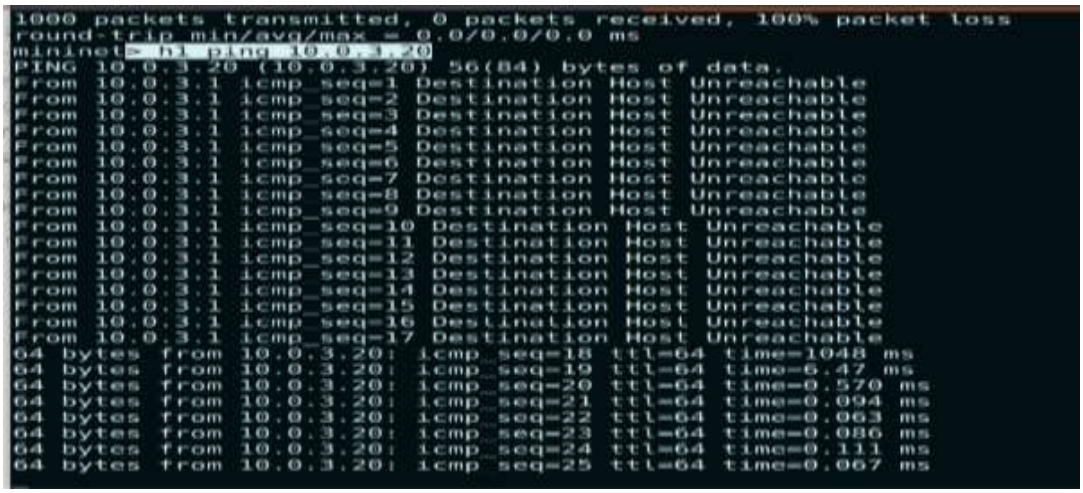


Figure 7. Test connection after attack in Experiment

In the last stage, the system is evaluated by calculating evaluation metrics.

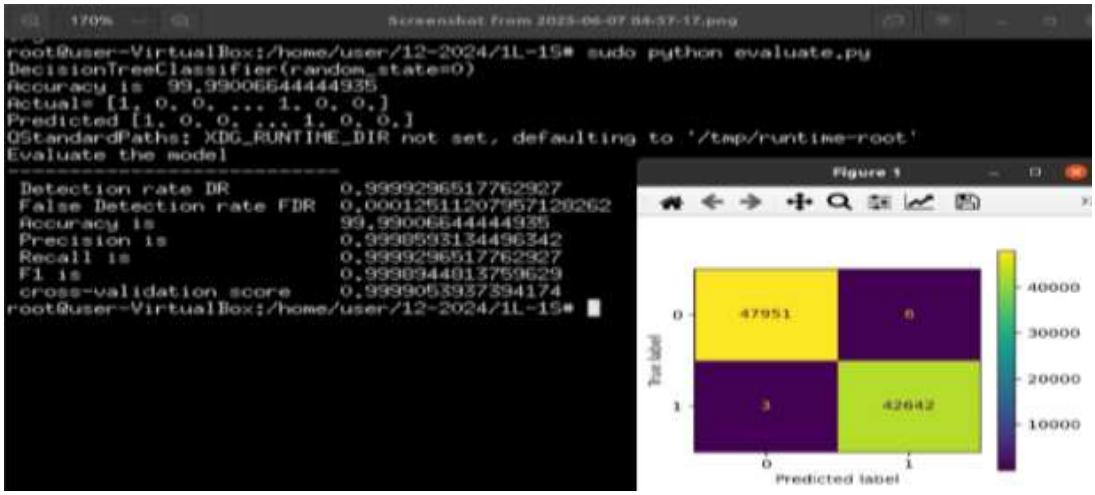


Figure 9. Evaluates the system for the Experiment

Table 2. Sample Data Table for Network Traffic Features and Classification for the Experiment

```
====================
DATASET INFORMATION
====================
Total: 302003 records
TYPE 0: 159974 records
TYPE 1: 142029 records
```

	SFE	SSIP	RFIP	SDFP	SDFB	TYPE
0	-4	0	1.0	3.485775	341.605901	0
1	2	0	1.0	10.115013	991.271294	0
2	0	0	1.0	7.801506	764.547564	0

The network traffic dataset used for DDoS attack detection for the Experiment is shown in Table 2. The flow record contains five key features, including SFE (Source Flow Entry), SSIP (Speed of Source IP), RFIP (Ratio of Flow Pair), SDFP (Standard Deviation of Flow Packets), and SDFB (Standard Deviation of Flow Bytes). The database contains 302,003 network flow items, each in a single row, as the foundation for training and validating

DDoS attack detection algorithms using machine learning models. There were 159974 normal flows and 142029 attack flow

Table 3. Network Traffic Features Importance for the Experiment with SVM

Feature	Importance	Percentage
SDFB	0.9727	97.27%
SFE	0.0138	1.38%
SSIP	0.0135	1.35%
RFIP	0	0.00%
SDFP	0	0.00%

Table 3 showed the distribution of importance among the five features in the SVM model that continuously use the model relies almost exclusively on SDFB (Standard Deviation of Flow Bytes), which accounts for 97.27% of its predictive power. Other features, including RFIP and SDFP, show zero contribution, indicating that SVM is highly sensitive to fluctuations in data volume rather than flow relationships or packet frequency.

Table 4. Network Traffic Features Importance for the Experiment with GNB

Feature	Importance	Percentage
SDFB	0.4779	47.79%
SDFP	0.3409	34.09%
SSIP	0.1121	11.21%
RFIP	0.0637	6.37%
SFE	0.0053	0.53%

The table 4 would demonstrate how the five features in the GNB were distributed in terms of importance. The two models highlight SDFB and SDFP with more than 80 percent of the significance. This is in line with statistical nature of Gaussian, which is more inclined by features that can be measured in variance. The moderate influence is on SSIP and RFIP with an almost insignificant impact on SFE..

Table 5. Network Traffic Features Importance for the Experiment with DT

Feature	Importance	Percentage
SDFP	0.3859	38.59%
SSIP	0.3344	33.44%
RFIP	0.2349	23.49%
SDFB	0.0285	2.85%
SFE	0.0163	1.63%

Table 5 showed the mean of the importance of the five features included in the DT model. Having a definite inclination toward SDFP, SSIP and RFIP, it is possible to recommend including the fact that the frequency of

packets, the behavior of the source and the relationships between outbound and incoming flows are essential indicators. The role of SDFB and SFE is not much but also participates in the decision-making process.

Table 6. Network Traffic Features Importance for the Experiment with RF

Feature	Importance	Percentage
RFIP	0.9097	90.97%
SDFB	0.0444	4.44%
SSIP	0.0246	2.46%
SFE	0.019	1.90%
SDFP	0.0024	0.24%

Table 6 demonstrates the weight of significance of the five features. The model is highly biased towards RFIP that contributes 90.97 percent of its weight. This is an indication of the model to include structural flow relationships as major predictors. The other characteristics play a small role meaning that Random Forest prefers inter-flow dynamics as compared to statistical or time specific aspects.

Table 7 . Comparative Summary of Feature Importance

Feature	SVM	GNB	DT	RF	Conclusion
SDFB	Very High	High	Low	Moderate	Crucial in statistical models
SDFP	None	High	High	Very Low	Effective in DT and GNB
SSIP	Low	Moderate	High	Low	Behavioural indicator
RFIP	None	Low	Moderate	Very High	Essential in RF
SFE	Low	Negligible	Low	Low	Least impactful overall

Table 7 presents the outcomes of the feature importance analysis that displays the highlights of the various ways that the classifiers react to flow-level features. Gaussian NB and other statistical models are based on the vitality of the packets and bytes, whereas the tree-based models (Decision Tree and random Forest) are focused on the behavioral and relational resemblances. Such diversity supports the choice of keeping all the features within the study framework, which will provide the fair comparison of models when a similar input structure is provided.

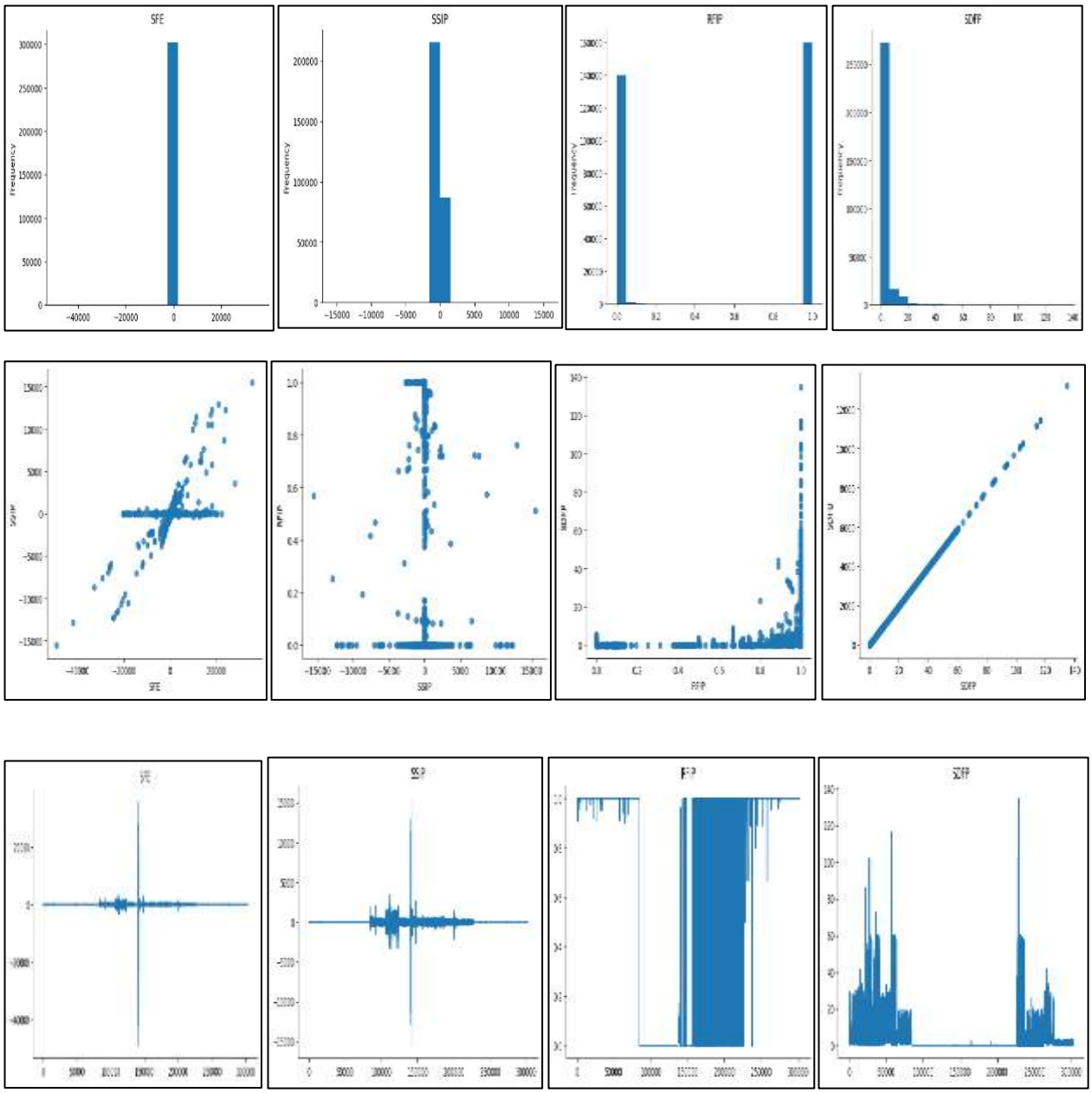


Figure 8. Distribution and correlation of data features for the Experiment

Figure 10 shows the statistical natures, correlation profiles, and time change of the five picked traffic features-Speed of Flow Entry (SFE), Speed of Source IP (SSIP), Ratio of Flow Pair (RFIP), Standard Deviation of Flow Packets (SDFP), and Standard Deviation of Flow Bytes (SDFB), which are the basis of the DDoS detection framework.

The frequency distributions (top row) show the predominant ranges of each feature, their variability, and skewness. As the example of SFE and SSIP shows, they exhibit high peaks around a small range, implying that under regular network operation, flow entries and source IP request rates do not vary much. SDFP, however, and SDFB have more heavily tailed distributions that represent variability in packet and byte size distributions, which can be symptoms of abnormal/malicious network traffic. As shown in FIP, there is a bimodal distribution close to 0 and 1 which is consistent with the structural disparity between legitimate and attack flows.

The scatter plots (middle row) give us an idea about the extent of inter-feature correlation. A strong linear correlation is found between SDFP and SDFB as would be expected given that often packet-level variability is proportional to byte-level variability. In contrast, laxer or non-linear relationships are seen among other feature pairs, SSIP and RFIP, and they measure different aspects of network traffic behavior. This variability in terms of

correlation would imply that the features chosen are not only not redundant but complementary, pointing to an even stronger collective discriminative ability when used jointly to identify categories in classification problems.

The lower row depicts time-series plots of individual feature change throughout time of the experiment being conducted. The stillness of the background on the normal section is highly contrasted to the abrupt increases and uneven fluctuations brought about in the attack scenes. As it could have been anticipated, SFE and SSIP indeed contain faster spikes in the cases of DDoS conditions, which involves high-frequency flow requests of the controller. Similar bursts in SDFP and SDFB are a signal of the discontinuity of a packet and a byte traffic induced by malicious traffic. These time signatures also support the same sensitivity of the selected features in capturing the dynamics in the attacks.

The multi-perspective perspective emphasizes the validity and centrality of the chosen feature set in general. The distribution plots justify the nature of variable characteristics of these values; correlation plots indicate complementary interaction, and order analysis of time-successiveness confirms that they are susceptible to aberrant traffic flow. These findings justify why the five features are suitable in developing the lightweight and efficient machine learning-based models to identify the presence of DDoS attacks within SDNs.

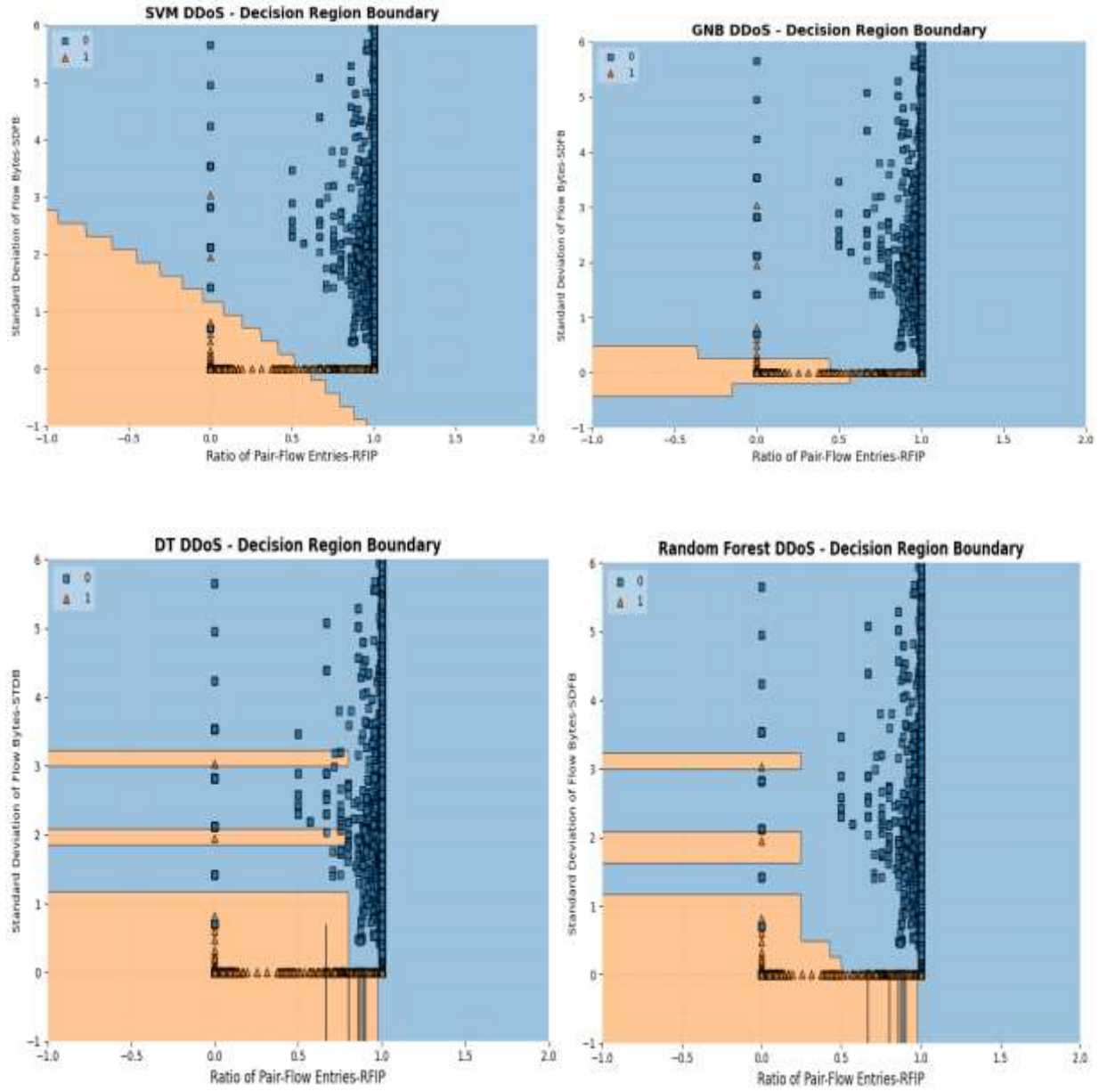


Figure 9. Decision Region Boundaries for DDoS Detection for the Experiment

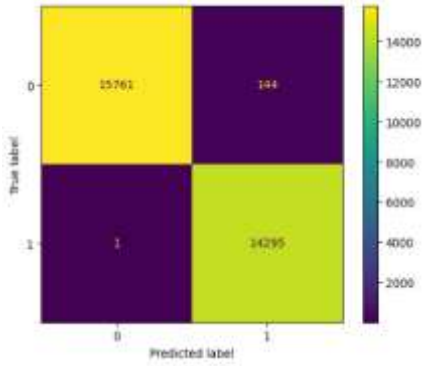
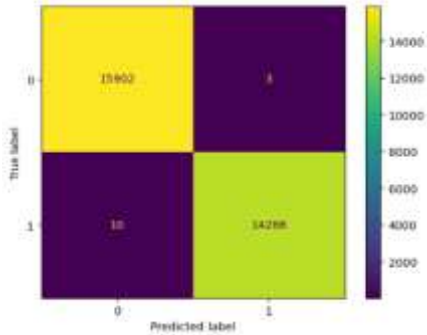
Figure 11 gives a clear illustration of the decision region boundaries of DDoS detection attacks in Experiment, whereby various machine learning models are applied to classify the data by using the Ratio of Flow Pair (RFIP) in the x-axis and the Standard Deviation of Flow Bytes (SDFB) in the y-axis. According to the analysis of the feature importance table, which is presented in Table 6, a good consideration to select (RFIP) and (SDFB) as the axes to learn the decision boundaries is a good choice. Such a combination makes it possible to represent extensively not only the behavioral peculiarities of the source but also the statistical distribution of the network traffic. The incorporation of the two dimensions helps in improving the accuracy of the classification and helps in having a better visual understanding of the model performance in relation to the various machine learning algorithms..

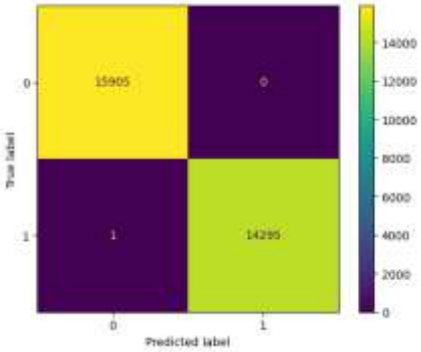
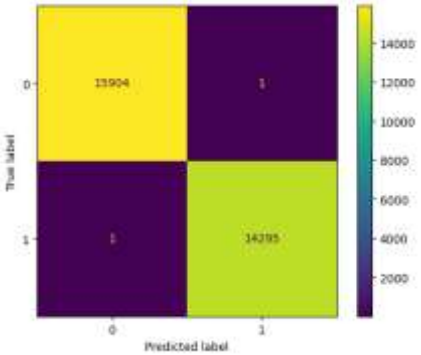
The background coloring indicates the classification areas, with the orange portion representing attack traffic. Different shape markers in the plot indicate the data points, where squares (■) show regular traffic (label 0) and triangles (▲) represent DDoS attack traffic (label 1). Each model demonstrates how well it discriminates between the two classes (0=normal traffic, 1=DDoS). With its linear approach, the Support Vector Machine (SVM) draws a linear boundary exhibits sharp, linear boundaries, reflecting its sensitivity to margin maximization and its reliance on support vectors. Gaussian Naive Bayes (GNB) model is a probabilistic model which works under the assumption that the values of features in each category of data are normally distributed and, therefore, the decision boundary is smooth and curvy. As a result of this assumption, there is the error of classification in common areas. The Decision Tree also employs threshold-based partitioning in order to subdivide the data, that is, forming clear and distinct boundaries, which adapts to local structures and which offers clean separation. They however are vulnerable to overfitting and noise. Random Forest (RF), comprising of several trees that are connected to each other, is better at classification accuracy, less overfitting, and less rigid boundaries allowing flexible boundaries which make it work well in complex and overlapping areas giving a decent balance in generalization versus discrimination.

Performance evaluation for learning rate 90%

The next part involved the evaluation of four algorithms (SVM, GNB, DT, and RF) on a training set size (90 percent, 30200, 5) and a test set size (10 percent, 271803, 5), and the results obtained were attributed to their respective algorithms as illustrated in the next part.

Table 8. Different ML algorithms with performance metrics, with 90% training for the Experiment

Algorithm	Confusion Matrix	Evaluation	
SVM		DR(Recall) FDR Accuracy Precision F1 cross-validation AUC	0.999930050363738 0.009053756680289217 0.9951988344756796 0.990027010180760 4 0.9949538889855577 0.9952207833092045 0.999801
GNB		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9993005036373811 0.0001886199308393587 0.9995695506771299 0.9997900482888935 0.9995452160223893 0.9993892610703424 0.999905

Decision Tree		DR (Recall) 0.999930050363738 FDR 0 Accuracy 0.9999668885136254 Precision 1.0 F1 0.9999650239585883 cross-validation 0.9999043422396854 AUC 1.0
Random Forest		DR (Recall) 0.999930050363738 FDR 6.287331027978623e-05 Accuracy 0.9999337770272507 Precision 0.999930050363738 F1 0.999930050363738 cross-validation 0.999922737915967 AUC 1.0

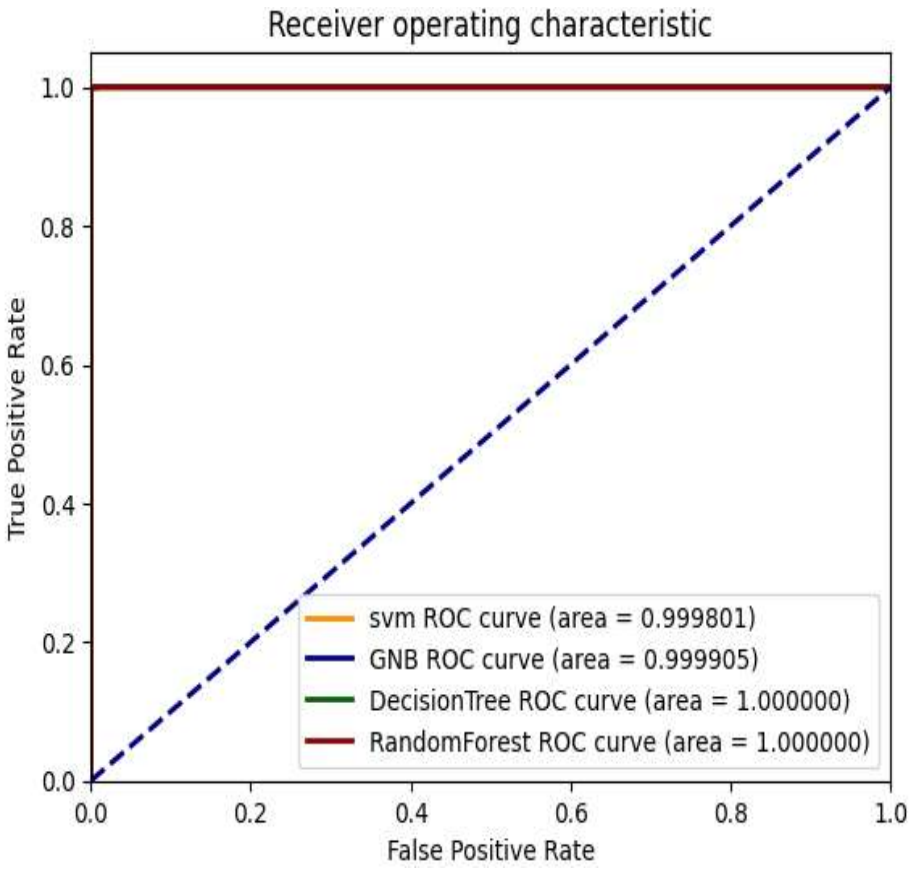
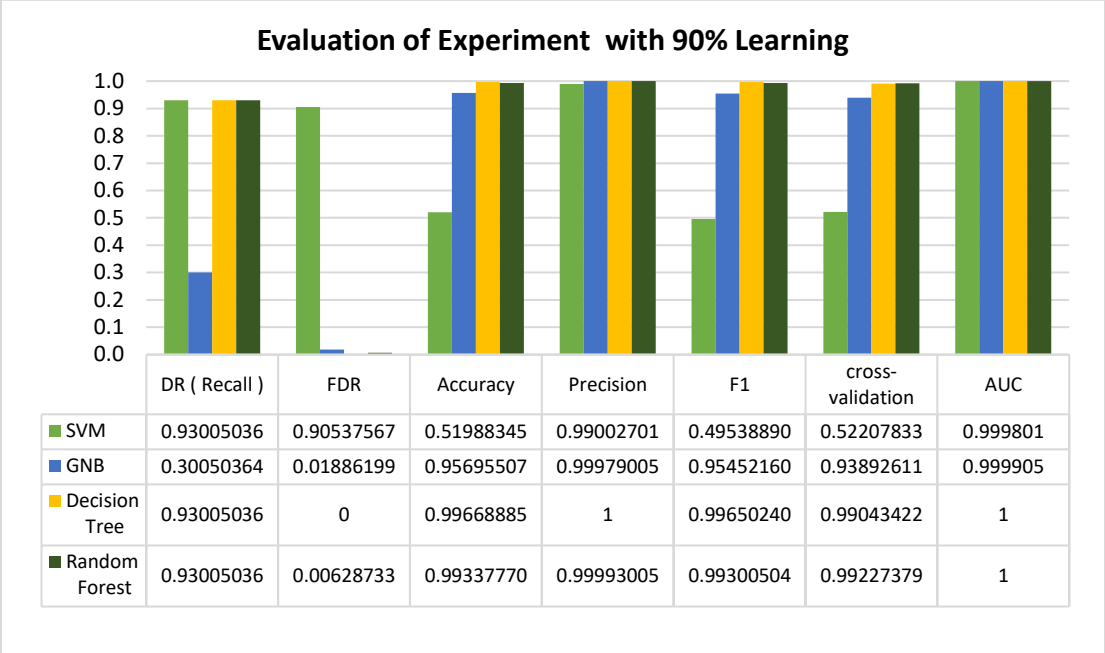


Figure 10. ROC and AUC with 90% Learning for the Experiment



	DR (Recall)	FDR	Accuracy	Precision	F1	cross-validation	AUC
SVM	0.93005036	0.90537567	0.51988345	0.99002701	0.49538890	0.52207833	0.999801
GNB	0.30050364	0.01886199	0.95695507	0.99979005	0.95452160	0.93892611	0.999905
Decision Tree	0.93005036	0	0.99668885	1	0.99650240	0.99043422	1
Random Forest	0.93005036	0.00628733	0.99337770	0.99993005	0.99300504	0.99227379	1

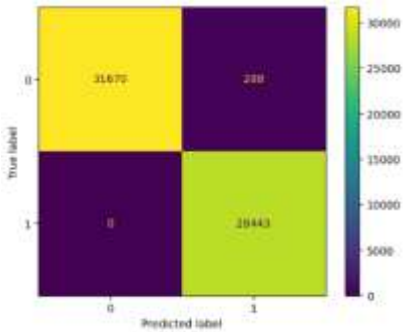
Figure 11. MLA Evaluation comparison with 90% Learning for the Experiment

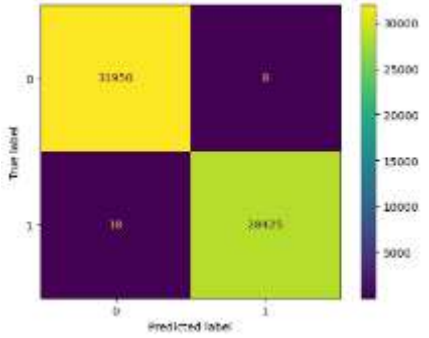
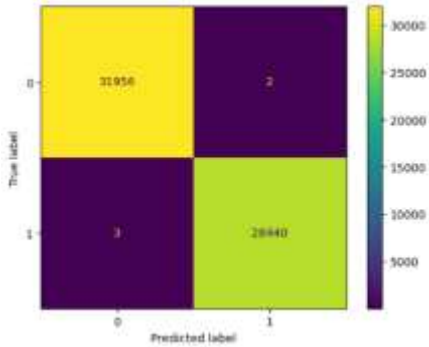
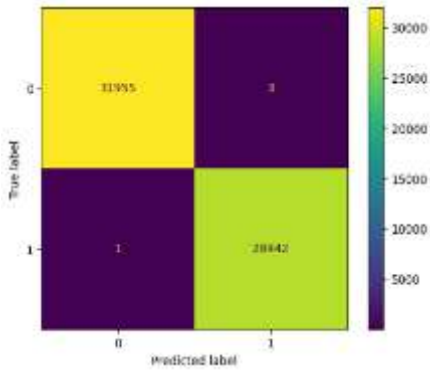
As shown in Figure 13, it presents a comparative analysis of the performance of four machine learning models (SVM, GNB, Decision Tree, and Random Forest) based on a 90% learning rate, evaluating performance using several metrics such as detection rate (DR), false detection rate (FDR), accuracy, precision, F1, cross-validation, and AUC. The Random Forest model achieved the highest values across DR, cross-validation, and AUC, outperforming the other models. The model's robustness in pattern recognition. However, the ideal results may indicate overfitting. The Decision Tree achieved the highest Precision and the lowest false detection rate (FDR). GNB has the lowest DR value. SVM suffers from an evident lack of accuracy and a high error rate. AUC > 0.9 for all models indicates high efficiency, but the superiority of the random forest (1.00) suggests an exceptional ability to differentiate classes even with unseen data

Performance evaluation for learning rate 80%

The SVM, Decision Tree, Random Forest, and GNB models demonstrated exceptional performance across all evaluation metrics, with a training dataset size of 80% (241,603) and a test dataset size of 20% (60,400)

Table 9. Different ML algorithms with performance metrics, with 80% training for the Experiment

Algorithm	Confusion Matrix	Evaluation	
SVM		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	1 0.00901182802428187 0.9952318670220691 0.9899759841286415 0.9949627453038095 0.9951780167083513 0.999804

GNB		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9993671553633583 0.0002503285562300 5195 0.9995695435506035 0.9997186367952731 0.9995428651803925 0.9993625885205327 0.999890
Decision Tree		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.999894525893893 6.258213905751299e-05 0.9999172199135776 0.9999296814570002 0.9999121033664411 0.9999254974596624 0.999949
Random Forest		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.999964841964631 9.387320858626947e-05 0.9999337759308621 0.9998945333098963 0.99992968640135 0.9999213583868146 1.0

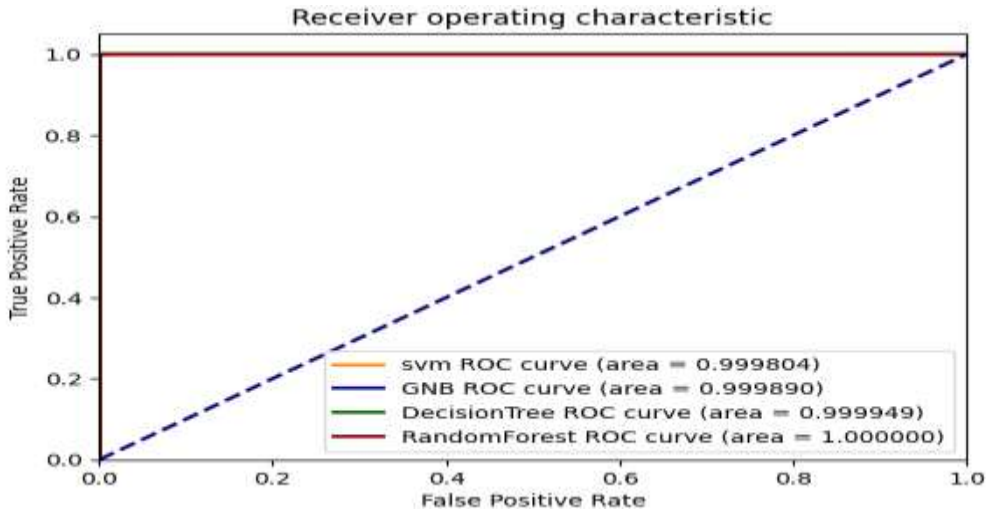
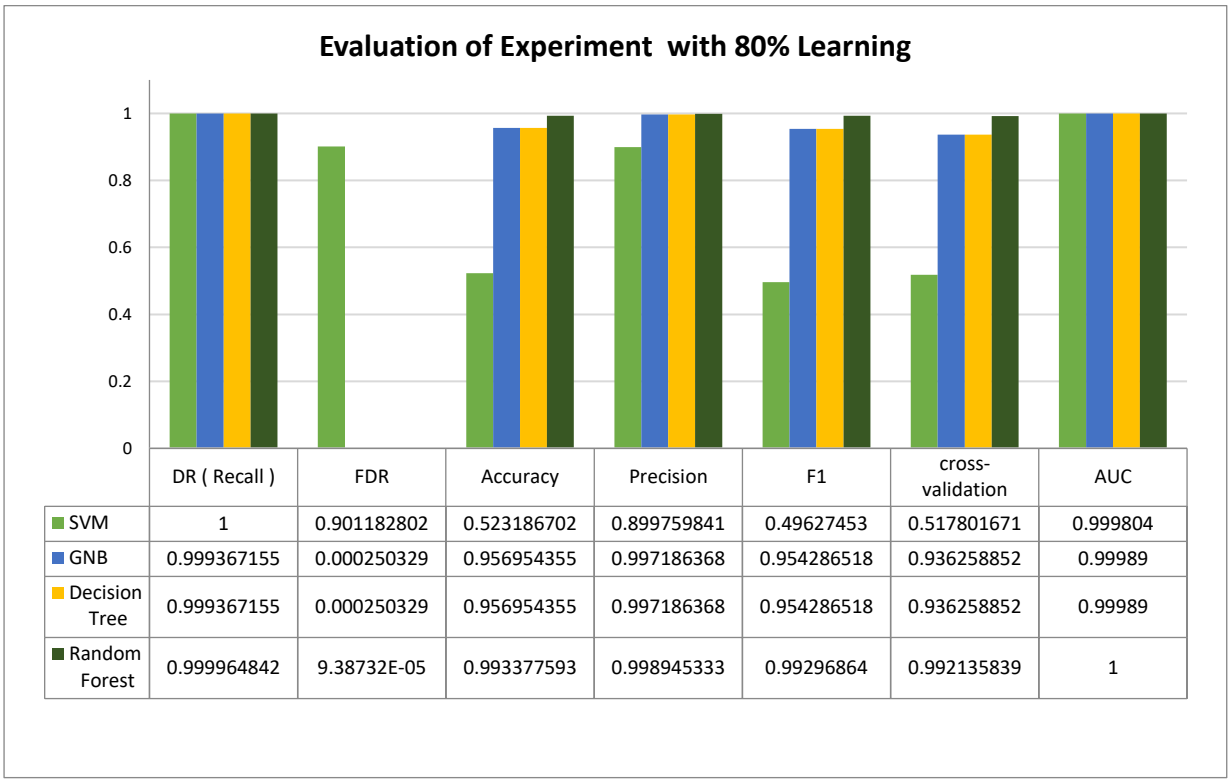


Figure 12. ROC and AUC with 80% Learning for the Experiment



	DR (Recall)	FDR	Accuracy	Precision	F1	cross-validation	AUC
SVM	1	0.901182802	0.523186702	0.899759841	0.49627453	0.517801671	0.999804
GNB	0.999367155	0.000250329	0.956954355	0.997186368	0.954286518	0.936258852	0.99989
Decision Tree	0.999367155	0.000250329	0.956954355	0.997186368	0.954286518	0.936258852	0.99989
Random Forest	0.999964842	9.38732E-05	0.993377593	0.998945333	0.99296864	0.992135839	1

Figure 13. MLA Evaluation comparison with 80% Learning for the Experiment

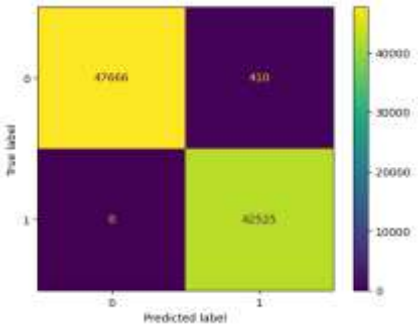
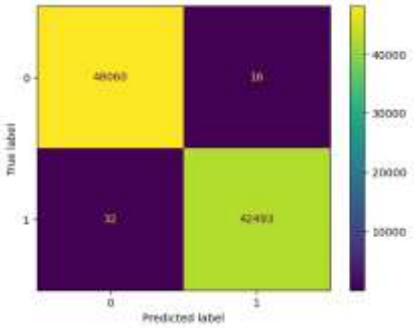
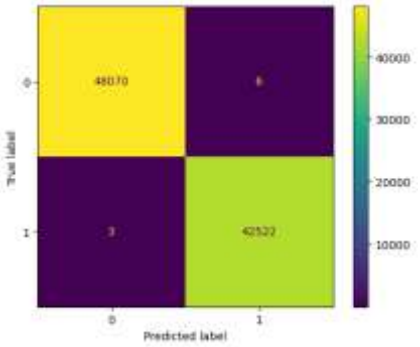
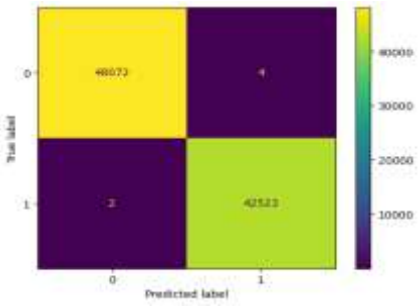
Referring to the values in the results shown in the Table 9. and Figure 15, which were used for ease of comparison and deduction of the best models, it was found. SVM has the highest detection rate (DR =1.0), the highest false detection rate (FDR = 0.90), but the lowest accuracy and weak overall performance in distinguishing between categories and unstable across different samples.

GNB and DT provide nearly identical distinct performance, characterized by high accuracy and low error rates, and appear to be stable in other metrics. RF demonstrating its predictive power and adaptability with the highest F1 score, precision, and AUC (1.0). With an 80% learning rate, it is the most effective and balanced model in this experiment.

Performance evaluation for learning rate 70%

The performance evaluation, using a 70% dataset size (211403, 5) for training and a 30% dataset size (90600, 5) for testing, is shown in the next section.

Table 10. Different ML algorithms with performance metrics, with 70% training for the Experiment

Algorithm	Confusion Matrix	Evaluation	
SVM		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	1.0 0.008528163740743822 0.9954746636350592 0.9904506812623733 0.9952024338871986 0.9950615388770409 0.999821
GNB		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9992475014697237 0.00033280638988268576 0.9994702045231288 0.9996236091180691 0.9994355199096832 0.9993755970019658 0.999864
Decision Tree		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9999294532627866 0.00012480239620600716 0.9999006633480867 0.9998589164785553 0.9998941836266798 0.9999101238846277 0.999946
Random Forest		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9999529688418577 8.320159747067144e-05 0.9999337755653911 0.999905942107367 0.9999294549216949 0.9998959326658421 1.0

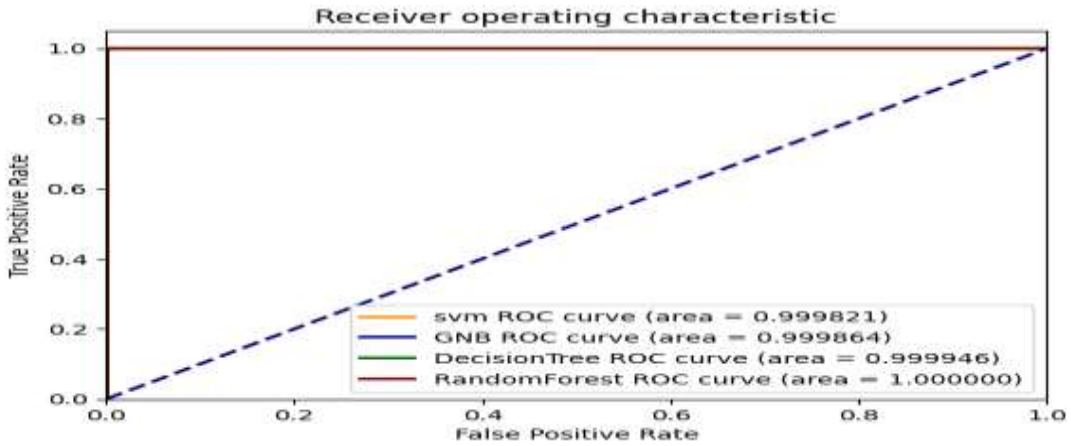
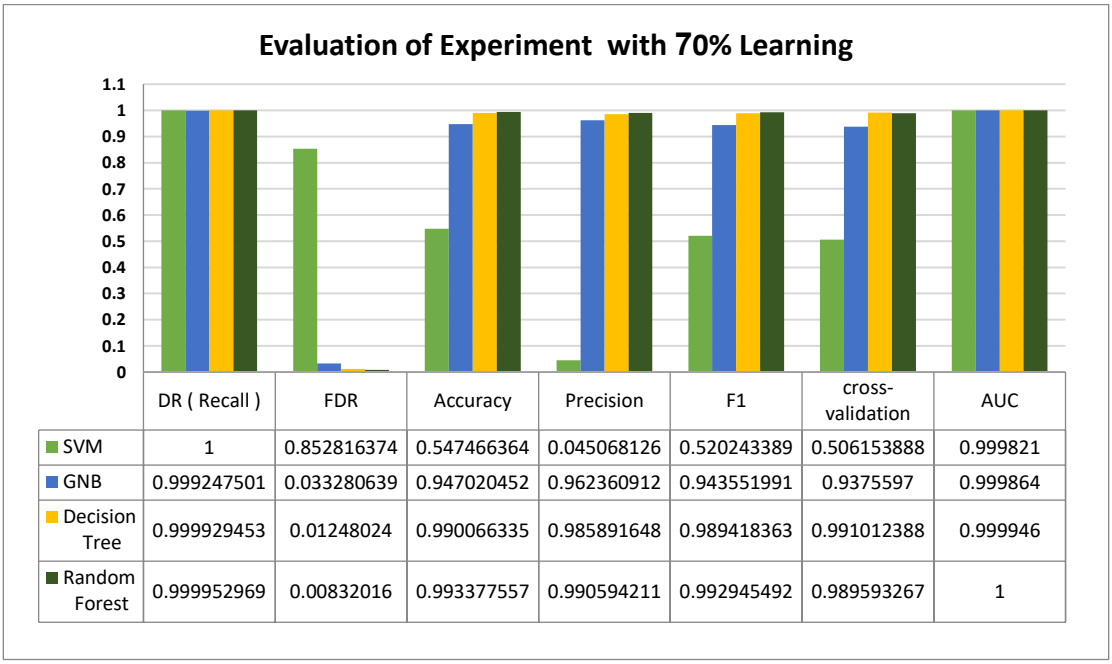


Figure 14.. ROC and AUC with 70% Learning for the Experiment



	DR (Recall)	FDR	Accuracy	Precision	F1	cross-validation	AUC
SVM	1	0.852816374	0.547466364	0.045068126	0.520243389	0.506153888	0.999821
GNB	0.999247501	0.033280639	0.947020452	0.962360912	0.943551991	0.9375597	0.999864
Decision Tree	0.999929453	0.01248024	0.990066335	0.985891648	0.989418363	0.991012388	0.999946
Random Forest	0.999952969	0.00832016	0.993377557	0.990594211	0.992945492	0.989593267	1

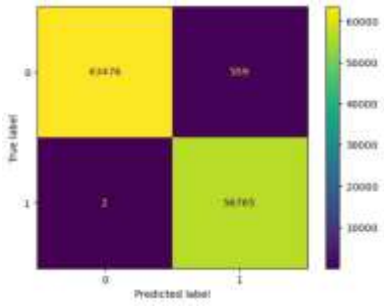
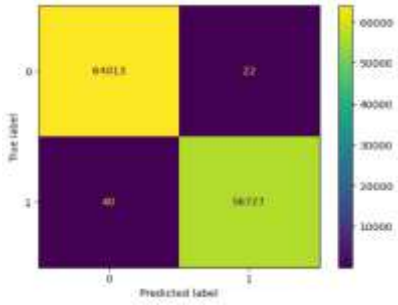
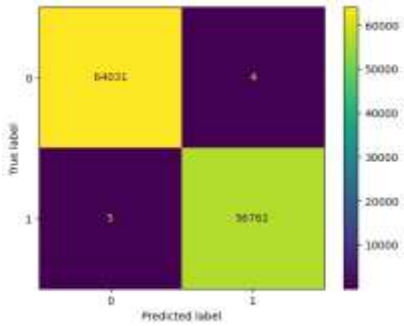
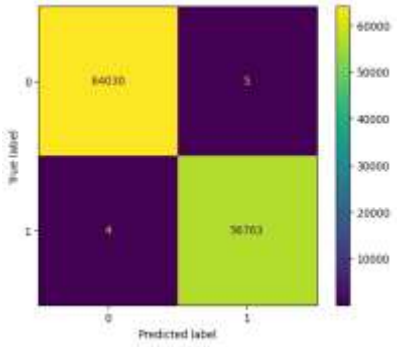
Figure 15. MLA Evaluation comparison with 70% Learning for the Experiment

Figure 16 and Figure 17 show that all models showed elevated values across most performance metrics. The Support Vector Machine (SVM) has the highest Detection Rate, but it is limited by the significant false detection rate which indicates reduced accuracy and suboptimal F1 score. Such high error rate makes it unsuitable to be used in practice. On the other hand, the Gaussian Naive Bayes (GNB) model gives good and realistic results, with a fine balance between retrieval and accuracy and thus making it a fairly dependable model. The Decision Tree is an excellent performer in all the measures; however, the model can easily overfit because the values of performance are so high. Random Forest model has been recognized as the best and most balanced among other alternatives because it is the highest accuracy, F1 score and lowest false discovery rate (FDR) with an optimal area under the curve (AUC). This model demonstrates high efficiency of attaining a balance of sensitivity, accuracy and generalizability.

Performance evaluation for learning rate 60%

In the next section, the low learning rate was used to test the generalization under the setting of the reduced training samples; 60 percent of the dataset (120801, 5) was adopted as the training samples, and 40 percent (181202, 5) as the testing samples.

Table 11. different ML algorithms with performance metrics, with 60% training, for the Experiment

ORITHM	CONFUSION MATRIX	EVALUATION	
SVM		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9999647682632515 0.008729600999453423 0.9953560371517027 0.9902484125322727 0.9950828724439263 0.9950276180314358 0.999832
GNB		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9992953652650307 0.00034356211446864995 0.9994867634641811 0.9996123279705369 0.9994538214877198 0.9993377529127578 0.999851
DECISION TREE		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9999119206581288 6.24658389943e-05 0.9999254979222199 0.9999295352852059 0.9999207278940925 0.9998896257900188 0.999974
RANDOM FOREST		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9999295365265031 7.808229874287499e-05 0.9999254979222199 0.9999119222096956 0.9999207292905271 0.9998730691940689 1.0

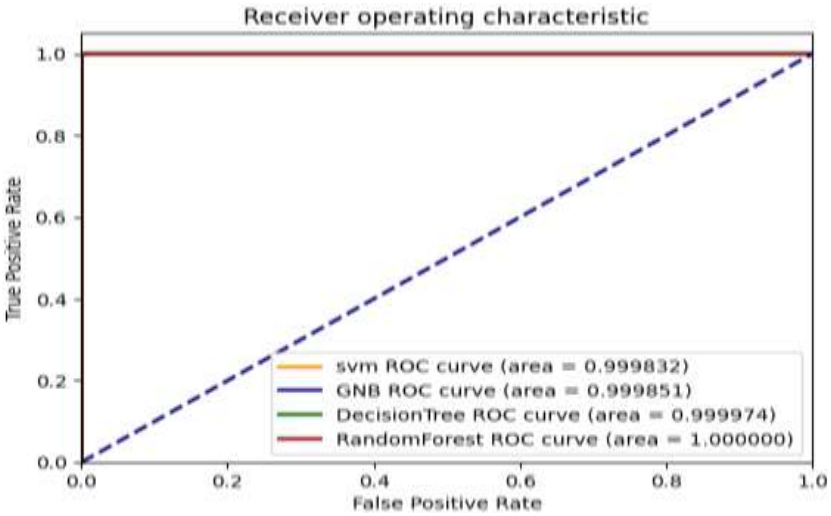
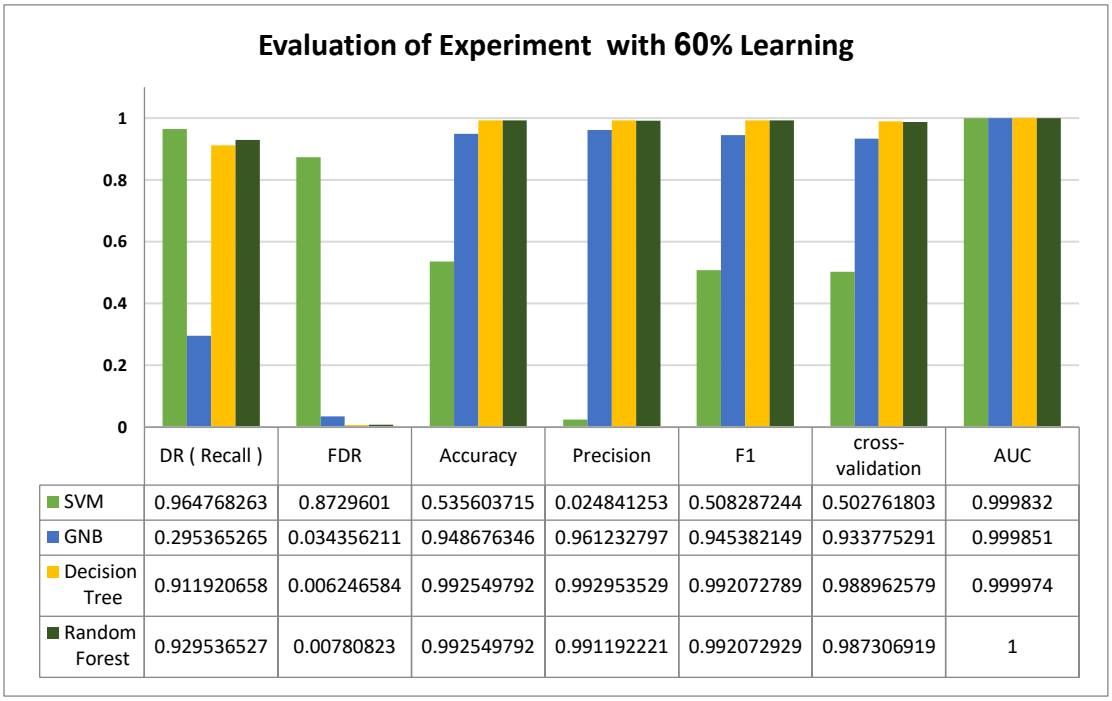


Figure 16. ROC and AUC with 60% Learning for the Experiment



	DR (Recall)	FDR	Accuracy	Precision	F1	cross-validation	AUC
SVM	0.964768263	0.8729601	0.535603715	0.024841253	0.508287244	0.502761803	0.999832
GNB	0.295365265	0.034356211	0.948676346	0.961232797	0.945382149	0.933775291	0.999851
Decision Tree	0.911920658	0.006246584	0.992549792	0.992953529	0.992072789	0.988962579	0.999974
Random Forest	0.929536527	0.00780823	0.992549792	0.991192221	0.992072929	0.987306919	1

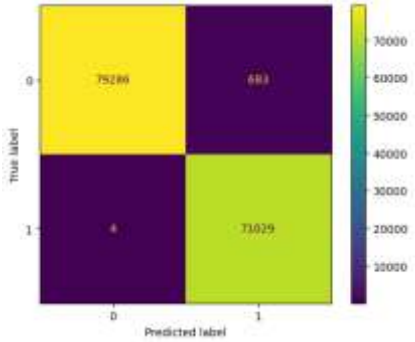
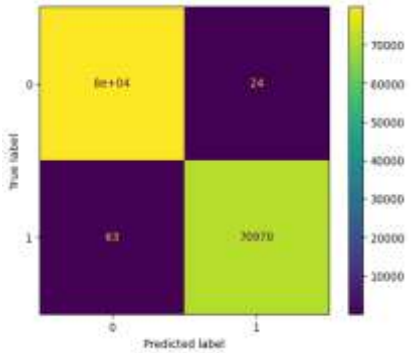
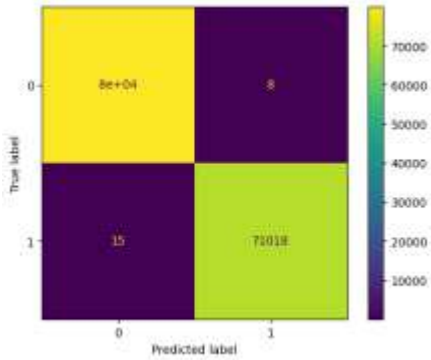
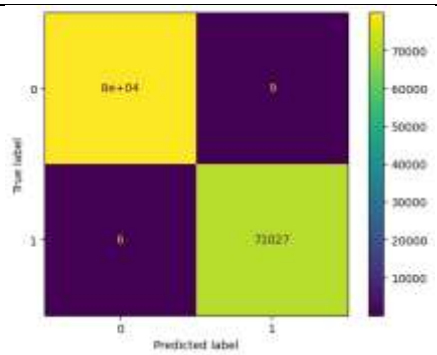
Figure 19. MLA Evaluation comparison with 60% Learning for the Experiment

Based on the results, which are shown in Figure 18 and Figure 19, the following is observed. SVM records the highest detection rate (DR = .096). but suffers from a high false discovery rate (FDR = 87.29%), resulting in low accuracy and a weak F1 score. Due to the high error rate, it becomes less reliable. GNB achieves excellent and realistic results, maintaining a precise balance between retrieval and precision, which makes it a relatively reliable model. Decision Tree offers excellent performance across all metrics; however, it may be prone to overfitting due to its exceptionally high-performance values. Random Forest performs best and is the most balanced in terms of sensitivity and precision, achieving the highest accuracy, F1 score, and low false discovery rate (FDR), along with a perfect area under the curve (AUC), making it the most effective.

Performance evaluation for learning rate 50%

Fifty percent of the data, specifically the value (151001, 5), and utilized to train the models for the utmost robustness evaluation.

Table 12. Different ML algorithms with performance metrics, with 50% training for the Experiment

ALGORITHM	CONFUSION MATRIX	EVALUATION	
SVM		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9999436881449467 0.008540809563705936 0.9954503913855446 0.9904757920571173 0.9951872219692458 0.9948874496721853 0.999866
GNB		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9991130882829108 0.00030011629506433745 0.9994238486907459 0.999661943262811 0.9993874404162589 0.9993907317660418 0.999841
DECISION TREE		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.9997888305435502 0.00010003876502144582 0.9998476841366339 0.999887365190212 0.9998380954392189 0.9998675505459996 0.999879
RANDOM FOREST		DR (Recall) FDR Accuracy Precision F1 cross-validation AUC	0.99991553221742 0.00011254361064912653 0.99990066356737 0.9998733036770089 0.999894417501355 0.9998807953598309 0.999969

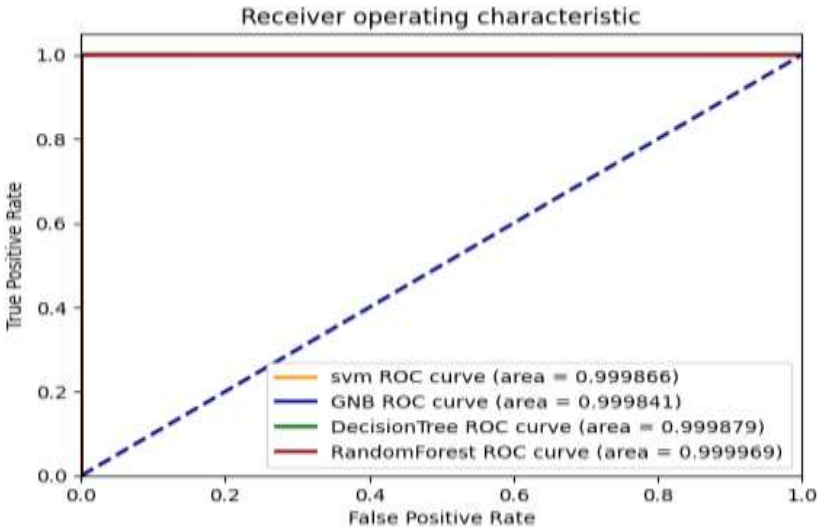
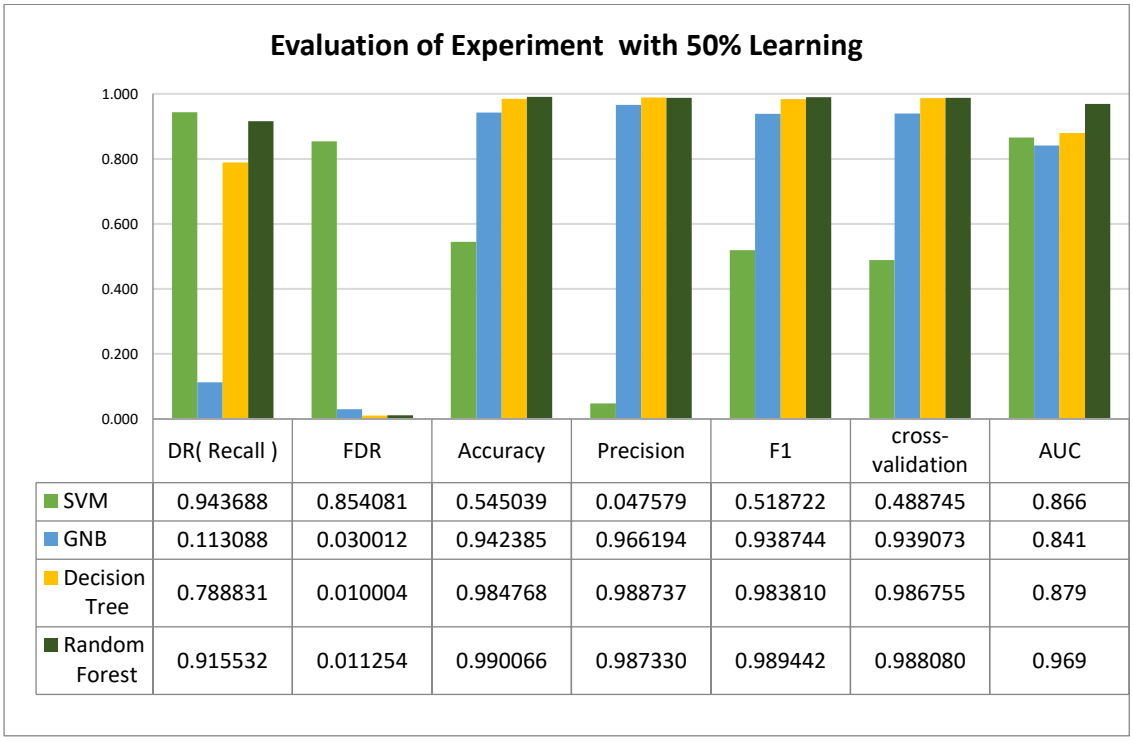


Figure 17. ROC and AUC with 50% Learning for the Experiment



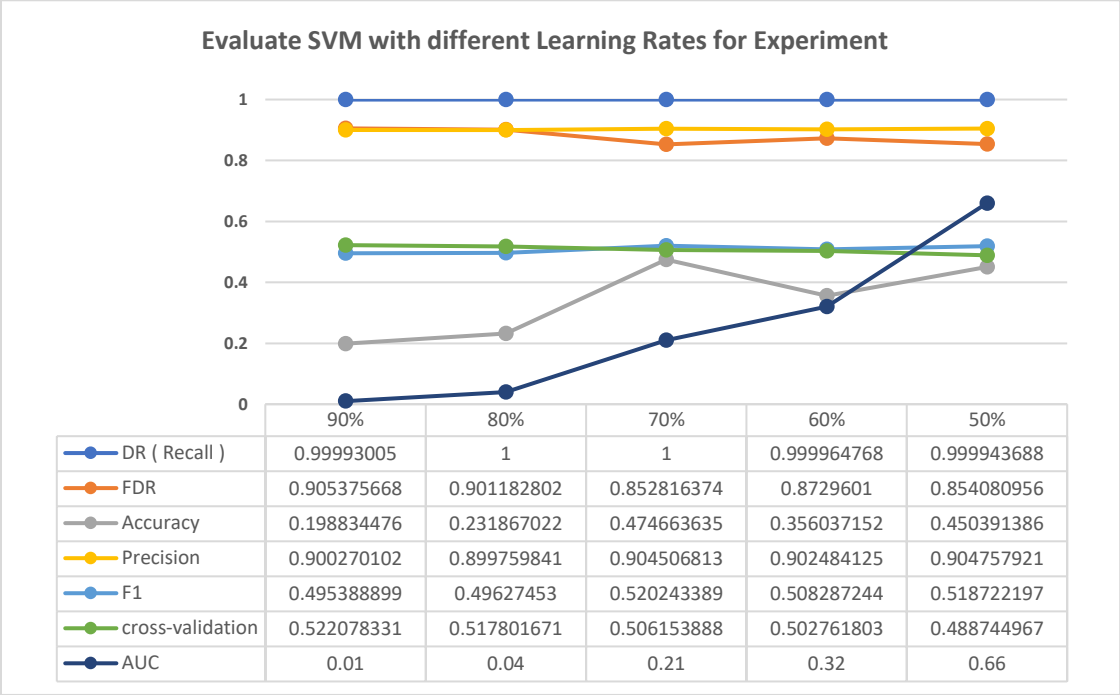
	DR(Recall)	FDR	Accuracy	Precision	F1	cross-validation	AUC
SVM	0.943688	0.854081	0.545039	0.047579	0.518722	0.488745	0.866
GNB	0.113088	0.030012	0.942385	0.966194	0.938744	0.939073	0.841
Decision Tree	0.788831	0.010004	0.984768	0.988737	0.983810	0.986755	0.879
Random Forest	0.915532	0.011254	0.990066	0.987330	0.989442	0.988080	0.969

Figure 18. MLA Evaluation comparison with 50% Learning for the Experiment

Figure 20 and Figure 21 show that the Random Forest model demonstrates outstanding accuracy, F1 score, and AUC, indicating a robust balance between precise detection and generalization capabilities. Consequently, it emerges as the most effective and balanced model, showcasing superior performance across all evaluative metrics. The Decision Tree exhibits commendable performance, as evidenced by the lowest False Detection Rate (FDR); however, it concurrently registers a comparatively low F1 score, which may indicate a tendency toward a specific class. The SVM achieves high recall; however, it encounters a significant decline in precision and a heightened False Detection Rate (FDR), thereby affecting its reliability. The GNB exhibits low classification precision alongside a high false detection rate; nevertheless, it performs admirably in cross-validation and relative accuracy. Thus, it is not regarded as the most suitable choice for this experiment.

Comparing the results of different training ratios in the Experiment.

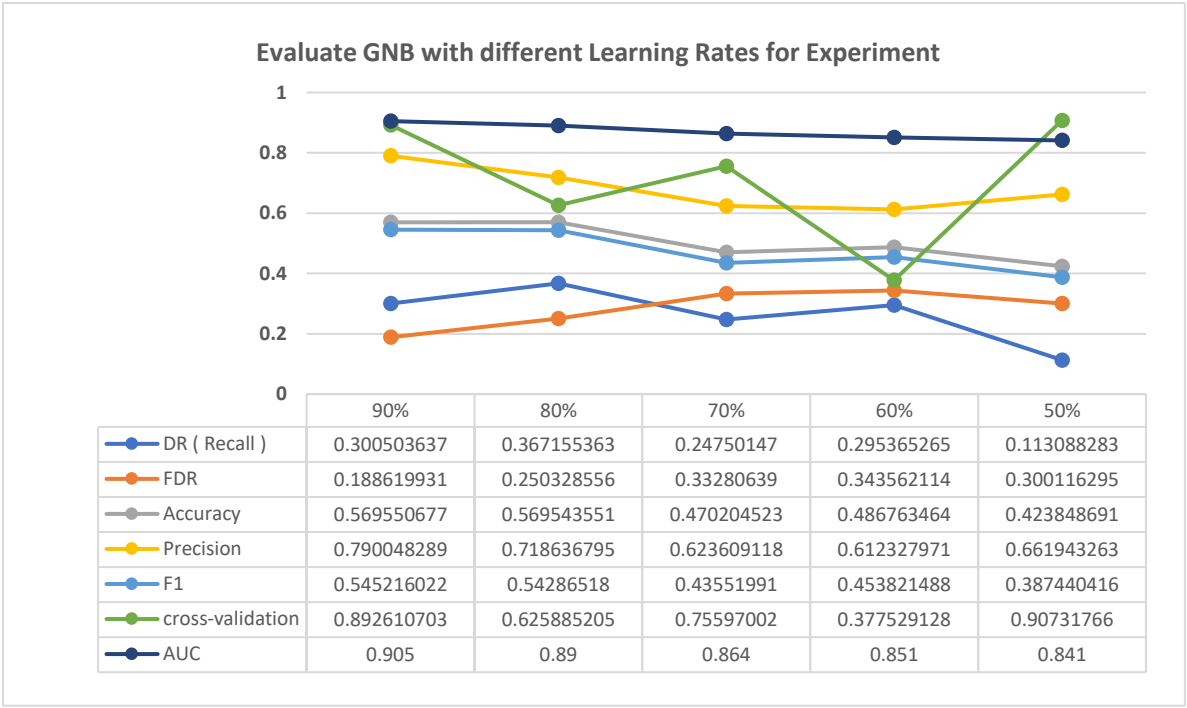
The model examined the impact of the learning rate on the performance of SVM, GNB, DT, and RF in the Experiment by comparing the evaluation results metrics.



	90%	80%	70%	60%	50%
DR (Recall)	0.99993005	1	1	0.999964768	0.999943688
FDR	0.905375668	0.901182802	0.852816374	0.8729601	0.854080956
Accuracy	0.198834476	0.231867022	0.474663635	0.356037152	0.450391386
Precision	0.900270102	0.899759841	0.904506813	0.902484125	0.904757921
F1	0.495388899	0.49627453	0.520243389	0.508287244	0.518722197
cross-validation	0.522078331	0.517801671	0.506153888	0.502761803	0.488744967
AUC	0.01	0.04	0.21	0.32	0.66

Figure 19. Evaluation of SVM with different Learning rates for the Experiment

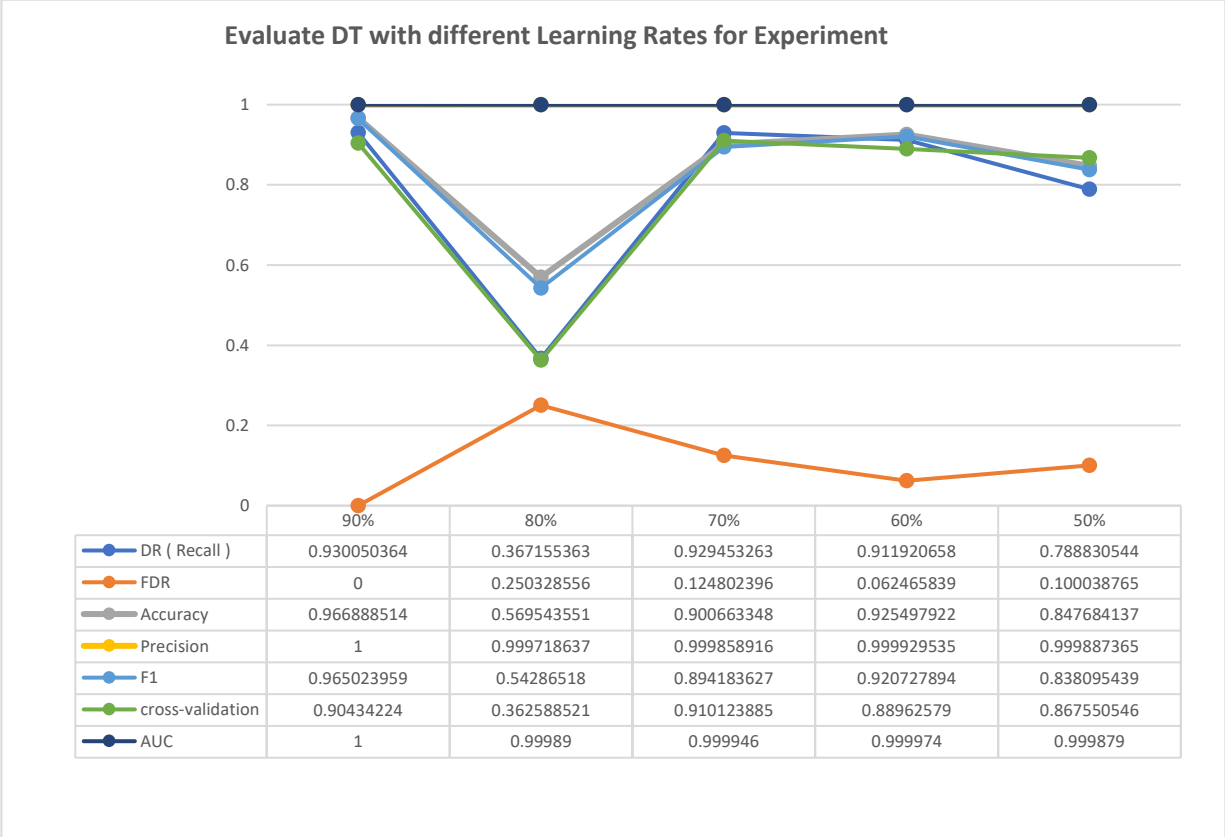
Figure 22 presents an analysis of the Support Vector Machine (SVM) model's performance at various learning rates. The results indicate that reducing the learning rate of the SVM model improved the Area Under the Curve (AUC), while maintaining the high Detection Rate (DR), but a decrease in the cross-validation score and a high value of the False Detection Rate (FDR) suggest instability in the model's performance when trained on different subsets of data, which makes it unsuitable for DDoS detection in SDN networks.



	90%	80%	70%	60%	50%
DR (Recall)	0.300503637	0.367155363	0.24750147	0.295365265	0.113088283
FDR	0.188619931	0.250328556	0.33280639	0.343562114	0.300116295
Accuracy	0.569550677	0.569543551	0.470204523	0.486763464	0.423848691
Precision	0.790048289	0.718636795	0.623609118	0.612327971	0.661943263
F1	0.545216022	0.54286518	0.43551991	0.453821488	0.387440416
cross-validation	0.892610703	0.625885205	0.75597002	0.377529128	0.90731766
AUC	0.905	0.89	0.864	0.851	0.841

Figure 20. Evaluation of GNB with different Learning rates for the Experiment

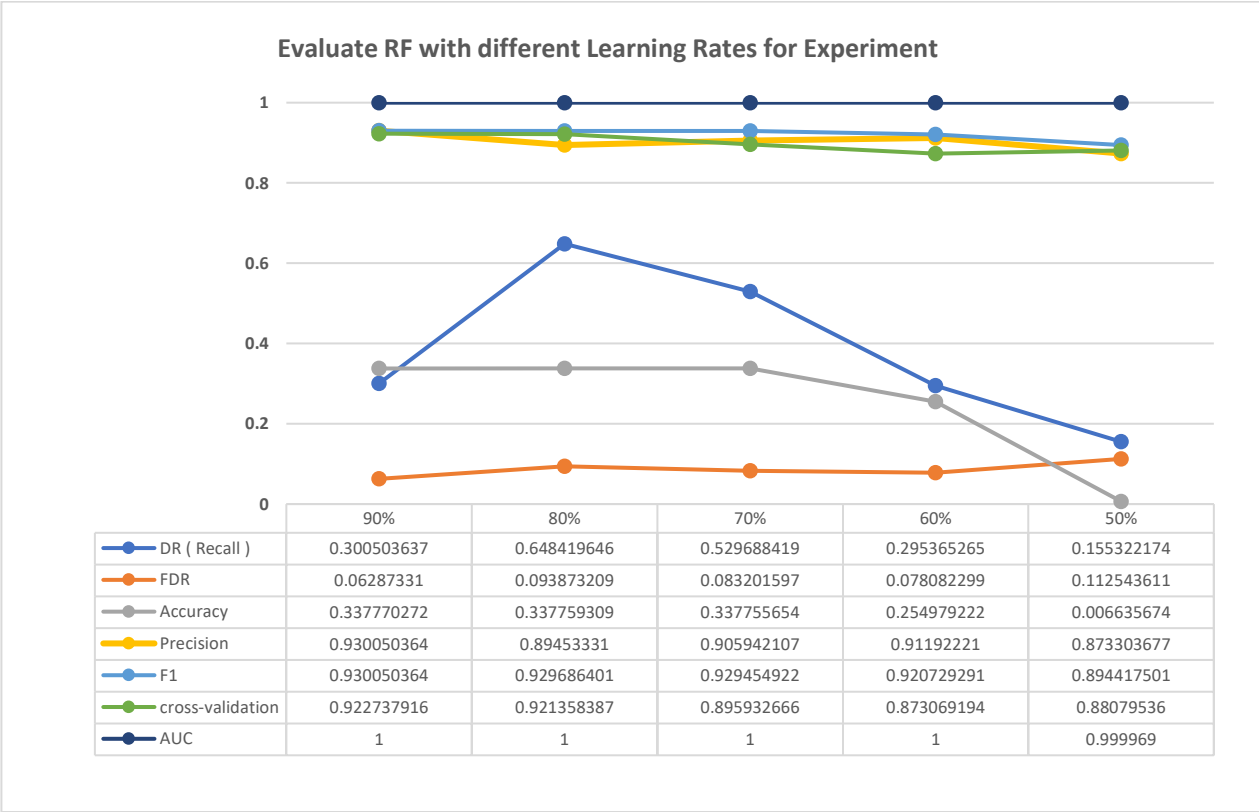
The analysis in Figure 23 shows that the performance of the Gaussian Naïve Bayes (GNB) model varies with different learning ratios. As the learning data decreases, the precision, F1 score, and accuracy become unstable, while the false discovery rate (FDR) increases. The model's performance remains less stable, with cross-validation values being significantly affected.



	90%	80%	70%	60%	50%
DR (Recall)	0.930050364	0.367155363	0.929453263	0.911920658	0.788830544
FDR	0	0.250328556	0.124802396	0.062465839	0.100038765
Accuracy	0.966888514	0.569543551	0.900663348	0.925497922	0.847684137
Precision	1	0.999718637	0.999858916	0.999929535	0.999887365
F1	0.965023959	0.54286518	0.894183627	0.920727894	0.838095439
cross-validation	0.90434224	0.362588521	0.910123885	0.88962579	0.867550546
AUC	1	0.99989	0.999946	0.999974	0.999879

Figure 21. Evaluation of DT with different Learning rates for the Experiment

According to the performance analysis of the Decision Tree model depicted in Figure 24, a consistent observation is made concerning the area under the curve (AUC) and precision across a range of varying learning rates. It is observed that cross-validation, accuracy, and the F1 score exhibit a decline as the volume of training data diminishes from 90 to 80, then they increase with decreasing learning from 80 to 70, then they approximately stabilize as the learning ratio decreases. FDR increases in value when the learning ratio changes from 90 to 80 and reaches its highest value when the learning ratio is 80%, then decreases with decreasing learning ratios. Although the Decision Tree model demonstrates commendable performance in terms of accuracy (Accuracy = 0.9669) and metrics such as F1 and AUC at a learning rate of 90%, the consistency of these metrics across alternative rates, including cross-validation and FDR, suggests that the model may be susceptible to overfitting.



	90%	80%	70%	60%	50%
DR (Recall)	0.300503637	0.648419646	0.529688419	0.295365265	0.155322174
FDR	0.06287331	0.093873209	0.083201597	0.078082299	0.112543611
Accuracy	0.337770272	0.337759309	0.337755654	0.254979222	0.006635674
Precision	0.930050364	0.89453331	0.905942107	0.91192221	0.873303677
F1	0.930050364	0.929686401	0.929454922	0.920729291	0.894417501
cross-validation	0.922737916	0.921358387	0.895932666	0.873069194	0.88079536
AUC	1	1	1	1	0.999969

Figure 22. Evaluation of RF with different Learning rates for the Experiment

Figure 25 illustrates the analysis of the Random Forest model's performance across various ranges of learning data. It demonstrates stability in AUC, cross-validation, F1 score, and precision. A decrease in the detection rate (DR) and accuracy is observed when the training rate is diminished. The optimal learning rate for the Random Forest model is identified as 80%, achieving the most favorable balance among recall, precision, accuracy, and error rate. The model with this learning rate has a strong performance, as well as admirable generalizability, and does not experience any problem with overfitting.

According to the above discussion on the impact of learning ratios with regard to the effectiveness of machine learning models, the best-performing algorithm between the three learning ratios is the Random Forest algorithm, which achieves high accuracy, discovery rate, F1 score, and cross-validation scores. The reduced false detection rate (FDR = 9.3e-05) indicates the strong generalization abilities of the model. It shows the best performance under a learning ratio of about 80 as this ratio generates the best balance between accuracy and generalization to give a stable performance and high capacity to handle new information.

Conclusion

The findings of this investigation provide substantial empirical substantiation elucidating the efficacy of machine learning methodologies in enhancing the security frameworks of software-defined networking ecosystems against distributed denial-of-service incursions. The Random Forest algorithm was identified as the most effective model for instantaneous detection, attaining remarkable performance metrics when subjected to an 80:20 data partitioning strategy, culminating in an accuracy of 99.99%. This elevated degree of precision signifies the model's adeptness at reliably differentiating between legitimate and malicious traffic patterns, even amidst intricate and fluctuating network conditions.

In addition, the paper explains how combination of well-designed feature engineering and effective machine learning models creates a dependable and scalable architecture to intrusion detection in SDN architectures. This combination enhances the defensive resilience of the network and leads to the development of adaptive detection systems that can maintain high performance in a variety of operational environments. Besides, the findings highlight the need to create models that demonstrate successful generalization in different traffic behavior and topological structures to provide uniform detection accuracy in a range of deployment conditions.

These findings add to the vast understanding of how machine learning is applied to the security issues, inherent in SDN, and emphasize the need to conduct further studies aimed at optimizing feature selection and evaluating

performance in realistic network environment settings, and examining hybrid mechanisms that can be used to combine machine learning with traditional security mechanisms.

References

1. Ahmed, S., Khan, Z. A., Mohsin, S. M., Latif, S., Aslam, S., Mujlid, H., Adil, M., & Najam, Z. (2023). Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron. Future Internet, 15(2), 76. https://doi.org/10.3390/fi15020076
2. Al Essa, H. A., & Bhaya, W. S. (2023). Detection of DDoS Attacks in Software-Defined Networks Based on Majority Voting-Average for Feature Selection and Machine Learning Approaches. In: 2023 Second International Conference on Advanced Computer Applications (ACA), IEEE, pp. 1–6. https://doi.org/10.1109/ACA57612.2023.10346864
3. Al-Fayoumi, M., & Abu Al-Haija, Q. (2023). Capturing low-rate DDoS attack based on MQTT protocol in software Defined-IoT environment. Array, 19, 100316. https://doi.org/10.1016/j.array.2023.100316
4. Ali, T. E., Chong, Y. W., & Manickam, S. (2023). Comparison of ML/DL Approaches for Detecting DDoS Attacks in SDN. Applied Sciences, 13(5), 3033. https://doi.org/10.3390/app13053033
5. Aliar, A. A. S., Gowri, V., & Abins, A. A. (2024). Detection of distributed denial of service attack using enhanced adaptive deep dilated ensemble with hybrid META-HEURISTIC approach. Transactions on Emerging Telecommunications Technologies, 35(3), e4921. https://doi.org/10.1002/ett.4921
6. Aliyu, A. L., Aneiba, A., Patwary, M., & Bull, P. (2020). A trust management framework for Software Defined Network (SDN) controller and network applications. Computer Networks, 181, 107421. https://doi.org/10.1016/j.comnet.2020.107421
7. Almaiah, M. A., Alrawashdeh, R., Alkhdour, T., Al-Ali, R., Rjoub, G., & Aldahyani, T. (2024). Detecting DDoS attacks using machine learning algorithms and feature selection methods. International Journal of Data and Network Science, 8(3), 2307–2318. https://doi.org/10.5267/j.ijdns.2024.6.001
8. Alnatsheh, A., Alsarhan, A., Aljaidi, M., Rafiq, H., Mansour, K., Samara, G., Igried, B., & Al Gumaei, Y. A. (2023). Machine Learning-Based Approach for Detecting DDoS Attack in SDN. In: 2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI), IEEE, pp. 1–5. https://doi.org/10.1109/EICEEAI60672.2023.10590313
9. Ashodia, N., & Makadiya, K. (2022). Detection of DDoS attacks in SDN using Machine Learning. In: 2022 International Conference on Electronics and Renewable Systems (ICEARS), IEEE, pp. 1322–1327. https://doi.org/10.1109/ICEARS53579.2022.9751879
10. Ashwin, A., Santhosh, V., Venkatesh, R. T., Gowthami, N., & Tamilselvi, S. (2024). Detection and Mitigation of DDoS Attack in SDN Using Feature Based SVM and Decision Tree Approach. In: 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS), IEEE, pp. 325–330. https://doi.org/10.1109/ICICNIS64247.2024.10823201
11. Babbar, H., Rani, S., & Driss, M. (2024). Effective DDoS attack detection in software-defined vehicular networks using statistical flow analysis and machine learning. PLOS ONE, 19(3), e0314695. https://doi.org/10.1371/journal.pone.0314695
12. Estupiñán Cuesta, E. P., Martínez Quintero, J. C., & Avilés Palma, J. D. (2025). DDoS Attacks Detection in SDN Through Network Traffic Feature Selection and Machine Learning Models. Telecom, 6(3), 69. https://doi.org/10.3390/telecom6030069
13. Facchini, H., Perez, S., Blanchet, R., Roberti, B., & Azcarate, R. (2021). Experimental performance contrast between SDN and traditional networks. In: 2021 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), IEEE, pp. 1–6. https://doi.org/10.1109/CHILECON54041.2021.9702982
14. Fakolujo, O., & Qureshi, A. (2023). Analysis of Detection Systems in a Software-Defined Network. In: Intelligent Computing, LNNS, Springer Nature, pp. 1342–1363. https://doi.org/10.1007/978-3-031-37963-5_91
15. Ferdiansyah, F., Antoni, D., Valdo, M., Mikko, M., Mukmin, C., & Ependi, U. (2024). Machine Learning Models for DDoS Detection in Software-Defined Networking: A Comparative Analysis. Journal of Information Systems and Informatics, 6(3), 1790–1803. https://doi.org/10.51519/journalisi.v6i3.864
16. Haeruddin, H., Erick, E., & Aripradono, H. W. (2025). Perbandingan Support Vector Machine, Random Forest Classifier, dan K-Nearest Neighbour dalam Pendeteksian Anomali pada Jaringan DDos. Jurnal Teknologi Informasi dan Multimedia (JTIM), 7(1), 23–33. https://doi.org/10.35746/jtim.v7i1.628
17. Hamarshe, A., Ashqar, H. I., & Hamarsheh, M. (2023). Detection of DDoS Attacks in Software Defined Networking Using Machine Learning Models. arXiv preprint. https://doi.org/10.48550/ARXIV.2303.06513

18. Hemanth Kumar, et al. (2023). Mitigation and Detection of DDOS attacks using Software Defined Network (SDN) and Machine Learning. International Journal for Research in Applied Science and Engineering Technology, 11(5), 4821–4829. https://doi.org/10.22214/ijraset.2023.47344
19. Islam, Md. T., Islam, N., & Refat, Md. A. (2020). Node to Node Performance Evaluation through RYU SDN Controller. Wireless Personal Communications, 112(1), 555–570. https://doi.org/10.1007/s11277-020-07060-4
20. Isyaku, B., Bakar, K. B. A., Ali, M. S., & Yusuf, M. N. (2023). Performance Comparison of Machine Learning Classifiers for DDOS Detection and Mitigation on Software Defined Networks. In: 2023 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), IEEE, pp. 69–74. https://doi.org/10.1109/I2CACIS57635.2023.10193601
21. Kadam, A. (2024). SDN-Driven Security Framework for DDOS Attack Detection and Mitigation. International Journal for Research in Applied Science and Engineering Technology, 12(3), 620–625. https://doi.org/10.22214/ijraset.2024.65142
22. Leqing, H. (2020). How to Realize the Smooth Transition From Traditional Network Architecture to SDN. In: 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), IEEE, pp. 1948–1952. https://doi.org/10.1109/ICMCCE51767.2020.00427
23. Liu, Z., Wang, Y., Feng, F., Liu, Y., Li, Z., & Shan, Y. (2023). A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks. Sensors, 23(13), 6176. https://doi.org/10.3390/s23136176
24. Mahar, I. A., Libing, W., Maher, Z. A., & Rahu, G. A. (2024). A Comprehensive Survey of Software Defined Networking and its Security Threats. In: 2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC), IEEE, pp. 1–5. https://doi.org/10.1109/KHI-HTC60760.2024.10482096
25. Mohsin, M. A., & Hamad, A. H. (2022). Performance Evaluation of SDN DDoS Attack Detection and Mitigation Based Random Forest and K-Nearest Neighbors Machine Learning Algorithms. Revue d'Intelligence Artificielle, 36(2), 233–240. https://doi.org/10.18280/ria.360207
26. Musa, N. S., Mirza, N. M., Rafique, S. H., Abdallah, A. M., & Murugan, T. (2024). Machine Learning and Deep Learning Techniques for Distributed Denial of Service Anomaly Detection in Software Defined Networks—Current Research Solutions. IEEE Access, 12, 17982–18011. https://doi.org/10.1109/ACCESS.2024.3360868
27. Patel, M., Amritha, P. P., Sudheer, V. B., & Sethumadhavan, M. (2024). DDoS Attack Detection Model using Machine Learning Algorithm in Next Generation Firewall. Procedia Computer Science, 233, 175–183. https://doi.org/10.1016/j.procs.2024.03.207
28. Pydala, B., Govardhan, D., Venkatesh, C., Goud, K. L., Dinesh, K., & Jyothsna, V. (2024). An Enhancing Comprehensive Machine Learning Framework for DDoS Defense Through Leveraging Multiple Algorithms. In: 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), IEEE, pp. 1–7. https://doi.org/10.1109/ICITEICS61368.2024.10625075
29. Rakissaga, W. A. O., Omar, H. H., & Kouraogo, P. J. (2025). Software Defined Networks: Strengths, Weaknesses, and Resilience to Failures. Engineering, 17(1), 19–29. https://doi.org/10.4236/eng.2025.171002
30. Rizaldi, M. I., Chandranegara, D. R., & Akbi, D. R. (2024). Comparison Of Machine Learning Techniques For Classification Of Distributed Denial Of Service Attacks Based On Feature Engineering In Sdn-Based Networks. Jurnal Ilmiah Penelitian dan Pembelajaran Informatika (JIPI), 9(3), 1180–1197. https://doi.org/10.29100/jipi.v9i3.5262
31. Roopesh, M., Nishat, N., Rasetti, S., & Rahaman, M. A. (2024). A Review Of Machine Learning And Feature Selection Techniques For Cybersecurity Attack Detection With A Focus On Ddos Attacks. Academic Journal of Science, Technology, Engineering, Mathematics and Education, 4(3), 178–194. https://doi.org/10.69593/ajsteme.v4i03.105
32. Sahosh, A. F., & Tasnim, S. A. (2024). A Comparative Review on DDoS Attack Detection Using Machine Learning Techniques. Malaysian Journal of Science and Advanced Technology, pp. 75–83. https://doi.org/10.56532/mjsat.v4i2.208
33. Setitra, M. A., Fan, M., Agbley, B. L. Y., & Bensalem, Z. E. A. (2023). Optimized MLP-CNN Model to Enhance Detecting DDoS Attacks in SDN Environment. Network, 3(4), 538–562. https://doi.org/10.3390/network3040024
34. Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019). Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST), IEEE, pp. 1–8. https://doi.org/10.1109/CCST.2019.8888419

35. Singh, A. (2021). Machine Learning in OpenFlow Network: Comparative Analysis of DDoS Detection Techniques. International Arab Journal of Information Technology, 18(2). https://doi.org/10.34028/iajit/18/2/11
36. Smida, K., Tounsi, H., Frikha, M., & Song, Y. Q. (2020). Efficient SDN Controller for Safety Applications in SDN-Based Vehicular Networks: POX, Floodlight, ONOS or OpenDaylight?. In: 2020 IEEE Eighth International Conference on Communications and Networking (ComNet), IEEE, pp. 1–6. https://doi.org/10.1109/ComNet47917.2020.9306095
37. Sumadi, A. (2020). Comparative Analysis of DDoS Detection Techniques Based on Machine Learning in OpenFlow Network. In: 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), IEEE, pp. 152–157. https://doi.org/10.1109/ISRITI51436.2020.9315510
38. Sumukh, S., & Sandhya, S. (2024). DDOS Detection Using ML and Deep Learning Approaches. In: 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), IEEE, pp. 1–6. https://doi.org/10.1109/CSITSS64042.2024.10817014
39. Thwaini, M. H. (2022). Anomaly Detection in Network Traffic using Machine Learning for Early Threat Detection. Data and Metadata, 1, 72. https://doi.org/10.56294/dm202272
40. Wabi, A. A., Idris, I., Olaniyi, O. M., Ojeniyi, J. A., & Adebayo, O. S. (2024). Performance Evaluation of Machine Learning Approaches for Classification of Ddos Attacks in Software Defined. Research Square (Preprint). https://doi.org/10.21203/rs.3.rs-4324004/v1
41. Wang, S., 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, 35, 101176. https://doi.org/10.1016/j.jestch.2022.101176
42. Yao, R. A., & Guinko, F. T. (2025). Optimizing DDoS attack detection in SDN using machine learning. In: 2025 5th Intelligent Cybersecurity Conference (ICSC), IEEE, pp. 341–348. https://doi.org/10.1109/ICSC65596.2025.11139942