IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 September 2024, accepted 15 October 2024, date of publication 24 October 2024,
date of current version 19 November 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3486034*

RESEARCH ARTICLE

# Detecting DDoS Threats Using Supervised Machine Learning for Traffic Classification in Software Defined Networking

**ABDINASIR HIRSI[1], (Graduate Student Member, IEEE), LUKMAN AUDAH[1,2], (Member, IEEE), ADEB SALH[3], (Member, IEEE), MOHAMMED A. ALHARTOMI[4], (Member, IEEE), AND SALMAN AHMED[5], (Graduate Student Member, IEEE)**

[1]Advanced Telecommunication Research Center (ATRC), Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja 86400, Malaysia
[2]Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja 86400, Malaysia
[3]Faculty of Information and Communication Technology, University Tunku Abdul Rahman (UTAR), Kampar 31900, Malaysia
[4]Department of Electrical Engineering, University of Tabuk, Tabuk 71491, Saudi Arabia
[5]VLSI and Embedded Technology (VEST) Focus Group, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja 86400, Malaysia

Corresponding authors: Lukman Audah (hanif@uthm.edu.my) and Adeb Salh (adebali@utar.edu.my)

This work was supported by the Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme under Grant FRGS/1/2022/TK07/UTHM/02/25.

**ABSTRACT** Software-Defined Networking (SDN) is a promising solution for large-scale network management that offers extensive opportunities for optimization. However, the centralized control inherent in SDN also exposes networks to security threats, notably Distributed Denial of Service (DDoS) attacks. To address these challenges, machine learning (ML) techniques have emerged as potent tools for anomaly detection and mitigation. This paper proposes a novel approach for traffic classification within SDN environments that distinguishes between benign and malicious traffic using supervised ML techniques. This study introduces a unique dataset tailored for DDoS attack detection, overcoming the limitations of existing datasets, such as unrealistic topologies and lack of public availability. Benchmarking against the CICDDoS2019 dataset validated the efficacy and relevance of the custom dataset. This research has significant implications for real-world applications, offering improved capabilities for detecting and mitigating DDoS attacks in SDN infrastructure. Experimental results demonstrated the effectiveness of the proposed random forest model, achieving a remarkable accuracy of 98.97% and a minimal False Alarm Rate (FAR) of 0.023. These findings underscore the potential of ML-based approaches in enhancing network security and resilience against DDoS attacks in SDN environments, paving the way for future advancements in network-defense strategies.

**INDEX TERMS** Anomaly detection, artificial intelligence, distributed denial of service (DDoS) attacks, machine learning, software-defined networking (SDN), supervised learning, traffic classification.

## I. INTRODUCTION

Software-Defined Networking (SDN) has emerged as a novel technology architecture that offers potential solutions to the ever-evolving challenges faced by industries in managing and optimizing the network infrastructure [1]. Moreover, with its centralized control and programmable features, SDN has garnered significant attention because of its ability to enhance network flexibility, scalability, and efficiency [2], [3]. The SDN architecture comprises three distinct layers: application, control, and data. Figure 1 illustrates this layered approach, which facilitates the decoupling of the traditionally tightly

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

integrated data, control, and application planes. In SDN, the data plane responsible for forwarding network traffic is decoupled from the control plane, which determines how traffic should be handled. This decoupling enables greater flexibility and agility in the network management. Furthermore, at the application layer, network administrators can exert control over the entire network by defining the rules and policies governing traffic behavior and resource allocation. This centralized control allows efficient network management and optimization. Open-source platforms, such as OpenFlow and OpenDaylight, have played crucial roles in SDN adoption by providing interoperable solutions that do not rely on any specific vendor [4], [5]. This fosters innovation and flexibility in network deployment.

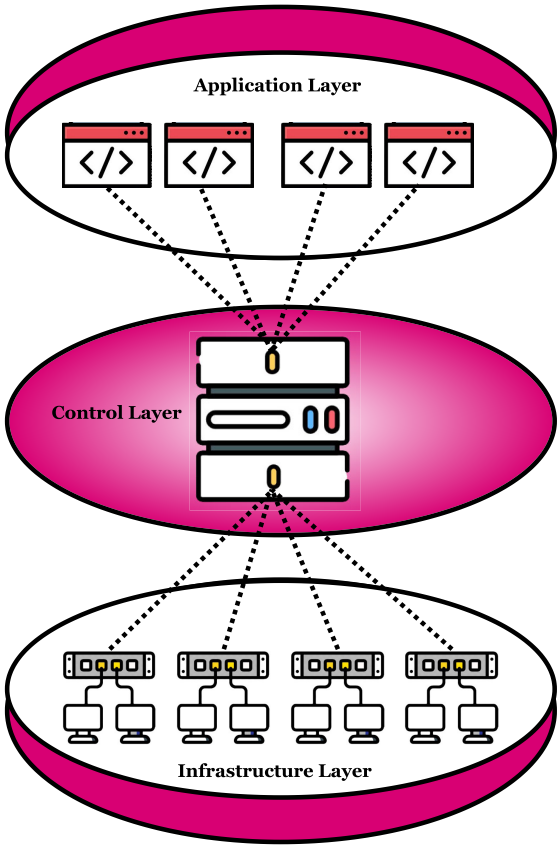


**FIGURE 1.** Software defined network architecture.

However, the centralized nature of SDN introduces significant security challenges, particularly the risk of Distributed Denial of Service (DDoS) attacks. A single controller governing the entire network can become a single point of failure, thereby exposing the network to security threats [6]. DDoS attacks, characterized by their distributed nature and orchestration from multiple sources, can overwhelm a target network or service, leading to severe disruptions in availability and performance [7]. As illustrated in Figure 2, centralized control in SDN magnifies the impact of these attacks because a compromised controller can lead to network-wide failures. This makes DDoS attacks a critical threat to SDN environments.

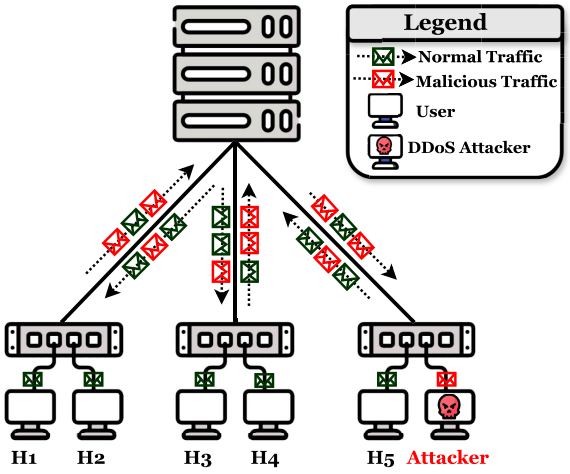


**FIGURE 2.** DDoS attack scenario.

Traditional methods of detecting and mitigating DDoS attacks in SDN are often inadequate owing to the unique characteristics of these networks. Techniques such as IP-address-based filtering struggle with tactics such as IP spoofing, making it difficult to distinguish between legitimate and malicious traffic [8]. In SDN environments, DDoS attacks targeting the data plane can cause congestion and service degradation [9], whereas attacks targeting the control plane can have even more severe consequences, disrupting the entire network operation by compromising the centralized controller [10]. 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 [11].

The detection of DDoS attacks in SDN remains a critical challenge owing to the complexity and dynamic nature of the network traffic. Traditional detection methods often struggle to identify attacks in their early stages, leaving networks vulnerable to prolonged disruptions and potential data breaches [12]. This has led to increased interest in the integration of Machine Learning (ML) techniques to enhance DDoS detection capabilities. ML algorithms can analyze vast amounts of network traffic data in real time, enabling the early detection of anomalous patterns indicative of DDoS attacks [13]. By learning from historical attack data, ML models can recognize subtle deviations from normal network behavior and trigger alerts before an attack fully manifests itself.

However, applying ML techniques to SDN environments presents specific challenges that must be addressed to ensure their effectiveness. One of the primary challenges is the requirement for real-time traffic classification in highly dynamic network environments [14]. SDNs generate a large

volume of diverse traffic patterns, making it difficult to differentiate between benign and malicious traffic in real-time. As a result, this complexity can lead to high false-positive rates, which degrade the network performance by misclassifying legitimate traffic as malicious [15]. In addition, the effectiveness of ML-based DDoS detection depends on the availability of high-quality training data. Existing datasets often suffer from limitations, such as outdated network topologies, unrealistic attack scenarios, and a lack of public availability, which undermines the generalization capabilities of ML models trained on them [16], [17].

To address these challenges, this study proposes a novel approach for improving traffic classification in SDN environments using supervised ML techniques. By focusing on realistic network configurations and contemporary DDoS attack scenarios, this study aims to overcome the limitations of existing datasets and enhance the detection capabilities of ML models for distinguishing between benign and malicious traffic [18]. Furthermore, the approach has been validated against well-established datasets such as CICDDoS2019, demonstrating its potential for improving the security and reliability of SDN-based network infrastructures.

### A. LITERATURE CONTEXT AND RESEARCH CONTRIBUTIONS

This section provides a critical review of the existing literature on DDoS attacks on SDN using ML techniques. Several studies have explored the application of ML to detect and mitigate DDoS attacks in SDN environments. However, a significant gap identified in these studies is the lack of attention paid to the selection of significant features within the dataset. Despite the proliferation of research in this area, existing studies have often overlooked the importance of feature selection, which is crucial for the effectiveness and efficiency of ML models for DDoS detection and mitigation. Furthermore, while numerous researchers have generated datasets for DDoS attack analysis in SDN, these datasets often lack detailed characteristics and are not publicly available. This lack of transparency and accessibility hinders the reproducibility and comparability of the research findings, thus limiting the advancement of knowledge in the field. Moreover, many studies failed to benchmark their datasets against existing public datasets or establish performance metrics, thereby limiting the evaluation and validation of their proposed methods. The absence of benchmarking not only undermines the credibility of research but also inhibits progress in the field by hindering meaningful comparisons and advancements. Overall, our research contributions stand out from previous studies in several respects.

Sahoo et al. [19] The dataset referred to as ''Dataset-I'' in this study, comprising 2,160,668 records with 27 features, is described as a modern DDoS dataset utilized for training and testing purposes. In contrast, Dataset-II, known as NSL-KDD, is explicitly mentioned as a refined version of the KDD'99 dataset. It is noteworthy that Dataset-I is not publicly available because its source or generation method is not explicitly stated in this study. This lack of accessibility to the dataset underscores the importance of transparency and reproducibility in research, highlighting the need for publicly available datasets to facilitate validation and comparison of results in the field of DDoS attack detection.

The study conducted by Kujur and Patel [20] was inadequate for providing detailed information regarding the characteristics and composition of the dataset. Without a thorough understanding of the properties of a dataset, it is difficult to determine its suitability and representativeness in assessing models for detecting DDoS attacks in SDN environments. Moreover, the study failed to mention any validation or verification processes for the Mendeley dataset. The absence of proper validation procedures raises concerns regarding the reliability and reproducibility of experimental findings.

Isyaku et al. [21] conducted a study comparing the effectiveness of eight ML classifiers in detecting and mitigating DDoS attacks in SDNs. They used a generated dataset, but the study lacked transparency regarding dataset sourcing and availability. This raises concerns regarding scalability and representativeness of the results. External validation of real-world datasets is crucial for enhancing the credibility and impact of the research. The study also evaluated ML classifiers based on the prediction time, learning time, and accuracy. However, it is important to consider the precision, recall, and F1-score, which offer a more comprehensive understanding of classifier performance, particularly for imbalanced datasets. Our study can improve this by incorporating additional evaluation metrics to provide a more comprehensive assessment of classifier performance, thus enhancing the overall quality of the research.

Rahman et al. [22] created an online dataset by generating normal and DDoS packets using the hping3 program in Python. They captured DDoS and normal traffic separately using Tshark to avoid confusion when labelling the dataset. DDoS traffic was flooded at a rate of 78 packets/s and captured for 30 min, whereas normal traffic was captured for 3 h to balance the dataset. Three supervised learning models, namely, Support SVM, Naive Bayes (NB), and nearest centroid (NC), were applied to classify network traffic based on applications in an SDN platform. This study lacks transparency regarding the dataset used for training and testing ML models, failing to specify its source, characteristics, and representativeness. In addition, while utilizing 24 packet-level features for DDoS detection, they do not provide a comprehensive list or description of these features, which hinders the evaluation of their effectiveness. Furthermore, the performance evaluation of the models lacks depth, as it only reports basic metrics without analyzing the model performance under different conditions or comparing it with existing approaches. Finally, the novelty of this work is diminished by the lack of comprehensive experimentation, benchmarking, or significant innovations in methodology, failing to advance state-of-the-art SDN security.

Raikar et al. [23] discussed accurate traffic classification in SDN using supervised learning techniques. This study highlights the limitations of traditional traffic classification techniques and the need for more accurate and scalable solutions. This manuscript lacks detailed information about the dataset used for training and testing supervised learning models. Without sufficient details regarding the dataset, including its size, diversity, and representativeness, it is challenging to assess the generalizability of the proposed models. Furthermore, analysis of evaluation metrics and comparisons with existing methods are lacking. The discussion of these challenges is incomplete, reducing the study's practical relevance. Finally, the study could benefit from improved organization and clarity to enhance understanding and coherence and explore the model's applicability in various network scenarios.

Ashodia and Makadiya [24] discussed the RF and DT algorithms to provide better accuracy and decision rates compared to other ML algorithms for detecting malicious traffic in SDN environments. This study does not discuss the validation strategies or techniques used to assess the quality, authenticity, and representativeness of the dataset. In addition, the dataset may lack diversity in terms of network topologies, traffic characteristics, and attack scenarios, potentially limiting its utility for a comprehensive performance evaluation and benchmarking of detection algorithms. In their study, the authors failed to disclose the provenance of the dataset employed, thereby compromising transparency and impeding their ability to replicate and validate their results. This lack of clarity regarding the origin of the dataset raises serious questions about its trustworthiness and appropriateness for the rigorous evaluation of DDoS detection methods.

Studies, such as those referenced in [25] and [26], which rely solely on individual datasets, such as CICDDoS2019 [25] and CIC DoS [26], run the risk of limiting the generalizability of their findings. By exclusively using a single dataset, these studies may overlook the broader landscape of cybersecurity threats and fail to capture the full spectrum of real-world scenarios, potentially affecting the applicability of their results. Without testing multiple datasets with varying characteristics, the robustness and applicability of the proposed system in different contexts remain uncertain.

Although Sudar et al. [27] achieved notable progress in improving DDoS attack detection in network security through the utilization of ML models, their reliance on the KDD99 dataset raises concerns regarding the comprehensiveness and robustness of their methodology. The absence of benchmarking against other widely recognized datasets commonly employed in intrusion detection systems hinders the ability to conduct a comparative analysis and assess the relative strengths and weaknesses of the proposed ML models. Consequently, the conclusions drawn from this study may be compromised in terms of their credibility and applicability.

Our study introduces a novel dataset specifically designed for DDoS attack classification that is publicly available for research purposes. In section IV can be found detailed descriptions of the features of the dataset and the network topologies used in its generation. The novelty of our work lies in the integration of three distinct modules.

- The first module focuses on creating a dataset, ensuring its comprehensiveness and relevance to DDoS attack scenarios in SDN environments.
- In the second module, we employed five machine learning models for DDoS attack classification. This approach allows a comprehensive evaluation of the effectiveness of the dataset for training and testing various classification algorithms.
- The third module benchmarks our dataset against other publicly available datasets, with a particular focus on comparing its performance with that of the widely used CICDDoS2019 dataset. This comparative analysis provides valuable insights into the strengths and limitations of our dataset and its potential utility in real-world applications.

The novelty of this study can be summarized as follows:

- **Novel Custom Dataset Creation:** One of the primary contributions of this study was the development of a novel custom dataset using the Mininet emulator. By leveraging this emulator, we were able to simulate realistic network environments and generate data that closely reflected real-world scenarios. This custom dataset fills a crucial gap in the literature by providing researchers with a new resource for evaluating network security algorithms and methodologies.
- **Evaluation of machine-learning algorithms:** We conducted a comprehensive evaluation of five supervised ML algorithms; logistic regression, support vector machine, random forest, K-nearest neighbor (KNN), and XGBooston our custom dataset. Through rigorous experimentation and analysis, we assessed the performance of these algorithms for detecting and classifying network intrusions and anomalies. Our findings shed light on the strengths and limitations of each algorithm in the context of our dataset, providing valuable insights for researchers and practitioners.
- **Benchmarking Against Public Datasets:** In addition to evaluating our custom dataset, we compared the performance of the machine learning algorithms on our dataset with that of CICDDoS2019, a widely used public dataset. This comparative analysis allowed us to contextualize the effectiveness of our custom dataset and algorithms relative to the established benchmarks. By identifying areas of improvement and potential research directions, our study contributes to the ongoing advancement of network security research and practices.
- **Implications for Real-World Applications:** Our research findings have practical implications for real-world network security applications. By identifying the most effective machine-learning algorithms for detecting network intrusions and anomalies,

organizations can enhance their cybersecurity defence and mitigate potential threats more effectively. Moreover, the availability of our custom dataset enabled researchers and practitioners to conduct further studies and develop innovative solutions tailored to specific network environments and requirements.

To the best of our knowledge, previous studies have not comprehensively integrated feature selection, dataset transparency, and benchmarking against established datasets in a single study that focuses on DDoS attack classification.

To address these objectives, The remainder of this paper is organized as follows. Section II reviews related research. In Section III, we explore the machine learning models selected for SDN threat detection. Section IV outlines the dataset creation process using Mininet emulator. Section V presents the proposed framework for DDoS detection in SDN using machine learning classification. The experimental setup details are provided in Section VI, followed by the results and discussion in Section VII. Finally, Section VIII concludes the paper.

## II. RELATED WORK

In this section, we review the existing literature and research relevant to our study. We surveyed the field, identified key findings, and discussed how previous works have contributed to our understanding of this topic. Specifically, we analyzed studies on DDoS attack detection and classification using ML models for traffic analyses.

Sahoo et al. [19] proposed an evolutionary SVM model for DDoS attack detection in SDNs, focusing on enhancing accuracy, reducing training time, and addressing the challenges posed by DDoS attacks in SDN environments. In this study, ML techniques were utilized for DDoS attack detection in SDNs. Specifically, it employs an SVM as the prime classifier for predicting malicious traffic. SVM is a supervised machine learning algorithm used for classification and regression tasks. Additionally, the study incorporated kernel principal component analysis (KPCA) with a genetic algorithm (GA) to enhance the performance of the SVM model. KPCA is a dimensionality reduction technique, whereas GA is a metaheuristic optimization algorithm inspired by the process of natural selection. Therefore, the ML techniques used in this study were primarily SVM for classification and KPCA with GA for feature extraction and optimization, respectively. The accuracy of the N-KPCA + GA + SVM model was reported to be 98.907%, outperforming other models such as hybrid models and single models such as SVM, KNN, and Random Forest classifiers. However, Dataset I, which was used for the evaluation, is not publicly available, limiting the reproducibility of the study's results.

Jawahar et al. [25] propose a novel system for real-time detection and mitigation of DDoS attacks using ML algorithms and blockchain technology. This system aims to enhance the security measures beyond traditional DDoS mitigation systems by leveraging advanced techniques. The simulation results show that the artificial neural network (ANN) model outperforms other ML algorithms with an accuracy score of 98.57%. The integration of blockchain technology enhances security and decentralization, providing transparency and immutability to maintain a blocklist of malicious IP addresses. Furthermore, the Mininet tool for building virtual networks was used for testing. A Python-based Open-Source and OpenFlow (POX) controller was used in conjunction with Mininet for traffic management. A CIC Flow Meter was employed to extract essential information from the incoming packets, generating 84 network traffic characteristics. The CICDDoS2019 dataset was used to train and test the classification ML techniques. In overall, this study lacks an evaluation of the performance of the proposed system in real-world scenarios. Additionally, the study does not benchmark the proposed approach against other state-of-the-art methods, which limits the generalizability of the results.

Perez-Diaz et al. [26] introduces a modular architecture for detecting and mitigating low-rate DDoS attacks in SDN using six ML techniques. These models include the J48, RT, REP tree, RF, multilayer perceptron (MLP), and SVM. This study emphasizes the development of a flexible and modular architecture that is capable of detecting and mitigating various DDoS attack scenarios. However, a key limitation of this work is the reliance on a single dataset, the Canadian Institute of Cybersecurity (CIC) DoS dataset, which may not fully capture the diversity of real-world DDoS attack patterns. The absence of benchmarking against other datasets limits the generalizability of the results. Additionally, while the proposed architecture is modular, the study lacks an in-depth analysis of the potential overhead introduced by the modular components, particularly in large-scale, real-world environments. The models were trained and evaluated using the CIC DoS dataset, achieving a detection rate of 95%. The architecture is designed to be flexible and modular, allowing for easy replacement or enhancement of components without affecting the overall system. The architecture is deployed in a simulated environment to mimic real-world SDN settings. They used an open network operating system (ONOS) controller running on a Mininet virtual machine to create a realistic testing environment. Furthermore, the reliance on a simulated environment means that the performance of the architecture in live, high-traffic networks remains untested.

Sudar et al. [27] have significantly enhanced the detection of DDoS attacks in network security through the identification and mitigation of malicious traffic using ML models. This study underscores the critical threat that DDoS attacks pose to SDNs owing to their capacity to overwhelm network resources. The study employed two prominent ML models: DT, which is utilized for the classification and differentiation of normal and malicious traffic, and SVM, which is employed for the precise classification and detection of DDoS attacks. The proposed models were evaluated using established metrics, such as precision, recall, accuracy, and F-measure. In addition, despite the solid performance metrics,

one major limitation of this study is the exclusive reliance on the KDD99 dataset. This dataset is known to be outdated and may not reflect the complexity and diversity of current DDoS attack patterns. Furthermore, the authors did not compare their models' performance with other benchmark datasets commonly used in intrusion detection systems, which limits the robustness of their findings. Addressing this gap would enhance the applicability of their approach in more contemporary settings.

Sahbi et al. [28] proposed intrusion detection systems (IDS) to enhance network function virtualizations (NFV) using ML algorithms. This study aims to address the emerging security challenges posed by SDN and propose an intelligent solution for identifying intrusions within these networks. The study employs a range of ML classifiers, including DT, RF, LR, Gradient Boosting (GB), Perceptron (Prc), Gaussian Naive Bayes (GNB), Stochastic Gradient Descent (SGD), and Linear SVC (LSVC). These algorithms have been applied to analyze network traffic data and identify various types of intrusion. Furthermore, the evaluation of the proposed solution involved metrics such as accuracy, precision, recall, and F1-score. The performance of each ML algorithm was assessed based on its ability to correctly classify different types of network traffic, including benign traffic, and various types of attacks, including DDoS attacks. This study utilizes a publicly available SDN-oriented dataset called ''SDN Intrusion'', which is provided by Cyber Cop and distributed under the GNU Affero General Public Licence 3.0. This dataset contains five traffic categories: DDoS, XSS intrusion, brute-force intrusion, SQL injection, and benign traffic. It consists of 79 features and over 1.1 million observations of network incursion and whitelisted traffic. The authors preprocessed and analyzed this dataset to train and evaluate their ML models. Furthermore, while multiple ML classifiers were employed, the absence of a relative comparative analysis against other state-of-the-art approaches reduces the strength of their findings.

Wang et al. [29] address the increasing security threats posed by DDoS attacks in SDN and aim to develop a solution that can effectively detect and mitigate such attacks. This study demonstrates the effectiveness of the proposed SL model in detecting flooding DDoS attacks against an SDN controller. Through simulations and measurements on real testbeds, the authors verified that the SL model, particularly the bagging tree algorithm, achieved a high accuracy of 99.64% in detecting DDoS attacks. Additionally, the study highlights the importance of selecting appropriate SL techniques and parameters to optimize the detection performance. The study compared the performance of different SL techniques in terms of detection accuracy using both experimental and real-world SDN datasets, including the DARPA and InSDN datasets. Furthermore, a key limitation of this study is that it primarily focuses on flooding DDoS attacks, which may not fully represent the wide variety of DDoS attack types seen in real-world scenarios. Moreover, the use of the DARPA and InSDN datasets, while useful, may not cover the latest attack patterns to SDN environments.

A recent study conducted by Garba et al. [30] focused on the Internet of Things (IoT) within an SDN to address DDoS attacks targeting IoT devices by classifying ML algorithms. The proposed framework specifically targets smart home networks, in which IoT devices are prevalent. This paper proposes a real-time DDoS attack detection and mitigation framework tailored for SDN-enabled smart home networks, emphasizing the importance of securing IoT devices within such environments. Additionally, the emphasis is on utilizing machine learning models such as DT, SVM, LR, and KNN to distinguish between benign and attack traffic, while also securing the SDN controller using a signature-based detection approach. This study employed several datasets: the IoT testbed dataset captured from OpenFlow switches, which contained both regular and attack traffic. UNSW-NB15 dataset: Capturing various attack types, including DDoS, reconnaissance, and exploits. The DT algorithm achieved a detection accuracy of 99.57% using the captured dataset from the smart-home IoT testbed. Additionally, when evaluated on other datasets, such as CICDDoS2019 and UNSW-NB15, the Decision Tree algorithm consistently outperformed the other algorithms, achieving accuracies of 99.95% and 98.2%, respectively. However, despite the high accuracy achieved by the proposed framework, the study lacks a discussion of potential limitations. For instance, the use of a signature-based detection approach may limit its ability to detect new or unknown DDoS attack patterns, as it relies on predefined signatures.

Limited utilization of actual IoT traffic features in building detection models. Khedr, et al. [31] presented another work based on SDN framework for detecting and mitigating DDoS attacks in IoT networks. The framework consists of four modules designed to efficiently detect and mitigate attacks while minimizing the computational burden on the IoT network nodes. The proposed model outperformed existing approaches across various evaluation metrics and effectively addressed the challenges posed by DDoS attacks on IoT networks. The ML-based detection module employs various ML algorithms, including Bayesian Logistic Regression (BLR), Gaussian Naive Bayes (GNB), SVM, KNN, DT, and RF. Among these, the Random Forest model performed best in terms of performance evaluation. The evaluation of the framework was conducted using real IoT traffic generated and deployed in the Mininet-IoT. Three test cases were considered: a single-node attack test case, and two multi-node attack test cases. In overall, one of the limitations of this study is the lack of diversity in the evaluated test cases. Only three test cases were considered, which may not fully capture the variety of DDoS attack scenarios present in real-world IoT networks.

Bhayo et al. [32] propose ML-based approach for DDoS attacks in an SDN IoT environment. The integration of SDN with IoT aims to improve security and access control

mechanisms. Three models, NB, SVM, and DT, were integrated into the SDN controller to classify network packets. The performance of the proposed framework was evaluated using different traffic simulation scenarios. Parameters such as CPU usage, attack detection time, and memory usage were analyzed to measure the performance of the framework. The network logs captured by the SDN controller were preprocessed and converted into a dataset for the training and testing of the ML models. The proposed framework achieved high accuracy rates, with a DT of 98.1%, NB of 97.4%, and SVM of 96.1%. Although the proposed framework focuses on detecting flooding types of DDoS attacks, it has the potential to extend research to cover other types of DDoS attacks and incorporate different types of IoT networks. However, the study lacks a detailed evaluation of how well the models generalize to different IoT networks, which could affect their applicability in more diverse real-world environments. Future work will include extending this research to incorporate other supervised learning algorithms, such as RF and XGBoost, and exploring unsupervised and semi-supervised learning approaches. Additionally, integrating a DDoS attack mitigation module and extending the framework to cover various types of DDoS attacks and IoT networks is suggested for further enhancement.

Anyanwu et al. [33] discussed the security challenges in intelligent transport systems (ITS) within the context of SDN. This study particularly addresses the susceptibility of SDN-based DDoS attacks owing to their centralized structure. The proposed solution involves integrating artificial intelligence tools, specifically a novel extension of the Radial Basis Function Support Vector Machine (RBF-SVM) kernel, for DDoS attack detection. The evaluation of the proposed solution involved the use of performance metrics to assess the effectiveness of different algorithms. Techniques such as grid search cross-validation (GSCV) and parameter optimization have been utilized to enhance the performance of the detection model. This study discusses the use of two datasets for evaluation: the SDN-DDoS and CICDDoS datasets. These datasets were used to validate the efficacy of the proposed framework under various conditions, including scaling conditions. The proposed model achieved high accuracy in DDoS attack detection, with 99.40% accuracy and 99.26% precision. However, the study does not address the robustness of the model against evolving attack strategies, which is crucial for long-term network security. Future research could consider real-world deployment, robustness against evolving attack strategies, and scalability to larger networks.

Limited research exists on integrating machine learning with SDN for DDoS defence in supply chain networks. Sebbar and Zkik [34] addressed this gap by proposing the use of machine learning techniques to detect and mitigate DDoS attacks in SDN-based supply chain networks. Their study demonstrated that integrating machine learning with SDN significantly enhances the effectiveness of DDoS attack detection and mitigation. They achieved high performance, with the Random Forest model achieving accuracy, precision, recall, and f1-score values all exceeding 98%. The evaluation used network traffic data collected from various nodes within an SDN-based supply chain network, thereby highlighting the practicality and effectiveness of the proposed solution in real-world scenarios. In addition, the study does not provide details on the scalability of the model in larger or more complex SDN environments, which could limit its applicability to broader use cases. Addressing these shortcomings would enhance the robustness and applicability of their findings.

**Insight:** Table 1 summarizes the related work. We employed LR, SVM, RF, KNN, and XGBoost for DDoS detection in SDN. We created a new dataset simulating attack scenarios and benchmarked it against established datasets. Importantly, our dataset was publicly available for further research. The authors of [28] used SVM, GLM, NB, DA, FNN, and BT for DDoS detection. They claimed to have created a custom dataset but did not provide detailed information about its generation process, and the dataset was not publicly available for testing by other researchers. While both studies explore DDoS detection using machine learning techniques, our study employs a wider range of algorithms and provides a publicly available dataset for benchmarking. Our work emphasizes transparency and reproducibility by sharing the dataset with the research community, thus enabling other researchers to validate and build upon our findings. In contrast, other studies lack transparency regarding the dataset creation process and do not offer it for public evaluation, limiting its usability and reproducibility.

## III. SELECTED MACHINE LEARNING MODELS FOR SDN THREAT DETECTION

In this section, we introduce our methodology for detecting DDoS attacks in SDN environments by using supervised ML techniques. Emphasizing the critical need to bolster network security against evolving cyber threats, our approach aims to harness the capabilities of machine learning to identify anomalous network behavior indicative of DDoS attacks, thus enhancing the resilience of the SDN infrastructure. To assess the performance of the proposed DDoS detection framework, we used a custom dataset containing diverse network traffic scenarios, including both normal patterns and simulated DDoS attacks. We conducted extensive experiments using five supervised ML known for their versatility, robustness, and effectiveness in classification. Here, we provide detailed explanations of why we chose each model and how they were used in our study.

### A. LOGISTIC REGRESSION (LR)

LR is a popular classification algorithm used in machine learning. Unlike linear regression, which predicts continuous values, logistic regression is used for binary classification tasks, where the outcome is either 0 or 1. LR models the probability that a given input belongs to a certain

**TABLE 1.** Comparison with other related works. C.: Custom dataset, P.: Public dataset, .:C&P This study used a custom public dataset.

| Reference | ML Techniques | Dataset Used | | | Evaluation Tools | Remarks |
|---|---|---|---|---|---|---|
| | | C | P | C&P | | |
| [19], 2020 | SVM, KPCA with GA | ★ | ✓ | ★ | **Emulator:** Mininet **Controller:** POX | While Dataset two, NSL-KDD, is available to the public, Dataset one is not publicly accessible, and discussions surrounding its origin are limited and unclear. |
| [25], 2024 | KNN, ANN, DT &RF | × | ✓ | × | **Emulator:** Mininet **Controller:** POX | The robustness and applicability of the system in various contexts unclear without testing on multiple datasets. |
| [25],2020 | J48, RT, REP Tree, RF, MLP & SVM | × | ✓ | × | **Emulator:** Mininet **Controller:** ONOS | It is essential to compare the performance of the proposed architecture with other existing approaches using the same dataset or alternative benchmark datasets. |
| [27], 2021 | DT & SVM | × | ✓ | × | **Emulator:** Mininet **Controller:** Not specified | Evaluating the performance of the two models on multiple datasets enhances the credibility and reliability of their findings. |
| [28], 2023 | DT, RF, LR, GB, Prc, GNB, SGD & LSVC | × | ✓ | × | Not specified tools | This study failed to mention the specific tools used for emulation, controllers, and switches in the SDN environment. The authors neglected to address the validity concerns regarding the dataset benchmarking. |
| [29], 2022 | SVM, GLM, NB, DA, FNN & BT | ✓ | ✓ | × | **Emulator:** Mininet **Controller:** RYU | The paper lacks a detailed explanation of the dataset generation process and does not make the dataset publicly available. |
| [30], 2024 | SVM, LR, DT & KNN | ✓ | ✓ | × | **Emulator:** Mininet **Controller:** RYU | Limited access to public datasets restricts the ability of the research community to evaluate and refine the proposed techniques. |
| [31], 2023 | BLR, GNB, SVM, KNN, DT & RF | ✓ | × | × | **Emulator:** Mininet **Controller:** POX | The paper does not provide details on how the dataset was collected, preprocessed, or annotated, which are crucial steps in ensuring the quality and reliability of the data used for evaluation. |
| [32], 2023 | DT, NB & SVM | ✓ | × | × | **Simulator:** Cooja and Contiki **Controller:** Not specified | The study did not provide the characteristics of the dataset used for training and testing ML models. |
| [33], 2022 | RBF-SVM | ✓ | ✓ | × | **Emulator:** Mininet **Controller:** Not specified | The details of the generated dataset were not discussed and are not available to the public to test other researchers. |
| [24], 2023 | RF | ✓ | × | × | **Emulator:** Mininet **Controller:** OpenDaylight | The dataset used in this study was not discussed in detail. Moreover, there is no benchmark for existing work. |
| **Proposed Model** | LR, SVM, RF, KNN & XGBoost | ✓ | ✓ | ✓ | **Emulator:** Mininet **Controller:** RYU **Switch:** OVS **Traffic Generator:** HPing3, MGEN | This study assessed DDoS detection in SDN, introduced a new dataset simulating attack scenarios, and evaluated it by using five supervised ML models. Benchmarking against established datasets validates its efficacy. Notably, our dataset is publicly available and provides a valuable benchmark for future research. |

✓ indicates the study utilize generated custom dataset, × the dataset is publicly available and ★ is Not specified.

class by using a logistic function. It fits a curve to the data points, allowing nonlinear relationships between the features and target variable. LR is widely used due to its simplicity, interpretability, and effectiveness in various domains, including cybersecurity. Numerous cybersecurity studies have utilized logistic regression for tasks such as intrusion detection, malware analysis, and DDoS attack detection [35], [36], [37]. Logistic regression plays a crucial role in SDN security solutions by aiding in the detection and mitigation of various threats, including DDoS attacks. Its ability to model the probability of malicious activity based on network traffic features makes it valuable for identifying suspicious behaviors and protecting SDN infrastructure from cyber threats. The equation for logistic regression is:

$$P(y = 1 \mid X) = 1/1 + e^{-(\beta_1 x_1 + \ldots + \beta_n x_n)} \quad (1)$$

where $P(y = 1|X)$ represents the probability of the positive class given input features $X$, $\beta_1$ is the intercept, $\beta_1, \ldots, \beta_n$ are the coefficients, and $x_1, \ldots, x_n$ are the feature values. Because logistic regression is crucial in cybersecurity and is widely applied, we ran tests on our specially designed SDN security dataset. Our experiments yielded promising results, with LR achieving an impressive accuracy of 84.31% in detecting and classifying security threats in SDN environments. This high accuracy underscores the effectiveness of logistic regression as a reliable tool for enhancing SDN security.

### B. SUPPORT VECTOR MACHINES (SVM)

SVM are widely used supervised learning algorithms for classification and regression. In classification, it discovers the optimal hyperplane that separates different classes with the largest margin in the feature space, thereby enhancing the generalization ability of the model. SVM is suitable for high-dimensional data and is robust to overfitting, making it suitable for complex datasets with

nonlinear boundaries. Although SVM is primarily used for classification tasks, it can also be used for regression tasks. In classification, SVM aims to maximize the margin between classes, leading to better generalization and resistance to noise in the data [38]. The decision plane, also known as the hyperplane, is determined by support vectors, which are the data points closest to the decision plane. This hyperplane separates the different classes in the feature space, as shown in Figure 3. Our primary goal was to utilize the hyperplane approach in our problem because it is well-suited for classification tasks. Furthermore, an SVM is essential for SDN security because it effectively classifies network traffic into normal and malicious activities. Our experiments with a custom dataset tailored for SDN security demonstrated the SVM's effectiveness, achieving an accuracy of 96.52% in classifying security threats. This high accuracy underscores SVM's trustworthiness and performance in enhancing SDN security by accurately identifying and responding to potential threats in real time.

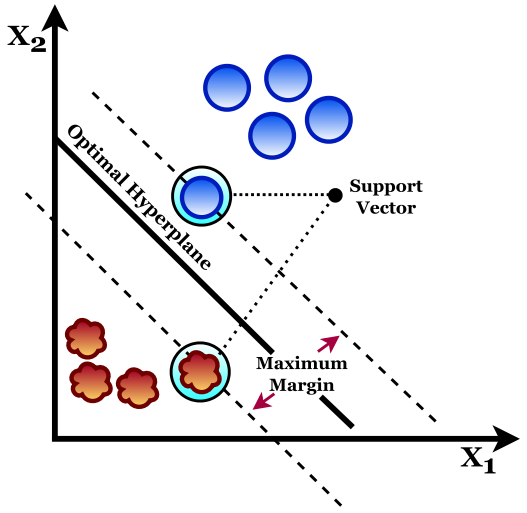


**FIGURE 3.** Visualization of optimal hyperplane with maximum margin and support vectors. The hyperplane serves as the decision boundary, maximizing the margin between classes, whereas support vectors (highlighted points) determine their position, aiding effective classification.

### C. RANDOM FOREST (RF)

Random Forest is a powerful ensemble learning method composed of decision trees [39]. Each tree in the forest was built using a random subset of training data and features. During classification, each tree independently predicted the class, and the final prediction was determined by combining the votes of all trees. As shown in Figure 4, ensemble decision trees comprise a collection of decision trees, where each tree contributes to the final prediction. These trees were trained independently and operated in parallel during the prediction. To evaluate the performance of the machine learning models, we employed k-fold cross-validation. This technique splits the dataset into k equal-sized folds, trains the model on k-1 folds, and evaluates the model on the remaining folds. This process is repeated k times, ensuring that the model's performance estimates are reliable and not overly sensitive to the choice of a single training-test split. RF has been extensively used in various studies on DDoS attack detection [40]. Its capability to handle large datasets and noisy data has enabled researchers to effectively detect and classify different types of DDoS attack. Moreover, by leveraging RF, SDN controllers can swiftly identify and mitigate DDoS attacks, thereby enhancing the network security and reliability. In our research, we applied Random Forest to our custom dataset for DDoS attack detection in SDN environments. Our experiments demonstrated that the Random Forest model achieved the highest accuracy of 98.97%, indicating its effectiveness in accurately identifying and classifying DDoS attacks in real-world scenarios.

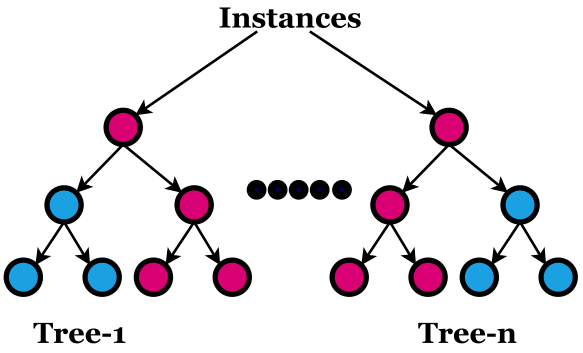


**FIGURE 4.** Random forest.

### D. K-NEAREST NEIGHBORS (KNN)

KNN has emerged as a compelling approach for detecting DDoS attacks within SDN environments. Our research explored the efficacy of KNN in discerning anomalous network behavior indicative of DDoS attacks by leveraging its simplicity and transparency. The fundamental principle of KNN, which classifies instances based on the majority class of their nearest neighbors in the feature space, aligns seamlessly with the dynamic nature of the network traffic in SDN. KNN demonstrates robustness in identifying and mitigating DDoS threats without the need for complex assumptions regarding data distribution [41]. The fundamental idea of the KNN algorithm is to classify or predict the label of a new data point based on the labels of its nearest neighbors in the feature space [42]. In other words, KNN identifies K samples (data points) closest to the new data point x in terms of their feature values. The closest samples, known as the nearest neighbors, are determined based on a distance metric, typically the Euclidean distance, as represented by the following equation:

$$d(x,u) = \sqrt{v^{-1}\sum_{i=1}^{n}(x_i - u_i)^2} \tag{2}$$

where $d(x, u)$ is the distance between vectors $x$ and $u$, and $v$ represents the dimensionality or length of the vectors, where $n$ is the number of dimensions in each vector. $x_i$ and $u_i$ are the corresponding elements of the vectors $x$ and $u$, respectively. The sum calculates the squared differences for each dimension and the entire expression is square-rooted to obtain the final distance measure.

Furthermore, KNN's interpretability and ease of implementation of KNN make it an advantageous choice for DDoS attack detection. Its transparent decision-making process allows for effective validation and interpretation of detection outcomes. Additionally, KNN's ability of KNN to handle high-dimensional data in SDN environments makes it well suited for analyzing network characteristics. This simplicity and interpretability make KNN a valuable asset in cybersecurity defenses, enabling organizations to proactively identify and mitigate DDoS threats in real-time. Our research on the use of the KNN algorithm for DDoS attack detection showed promising results, with an accuracy of 97.60%. This demonstrates the potential of KNN to distinguish between regular network traffic and malicious DDoS attacks. By incorporating KNN into our detection framework and using rigorous feature selection techniques, we lay the groundwork for improving the security and resilience of SDN infrastructure against evolving cyber threats.

### E. XTREME GRADIENT BOOSTING (XGBoost)

XGBoost is a key component of current ML practices and is particularly recognized for its proficiency in managing structured data and delivering superior predictive results [43]. In our research, which focused on identifying DDoS attacks within SDN settings, we capitalized on the strengths of XGBoost, owing to its resilience and scalability. By employing a gradient boosting scheme, XGBoost constructs a sequence of hundreds of decision trees and progressively refines predictions to minimize the loss function. This iterative process not only enhances predictive accuracy, but also ensures the adaptability of the model to complex, dynamic network environments. Thus, the importance of XGBoost in SDN security solutions cannot be overlooked. Its capacity to handle large datasets and resistance to overfitting make it an ideal tool for detecting DDoS attacks in real time [44]. Unlike traditional methods, XGBoost offers a versatile framework that can adapt to new threats and effectively mitigate risks. Furthermore, its interpretability allows network administrators to gain valuable insights into the features driving the detection process, enabling them to make informed decisions and implement proactive defence strategies. In our experiment, we evaluated the performance of XGBoost by using a custom dataset. The results showed an impressive 92.36% accuracy rate, indicating its effectiveness in distinguishing malicious traffic from legitimate network activities. In addition, XGBoost performed well across other metrics such as precision, recall, and F1-score, demonstrating its value as a key component of our DDoS detection framework. This strong performance highlights the relevance of XGBoost in SDN security and its potential to bolster the network infrastructure against emerging cyber threats.

### F. DATA PREPROCESSING AND HYPER-PARAMETER TUNING

In this study, the data preprocessing phase was carefully designed to ensure that ML algorithms could perform optimally on the dataset, while avoiding any potential biases. The preprocessing steps were applied independently to the training and test datasets after the data partitioning step to ensure that the test partition remained unseen during the model training process. The following steps were performed:

- **Handling of Missing Values:** Missing or null values in the dataset were handled by either removing or imputing them. This step ensured that the dataset was complete and free from inconsistencies that could hinder the learning process. The imputation strategy was applied separately to the training and test sets to prevent data leakage.
- **Categorical Value Encoding:** Categorical variables, such as Source_IP and Destination_IP, were encoded into numerical formats to be processed using the ML approach. We used dummy variable encoding to convert these categorical features into a series of binary variables. This transformation expanded the dataset and enabled the algorithms to handle categorical data effectively. Notably, this encoding was performed independently for the training and test datasets to ensure that no information from the test set was leaked during the training process.
- **Normalization:** To ensure that features with large value ranges did not disproportionately influence the learning process, normalization was performed. All the features were scaled down to a common range, typically between 0 and 1. This normalization step prevents features with larger scales from dominating the learning process and ensures that all the features contribute equally to the model's decision-making. As in the previous steps, normalization was conducted separately for the training and test datasets.
- **Expansion of Dataset:** After preprocessing, the dataset expanded from 16 to 57 columns owing to the addition of dummy variables for categorical features. This expansion increases the dimensionality of the dataset, allowing ML methods to capture more complex relationships between features and improve classification performance.

Thereafter, this preprocessing step is essential to ensure that the algorithms work effectively on the dataset without being biased by the original scales or missing data. By applying these preprocessing steps after partitioning the dataset into training and test sets, we mitigated the risk of introducing a positive bias and ensured that the evaluation of the models was fair and unbiased.

In addition to evaluating the performance of the ML frameworks using k-fold cross-validation, hyper-parameter tuning was also performed to optimize the performance of each algorithm. Hyper-parameter tuning was carried out using grid search combined with k-fold cross-validation. This method allowed us to explore different combinations of hyper-parameters and select the best-performing configurations based on the average performance across the folds.

### G. COMPARATIVE ADVANTAGES OF THE PROPOSED METHOD

The proposed method offers distinct improvements over the existing ML techniques by focusing on several critical areas.

- **Enhanced Model Performance Through Data Representation:** The chosen ML methods, such as RF and XGBoost, demonstrate superior performance owing to their ability to effectively leverage the selected features. The emphasis on robust feature engineering and the use of models that handle complex patterns in the data contribute to achieving a higher accuracy (98.97%) and a lower FAR compared with existing techniques, which often suffer from limited adaptability to dynamic network environments.
- **Optimized Handling of Imbalanced Data:** Unlike previous studies, the proposed method employs models specifically chosen for their robustness in handling imbalanced datasets. RF is known for its ability to manage data imbalances without overfitting, leading to better precision and recall rates. This approach ensures a more balanced and reliable detection, which has been a common challenge in prior research that relies on techniques that are less capable of dealing with data imbalance.
- **Targeted Feature Selection:** The use of the chi-square (Chi2) algorithm for feature selection ensures that the model focuses on the most relevant data, thereby improving both accuracy and efficiency. This process optimizes the performance of the model by concentrating on key features, addressing a limitation in earlier studies that did not prioritize targeted feature selection, resulting in suboptimal performance.
- **Comprehensive Evaluation Using Multiple Metrics:** Unlike earlier studies, which often limited their evaluations to a narrow set of metrics, the proposed method includes a broad evaluation across multiple performance indicators, such as accuracy, precision, recall, and F1-score. This ensures a more complete assessment of the model's capabilities and robustness, particularly in real-world scenarios, where multiple metrics are crucial for performance validation. The comprehensive results summarized in Table. 7 highlight the superior performance of the proposed method across these metrics.

## IV. DATASET CREATION USING MININET EMULATOR

In this section, we outline the steps taken to create a custom dataset by using the Mininet emulator to capture the intricate dynamics of an SDN environment. The resultant dataset comprised 1,048,575 rows and 21 columns, reflecting a diverse array of network behaviors, configurations, and performance characteristics. The network architecture used for data collection consists of 12 switches managed by the Ryu controller, facilitating connectivity for 24 interconnected devices. The steps involved in the dataset creation are listed below.

- **Topology Design:** The first step involved designing a network topology that encapsulates the desired characteristics and complexities of SDN environments. We crafted a topology comprising 12 switches and 24 hosts interconnected in a manner conducive to diverse traffic patterns and network behaviours, as shown in Figure 5. Figure 5(a) shows the first topology scenario, in which the network topology comprises various hosts interconnected by switches. PC5, PC8, and PC17 were identified as attacker hosts, collectively launching an attack against PC12, the targeted host. This topology illustrates the initial stage of the attack scenario, highlighting the interactions between the attacker and target hosts within the network environment. Figure 5(b) illustrates the second topology scenario, where a different network topology scenario is depicted, wherein PC15 and PC22 are identified as the attacker hosts. These hosts perpetrate an attack against PC20, the target host. This depicts an alternate scenario in which different hosts are involved in an attack against a distinct target within the network infrastructure.
- **Mininet Setup:** With the topology defined, we proceeded to set up the Mininet emulator environment. Leveraging the flexibility and scalability offered by Mininet, we instantiated virtual network elements, including switches, hosts, links, and controllers, to emulate real-world network scenarios.
- **Ryu Controller Integration:** Integration of the Ryu controller facilitates centralized control and orchestration of network operations in the Mininet environment. By deploying Ryu, we exercised fine-grained control over network behavior, implemented SDN policies, and collected real-time data for subsequent analysis.
- **Traffic Generation:** Generating diverse traffic flows within a network is pivotal for capturing a broad spectrum of network behaviors and performance metrics. We simulated various traffic patterns, including normal and abnormal traffic, to simulate real-world workloads and scenarios. The MGEN tool generated various traffic types, including TCP, UDP, and ICMP, at controlled rates to simulate legitimate network traffic. For DDoS attack traffic, hping3 was used to simulate SYN floods, UDP floods, and ICMP floods. Attack traffic

was generated at varying intensities, from moderate (100 packets/second) to high (1000 packets/second), to represent different levels of DDoS attacks. This variation ensures that the dataset captures different stages and severities of attacks.

- **Feature Overview:** From the initial 26 features extracted from network flows, we applied the Chi-square (Chi2) feature selection algorithm. This method prioritized features most relevant to DDoS detection based on their statistical significance. As a result, we retained 16 essential features, such as flow duration, packet-per-flow (PPF), bytes-per-flow (BPF), and packet rate, which were used to train and evaluate the supervised learning models.
- **Data Collection:** Real-time data collection mechanisms were deployed to capture pertinent network parameters, performance metrics, and flow statistics every 30 s.
- **Data Storage and Structuring:** The collected data were stored in a structured format suitable for subsequent analysis and processing. We structured the dataset into 1,048,575 rows and 21 columns, with each row representing a distinct network observation, and each column encapsulating specific network features. The dataset was saved in a comma-separated value (CVS) file named "SDN-DDoS_Traffic_Dataset.cvs."
- **Data Preprocessing:** Preprocessing steps were applied to clean, normalize, and transform the raw data into a usable format.

### A. FEATURE OVERVIEW: UNDERSTANDING THE DATASET COMPOSITION

In this section, we provide a comprehensive overview of the key attributes collected in our dataset for testing the DDoS traffic classification using ML. Each feature encapsulates essential information about network traffic, ranging from basic identifiers, such as switches and hosts, to more intricate metrics, such as packet rates and data transfer rates. This section serves as a foundation for understanding the dataset composition and metrics used for subsequent analysis and classification tasks. The following list provides the characteristics of these features:

- **Flow (F):** A flow in networking is a collection of packets sent between a specific source and destination IP addresses, typically within a certain time window, using common protocols and port numbers. Network devices such as routers and switches maintain flow tables to store information about active flows passing through the device. Flow tables include information, such as source and destination IP addresses, protocols, port numbers, packet and byte counts, and timestamps. As packets travel through a network device, the flow table is updated using new entries and statistics. If a new packet matches an existing flow entry, relevant statistics are updated. Otherwise, a new entry is created to represent a new flow.

$$f = length(flowtable) \tag{3}$$

Furthermore, the above expression calculates the length or size of the flow table, which represents the total number of active flows stored in the table. Where, length(*flowtable*) denotes the function that returns the number of entries or rows in the flow table.

- **The packet per flow (PPF)** represents the average number of packets in the network communication flow. This is calculated by dividing the total number of packets in a flow by the total number of flows, as expressed below.

$$PPF = packet\_count/flows \tag{4}$$

This feature provides insight into the typical packet density within individual communication flows. In the context of DDoS attack classification, anomalies in the *PPF* values can indicate potential malicious activities. For example, a sudden spike or drop in *PPF* compared to the normal baseline could indicate the presence of a DDoS attack. Attackers often generate abnormal packet rates to overwhelm the target system or network, and monitoring the PPF helps detect such anomalies.

- **Byte Per Flow (BPF):** Unusually high BPF values may indicate abnormal behavior, such as large data transfers associated with DDoS attacks or other malicious activities. Therefore, BPF can be used as an indicator for detecting anomalous traffic patterns. BPF represents the average number of bytes transferred per flow in the network traffic dataset. This was calculated by dividing the total number of bytes transferred in all flows by the total number of flows. Mathematically, the BPF is represented as.

$$\frac{\sum_{l=1}^{n} bc_i}{\text{flows}} \tag{5}$$

where $\text{bc}_i$ is the number of bytes transferred in the $i$th flow and flows is the total number of flows in the dataset. A higher BPF value indicates that, on average, larger amounts of data are transferred in each flow, whereas a lower value suggests smaller data transfer sizes per flow.

- **Packet rate (PR):** PR is an important feature for attack classification because it can indicate abnormal behavior in network traffic. For instance, DDoS attacks often involve a significantly higher packet rate than that of normal traffic. By monitoring the packet rate, anomalies in the network behavior can be detected, allowing for the identification and mitigation of potential attacks. The packet rate was calculated as follows:

$$\boldsymbol{Pr} = \frac{\Delta p}{\Delta t} \tag{6}$$

where $\Delta p$ represents the change in packet count over a time interval and $\Delta t$ represents the elapsed time during which the change in packet count occurs. Suppose that

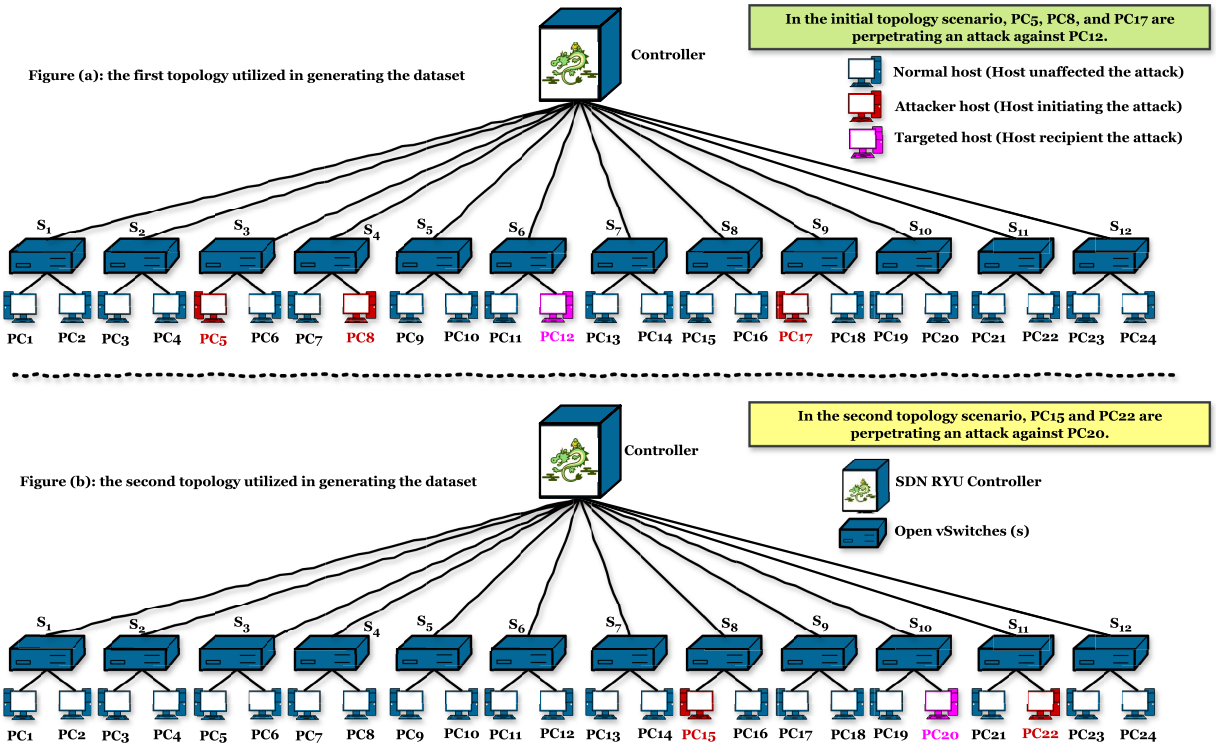


**FIGURE 5.** Network topologies used for dataset generation. (a): Attackers PC 5, PC 8, and PC 17 target PC 12. (b): Attackers PC 15 and PC 22 target PC 20.

the network flow transmits 1000 packets in 10 s. The packet rate was 100 packets per second (PPS).

- **Protocol:** This feature provides valuable information for understanding network traffic patterns, identifying potential security threats, and developing effective defence strategies against attacks. This analysis contributes to a comprehensive classification and detection of network-based attacks. For example, an increase in the use of ICMP (Internet Control Message Protocol) traffic, which is often associated with network scanning, could signal an ongoing DDoS attack.
- **Delay:** A delay is the time taken for a data packet to reach its destination, and includes factors such as propagation, transmission, queuing, and processing delays. This is a crucial factor for identifying malicious activities in network communication. For instance, in a DDoS attack, excessive traffic floods the network, causing congestion and increased delay in legitimate traffic. Monitoring delay metrics enables the detection of abnormal patterns and help identify and mitigate DDoS attacks.
- **Port number (PNO):** A port is a communication endpoint identified by using a numerical value. Ports allow different applications and services to share a single physical or virtual connection within a network. PNO ranged from 0 to 65535, with certain ranges reserved for specific purposes. Moreover, unexpected port usage can indicate potentially malicious activity. For instance, if a port commonly associated with one protocol is suddenly used for a different type of traffic, it may indicate an attempt to evade detection or to exploit a vulnerability.
- **Total Duration of Communication (TDC):** This feature is important for attack classification because it provides insight into the behavior of the network traffic. For example, a DDoS attack may involve a flood of packets targeting a specific service or server, resulting in unusually long or short communication duration. Furthermore, to identify attacks using the duration feature, one typically looks for deviations from normal behavior. To compute the total TDC flow in microseconds, we integrated the duration of the flow in seconds ($d$) with its duration in microseconds ($d_{\mu s}$) using the following equation:

$$TDC = \left(d * 10^6\right) + \left(d_{\mu s}\right) \tag{7}$$

Suppose we have a communication flow with durations of 3s and 500 microseconds. Using the above equation, the total duration of the communication flow is 3,000,500 microseconds.

- **Switch:** The switch feature serves as a network conduit for routing traffic and helps to identify attack sources.

Traffic patterns from compromised devices that are connected to the same switch may exhibit similar characteristics. Abnormal traffic patterns consistently emanating from a specific switch may suggest a coordinated attack.

- **Host:** The host feature identifies the source or destination host involved in the communication. In the case of an attack, identifying a compromised host can help to isolate and mitigate attacks. For instance, if traffic from a specific host suddenly increases abnormally or exhibits suspicious behavior such as scanning multiple ports, it could indicate a potential attack.
- **Source Internet Protocol (SIP) and Destination Internet Protocol (DIP):** SIP and DIP addresses are fundamental features in network traffic analysis. Anomalous traffic patterns, such as a high volume of traffic originating from single or multiple suspicious source IP addresses towards a specific destination IP address, can be indicative of a DDoS attack or an unauthorized access attempt.
- **Packet per message (PPM):** PPM refers to the number of packets required to transmit a single message or data unit over a network. DoS attacks often involve flooding a target network or server with a high traffic volume. Monitoring the PPM ratio allows for the detection of unusual patterns, where a significant increase in the number of packets per message occurs, indicating potential malicious activity.
- **Jitter:** Jitter refers to the variation in packet arrival times. High jitter values may indicate a congested or obstructed network, which could be attributed to a security breach, such as a DDoS attack directed at the network infrastructure.
- **Packet Loss Rate (PLR):** This feature represents the rate of packet loss in the communication flow. A significant PLR can degrade network performance and indicate potential network attacks such as packet fragmentation or ICMP flood attacks.
- **Label:** The label feature categories traffic into different classes, such as normal and attack traffic. This serves as the ground truth for training the ML models to accurately classify and detect attacks. By analyzing the characteristics of labelled attack traffic, such as patterns in the aforementioned features, machine-learning models can identify and classify attacks effectively.
- **Data Transfer Metrics (DTM):** Other features include transmission bytes (*tx_bytes*), received bytes (*rx_bytes*), transmission kilobits per second (*tx_kbps*), received kilobits per second (*rx_kbps*), and total kilobits per second (*tot_kbps*), which provide insight into the transmission and reception rates and volumes of data. Anomalous increases in *tx_kbps* or *tot_kbps* compared with the normal baseline levels could indicate a potential attack, such as data exfiltration or volumetric DDoS attacks.

### B. DIFFERENTIATION FROM EXISTING DATASETS

Nevertheless, the SDN-DDoS_Traffic_Dataset distinguishes itself from existing datasets by addressing the key limitations that have constrained previous research on DDoS detection. Unlike datasets such as KDD99 and NSL-KDD, which are restricted by outdated network architectures, this dataset was generated using the Mininet emulator to simulate realistic SDN topologies that accurately mirror modern network infrastructure. To ensure relevance, contemporary attack scenarios were integrated, and detailed flow-based metrics were captured to address the outdated attack types and simplistic features found in the older datasets. Furthermore, while traditional datasets often focus on a narrow range of attack types, the novel dataset incorporates a broader spectrum of sophisticated and multi-vector DDoS attacks, reflecting the evolving nature of cybersecurity threats and making it more applicable to the current security challenges. In contrast to datasets primarily centered on packet-level metrics, this dataset emphasizes flow-based features such as packet-per-flow (PPF) and byte-per-flow (BPF), which are crucial for detecting subtle anomalies within network flows. This focus enables a more detailed analysis of the network behavior, thereby enhancing the effectiveness of ML techniques trained on these data. Additionally, to overcome the accessibility challenges that have historically limited the usability of many datasets, a custom dataset has been made publicly available on Mendeley Data, promoting transparency and enabling broader research collaboration. Moreover, through rigorous benchmarking against widely used datasets, such as CICDDoS2019, this dataset has demonstrated superior performance in detecting DDoS attacks, particularly in terms of accuracy and false alarm rates. The combination of realistic simulation, contemporary attack representation, flow-based metrics, and public accessibility establishes the SDN-DDoS_Traffic_Dataset as a significant advancement over the existing alternatives.

### C. ADDRESSING THE LIMITATIONS OF SYNTHETIC TRAFFIC GENERATION

Although synthetic traffic generation tools such as MGEN and hping3 were useful for simulating DDoS attack scenarios in this study, they do not fully capture the complexity and unpredictability of real-world network traffic. This can lead to models that perform well in controlled environments, but may not generalize as effectively to real-world conditions, where traffic patterns and attack behaviors are more diverse and sophisticated [19], [23], [32]. To address these limitations, we recognize that the lack of real-world variability in synthetic datasets may affect the accuracy of our results when deployed in operational environments [25], [27]. Benchmarking against the CICDDoS2019 dataset provides some validation, but future work should incorporate hybrid datasets that combine synthetic traffic with real-world captures from sources, such as MAWI or CAIDA, to better reflect live network conditions. Additionally, the use of

advanced emulation platforms, such as GENI or DETERLab, could provide more dynamic and realistic traffic scenarios, enhancing the robustness of machine learning models for DDoS detection.

## V. PROPOSED FRAMEWORK FOR DDoS DETECTION IN SDN USING MACHINE LEARNING CLASSIFICATION

In this section, we present a comprehensive framework for detecting DDoS attacks in SDN environments. DDoS attacks pose a significant threat to network infrastructure, disrupting service availability and causing financial losses [45]. By leveraging ML classification techniques, the proposed framework aims to effectively identify and mitigate attacks in SDN environments. Figure 6 illustrates the proposed DDoS detection framework for SDN. This framework comprises of several key steps, each crucial for the successful detection and mitigation of DDoS attacks.

The process of creating a labelled network traffic dataset comprises of several stages. The first stage involved data collection, which entailed gathering network traffic data from the MGEN (Multi-Generator) and hping3 tools. MGEN is a versatile instrument for generating various types of benign network traffic including TCP, UDP, and ICMP traffic. In contrast, hping3 is used to simulate different types of DDoS attacks such as SYN, UDP, and ICMP. Both tools were used to ensure that the collected data encompassed a diverse range of network activities, including both normal traffic and DDoS attacks. The second stage was annotation, which involved manual or automatic labelling of each instance in the dataset, with labels indicating its class: normal traffic or DDoS attack. In this stage, DDoS attack instances are annotated based on known attack patterns, signatures, or behavior anomalies. The third stage is Feature Extraction, which involves extracting relevant features from network traffic data to represent each instance. Common features for DDoS detection include packet-level features such as packet size and protocol type; flow-level features such as flow duration, packet count, byte count, packet rate, and flow rate; and statistical features such as entropy of packet sizes or inter-arrival times. In addition, domain-specific features or metadata such as source/destination IP addresses, port numbers, and packet timing should be considered. The fourth stage was Data Preprocessing, which involved cleaning and preprocessing the annotated traffic dataset to ensure data quality and consistency. The fifth stage is Dataset Splitting, which involves splitting the annotated dataset into training, validation, and testing sets. The training set was employed to train the machine learning models, the validation set was utilized for hyperparameter tuning and model selection, and the testing set was used to evaluate the performance of the final model.

Once the annotated traffic dataset was obtained, the next crucial step was to prepare the data for training the ML models. This ensured that the training data was properly organized and ready to be used effectively. The quality of our training data significantly influenced the performance of the ML models. The training data plays a vital role in teaching the models to recognize patterns and relationships within the data, enabling them to make accurate predictions or classifications of unseen data. Several phases are involved in the preparation of training data. First, we perform feature selection/extraction to identify the most relevant features for training the models. In this study, we utilized the chi-square (Chi2) algorithm to select the most informative features from the dataset. Following the feature selection, we split the dataset into features (X) and labels (y). Subsequently, we divided the dataset into training and testing datasets. The typical split ratio used in our study was 80-20, where 80% of the data was used for training the models, and the remaining 20% was reserved for testing the model's performance.

In the training stage, we trained our ML models using training data from the annotated traffic dataset. We employed various ML algorithms such as LR, SVM, RF, KNN, and XGBoost to identify patterns and relationships within the data. Each algorithm is based on distinct principles and mathematical techniques for learning from data, as discussed in section IV-A. After the completion of training for each algorithm, we evaluated and compared their performance using metrics such as the ACC, PRC, RCL, and F1-score (see section VII). Saving the trained models allows deployment in SDN environments to predict unseen data. For the testing data, we prepared test data from our custom dataset, consisting of 69 columns and approximately 1,048,575 rows, after preprocessing. The test data were split from the preprocessed dataset using a standard train-test split. Each row in the test data represents a single instance from the dataset with annotated labels for the target variable. During the evaluation, test data were used to assess the performance of the five ML models. In the context of DDoS attack classification, performance evaluation encompasses an assessment of the ability of ML models to accurately classify network traffic into two categories: normal and attacked. Through rigorous evaluation metrics and techniques such as accuracy, precision, recall, and F1-score, the effectiveness of these models in distinguishing between benign and malicious traffic patterns was comprehensively analyzed, facilitating informed decisions for enhancing network security measures.

Furthermore, the lack of publicly available datasets with realistic traffic and flow information poses a challenge for researchers developing and evaluating IDS systems in SDN environments. In this study, we address this gap by proposing an algorithm for generating a dataset with traffic and flow information in SDN. Algorithm 1 outlines the process of generating an SDN traffic dataset using the Mininet experiments. The algorithm consists of two main phases: initialization and Mininet experiments.

- **Initialization:** In this phase, we define the parameters for extracting traffic statistics from Mininet and specify the criteria for labelling attacks and normal traffic. This involves setting up a Mininet topology with the desired configurations, including controllers, switches,

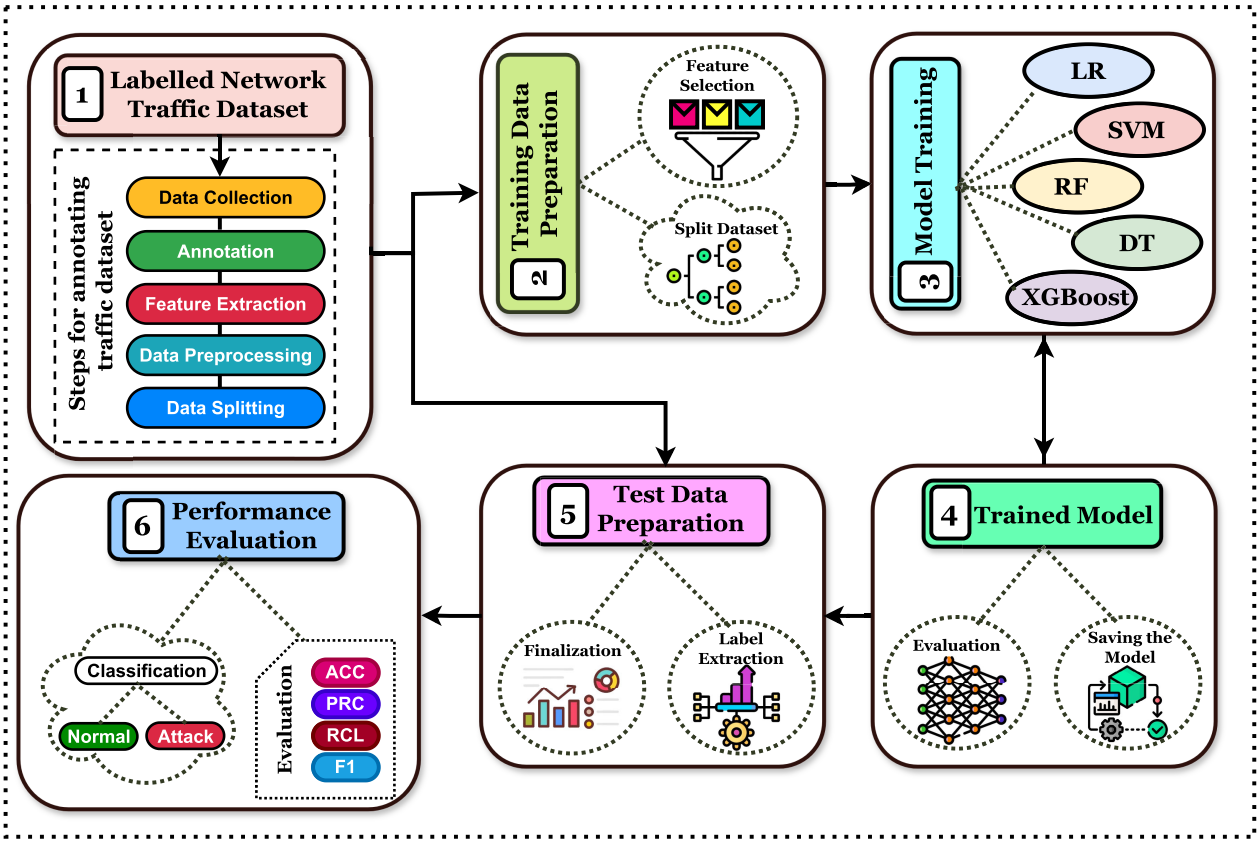


**FIGURE 6.** Illustrates the proposed DDoS Detection Framework for SDN, leveraging machine learning classification techniques to effectively identify and mitigate DDoS attacks. Step 1 depicts the labelled network traffic dataset, Step 2 illustrates the training dataset preparation, Step 3 shows the model training, Step 4 visualizes the trained model, Step 5 demonstrates the test data preparation, and Step 6 highlights the performance evaluation.

and hosts. In addition, we implemented attack scenarios such as DDoS attacks to simulate realistic network traffic.

- **Conduct Mininet experiments:** In the Mininet experiments phase, we execute traffic generation and capture traffic statistics at regular intervals. For each flow in the network, we extracted the flow and port statistics from the SDN switches and calculated additional features, such as delay, jitter, and packet loss rate. These statistics are then compiled into a structured dataset.

After collecting the statistics, we analyzed the data to distinguish the attack traffic from the normal traffic. Based on predefined criteria, we label the dataset and accordingly assign label 1 to instances representing attack traffic, and label 0 to instances representing normal traffic. Finally, the annotated SDN traffic dataset derived from the Mininet experiment is returned.

Figure 7 illustrates a custom dataset creation block diagram for evaluating IDS systems and anomaly detection in SDN. The steps involved are as follows:

- **Topology design:** The network topology, including controllers, switches, and hosts, is defined to simulate the SDN environment.
- **Mininet Setup:** Configure the Mininet environment with the defined network topology to ensure proper connectivity and functionality.
- **Controller Integration:** Integrate the SDN controller(s) with the Mininet topology to manage network operations and control traffic flow.
- **Traffic Generation:** Implement various traffic generation techniques, including normal traffic and attack scenarios (e.g. DDoS attacks), to simulate realistic network conditions.
- **Data Collection:** Capture traffic statistics and flow information from SDN switches at regular intervals, including flow and port statistics, delay, jitter, and packet loss rate.
- **Storage and structuring:** Collected data are stored in a structured format and organized into a dataset suitable for analysis and model training.

## VI. EXPERIMENTS SETUP

To conduct experiments, the selection of an appropriate controller for simulation plays a crucial role in influencing the effectiveness and authenticity of the research. Controller selection acts as a critical factor influencing the governance

**Algorithm 1** : The Algorithm Produces a Dataset Containing Traffic and Flow Information in SDN

**Initialization:**
Define parameters for extracting traffic statistics from Mininet.
Specify criteria for labeling attack and normal traffic.
**Conduct Mininet experiments:**
Set up Mininet topology with desired configurations (controllers, switches, hosts).
Implement attack scenarios (e.g., DDoS attacks).
Execute traffic generation and capture traffic statistics at regular intervals.
**for** each flow **do**
    Extract flow and port statistics from SDN switches.
    Calculate additional features (delay, jitter, packet loss rate).
    Compile statistics into a structured dataset format.
**end for**
**Annotate the dataset:**
Analyze collected statistics to distinguish attack traffic.
Label the dataset:
Assign label 1 to attack traffic instances.
Assign label 0 to normal traffic instances.
**Return:** Annotated SDN traffic dataset.

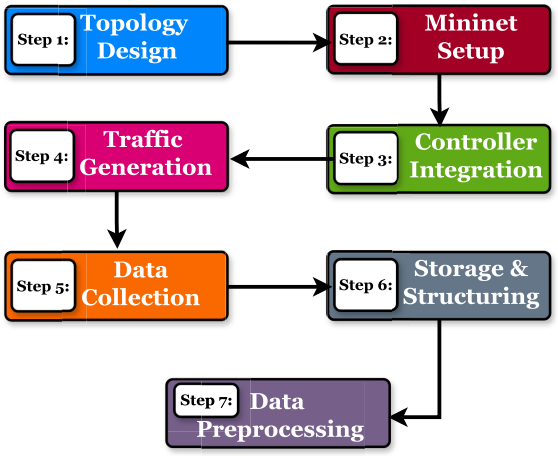


**FIGURE 7.** Custom Dataset Generation Framework for SDN Security Evaluation.

of network resources and orchestration, thereby directly affecting the operational behavior and performance indicators of the SDN network undergoing evaluation. The controller serves as the brain of SDN, orchestrating the flow of data and commands between the control and data planes. Its selection profoundly influences various aspects such as network scalability, resource utilization, latency, and fault tolerance. Therefore, careful consideration is imperative to ensure that the chosen controller aligns with the specific requirements and objectives of an experiment. After evaluating various options, we selected the RYU controller because of its numerous advantages. RYU is an open-source Python-based SDN controller with excellent adaptability, extensibility, and dependability [46]. Its extensive community support and ongoing development have made it appealing to both researchers and professionals. Compatibility of the controller with the OpenFlow protocol also allows smooth integration with different network components, facilitating extensive experimentation and personalization.

**TABLE 2.** Parameters used Simulation environment.

| S. No | Component and Specification |
|---|---|
| 1 | **Operating System:** Ubuntu 22.04.3 LTS |
| 2 | **Graphics:** AMD Caicos |
| 3 | **Emulator:** Mininet |
| 4 | **Disk capacity:** 1TB |
| 5 | **Memory:** 32 Gigabytes |
| 6 | **Processor:** Intel Core i5 CPU @3.40GHz (4 cores) |
| 7 | **Controller:** RYU |
| 8 | **Switches:** Open VSwitch (OVS) |
| 9 | **Number of Controller:** 1 |
| 10 | **Number of Switches:** 3 |
| 11 | **Protocol:** OpenFlow |
| 12 | **Visualization:** MiniEdit |
| 13 | **Port for Controller:** 6653 |
| 14 | **Network Packet Crafting Tool:** Hping3 |
| 15 | **Multi-Generator Packet Crafter:** MGEN |
| 16 | **IP for Host 1 to Host 6:** 10.0.0.1 to 10.0.0.6 |
| 17 | **Bandwidth:** 30 s per plot interval |
| 18 | **Traffic Type:** Benign and Malicious |
| 19 | **Simulation Time:** 300 s |
| 20 | **Packet Loss:** 0% |
| 21 | **Data collected:** Statistically every 30 seconds |

To emulate the SDN environment and facilitate the experimentation, we employed a Mininet emulator. Mininet provides a lightweight and scalable platform for creating virtual SDN networks with customizable topologies and realistic traffic patterns [47]. Its integration with the RYU controller streamlines the simulation process, enabling researchers to emulate complex network scenarios and to evaluate the performance of SDN applications and algorithms in controlled environments. The simulation environment was hosted on a desktop machine running Ubuntu 22.04.3 LTS, equipped with robust hardware specifications including a 1TB disk capacity, AMD Caicos graphics, 32 gigabytes of memory, and an Intel Core i5 CPU @3.40GHz processor with four cores. The network topology comprises three switches and six hosts, meticulously designed using the graphical package MiniEdit to emulate real-world network configurations and facilitate comprehensive experimentation. The details of each parameter used in the experiment are listed in Table 2.

Figure 8 shows a network topology consisting of six hosts connected to three open vSwitch (OVS) switches. Each

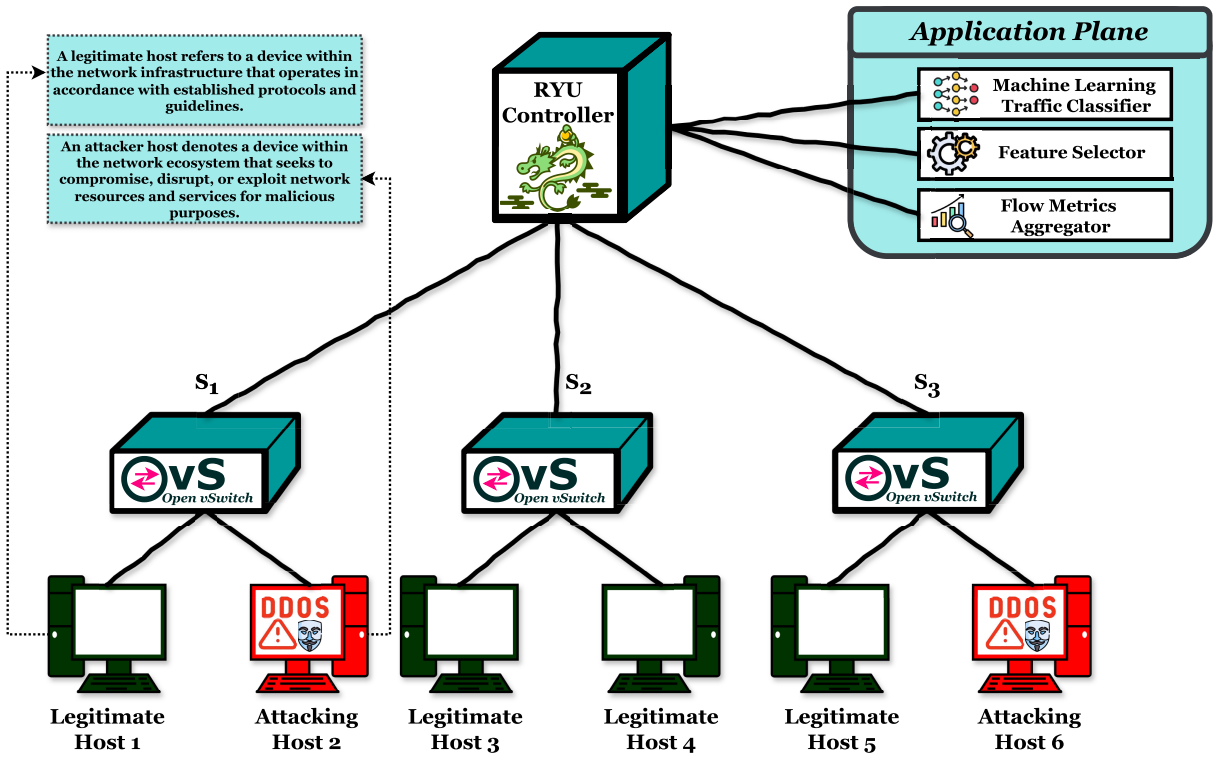


**FIGURE 8.** SDN topology (experimental setup).

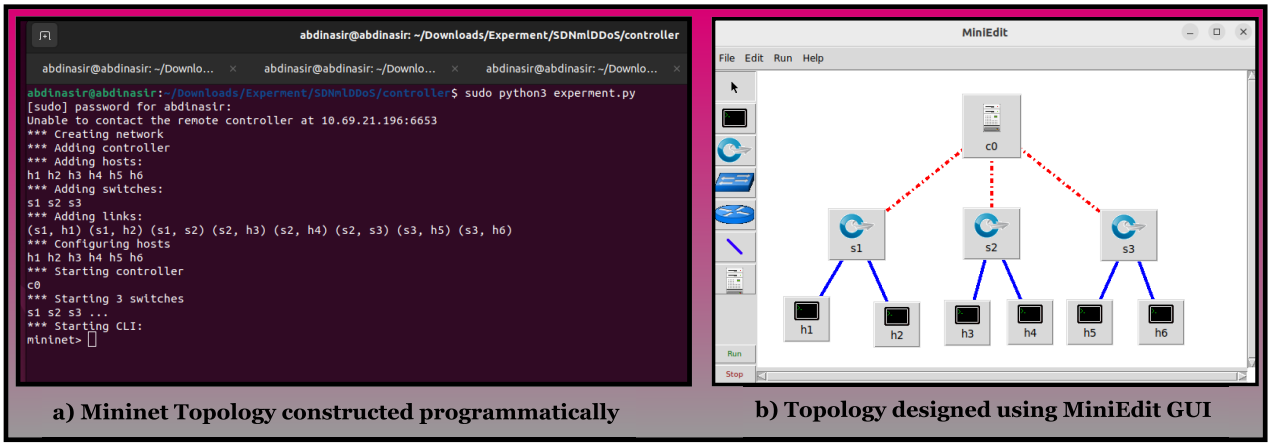


**FIGURE 9.** Mininet Topology with One Controller and Three Switches.

OVS switch is connected to a single RYU controller. The application plane comprises three main components: the ML traffic classifier, the feature selector, and the flow metric aggregator. The ML Traffic Classifier employs machine learning algorithms to categorize network traffic based on predefined criteria. By examining incoming data packets, the component distinguishes between benign and malicious traffic, enabling intelligent decision making and traffic management. The Feature Selector component identifies relevant features from incoming network traffic data to enhance the efficiency of subsequent operations. This reduces the dimensionality of the data while retaining crucial information for analysis or decision-making processes. The Flow Metrics Aggregator component summarizes the flow-related metrics for different network flows, such as the packet count, byte count, and duration. This aggregation provides valuable insights into the overall network, facilitating performance monitoring and troubleshooting.

In the experiment, hosts 2 (H2) and H6 were designated as attackers in an attempt to compromise other hosts within the network. We utilized the OpenFlow protocol for communication between the controller and network elements, with the controller port number set to 6653 for seamless integration and control. To generate benign and malicious traffic, we leveraged two distinct tools: hping3 and MGEN. Hping3, a command-line network tool, was employed to generate and analyze TCP/IP, UDP, and ICMP packets to simulate malicious traffic. This encompasses network data designed to cause harm, disruption, or unauthorized access. Conversely, MGEN was utilized to generate benign traffic, representing network data generated by legitimate users or devices for normally authorized purposes.

Furthermore, Figure 9 shows two Mininet topologies: a) a programmatically constructed topology and b) a topology designed using the MiniEdit package. The first topology illustrates a network setup created through code, showcasing three switches interconnected with hosts, whereas the second topology demonstrates a visually designed network layout using the MiniEdit graphical user interface. These representations offer insights into different methodologies for creating and visualizing network topologies within the Mininet framework, catering to diverse research and experimental needs in SDN.

We employed a 5-fold cross-validation technique to evaluate the performance of our models. This technique ensures that each data subset serves as a testing set once while the remaining data is used for training, providing a reliable estimate of the model's performance.

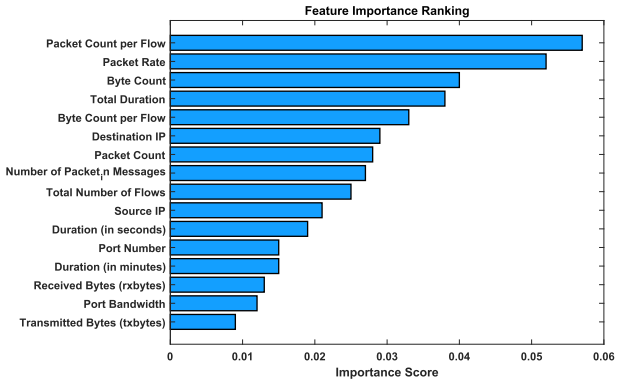


**FIGURE 10.** Selected features for the proposed dataset.

### A. DATASET FEATURES SELECTION FOR MODEL TRAINING

Figure 10 provides a visual representation of the feature importance ranking, highlighting the selected variables derived through the rigorous Chi2 feature selection process. Initially, the study comprised of 26 features, as detailed in Section IV. Utilizing the Chi2 feature selection methodology, Table 3 presents a curated list of selected features, along with their respective descriptions. This method serves as a crucial tool for discerning informative features from non-informative ones, thereby enhancing the robustness of the generated dataset. It is crucial to emphasize the meticulous curation process, which aims to prevent the detrimental effects of overfitting, by condensing the feature set into 16 essential variables. This approach is supported by prior research, such as that of *Zulhipni* et al. [48], which demonstrated its efficacy in similar experimental contexts. The systematic computation of chi-square assiduously evaluates features based on their relevance to the target variable, thereby providing a rigorous framework for ascertaining feature importance. Overall, this method bolsters the efficacy of DDoS attack-detection mechanisms, leading to more precise and effective threat mitigation strategies.

**TABLE 3.** Selected features of the dataset.

| S. No | Features used along with description |
|---|---|
| 1 | **Packet Count per Flow:** The average number of packets transmitted per communication stream. |
| 2 | **Duration (in minutes):** The duration of each communication stream in minutes. |
| 3 | **Source IP:** The IP address of the sender. |
| 4 | **Port Bandwidth:** The maximum data transfer rate supported by the network port. |
| 5 | **Total Duration:** The total duration of all communication streams. |
| 6 | **Destination IP:** The IP address of the receiver. |
| 7 | **Packet Rate:** The rate at which packets are transmitted per unit time. |
| 8 | **Total Number of Flows:** The count of distinct communication streams within the network. |
| 9 | **Number of Packet_in Messages:** The count of messages indicating packets received by the network switch. |
| 10 | **Byte Count per Flow:** The average number of bytes transmitted per communication stream. |
| 11 | **Port Number:** The identifier for the network port. |
| 12 | **Duration (in seconds):** The duration of each communication stream in seconds. |
| 13 | **Packet Count:** The total number of packets transmitted. |
| 14 | **Transmitted Bytes (txbytes):** The total number of bytes transmitted. |
| 15 | **Byte Count:** The total number of bytes transmitted. |
| 16 | **Received Bytes (rxbytes):** The total number of bytes received. |

### B. EVALUATION METRICS FOR ML ALGORITHMS IN DDoS ATTACK MITIGATION

In this section, we describe the evaluation parameters used to assess the effectiveness of ML algorithms in mitigating SDN against DDoS attacks. Each ML model was evaluated based on key metrics, including accuracy, precision, recall, and F1 score. Understanding the parameters of the confusion matrix is essential for evaluating the performance of the ML algorithms. A high value for true positives and true negatives, along with a low value for false positives and negatives, is crucial for an effective intrusion detection system (IDS) in

SDN environments. The definitions of each parameter of the confusion matrix are as follows.

- **True positive state (Tpst):** A true positive represents instances in which the model correctly predicts the positive class when the actual class is positive. In the context of SDN DDoS detection, a true positive occurs when the ML algorithm correctly identifies a DDoS attack during the actual attack instances.
- **True negative state (Tnst):** A true negative indicates instances in which the model correctly predicts the negative class when the actual class is negative. For example, in SDN DDoS detection, a true negative occurs when the ML algorithm correctly identifies normal network traffic as non-malicious during regular operation.
- **False-positive state (Fpst):** False-positives occur when the model incorrectly predicts a positive class when the actual class is negative. For instance, in SDN DDoS detection, a false positive occurs when the ML algorithm mistakenly identifies normal network traffic as malicious, thereby triggering an unnecessary response or an alarm.
- **False-negative state (Fnst):** False-negatives represent instances in which the model incorrectly predicts the negative class when the actual class is positive. For example, in SDN DDoS detection, a false negative occurs when the ML algorithm fails to identify a genuine DDoS attack, potentially allowing malicious traffic to remain undetected.

An IDS plays a critical role in achieving high levels for these performance parameters. By continuously monitoring network traffic and applying ML algorithms, an IDS can accurately detect and classify suspicious activities, thereby enabling prompt responses to potential DDoS attacks. In the confusion matrix, *Tpst* represents correctly classified positive instances, *Tnst* represents correctly classified negative instances, *Fpst* indicates negative instances incorrectly classified as positive, and *Fnst* represents positive instances incorrectly classified as negative instances.

Accuracy (ACC) measures the ratio of correctly predicted instances to the total number of instances in the dataset. This provided an overall assessment of the accuracy of the model. ACC is crucial in evaluating the overall effectiveness of an ML model in correctly identifying both attack and normal instances. The high accuracy indicates that the model makes fewer mistakes in classifying instances, leading to better detection and mitigation of DDoS attacks. The mathematical equations are as follows.

$$ACC = \left(T_{pst} + T_{nst}\right) / T_{pst} + T_{nst} + F_{pst} + F_{nst} \quad (8)$$

Precision (PRC) measures the ratio of correctly predicted positive observations (*Tpst*) to total predicted positive observations (*Tpst* + *Fpst*). This indicates the ability of the model to accurately classify positive instances. PRC is essential for evaluating the reliability of the ML model in correctly identifying DDoS attacks. A high-precision score signifies that the model has a low false-positive rate, minimizing the chances of misclassifying normal instances as attacks. The formula for PRC is as follows:

$$PRC = T_{pst} / \left(T_{pst} + F_{pst}\right) \quad (9)$$

Recall (RCL), also known as sensitivity, measures the ratio of correctly predicted positive observations (*Tpst*) to all observations in the actual positive class (*Tpst* + *Fnst*). This highlights the ability of the model to correctly identify positive instances. RCL is critical for assessing the ability of the model to capture all instances of DDoS attacks without missing any instances. A high recall score indicates that the model has a low false-negative rate, ensuring that it detects the majority of DDoS attacks present in network traffic. The RCL equation is as follows:

$$RCL = T_{pst} / \left(T_{pst} + F_{nst}\right) \quad (10)$$

The F1 score (F1) is the harmonic mean of the PRC and RCL and provides a balanced measure of the model's performance. It combines precision and recall into a single metric, thereby offering a comprehensive evaluation of the effectiveness of the model. The *F1* score is crucial for assessing the overall performance of the ML model for DDoS attack detection. It accounts for both false positives and negatives, making it a reliable indicator of a model's ability to maintain a balance between precision and recall. The equation below represents the *F1* score:

$$F1 = 2 * (PRC * RCL)/(PRC + RCL) \quad (11)$$

## VII. RESULT AND DISCUSSION

The threat of DDoS attacks is growing in the online environment; therefore, effective defense mechanisms are crucial. Numerous studies have examined the application of ML techniques to counter these attacks by leveraging diverse datasets to improve the detection accuracy. However, previous studies [19], [23], [30], [32], [33], [34] generated datasets without adequately explaining their unique characteristics. This study addressed this gap by introducing a transparent and publicly available dataset. By meticulously documenting the dataset details, we provided researchers with a clear foundation for experimentation and validation. Using this dataset, we evaluated the five ML models ( Table 4) based on their ACC, PRC, REC, and F1 scores. The result analysis identifies random forest as the most effective model for DDoS attack detection.

Figure 11 shows the distribution of benign and malicious instances across the different traffic classes (ICMP, UDP, and TCP) in the dataset. This visualization is crucial for comprehending the dataset because it provides insights into the relative prevalence of benign and malicious traffic within each traffic class. By examining Figure 11, we observe the following.

- **Traffic Class Distribution:** The horizontal bars represent the different traffic classes, with each bar segmented

**TABLE 4.** Performance evaluation metrics for five supervised learning algorithms using the proposed dataset.

| Dataset | Algorithm | ACC | PRC | RCL | F1-Score |
|---|---|---|---|---|---|
| Proposed Dataset | LR | 84.31% | 83.93% | 84.59% | 84.26% |
| | SVM | 96.52% | 95.75% | 95.75% | 95.75% |
| | RF | 98.97% | 98.33% | 96.37% | 97.34% |
| | KNN | 97.60% | 99.21% | 94.72% | 96.91% |
| | XGBoost | 92.36% | 92.29% | 90.57% | 91.42% |

into two parts: one for benign instances and the other for malicious instances. This breakdown allows for a clear understanding of how each traffic class contributes to the overall dataset.

- **Relative Importance of Traffic Classes:** The length of each bar indicates the total instance count for the corresponding traffic class. A comparison of the lengths of bars across traffic classes helps understand which traffic classes are more prevalent in the dataset.
- **Benign versus Malicious Composition:** The Segmentation of bars into benign and malicious instances enables a direct comparison of their proportions within each traffic class. This information is valuable for understanding the prevalence of malicious activities in different types of network traffic.

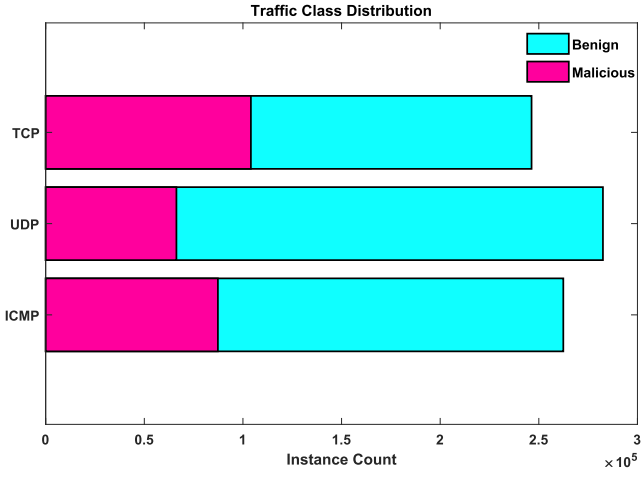


**FIGURE 11.** Performance Metrics Across Traffic Categories for Each Traffic Instance.

Table 5 presents a summary of the network traffic instances, which are categorized into two main categories: ambiguous traffic and network protocols.

- **Category I (Ambiguous Traffic):** This category includes traffic instances classified as both normal and suspicious. The Instances Count column reveals that there are 523,195 instances classified as normal and 523,380 instances classified as suspicious. The Total Instance column shows the cumulative count, where the instance counts for Normal and Suspicious instances are combined, resulting in a total of 1,048,575 instances.

**TABLE 5.** Summary of network traffic instances.

| Category | Traffic | Instances |
|---|---|---|
| Category I: Ambiguous Traffic | Normal | 523,195 |
| | Suspicious | 523,380 |
| Category II: Network Protocol | TCP | 350,358 |
| | UDP | 348,790 |
| | ICMP | 349,727 |

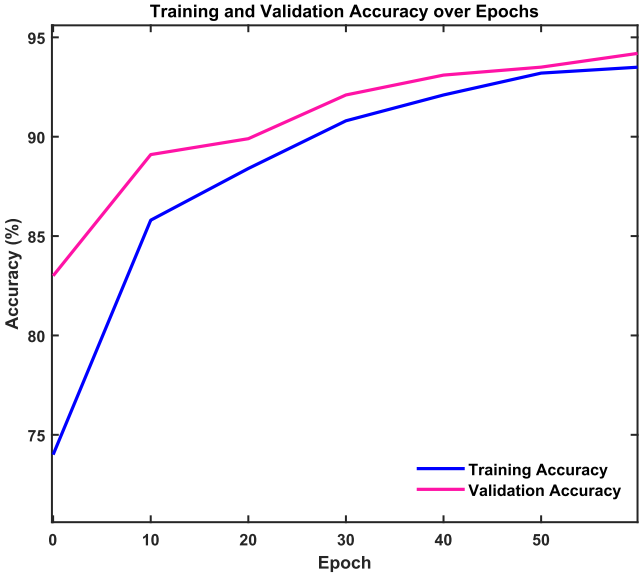


**FIGURE 12.** Training and validation accuracies of KNN model over epochs.

- **Category II (Network Protocol):** This category focuses on instances categorized by network protocols such as TCP, UDP, and ICMP. The Instances Count column provides the count for each specific protocol, with 350,358 instances attributed to TCP, 348,790 to UDP, and 349,727 to ICMP. The Total Instance column shows the aggregate count for all protocols, resulting in 1,048,575 instances when considering TCP, UDP, and ICMP collectively.

Figure 12 illustrates how well our KNN model learned from the data over time. The blue line shows how the accuracy of the model improved as it learned from the training data, starting at 74% and reaching 93.5% after 60 epochs. This indicates that our model adapts quickly to patterns in the data. Similarly, the red line represents the accuracy of the model for new, unseen data (validation accuracy). It started at 83% and steadily increased to 94.2% over the same 60 epochs. The close alignment between the training and validation accuracies suggests that our KNN model effectively learns from the data without overfitting or underfitting, thereby demonstrating its ability to generalize well to new instances.

### A. MODEL PERFORMANCE ANALYSIS AND COMPARISON

Figure 13(a) shows a comprehensive analysis of the accuracies of five distinct supervised models. The varying degrees of accuracy among the models provided insights into their efficacy in the detection task. Random Forest emerged

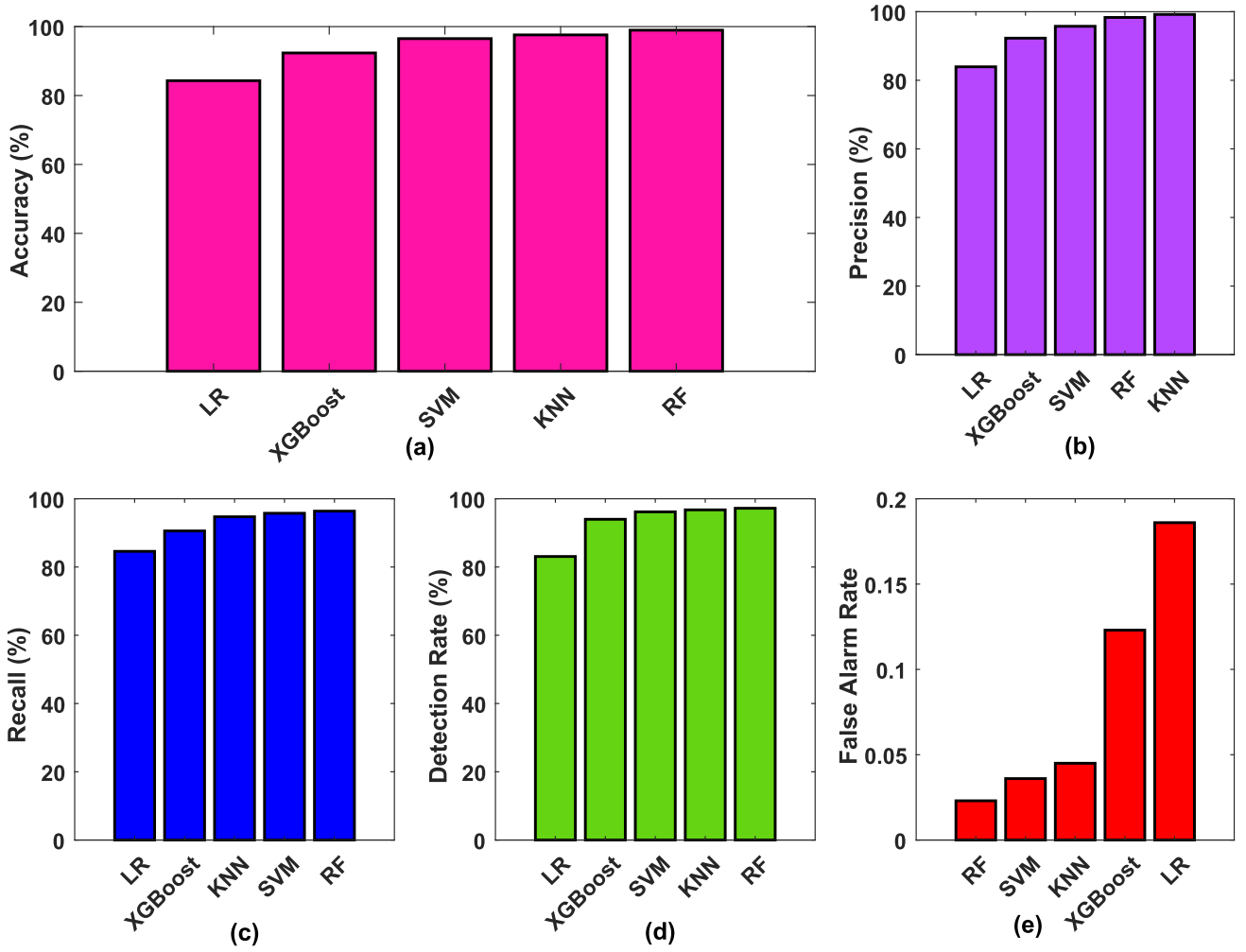


**FIGURE 13.** Comparative Analysis of Algorithm Performance Metrics. Subfigures (a)-(e) depict the performance comparison of different algorithms across key metrics: (a) accuracy, (b) precision, (c) recall, (d) Detection Rate, and (e) False Alarm Rate. Each subfigure presents the metrics sorted from the lowest to the highest values, offering valuable insights into the effectiveness of algorithms in detection and classification tasks.

as a standout performer, with an accuracy of 98.97%. This exceptional accuracy can be attributed to the RF ensemble learning approach that combines multiple decision trees to produce robust predictions. Unlike single decision trees, RF mitigates the risk of overfitting by aggregating the predictions of individual trees, resulting in a more generalized and reliable model. Furthermore, RF excels in capturing complex data interactions, making it particularly well-suited for tasks involving high-dimensional data or nonlinear relationships. Its superior performance underscores the effectiveness of ensemble learning in classification tasks, and positions RF as the model of choice for accurate and reliable predictions. Logistic Regression exhibited the lowest accuracy among the models, with a score of 84.31%. LR simplicity and interpretability make it a popular choice for binary-classification tasks. However, its linear nature may limit its ability to capture complex relationships present in the data, resulting in lower predictive accuracy. LR operates under the assumption of linearity between the independent variables and log-odds of the dependent variable, which may not hold true in datasets with intricate patterns or nonlinear relationships. Despite its shortcomings in accuracy, LR serves as a foundational baseline for comparison and remains a valuable tool in situations where model interpretability is paramount. Next, we explain these models as follows.

- LR is the simplest model of the ensemble, as shown in Figure 13(a). Despite its simplicity, LR provides a foundational understanding of the linear relationships inherent in data. However, its accuracy of 84.31% suggests constraints in capturing the nuanced patterns within the dataset. Although LR serves as a valuable baseline, more sophisticated models may offer superior predictive efficacy.
- The SVM, as illustrated in Figure 13(a), exhibited a formidable accuracy of 96.52%. By leveraging optimal

hyperplanes to delineate between classes, the SVM excels in discerning intricate data patterns. However, its performance may be contingent on kernel selection and parameter tuning (Figure 13(a)). Despite these considerations, the SVM remains a robust model for classification tasks, as evidenced by its competitive accuracy.

- As depicted in Figure 13(a), the RF emerged as a standout performer with an accuracy of 98.97%. Its ensemble approach, aggregating multiple decision trees, enables RF to effectively capture complex data interactions. Moreover, the resilience of the RF to overfitting bolsters its reliability in real-world applications, solidifying its position as the top-performing model.
- The KNN demonstrated a reasonable accuracy of 97.60%. By classifying objects based on the majority class of their nearest neighbors, KNN adeptly discerns intricate data patterns. However, its efficacy may be contingent on the parameter settings, affecting its performance in certain scenarios (Figure 13(a)). Nevertheless, the KNN remains a versatile and intuitive model for classification tasks.
- XGBoost yielded an accuracy of 92.36%. Owing to its scalability and performance, XGBoost sequentially enhances the predictive accuracy of weak learners (Figure 13(a)). Although XGBoost offers a competitive performance, optimal results require meticulous parameter tuning to mitigate overfitting and enhance model robustness.

Figure 13(e) shows the false alarm rate (FAR) associated with each model. RF boasts the lowest FAR (0.023), followed by SVM (0.036) and KNN (0.045) (Figure 13(e)). Conversely, LR and XGBoost exhibited comparatively higher FAR values, indicating a heightened propensity for false alarms (Figure 1(e)). The FAR analysis underscores the importance of model reliability and accuracy in minimizing false positives in detection tasks.

### B. ANALYSIS OF BANDWIDTH USAGE DURING ATTACK AND MITIGATION

Figure 14 illustrates the bandwidth usage over time for different ports during an orchestrated cyberattack, and the subsequent implementation of mitigation measures. Each line on the graph represents the bandwidth utilization of a specific port over a 300-second timeframe. During the attack phase depicted in the graph, there is a discernible increase in the bandwidth usage across all ports. Port 1 experienced a notable surge in bandwidth, reaching a peak of approximately 3050 Kbps. Similarly, Ports 2 and 3 also exhibited elevated bandwidth consumption, with peaks of approximately 950 and 2100 Kbps, respectively. Following the implementation of mitigation measures, as indicated by the dashed lines in the graph, there is a visible reduction in the bandwidth usage for all ports. In particular, Port 1 demonstrated a significant decrease in bandwidth consumption, effectively mitigating the impact of the attack. Ports 2 and 3 also showed a decrease in bandwidth usage post-mitigation, albeit to a lesser extent. A comparison of bandwidth usage before and after mitigation highlights the effectiveness of the implemented strategies. The graph clearly illustrates how the mitigation measures succeeded in mitigating the impact of the attack, leading to lower bandwidth consumption across all the ports. The observed reduction in bandwidth usage post-mitigation underscores the effectiveness of the implemented strategies in mitigating the impact of cyberattacks. The significant decrease in bandwidth consumption for Port 1, in particular, indicates the successful containment of the effects of the attack on network performance. Furthermore, the analysis in Figure 14 provides valuable insights into the dynamics of network traffic during both attack and mitigation phases. This underscores the importance of proactive mitigation measures to safeguard network integrity and stability, particularly in the face of evolving cyberthreats. In conclusion, Figure 14 serves as a visual representation of the effectiveness of mitigation strategies for mitigating the impact of a simulated cyberattack on network performance. The graph reinforces the critical role of proactive measures in enhancing network security and resilience, thereby ensuring uninterrupted operation of critical infrastructure and services.

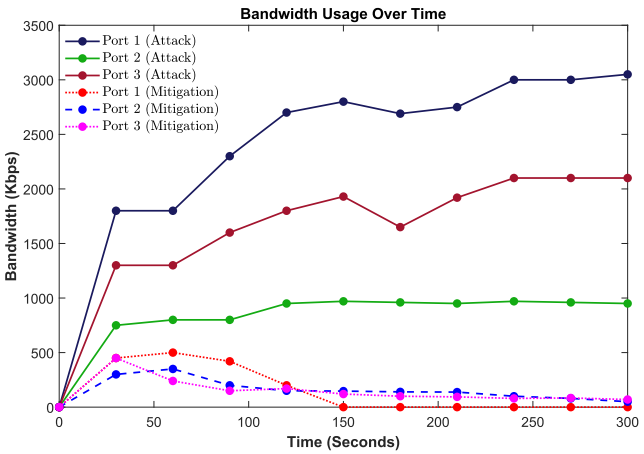


**FIGURE 14.** Bandwidth usage over time during attack and mitigation.

### C. COMPARATIVE ANALYSIS: NOVEL DATASET VS. CICDDoS2019

CICDDoS2019 contains both DDoS attacks and benign network traffic, making it a valuable resource for studying and comparing malicious and legitimate network behaviors [49]. This duality enables researchers to develop and evaluate intrusion detection systems effectively. Owing to its comprehensive features, the CICDDoS2019 dataset has been widely used in cybersecurity research. Researchers have relied on this to test the performance of various machine-learning models to distinguish between benign and malicious network activities. For evaluation purposes,

**TABLE 6. Performance comparison of the RF model using novel dataset and CICDDoS2019.**

| Dataset | Year | ACC | PRC | RCL | F1-Score |
|---|---|---|---|---|---|
| Proposed | 2024 | 98.97% | 98.33% | 96.37% | 97.34% |
| CICDDoS2019 | 2019 | 97.62% | 98.70% | 97.73% | 97.69% |

evaluation metrics such as accuracy, precision, recall, and F1-score were used to compare the performance of the models (see Table 6). The RF model achieved notable results when applied to CICDDoS2019, with an accuracy of 97.62%, precision of 98.70%, recall of 97.73%, and F1-score of 97.69%. Comparatively, on the novel dataset, the RF model exhibited superior performance, with an accuracy of 98.97%, precision of 98.33%, recall of 96.37%, and F1-score of 97.34%. These findings underscore the importance of custom datasets in cybersecurity research, offering enhanced accuracy and robustness compared with CICDDoS2019. This novel dataset provides researchers with a valuable resource for advancing intrusion detection methodologies and exploring advanced techniques such as deep learning.

In summary, to assess the impact of feature selection on model performance, we conducted an ablation study comparing model accuracy before and after applying feature selection techniques. The results showed that models such as RF, SVM, and KNN significantly improved in accuracy, precision, and recall when using Chi-square and recursive feature elimination (RFE) techniques. This analysis demonstrates that feature selection not only enhances the model's ability to detect DDoS attacks but also reduces overfitting risks by eliminating irrelevant features. By minimizing noise in the data, the selected features enable the models to generalize better to unseen traffic patterns.

### D. COMPARATIVE ANALYSIS OF NOVEL AND ESTABLISHED DATASETS

In addition to evaluating the performance of the proposed dataset, a comparison was conducted with the CICDDoS2019 dataset. This comparison was selected because CICDDoS2019 encompasses more recent and diverse network traffic, which makes it particularly relevant for contemporary DDoS detection. In contrast, although NSL-KDD and KDD datasets have been extensively used in previous studies, they present limitations that reduce their applicability to current threat landscapes. Specifically, the NSL-KDD dataset, which is derived from the older KDD Cup 1999 dataset, does not accurately reflect modern network traffic patterns or attack strategies. Both KDD and NSL-KDD are static datasets, meaning that they do not evolve over time. As network traffic patterns change and new attack methods emerge, these datasets may fail to provide relevant training data for modern IDS, potentially leading to decreased performance in real-world applications. Additionally, they lack the complex and varied attack vectors found in current networks, which are more effectively represented in CICDDoS2019. Previous research has investigated ML approaches using NSL-KDD and KDD datasets [19], [27]. These studies indicate that while high accuracy can be achieved, the results often do not generalize well to realistic, modern scenarios. This underscores the need for updated datasets that address the challenges of detecting sophisticated DDoS attacks in contemporary SDN environments. Consequently, this study focuses on developing and evaluating a new dataset that more accurately reflects current threats. The findings demonstrate that this dataset offers superior detection capabilities for a broad range of DDoS attack scenarios compared with CICDDoS2019.

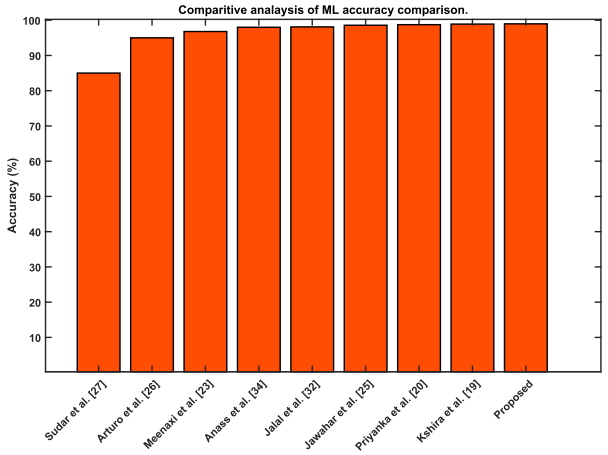


**FIGURE 15. Accuracy comparison with other studies.**

### E. COMPARISON OF DDoS ATTACK DETECTION METHODS USING MACHINE LEARNING IN SDN SECURITY

This study presents a significant advancement in DDoS attack detection in SDN environments compared to existing studies (see in Table 7). Unlike previous studies, which often lack transparency and restrict access to their datasets, a novel dataset is meticulously crafted to simulate realistic DDoS attack scenarios. Importantly, this dataset is publicly available and facilitates transparency, reproducibility, and collaboration within the research community. For instance, consider the study by Jawahar et al. [25], which achieved an accuracy of 98.57% using an ANN. While the incorporation of multiple public datasets enhances the validity of their findings, the approach in this study surpasses theirs by providing a custom dataset that allows for a more targeted and comprehensive evaluation of DDoS detection methods in SDN. Furthermore, the results demonstrated superior performance compared to other studies. As illustrated in Figure 15, the proposed models achieved an accuracy of 98.97%, which is the highest among all the compared methods. This exceptional accuracy underscores the effectiveness of the approach for accurately detecting and mitigating DDoS attacks in SDN environments.

Due to resource constraints, precise computational cost metrics were not measured in this study. Future work

**TABLE 7. Comparison with other studies.**

| Reference | Dataset and Method Characteristics | ACC |
|---|---|---|
| *Kshira et al.* [19] | Dataset I: Not publicly available and discussion is limited. | 98.90% |
| *Priyanka et al.* [20] | The limited diversity of attack scenarios for testing the dataset may restrict the performance evaluation of the model. | 98.74% |
| *Meenaxi et al.* [23] | The study lacks specific dataset details and public availability. | 96.79% |
| *Jawahar et al.* [25] | Incorporating multiple public datasets enhanced the validity of the findings. | 98.57% |
| *Arturo et al.* [26] | Benchmarking against multiple datasets assesses the effectiveness across various environments. | 95% |
| *Sudar et al.* [27] | Relying solely on one dataset for training and testing may introduce limitations and bias. | 85% |
| *Jalal et al.* [32] | Lack of dataset details hinders replication & verification. | 98.1% |
| *Anass et al.* [34] | The paper does not provide specific details about the dataset itself, such as its size, composition, or source. | 98% |
| **Proposed** | A novel custom dataset tailored for DDoS detection in SDN is publicly available for transparency and reproducibility. Benchmarking against the CICDDoS2019 dataset validated its robustness. It achieves high accuracy (98.97%), scalability, and adaptability to evolving threats. | 98.97% |

should focus on detailed evaluations of the computational requirements of these models to better assess their real-time applicability in SDN environments. Furthermore, studies such as those by Sahoo et al. [19] and Kujur and Patel [20], reported high accuracies, the limited availability and diversity of their datasets could potentially hinder the generalizability and robustness of their models. In contrast, the publicly available dataset encompasses a wide range of attack scenarios, ensuring comprehensive evaluation and validation of the proposed methods. Moreover, most studies in the field have traditionally relied on PCA for feature selection, whereas the adoption of the chi-squared (Chi2) algorithm in this study is crucial. The Chi2 algorithm is specifically designed for categorical feature selection, making it well-suited for the dataset, which comprises various network traffic attributes. Unlike PCA, which assumes linear relationships between variables, Chi2 identifies the most significant attributes based on their independence from the target variable, thereby better capturing the nonlinear relationships inherent in network traffic data. Additionally, this study addressed the limitations identified in previous studies, such as the lack of dataset details and diversity, reliance on a single dataset for evaluation, and limited transparency. By providing a detailed and publicly available dataset, along with achieving exceptional accuracy in DDoS detection, this study establishes a benchmark for future research on SDN security. Further details and comparisons are presented in Table 7, highlighting the superiority and value of this contribution in the SDN domain.

## VIII. CONCLUSION

Software-Defined Networking (SDN) is poised as a transformative technology architecture with the potential to revolutionize network management. However, this innovation has introduced new security challenges. The centralized control inherent in SDN exposes networks to vulnerabilities, including DDoS attacks, which can have devastating effects on network performance and availability. The findings of this study are of significant importance. By addressing the security challenges inherent in SDN environments, this study contributes to filling critical gaps in knowledge and tackling existing issues. Through the development of a novel dataset and comprehensive evaluation of five machine learning techniques, along with benchmarking against established datasets such as CICDDoS2019, valuable insights into effective DDoS detection and mitigation strategies within SDN are provided. The generated dataset serves as a valuable resource for evaluating intrusion detection techniques and anomaly detection algorithms in SDN. This study yielded several key findings that supported its objectives. A tailored dataset was successfully developed to simulate realistic DDoS attack scenarios, providing researchers with valuable resources for evaluating network security algorithms. Furthermore, the evaluation of machine learning techniques, including logistic regression, support vector machine, random forest, K-nearest neighbor, and XGBoost, sheds light on their effectiveness in DDoS attack classification in SDN environments. Benchmarking against established datasets, such as CICDDoS2019, further validated the efficacy of the approach. The implications of the findings extend to real-world applications, informing the decision-making processes of network administrators and cybersecurity professionals. By identifying effective DDoS detection and mitigation strategies in SDN environments, this research enhances the resilience of SDN infrastructures against DDoS attacks and safeguards network performance and availability. To acknowledge the significance of this study, it is important to address its limitations. One potential constraint is the scope of the dataset, which may not encompass all the possible DDoS attack scenarios. Moreover, the performance of the ML approaches could be affected by factors such as the size and composition of the dataset..

Future research should explore deep learning models, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. These models can further enhance detection accuracy by capturing complex patterns and temporal dependencies in network traffic. Additionally, anomaly detection techniques, such as autoencoders and isolation forests, could be integrated with supervised learning models to identify previously unseen attack patterns,

**TABLE 8.** List of abbreviations.

| Acronyms | Description |
|---|---|
| ACC | Accuracy |
| ANN | Artificial Neural Network |
| BPF | Byte Per Flow |
| BLR | Bayesian Logistic Regression |
| DIP | Destination Internet Protocol |
| DDoS | Distributed Denial of Service |
| DT | Decision Tree |
| DTM | Data Transfer Metrics |
| FAR | False Alarm Rate |
| GNB | Gaussian Naive Bayes |
| GA | Genetic Algorithm |
| GB | Gradient Boosting |
| GN | Nearest Centroid |
| IDS | Intrusion Detection Systems |
| KNN | K-Nearest Neighbor |
| LR | Logistic Regression |
| MLP | Multilayer Perceptron |
| ML | Machine Learning |
| NC | Nearest Centroid |
| NB | Naive Bayes |
| ONOS | Open Network Operating System |
| PCA | Principal Component Analysis |
| PPM | Packet Per Message |
| PR | Precision |
| RCL | Recall |
| RF | Random Forest |
| SGD | Stochastic Gradient Descent |
| SIP | Source Internet Protocol |
| SVM | Support Vector Machine |
| TDC | Total Duration of Communication |
| XGBoost | Xtreme Gradient Boosting |

offering a more holistic approach to DDoS mitigation. Future work will also expand the scope of attack scenarios to include more sophisticated DDoS vectors, such as amplification attacks (e.g., DNS and NTP amplification), application-layer attacks (e.g., HTTP flood), and botnet-driven multi-vector attacks. This enhances the generalizability of the models by ensuring their robustness across diverse real-world scenarios.

Another promising direction involves investigating the scalability of detection systems in large-scale SDN environments. Research into how different network topologies affect detection performance could provide insights into optimizing DDoS defense mechanisms. Additionally, real-world traffic data from sources such as MAWI or CAIDA are integrated to ensure that the models are trained and validated under live network conditions, further enhancing their applicability in practical deployments. To further strengthen this aspect, we aim to collaborate with industry partners to access real SDN traffic, providing critical insights into the operational performance of our framework and identifying deployment challenges.

Finally, examining the integration of hybrid detection frameworks that combine signature-based detection with ML techniques could offer robust real-time defense strategies. By pursuing these avenues, future studies can continue to advance SDN security, ensuring that networks remain resilient to evolving DDoS threats.

## APPENDIX

Table 8 provides references for the acronyms used in this study.

## REFERENCES

[1] V. Balasubramanian, M. Aloqaily, and M. Reisslein, "An SDN architecture for time sensitive industrial IoT," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107739.
[2] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)," *Comput. Netw.*, vol. 112, pp. 279–293, Jan. 2017.
[3] A. Abuarqoub, "A review of the control plane scalability approaches in software defined networking," *Future Internet*, vol. 12, no. 3, p. 49, Mar. 2020.
[4] J. Miguel-Alonso, "A research review of OpenFlow for datacenter networking," *IEEE Access*, vol. 11, pp. 770–786, 2023.
[5] A. SR, K. Mahadev, S. Prasad, S. Eswaran, and P. Honnavalli, "OpenDaylight as software defined networking controller: Shortcomings and possible solutions," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Jul. 2022, pp. 1–6.
[6] Z. A. Bhuiyan, S. Islam, M. M. Islam, A. B. M. A. Ullah, F. Naz, and M. S. Rahman, "On the (in)security of the control plane of SDN architecture: A survey," *IEEE Access*, vol. 11, pp. 91550–91582, 2023.
[7] S. Khorsandroo, A. G. Sánchez, A. S. Tosun, J. Arco, and R. Doriguzzi-Corin, "Hybrid SDN evolution: A comprehensive survey of the state-of-the-art," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 107981.
[8] A. L. Aliyu, A. Aneiba, M. Patwary, and P. Bull, "A trust management framework for software defined network (SDN) controller and network applications," *Comput. Netw.*, vol. 181, Nov. 2020, Art. no. 107421.
[9] M. T. Islam, N. Islam, and M. A. Refat, "Node to node performance evaluation through RYU SDN controller," *Wireless Pers. Commun.*, vol. 112, no. 1, pp. 555–570, May 2020.
[10] H. Leqing, "How to realize the smooth transition from traditional network architecture to SDN," in *Proc. 5th Int. Conf. Mech., Control Comput. Eng. (ICMCCE)*, Dec. 2020, pp. 1948–1952.
[11] H. Facchini, S. Perez, R. Blanchet, B. Roberti, and R. Azcarate, "Experimental performance contrast between SDN and traditional networks," in *Proc. IEEE CHILEAN Conf. Electr., Electron. Eng., Inf. Commun. Technol. (CHILECON)*, Dec. 2021, pp. 1–6.
[12] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, and Y. Deng, "A new framework for DDoS attack detection and defense in SDN environment," *IEEE Access*, vol. 8, pp. 161908–161919, 2020.
[13] R. K. Batchu and H. Seetha, "On improving the performance of DDoS attack detection system," *Microprocessors Microsyst.*, vol. 93, Sep. 2022, Art. no. 104571.
[14] H. S. Abdulkarem and A. Dawod, "DDoS attack detection and mitigation at SDN data plane layer," in *Proc. 2nd Global Power, Energy Commun. Conf. (GPECOM)*, Oct. 2020, pp. 322–326.
[15] K.-Y. Chen, S. Liu, Y. Xu, I. K. Siddhrau, S. Zhou, Z. Guo, and H. J. Chao, "SDNShield: NFV-based defense framework against DDoS attacks on SDN control plane," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, pp. 1–17, Feb. 2022.
[16] L. F. Eliyan and R. Di Pietro, "DoS and DDoS attacks in software defined networks: A survey of existing solutions and research challenges," *Future Gener. Comput. Syst.*, vol. 122, pp. 149–171, Sep. 2021.
[17] M. Revathi, V. V. Ramalingam, and B. Amutha, "A machine learning based detection and mitigation of the DDOS attack by using SDN controller framework," *Wireless Pers. Commun.*, vol. 127, no. 3, pp. 2417–2441, Dec. 2022.

[18] A. Hirsi, L. Audah, and A. Salh, "SDN-DDoS traffic dataset," Mendeley Data, V1, 2024, doi: 10.17632/b7vw628825.1. [Online]. Available: https://data.mendeley.com/datasets/b7vw628825/1

[19] K. S. Sahoo, B. K. Tripathy, K. Naik, S. Ramasubbareddy, B. Balusamy, M. Khari, and D. Burgos, "An evolutionary SVM model for DDOS attack detection in software defined networks," *IEEE Access*, vol. 8, pp. 132502–132513, 2020.

[20] P. Kujur and S. Patel, "Comparison of various ML approaches for detection of DDoS attacks in SDN," in *Proc. IEEE 15th Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Dec. 2023, pp. 245–249.

[21] B. Isyaku, K. B. A. Bakar, M. S. Ali, and M. N. Yusuf, "Performance comparison of machine learning classifiers for DDoS detection and mitigation on software defined networks," in *Proc. IEEE Int. Conf. Automat. Control Intell. Syst. (I2CACIS)*, Jun. 2023, pp. 69–74.

[22] O. Rahman, M. A. G. Quraishi, and C.-H. Lung, "DDoS attacks detection and mitigation in SDN using machine learning," in *Proc. IEEE World Congr. Services (SERVICES)*, vol. 2642, Jul. 2019, pp. 184–189.

[23] M. M. Raikar, M. M. Mulla, M. M. Mulla, N. S. Shetti, and M. Karanandi, "Data traffic classification in software defined networks (SDN) using supervised-learning," *Proc. Comput. Sci.*, vol. 171, pp. 2750–2759, Jan. 2020.

[24] N. Ashodia and K. Makadiya, "Detection of DDoS attacks in SDN using machine learning," in *Proc. Int. Conf. Electron. Renew. Syst. (ICEARS)*, Mar. 2022, pp. 1322–1327.

[25] A. Jawahar, P. Kaythry, V. C. Kumar, R. Vinu, R. Amrish, K. Bavapriyan, and V. Gopinaath, "DDoS mitigation using blockchain and machine learning techniques," *Multimedia Tools Appl.*, vol. 83, no. 21, pp. 60265–60278, Jan. 2024.

[26] J. A. Pérez-Díaz, I. A. Valdovinos, K. R. Choo, and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020.

[27] K. M. Sudar, M. Beulah, P. Deepalakshmi, P. Nagaraj, and P. Chinnasamy, "Detection of distributed denial of service attacks in SDN using machine learning techniques," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Jan. 2021, pp. 1–5.

[28] A. Sahbi, F. Jaidi, and A. Bouhoula, "Machine learning algorithms for enhancing intrusion detection within SDN/NFV," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2023, pp. 602–607.

[29] S. Wang, J. F. Balarezo, K. G. Chavez, A. Al-Hourani, S. Kandeepan, M. R. Asghar, and G. Russello, "Detecting flooding DDoS attacks in software defined networks using supervised learning techniques," *Eng. Sci. Technol., Int. J.*, vol. 35, Nov. 2022, Art. no. 101176.

[30] U. H. Garba, A. N. Toosi, M. F. Pasha, and S. Khan, "SDN-based detection and mitigation of DDoS attacks on smart homes," *Comput. Commun.*, vol. 221, pp. 29–41, May 2024.

[31] W. I. Khedr, A. E. Gouda, and E. R. Mohamed, "FMDADM: A multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks," *IEEE Access*, vol. 11, pp. 28934–28954, 2023.

[32] J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106432.

[33] G. O. Anyanwu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "RBF-SVM kernel-based model for detecting DDoS attacks in SDN integrated vehicular network," *Ad Hoc Netw.*, vol. 140, Mar. 2023, Art. no. 103026.

[34] A. Sebbar and K. Zkik, "Enhancing resilience against DDoS attacks in SDN-based supply chain networks using machine learning," in *Proc. 9th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Jul. 2023, pp. 230–234.

[35] B. Subba, S. Biswas, and S. Karmakar, "Intrusion detection systems using linear discriminant analysis and logistic regression," in *Proc. Annu. IEEE India Conf. (INDICON)*, Dec. 2015, pp. 1–6.

[36] Z. Akram, M. Majid, and S. Habib, "A systematic literature review: Usage of logistic regression for malware detection," in *Proc. Int. Conf. Innov. Comput. (ICIC)*, Nov. 2021, pp. 1–8.

[37] S. Yadav and S. Selvakumar, "Detection of application layer DDoS attack by modeling user behavior using logistic regression," in *Proc. 4th Int. Conf. Rel., Infocom Technol. Optim. (ICRITO) (Trends Future Directions)*, Sep. 2015, pp. 1–6.

[38] Z. Long and W. Jinsong, "A hybrid method of entropy and SSAE-SVM based DDoS detection and mitigation mechanism in SDN," *Comput. Secur.*, vol. 115, Apr. 2022, Art. no. 102604.

[39] V. Sonai and I. Bharathi, "Packet classification using improved random forest algorithm," in *Proc. Int. Conf. Mach. Learn., Deep Learn. Comput. Intell. Wireless Commun.* Cham, Switzerland: Springer, 2023, pp. 157–168.

[40] Y. Chen, J. Hou, Q. Li, and H. Long, "DDoS attack detection based on random forest," in *Proc. IEEE Int. Conf. Prog. Informat. Comput. (PIC)*, Sep. 2020, pp. 328–334.

[41] S. Dong and M. Sarem, "DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks," *IEEE Access*, vol. 8, pp. 5039–5048, 2020.

[42] S. Kumar, G. Bansal, and V. S. Shekhawat, "A machine learning approach for traffic flow provisioning in software defined networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2020, pp. 602–607.

[43] A. Zainudin, L. A. C. Ahakonye, R. Akter, D.-S. Kim, and J.-M. Lee, "An efficient hybrid-DNN for DDoS detection and classification in software-defined IIoT networks," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8491–8504, May 2023.

[44] H. A. Alamri and V. Thayananthan, "Analysis of machine learning for securing software-defined networking," *Proc. Comput. Sci.*, vol. 194, pp. 229–236, Jan. 2021.

[45] A. Abhishta, R. Joosten, S. Dragomiretskiy, and L. J. M. Nieuwenhuis, "Impact of successful DDoS attacks on a major crypto-currency exchange," in *Proc. 27th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, Feb. 2019, pp. 379–384.

[46] S. Bhardwaj and S. N. Panda, "Performance evaluation using RYU SDN controller in software-defined networking environment," *Wireless Pers. Commun.*, vol. 122, no. 1, pp. 701–723, Jan. 2022.

[47] D. Dholakiya, T. Kshirsagar, and A. Nayak, "Survey of mininet challenges, opportunities, and application in software-defined network (SDN)," *Proc. Int. Conf. Inf. Commun. Technol. Intell. Syst.*, vol. 2, 2021, pp. 213–221.

[48] Z. R. Saputra Elsi, D. Stiawan, A. F. Oklilas, K. Susanto, Y. N. Kunang, M. Y. Idris, and R. Budiarto, "Feature selection using chi square to improve attack detection classification in IoT network: Work in progress," in *Proc. 9th Int. Conf. Electr. Eng., Comput. Sci. Informat. (EECSI)*, Oct. 2022, pp. 226–232.

[49] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8.

**ABDINASIR HIRSI** (Graduate Student Member, IEEE) received the B.S. degree in telecommunication engineering from Mohammad Ali Jinnah University (MAJU), Karachi, Pakistan, in 2019, and the M.S. degree in electrical engineering, specializing in communication engineering from Bahria University, Karachi, in 2021. He is currently pursuing the Ph.D. degree in electrical engineering with Universiti Tun Hussein Onn Malaysia (UTHM), Johor, Malaysia. He is currently a Graduate Research Assistant (GRA) with the Advance Telecommunication Research Center (ATRC), FKEE, UTHM. His research interests include software-defined networking (SDN) security, DDoS attack detection and mitigation, cybersecurity, and AI techniques, such as machine learning and deep learning techniques for network intrusion detection.

**LUKMAN AUDAH** (Member, IEEE) received the Bachelor of Engineering degree in telecommunications from Universiti Teknologi Malaysia, in 2005, and the M.Sc. degree in communication networks and software and the Ph.D. degree in electronic engineering from the University of Surrey, U.K. He is currently a Senior Lecturer with the Communication Engineering Department, Universiti Tun Hussein Onn Malaysia. His research interests include wireless and mobile communications, internet traffic engineering, network system management, data security, and satellite communications.

**ADEB SALH** (Member, IEEE) received the bachelor's degree in electrical and electronic engineering from IBB University, Yemen, in 2007, and the master's and Ph.D. degrees in electrical and electronic engineering from University Tun Hussein Onn Malaysia, in 2015 and 2020, respectively. From 2007 to 2012, he was a Lecturer Assistant with the Yareem Community College. From 2020 to 2023, he was a Postdoctoral Researcher with UTHM and UTM, respectively. Currently, he is an Assistant Professor with the Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman. His research interests include 5G, 6G wireless communications, massive MIMO, artificial intelligence (AI), and the Internet of Things (IoT).

**MOHAMMED A. ALHARTOMI** (Member, IEEE) received the Ph.D. degree in electronic and electrical engineering from Leeds University, U.K., in 2016. He is currently an Assistant Professor with the Department of Electrical Engineering, University of Tabuk. His research interests include wireless and mobile communications, signal processing, optical wireless systems design, and visible light communications.

**SALMAN AHMED** (Graduate Student Member, IEEE) received the bachelor's degree in electronic engineering from the Mehran University of Engineering and Technology Jamshoro, Pakistan, in 2015, and the Master of Engineering degree in industrial automation and control from the Quaid e Awam University of Science and Technology Nawabshah, Pakistan, in 2021. He is currently pursuing the Ph.D. degree with the Faculty of Electrical and Electronics Engineering, Universiti Tunn Hussein, Malaysia (UTHM). He is currently a Graduate Research Assistant (GRA) with the Faculty of Electrical and Electronics Engineering, UTHM. He was a Lecturer in electronics engineering with Sukkur IBA University, Pakistan, from 2017 to 2022. His research interests include network security, cryptographic, and resource-constrained Internet of Things devices.