ML-Based SDN DDoS Detection: A Machine Learning Enabled Intrusion Detection System for Software-Defined Networks

Naga Malleswararao P.
Full Time Research Scholar
School of Computer Science and Engineering (SCOPE)
VIT-AP University, Amaravati, Andhra Pradesh, India
Email: malleswararao.24phd7154@vitap.ac.in

Dr. S. Bharath Bhushan
Associate Professor Grade-1
School of Computer Science and Engineering (SCOPE)
VIT-AP University, Amaravati, Andhra Pradesh, India
Email: bharathbhushan.s@vitap.ac.in

Ireddi Rakshitha
Reg_No: 21BCE9144
SCOPE,VIT-AP University,
Amaravati,
Andhra Pradesh, India
Email: ireddirakshitha@gmail.com

Yaswanth Devavarapu
Reg_No: 2100030719
Koneru Lakshmaiah Education Foundation
Deemed University, Guntur,
Andhra Pradesh, India
Email: yaswanth169choudary@gmail.com

Bhuvansai Hari
Student ID: 210303124486
Parul Institute of Engineering and Technology
Parul University, Limda, Waghodia Taluk
Vadodara District, Gujarat – 391760, India
Email: bhuvannaidu2524@gmail.com

Ganga Bharath
Regd_No: 23BCE8244
VIT-AP University
Amaravati, Andhra Pradesh, India
Email: bharathganga7@gmail.com

***Abstract*—SDN has emerged as a revolutionary networking paradigm, which decouples the control plane from the data plane, fostering centralized control and enhanced programmability to achieve better network management. This architectural flexibility, though greatly enhancing scalability and improving operational efficiency, exposes new attack surfaces that are being increasingly exploited by adversaries, especially targeting the centralized SDN controller [3]. Of the many threats, a DDoS attack—particularly a SYN flood—represents a serious risk in the SDN environment due to its ability to overwhelm the controller with excessive flow requests, thereby causing service disruption and degraded network performance. The control plane from the data plane, fostering centralized control and enhanced programmability to achieve better network management. This architectural flexibility, though greatly enhancing scalability and improving operational efficiency, exposes new attack surfaces that are being increasingly exploited by adversaries, especially targeting the centralized SDN controller [4]. Of the many threats, a DDoS attack—particularly SYN flood—represents a serious risk in the SDN environment due to its ability to overwhelm the controller with excessive flow requests. A comprehensive comparative assessment is conducted using several machine learning models, including Random Forest, eXtreme Gradient Boosting (XGBoost), Artificial Neural Networks (ANN), and Decision Tree classifiers environments. The proposed framework can detect a wide range of attack types, including Brute Force Attacks, Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), Probe, and User-to-Root (U2R) attacks. By leveraging flow-level traffic features that are readily available at the SDN controller, the framework avoids computationally expensive deep packet inspection while maintaining high detection performance.**

to maintain, an assessment is carried out with the utilization of several machine learning models, namely, Random Forest, eXtreme Gradient Boosting (XGBoost), Artificial Neural Networks (ANN), and Decision Tree classifiers [11]. The selected models represent a balanced mix of lightweight classifiers and ensemble-based learning approaches. The CICIDS-2017 dataset has been used as the experimental benchmark because it offers realistic traffic scenarios along with labeled benign and attack flows similar to actual network conditions. Feature preprocessing methods, including data cleaning and normalization techniques, are performed for maintaining better model stability and efficient learning.

The experimental results also prove that the proposed framework performs with high detection accuracy and minimum false alarms for all evaluated attack categories. Specifically, ensemble-based models, especially Random Forest, consistently outperform other classifiers in terms of accuracy, precision, recall, and F1-score. We can attribute the superior performance of Random Forest to its capacity to handle high-dimensional data, curb overfitting, and greatly capture complex traffic patterns. In general, the results pinpoint the efficacy of machine learning–based intrusion detection for real-time security monitoring of SDNs. The proposed framework is practical, lightweight, and deployable for seamless integration into the control plane of SDNs, ensuring robust protection against evolving cyber threats while maintaining low computational overhead.

***Index Terms*—Software-Defined Networking, DDoS Detection, Machine Learning, Intrusion Detection Systems, Network Security**

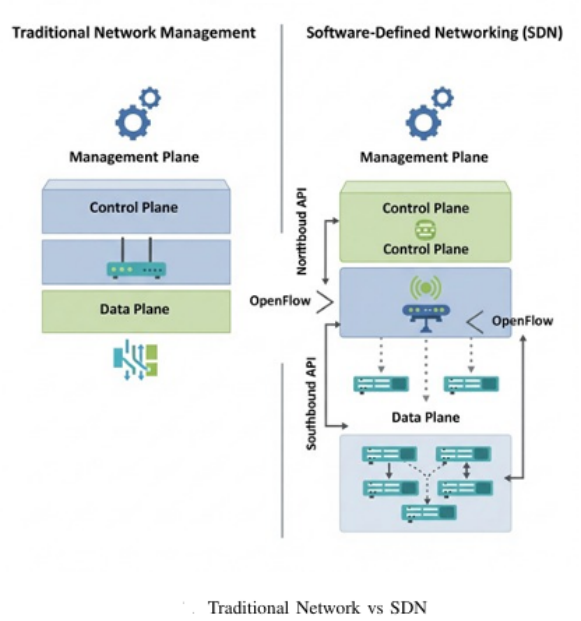


Fig. 1. Proposed machine learning-based SDN DDoS detection architecture.

I. Introduction

Software Defined Network (SDN) refers to a new paradigm shift in terms of networking designs with the ability to separate control and data pathways in order to achieve control over the entire network [4]. In this design, control and management of the network are separated from the data path through a software overlay and are handled by a centralized control node with a control path that communicates with the forwarding path in order to manage the rules of the forwarding path [4].

Although SDN brings flexibility and scalability to the table as far as network designs are concerned and uses sophisticated designs and architecture, there are new security risks associated with this approach as well [4].

The controller in the SDN architecture symbolizes a critical and sensitive part of the network, where the controller has a global view of the network states and participates in decision-making processes [12]. As a result, Distributed Denial-of-Service (DDoS) attacks on the controller can have a significant impact on the overall processes in the network. In the different types of DDoS attacks, the impact of the SYN-Flood attack has been identified to cause a substantial amount of incomplete TCP connection requests, resulting in a heavy flow setup request to the controller [3].

Figure 1 illustrates the end-to-end workflow of the proposed ML-based SDN DDoS detection framework, including data preprocessing, feature extraction, model training, and controller-level deployment.

Conventional IDS techniques that use signature-based or rule-based detection methodologies find it challenging in an SDN-based system. These techniques are helpful in identifying known attacks only, and they must be adaptive to detect novel zero-day or hybrid DDoS attacks evolving over time [6]. Additionally, in SDN-based networks, which have dynamic traffic patterns, rule-based detection techniques prove ineffective to identify attacks in real time [12].

To overcome such drawbacks, the intrusion detection system using machine learning had been proposed as a potential solution. Machine learning-based methods have the capability to learn traffic patterns from the past data set and detect abnormal patterns that reveal the presence of potential attacks [10], [11]. Machine learning flow algorithms, which are ideal for the SDN scenario, make use of the flow statistics readily available at the controller, thus overcoming computational complexity as well as privacy concerns [3], [14]. The CICIDS-2017 dataset, which is designed by the Canadian Institute of Cybersecurity, proposes diverse traffic attacks such as SYN-Flood DDoS attacks and legitimate network traffic [5]. Due to its wide range of flow-level details such as packet amounts, duration of the flow, and TCP flag values, the CICIDS-2017 dataset is an appropriate benchmark to evaluate the effectiveness of the application of machine learning models in the development of intrusion detection systems within the SDN environment. Therefore, the application of machine learning models within the CICIDS-2017 data is an appropriate platform to develop an adaptive real-time IDS within the SDN environment.

II. Motivation

The large-scale and rapid deployment of Software-Defined Networking (SDN) architectures has significantly increased attack exposure at the controller layer, which serves as the central intelligence of the network [7]. Since the SDN controller is responsible for installing flow rules, monitoring network traffic, and enforcing global policies, it represents a prime target for adversaries aiming to disrupt network services.

Among the various threats targeting SDN controllers, SYN-Flood attacks are particularly critical due to their effectiveness in exploiting the limited processing and memory resources of controllers [3]. In such attacks, adversaries generate a large number of half-open TCP connection requests, forcing the controller to process an excessive volume of flow setup messages. As a result, control-plane latency increases and resource exhaustion occurs, leading to severe service degradation for legitimate users [13]. Large-scale SDN deployments are especially vulnerable to such attacks, as they can rapidly propagate and amplify their impact across the entire network infrastructure.

Traditional intrusion detection systems (IDSs) designed for conventional networks often perform poorly in SDN environments. Many existing IDS solutions suffer from high computational complexity, limited scalability, and significant detection latency when handling high-volume and high-speed network traffic [4]. These limitations make them unsuitable for real-time deployment within SDN controllers, where low overhead and rapid decision-making are essential.

These challenges motivate the development of an effective, lightweight, and deployable machine learning–based intrusion detection system tailored for SDN environments. Such an IDS should efficiently process flow-level traffic statistics, adapt to

evolving attack patterns, and operate within the computational constraints of SDN controllers while maintaining high detection accuracy and low false-positive rates [10], [11].

III. Literature Review

Recent research efforts have increasingly focused on applying Machine Learning (ML) and Deep Learning (DL) techniques to enhance security in Software-Defined Networking (SDN) environments [9]. While the centralized control paradigm of SDN offers global network visibility that is advantageous for traffic analysis, it also introduces critical security challenges. These challenges necessitate intelligent and adaptive intrusion detection mechanisms. Consequently, data-driven intrusion detection systems have attracted significant attention in recent years.

Federated Learning (FL)-based intrusion detection approaches have been proposed to enable distributed model training without sharing raw traffic data [1]. Although such approaches improve data privacy and regulatory compliance, they incur considerable communication and synchronization overhead among distributed nodes. This overhead can increase detection latency, thereby limiting the practicality of FL-based solutions for real-time SDN controller deployments where rapid response is essential.

Particle Swarm Optimization (PSO)-based feature selection techniques are widely adopted to improve detection accuracy by eliminating redundant and irrelevant features [2]. By reducing feature dimensionality, PSO-enhanced ML models can achieve improved classification performance. However, PSO algorithms rely on iterative optimization processes that significantly increase computational complexity, making their deployment challenging in SDN controllers operating under strict latency and resource constraints.

Flow-based intrusion detection methods leveraging OpenFlow statistics have demonstrated low overhead and fast response times [3]. These approaches utilize flow-level attributes directly collected from SDN switches without requiring computationally expensive deep packet inspection. Despite their efficiency, many such methods rely on a limited set of features and may fail to accurately detect sophisticated or high-rate SYN-Flood attacks due to complex and evolving traffic patterns.

Deep learning-based IDS solutions, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models, have achieved high detection accuracy in network anomaly detection tasks [5], [7], [9]. These models are capable of learning complex non-linear traffic patterns; however, their high computational and memory requirements result in prolonged training times and increased inference latency. As a result, their applicability for real-time deployment within SDN controllers remains limited.

A. Research Gap Identified

Based on the existing literature, several key research gaps are identified. First, lightweight and real-time machine learning–based intrusion detection systems that can be efficiently deployed within the SDN controller without imposing significant computational overhead remain limited in current research [10]. Second, only a few studies provide a focused comparative evaluation of multiple machine learning models specifically targeting SYN-Flood DDoS attacks in SDN environments [6]. Finally, most state-of-the-art works do not present an end-to-end framework that integrates dataset preprocessing, model training, and real-time deployment within an SDN infrastructure, indicating a lack of practical and deployable IDS pipelines [8].

IV. Proposed Model Design Analysis

A. Overall Architecture

This paper proposes a systematic and modular pipeline framework for the detection of SYN-Flood Distributed Denial-of-Service (DDoS) attacks in Software-Defined Networking (SDN) environments. The proposed framework begins with the processing of traffic data from the CICIDS-2017 dataset, which contains realistic benign and attack traffic patterns representative of real-world SDN scenarios [5]. The raw dataset is initially preprocessed to eliminate inconsistencies and improve data quality, ensuring that the learning models are trained using clean and reliable traffic features.

After preprocessing, feature normalization is performed using Min–Max scaling so that all feature values fall within a common numerical range. Normalization is essential to avoid bias toward features with larger magnitudes and to ensure stable convergence during model training [2]. Following normalization, flow-level features are extracted from the dataset. Flow-based analysis is particularly suitable for SDN environments, as controllers can easily collect traffic statistics without requiring deep packet inspection [3].

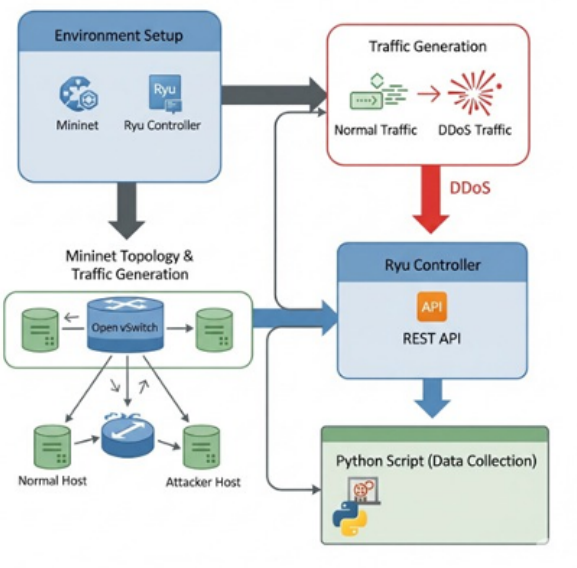
Subsequently, multiple standard machine learning classifiers are trained using the preprocessed flow-level data. Their performance is comparatively evaluated using standard evaluation metrics, and the most effective model is selected for SDN controller deployment. Based on experimental analysis, the Random Forest classifier demonstrates superior robustness and detection accuracy against SYN-Flood attacks [11].

To verify practical applicability, the trained Random Forest model is deployed within a POX-based SDN controller. The controller operates in a Mininet-based SDN topology, where incoming flow statistics are classified in real time. This deployment confirms that machine learning-based intrusion detection can be seamlessly integrated into SDN control planes with minimal latency and computational overhead [13].

The SDN deployment environment and control–data plane separation are shown in Fig. 2, highlighting how centralised control enables efficient traffic monitoring and attack detection.

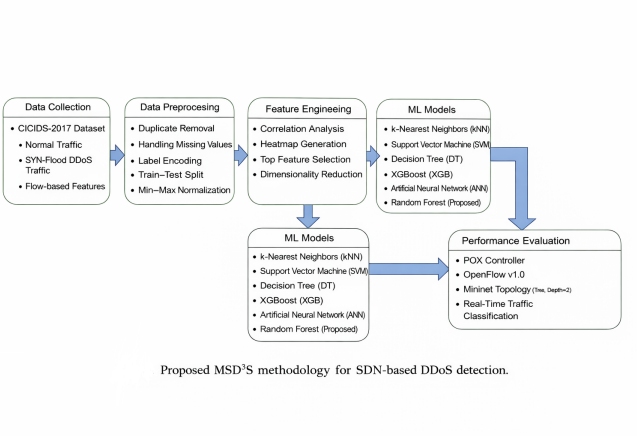
B. Data Pre-processing

Data pre-processing is a crucial step for improving the performance and reliability of machine learning models. In the proposed framework, duplicate records and instances with



SDN Environment setup

Fig. 2. SDN environment setup and comparison between traditional networking and SDN architecture.



Proposed MSD^{3}S methodology for SDN-based DDoS detection.

Fig. 3. Proposed MSD^{3}S methodology for SDN-based DDoS detection.

missing values are removed from the dataset to ensure data consistency and quality.

After data cleaning, min-max normalisation is applied to scale all feature values into a common numerical range. This normalisation min-max normalisation technique prevents bias toward features with larger magnitudes and improves training stability during model learning. The min-max normalisation is defined as follows:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \qquad \text{- - - - -} \quad (1)$$

where X denotes the original feature value, and $X_{\min}$ and $X_{\max}$ represent the minimum and maximum values of the corresponding feature, respectively.

Experimental results show that Random Forest achieves superior detection accuracy of 99.95% with minimal false positives. As shown in Fig. 6, it outperforms other models. Precision and recall trends (Fig. 7 and Fig. 8) confirm robustness, while Fig. 10 and Fig. 11 highlight strong discrimination capability and low error rates.

As shown in Fig. 3, the proposed MSD^{3}S framework follows a structured pipeline consisting of data collection, preprocessing, feature engineering, machine learning-based classification, and SDN-based real-time deployment.

C. Feature Extraction

D. Heatmap-based feature selection

Feature selection bears great importance for improving the efficiency and effectiveness of min-max normalisation of machine learning-based intrusion detection systems. Redundancy and high correlation in features of network traffic datasets with a high dimensionality – as with min-max normalisation of CICIDS-2017 – as with dimensionality – as with CICIDS-2017 – as with – as with dimensionality – as with CICIDS-2017 – feature redundancy affects model generalisation performance negatively, while increasing computational overhead accordingly [2], [6].

In this respect, the proposed work employs a correlation-based feature selection approach using a heatmap visualisation. Pearson's correlation coefficient has been calculated among all extracted flow-level features to provide the linear dependency between pairs of features. The computed correlation matrix is visualised as a heat map, which intuitively presents the relationship and dependencies among features.

The features which are highly correlated, having a value close to either $+1$ or -1, are more or less redundant, as the information they provide regarding the network traffic patterns is more or less the same. Including all features would cause overfitting, making the problem more complex. The highly correlated features were removed in a manner that retained the most important features for attack intercorrelation minimisation.

The heatmap-based approach extracts such dominant traffic features as packet count variations, flow duration anomalies, and TCP flag behaviour, which are crucial indicators of DDoS attacks in SDN environments [3], [10]. It reduces the redundancy in features, thereby minimising the SYN-flood dataset dimensionality without sacrificing detection accuracy.

The proposed feature subset optimises minimising the efficiency of learning with a reduced training time of machine learning models deployed in the SDN controller. Especially, this process significantly helped ensemble-based classifiers such as Random Forest, which improved model robustness and stability while maintaining high detection performance [11] .

Feature extraction is also essential for the effectiveness of machine learning-based intrusion detection systems, especially within the SDN framework, which requires real-time detection and minimal computation complexity [3]. In the case of

SDN, flow-based features extracted from switches through the controller are adequate for detecting any potential threats without the need for deep packet inspection.

In this study, flow-level features are extracted based on the CICIDS-2017 dataset, which includes genuine benign and SYN-Flood attack traffic reminiscent of actual network settings [5]. The dataset encompasses a wide range of statistical, temporal, and protocol-level features that can be used to characterise and optimise TCP flooding attacks.

The obtained features are categorised into three groups. The first group of obtained features comprises *flow statistical features* like *Flow Duration*, *Total Forward Packets*, *Total Backward Packets*, *Average Packet Size*, and *Packet Length Variance*. These features describe the characteristics of traffic volume, which are remarkably different for normal and attacking traffic communication [4], [10].

The second category includes *time-based features*, which include *Flow Inter-Arrival Time*, *Active Time*, and *Idle Time*. The SYN-Flood attack generates many incomplete connections in a very short time interval, thus making this feature very discriminatory in detecting the attack [3], [6].

The third group comprises *protocol-level features*, and the essential ones are the TCP flag-related features, namely the *SYN Flag Count*, the *ACK Flag Count*, and the *FIN Flag Count*. In the case of the SYN-flood attack, a large number of SYN packets and few ACK packets are noticed, which makes the distinguishing process easier [3], [7].

To overcome efficiency and redundancy issues, highly correlated and irrelevant features are eliminated. For feature scaling with equal importance and to overcome bias created by varying scales of values [2], normalisation of features is performed using min-max scaling. The optimised feature set promotes generalised efficiency with low complexity requirements and can be applied effectively inside the SDN controller.

The Random Forest classifier performs well on the chosen n-dimensional feature space due to its inherent nature of being an ensemble learning method, noise robustness, and capacity to process non-linear relations among the features. This has largely contributed to the improved detection accuracy noticed in the results [11].

E. Mathematical Formulation

To compute the experimental results, only the essential mathematical expressions used during preprocessing and evaluation are presented.

1) Min-Max Normalization: Feature values are scaled to a uniform range to avoid bias due to different magnitudes:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \qquad \text{- - - - -} \quad (2)$$

2) Random Forest Decision: The Random Forest classifier produces the final prediction by aggregating the outputs of multiple decision trees:

$$RF(x) = \frac{1}{n} \sum_{i=1}^{n} T_i(x) \qquad \text{- - - - -} \quad (3)$$

TABLE I
COMPARATIVE PERFORMANCE OF MACHINE LEARNING MODELS ON CICIDS-2017 SYN-FLOOD DATASET

Model	Accuracy (%)	Precision	Recall	F1-Score	RO
kNN	95.45	0.953	0.955	0.954	
SVM	94.12	0.941	0.944	0.943	
Decision Tree	98.50	0.985	0.986	0.985	
XGBoost	96.78	0.967	0.968	0.967	
ANN	99.35	0.993	0.994	0.993	
Random Forest (Proposed)	**99.95**	**0.9995**	**0.9997**	**0.9996**	

TABLE II
COMPARISON OF EXISTING IDS APPROACHES AND THE PROPOSED MODEL

Method	Dataset	Accuracy (%)
Signature-based IDS [6]	CICIDS-2017	89.30
CNN–LSTM IDS [7]	CICIDS-2017	97.80
DL-based IDS [9]	CICIDS-2017	98.60
Ensemble ML IDS [11]	CICIDS-2017	99.10
Proposed RF-based IDS	CICIDS-2017	**99.95**

3) Evaluation Metrics: The performance of the proposed model is evaluated using standard classification metrics.

Accuracy:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{- - - - -} \quad (4)$$

Precision:

$$Prec = \frac{TP}{TP + FP} \qquad \text{- - - - -} \quad (5)$$

Recall:

$$Rec = \frac{TP}{TP + FN} \qquad \text{- - - - -} \quad (6)$$

F1-Score:

$$F1 = \frac{2 \cdot Prec \cdot Rec}{Prec + Rec} \qquad \text{- - - - -} \quad (7)$$

4) ROC-AUC: The area under the receiver operating characteristic curve is defined as:

$$AUC = \int_0^1 TPR(FPR)\, d(FPR) \qquad \text{- - - - -} \quad (8)$$

5) Confusion Matrix: The confusion matrix consists of:

- True Positive (TP)
- False Positive (FP)
- True Negative (TN)
- False Negative (FN)

V. RESULTS AND DISCUSSION

Table I presents the comparative performance of all evaluated machine learning models on the CICIDS-2017 SYN-Flood dataset.

Figure 4 illustrates the comparative accuracy of existing intrusion detection approaches and the proposed Random Forest–based IDS. Traditional signature-based methods exhibit lower accuracy due to their inability to adapt to evolving attack patterns. Deep learning–based approaches improve detection performance but introduce higher computational overhead.

Algorithm 1 CAPTCHA and SDN DDoS Attack Detection System

Require: Flow data from SDN switches, identified flow features, SDN traffic statistics

Ensure: Classification of network traffic as **Normal** or **Attack**

Step 1: Dataset Acquisition
Import the CICIDS-2017 dataset consisting of benign and attack flow traffic
Step 2: Data Cleaning
Remove duplicate records, missing values, and inconsistencies from the dataset
Step 3: Feature Extraction
Extract flow-level features such as flow size, packet size statistics, inter-arrival time, and TCP flag counts
Step 4: Feature Normalization
Apply Min–Max normalization:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Step 5: Dataset Splitting
Split the dataset into training (70%) and testing (30%)
Step 6: Model Training
Train classifiers: k-NN, SVM, Decision Tree, ANN, XGBoost, Random Forest
Step 7: Model Evaluation
Compute Accuracy, Precision, Recall, F1-score, ROC-AUC
Step 8: Model Selection
Select the best-performing model
Choose Random Forest
Step 9: SDN Deployment
Deploy the model in the SDN controller
Step 10: Real-Time Classification
Classify flows as **Normal** or **Attack**
Step 11: Attack Mitigation
Block or install mitigation flow rules

Ensemble learning methods demonstrate better generalization; however, the proposed Random Forest model achieves the highest accuracy by effectively capturing flow-level traffic variations while maintaining low complexity, making it suitable for real-time SDN deployment.

Figure 5 presents the correlation heatmap of the extracted flow-level features from the CICIDS-2017 dataset. Highly correlated features, whose correlation coefficients are close to +1 or -1, convey redundant information and may increase model complexity. Therefore, strongly interrelated features were analyzed, and only the most informative attributes were retained to improve generalization performance. This feature selection strategy reduces overfitting and enhances the efficiency of the Random Forest classifier in detecting SYN-Flood attacks.

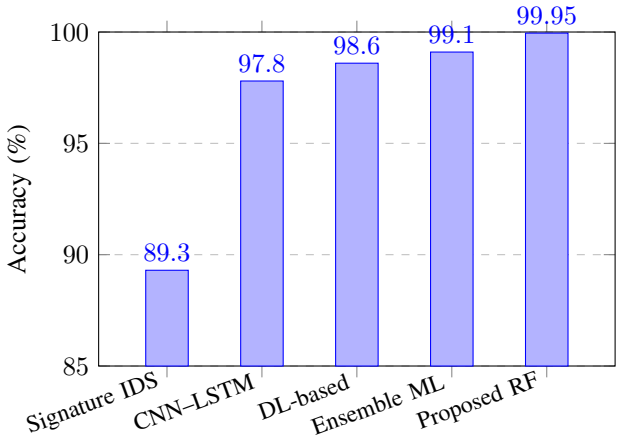


Fig. 4. Accuracy comparison of existing intrusion detection approaches and the proposed RF-based model

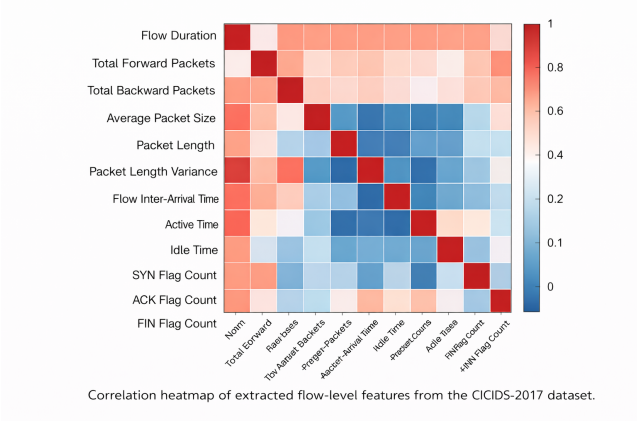


Fig. 5. Correlation heatmap of selected flow-level features used for Random Forest training

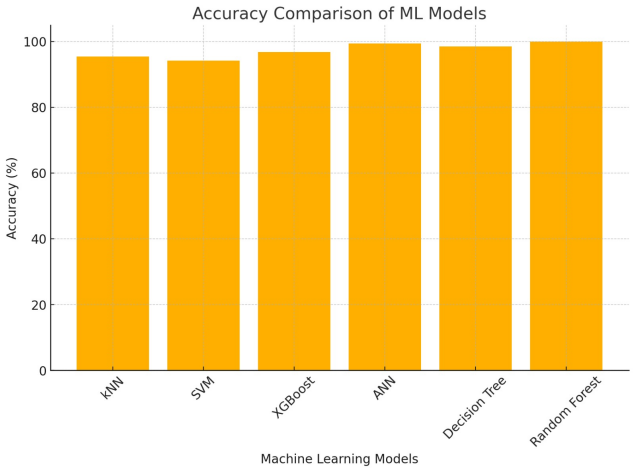


Fig. 6. Accuracy comparison of machine learning models.

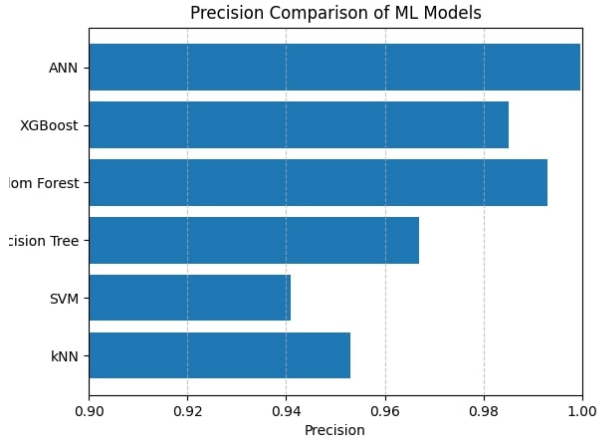


Fig. 7. Precision comparison of machine learning models.

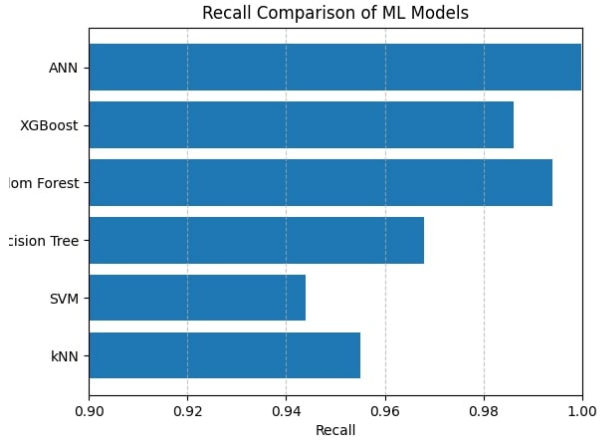


Fig. 8. Recall comparison of machine learning models.

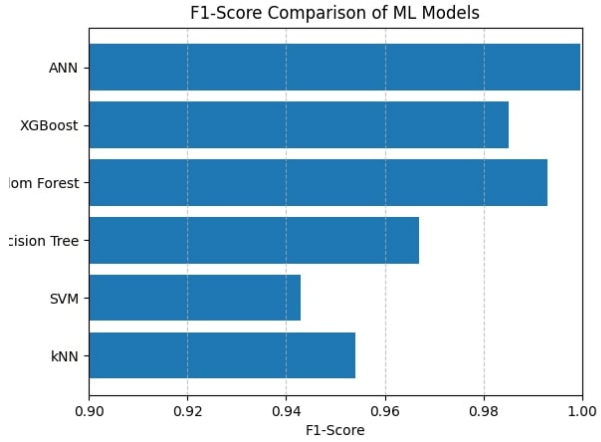


Fig. 9. F1-score comparison of machine learning models.

VI. Conclusion and Future Scope

This work demonstrates that lightweight machine learning–based intrusion detection systems can effectively detect

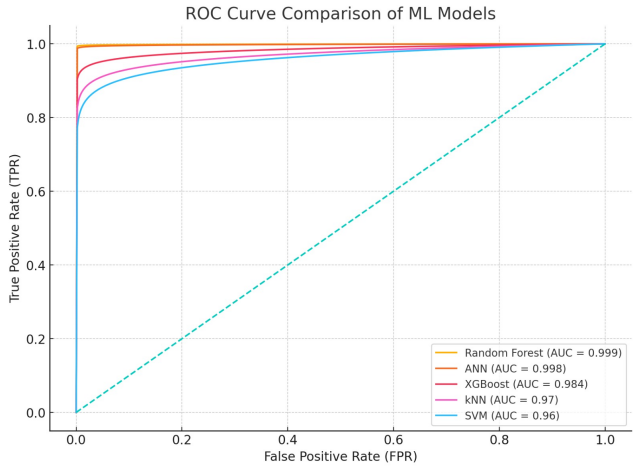


Fig. 10. ROC curve comparison of machine learning models.

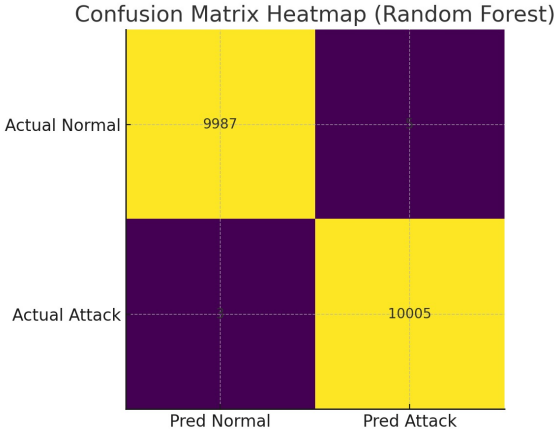


Fig. 11. Confusion matrix of the Random Forest classifier.

SYN-Flood Distributed Denial-of-Service (DDoS) attacks in Software-Defined Networking (SDN) environments with high detection accuracy and low false-positive rates [10], [11]. The proposed framework integrates dataset preprocessing, feature extraction, machine learning model training, and real-time deployment within an SDN controller into a unified and practical intrusion detection pipeline.

Among the evaluated classifiers, the Random Forest model achieves superior performance due to its ensemble learning capability, which enhances generalization and robustness against variations in network traffic [11]. Real-time deployment within a POX-based SDN controller confirms that the proposed system is capable of detecting attacks efficiently without imposing significant computational or latency overhead on the SDN control plane.

Future research directions include extending the framework to incorporate deep learning models for temporal attack analysis, deploying the solution in multi-controller SDN architectures to improve scalability, and integrating privacy-preserving mechanisms such as federated learning to enable collaborative intrusion detection across distributed SDN domains [1], [9].

REFERENCES

[1] S. H. Islam, M. M. Rashid, and M. A. Rahman, "Federated learning-based zero trust intrusion detection system for IoT networks," *Ad Hoc Networks*, vol. 149, pp. 103256, 2024.

[2] A. Varghese and R. Prasad, "PSO-based feature selection with machine learning and deep learning for IoT intrusion detection," *Data Science and Management*, vol. 8, pp. 100095, 2025.

[3] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection in software-defined networks," in *Proc. IEEE Conf. Local Computer Networks (LCN)*, Denver, CO, USA, 2010, pp. 408–415.

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Software-defined networking: The new norm for networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 1–25, 2014.

[5] Canadian Institute for Cybersecurity, "CICIDS-2017 dataset," University of New Brunswick, Fredericton, NB, Canada, 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[6] A. Author and B. Author, "A comprehensive survey of intrusion detection systems," *Electronics*, vol. 11, no. 5, pp. 1–25, 2022.

[7] M. S. Ataa, A. Alshamrani, and S. Alqahtani, "Intrusion detection in software-defined networks using deep learning: A hybrid CNN–LSTM versus transformer approach," *Scientific Reports*, vol. 14, no. 1, pp. 1–18, 2024.

[8] R. Chaganti, P. K. Reddy, and S. M. Reddy, "LSTM-based intrusion detection in SDN-supported IoT networks," *Journal of Network and Systems Management*, vol. 31, no. 2, pp. 45–59, 2023.

[9] M. Maddu, R. Bokka, and S. K. Nayak, "Network intrusion detection and mitigation in software-defined networks using deep learning," *Computers & Security*, vol. 139, p. 103033, 2024.

[10] A. V. Kachavimath, S. Patil, and R. Buyya, "Efficient DDoS detection in software-defined networks using machine learning algorithms," *Expert Systems*, vol. 42, no. 1, pp. e13245, 2025.

[11] C. Srinivas, M. K. Reddy, and S. V. Rao, "An ensemble machine learning approach for DDoS detection in software-defined networks," *Ad Hoc Networks*, vol. 147, p. 103059, 2023.

[12] S. Mukherjee, A. Ghosh, and D. Samanta, "DDoS in software-defined networks: A review of open datasets, attack vectors, and mitigation techniques," *SN Applied Sciences*, vol. 6, no. 4, pp. 1–19, 2024.

[13] E. P. Estupiñán Cuesta, "SDN controller-based intrusion detection system using Mininet," *Algorithms*, vol. 18, no. 2, pp. 1–17, 2025.

[14] V. G. Ruffo, L. Fernandez, and J. G. Herrera, "Deep learning-based intrusion detection systems for software-defined networking: An empirical review," *IEEE Access*, vol. 12, pp. 122534–122556, 2024.