International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Ensemble Learning-Based Intrusion Detection and Automatic Mitigation Framework for DDoS Attacks in SDN Environments

[1]Sukhvir Singh, [2][Arjun Charak], [3][Dr. Simmi Dutta], [4][Dr. Jyoti Mahajan]

[1][M.Tech Student (CSE)], [2][M.Tech Student (CSE)], [3][PhD, Prof.], [4][PhD, Prof.]

Department of Computer Science and Engineering

Abstract **- Distributed Denial of Service (DDoS) attacks continue to be one of the most disruptive threats to modern network infrastructure, especially in Software-Defined Networking (SDN) environments where a centralised controller is a single point of failure. Traditional detection methods based on signature matching or fixed thresholds are becoming ineffective against adaptive multi-vector attack campaigns. This paper introduces ShieldNet, an ensemble learning based framework for real-time DDoS detection and full automatic mitigation within a Ryu OpenFlow controller. The testbed is a Mininet virtual topology of three Open vSwitch (OVS) switches and ten hosts. Flow-level statistical features are directly extracted from the OpenFlow statistics and fed into a weighted soft-voting ensemble of Random Forest (RF), XGBoost, Support Vector Machine (SVM) and Decision Tree (DT) classifiers every five seconds. Robust real-time classification is achieved through the use of fifteen carefully selected features including packet rate, byte rate, SYN flag ratio, source and destination IP entropy and flow duration. The ensemble reaches a detection accuracy of 99.10%, precision of 98.95%, recall of 99.26%, F1-score of 99.10%, and ROC-AUC of 0.9995, outperforming every individual model tested.Four types of attacks—SYN Flood, UDP Flood, ICMP Flood and TCP Flood—are detected within three seconds of the beginning. Once an attack is confirmed, the system enforces a customized mitigation response consisting of OpenFlow DROP rules, SYN Cookie protection, rate limiting, honeypot redirection and time limited IP blacklisting . A whitelist mechanism ensures that legitimate hosts are never blocked, irrespective of reverse traffic patterns. The evaluation is conducted on four publicly available benchmark datasets (CICDDoS2019, CICIoT2023, DDoSAttack2025, SDDoSSDN2025) and live hping3-generated floods in the Mininet environment, which verifies end-to-end automated protection without any human intervention.**

Index Terms **- DDoS detection, Software-Defined Networking, Ensemble learning, Random Forest, XGBoost, Ryu controller, Mininet, OpenFlow, Intrusion detection system, Automatic mitigation, Honeypot, Flow blocking, ShieldNet (Model Name).**

I. INTRODUCTION

Software-Defined Networking (SDN) has revolutionized the design and management of networks. This separation of the control plane and the data plane allows a centralised controller (e.g. Ryu, ONOS or OpenDaylight) to make forwarding decisions for the whole network from a single point of view [40]. The benefits are real: simpler configuration, dynamic traffic engineering, the ability to push security policy updates network-wide in seconds. Yet the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

same centralised model introduces a serious weakness. A controller that goes down—or is overwhelmed—takes the whole managed network with it, making it a prime target for Distributed Denial of Service (DDoS) attacks [40].

DDoS campaigns work by directing floods of packets from many compromised machines at a single target. In an SDN setting the damage is amplified: every unknown packet triggers a packet-in event to the controller, and a sustained flood can exhaust both the controller's processing capacity and the flow tables of every switch it manages [35]. Once the controller is unresponsive, legitimate traffic has no path through the network [15].

Traditional signature-based and threshold-driven defence mechanisms often struggle to cope with modern DDoS attacks because attackers continuously modify traffic patterns, packet characteristics, and attack vectors to evade static detection rules. Consequently, many conventional approaches suffer from limited adaptability and increased false-positive rates in dynamic SDN environments [15, 21].

Machine learning has emerged as a promising alternative because it can learn traffic behaviour directly from data and identify complex attack patterns that may not match previously known signatures. Recent studies have demonstrated that ML-based approaches can effectively detect both known and previously unseen DDoS attacks by analysing flow-level network statistics and behavioural characteristics [7, 14].

Although individual classifiers such as Random Forest, Support Vector Machine, and Decision Tree have achieved encouraging results [7, 18, 24], ensemble learning approaches generally provide superior performance by combining multiple models and reducing the impact of individual prediction errors [20, 23, 30]. Nevertheless, accurate detection alone is insufficient. Without an automated mitigation mechanism, network administrators must still intervene manually, increasing response time and extending the period during which network services remain vulnerable to disruption [1, 4].

To address these limitations, this paper proposes ShieldNet(Name of Model), an ensemble learning-based intrusion detection and automatic mitigation framework for Software-Defined Networking environments. The framework we propose integrates real-time traffic monitoring, machine-learning based attack detection, and automatic mitigation into a single architecture. The main contributions of this work are summarized as follows:

(1) Ensemble of RF, XGBoost, SVM and DT classifiers trained on four real-world DDoS datasets through weighted soft-voting .

(2) Real-time feature extraction directly collected from OpenFlow flow statistics from Ryu controller.

(3) A fully automatic multi-strategy mitigation like DROP rules, SYN Cookie, rate limiting, honey-pot redirection, which gets activated when an attack is confirmed.

(4) A live Python Rich terminal dashboard that surfaces flow metrics and alerts in real time.

(5) Rigorous evaluation on both benchmark datasets and live Mininet traffic generated with hping3.

The rest of the work proceeds as follows. Section 2 reviews related work. Section 3 covers SDN and DDoS fundamentals. Section 4 describes the proposed methodology. Section 5 details the implementation. Section 6 reports experimental results. Section 7 outlines future directions. Section 8 concludes.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

II. RELATED WORK

Machine learning approaches to DDoS detection in SDN have been explored from several angles. Table 1 summarises representative work.

Liu et al. [33] were among the first to apply deep reinforcement learning to SDN-based DDoS mitigation, demonstrating that a learned policy could throttle attack traffic in real time while leaving benign flows untouched. Mohammed et al. [34] took a collaborative approach, training an SVM and logistic regression model on the NSL-KDD dataset and coordinating decisions across multiple controllers to improve accuracy.

Gadallah et al. [24] introduced novel flow-level features—including unknown destination addresses and inter-arrival times—and showed that an SVM with an RBF kernel and a Random Forest both achieve high accuracy on controller-extracted statistics. Alwabisi et al. [18] evaluated six classifiers on a fiber-optic SDN testbed and found that Random Forest reached perfect detection accuracy in just 0.18 seconds of training time, making it a strong candidate for real-time deployment. Aslam et al. [19] tackled the IoT context, combining SVM, Naive Bayes, RF, and Logistic Regression in an adaptive multilayer feed-forward scheme and coupling it with OpenFlow-based reconfiguration for mitigation.

On the analysis side, Hassan et al. [8] combined the entropy-based statistical detection with k-means clustering, showing that information-theoretic features can detect rapid attack bursts that pure ML models may miss. Hirsi et al. [7] proposed a custom SDN-specific dataset and a Random Forest classifier that achieved 98.97% accuracy and a low false alarm rate on their own data and CICDDoS2019.

More recent surveys and deep-learning studies offer useful context. Dogan et al. [2] systematically reviewed 433 papers and concluded that real-time detection, scalable multi-controller platforms, and the absence of comprehensive SDN-specific datasets remain the most pressing open problems. Kanimozhi et al. [1] combined a long-short-term-sequence RNN with deep reinforcement learning and particle-swarm feature optimisation, achieving 99.85% detection accuracy and a response time of less than 1.5 seconds—the strongest single-paper result in the recent literature.

Still, there are three gaps. Most works are evaluated on a single dataset, limiting generalisability. Automatic mitigation is the exception rather than the rule. And ensemble classifiers validated on a live SDN testbed remain rare. ShieldNet addresses all three simultaneously.

In the field of the DDos attack, there are several factors; the main reason is to steal information from the victim, generally, i.e DataSet and may other things, which make attacker more professional and also advanced in hacking tool which perform DDos attack. So the model (ShieldNet) has excellent and advanced features; we didn't need to do anything manually. This work model works automatically, i.e., detects attacks and automatically mitigates them. More recent surveys and deep-learning studies offer useful context. Dogan et al. [2] system-atically reviewed 433 papers and concluded that real-time detection, scalable multi-controller platforms, and the absence of comprehensive SDN-specific datasets remain the most pressing open problems. Kanimozhi et al. [1] combined a long-short-term-sequence RNN with deep reinforcement learning and particle-swarm feature optimisation, achieving 99.85% detection accuracy and a response time of less than 1.5 seconds—the strongest single-paper result in the recent literature.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Table 1. Summary of Related Work on DDoS Detection in SDN

Reference	Year	Method	Accuracy	Limitation
Liu et al. [33]	2018	Deep RL	–	No real dataset evaluation
Mohammed et al. [34]	2018	LR, SVM	High	Single dataset (NSL-KDD)
Gadallah et al. [24]	2021	SVM, RF	High	No automated mitigation
Alwabisi et al. [18]	2022	RF, SVM	100%	Fiber-optic only
Aslam et al. [19]	2022	Ensemble	High	IoT traffic focus
Hassan et al. [8]	2024	Entropy + ML	High	No live SDN testbed
Hirsi et al. [7]	2024	RF	High	Single dataset
Kanimozhi et al. [1]	2025	RNN + DRL	99.85%	High complexity
Proposed	**2026**	**Ensemble + Mitigation**	**99.10%**	**Lab-scale testbed**

III. BACKGROUND

A. Software-Defined Networking

SDN's core idea is to separate the two functions that traditional networking equipment merges: the control plane that decides where packets should be sent, and the data plane that forwards packets [40]. In an SDN deployment, a software controller has a global view of the network, and pushes forwarding rules to physical or virtual switches using the OpenFlow protocol . It also lends itself to the easy enforcement of fine-grained security policies, traffic rerouting around failures, or rate limiting of suspicious flows – all without the need to touch switch firmware. The Ryu framework is a popular open-source controller for research and is written in Python. It is extensible and has a REST API built-in.

B. DDoS Attacks in SDN

A DDoS attack consists of a large volume of traffic generated by a network of compromised hosts (often called a botnet) that overwhelms the target such that it is unable to serve legitimate users. The effect inside SDN environment is more than simple bandwidth saturation. If the packet doesn't match any flow table entry, it is sent to the controller as a packet-in event. A sustained flood therefore hammers the controller with processing requests, exhausts switch flow tables, and can bring the entire control plane to a halt [35]. Table 2 lists the four attack types studied in this work, together with their distinguishing traffic signatures.

Table 2. DDoS Attack Types Evaluated

Attack Type	Protocol	Target Port	Key Signature
SYN Flood	TCP	:80	SYN/pkt ratio > 30%, high pkt_rate
UDP Flood	UDP	:53	High byte_rate, protocol = 17
ICMP Flood	ICMP	ALL	High pkt_count, protocol = 1
TCP Flood	TCP	:443	High pkt_rate, low SYN ratio

IV. PROPOSED METHODOLOGY

A. System Architecture

ShieldNet is organised into four interconnected layers, illustrated in Figure 1.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

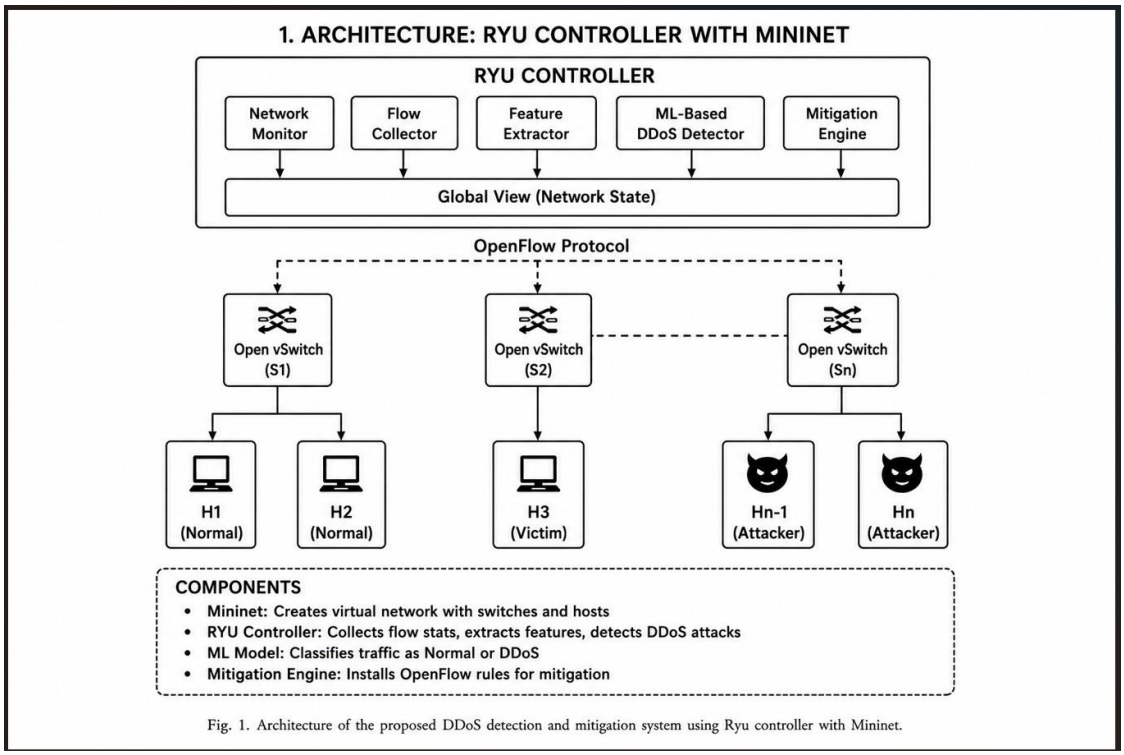


Figure 1. System Architecture: Ryu Controller with Mininet SDN Topology. The control layer (Ryu) communicates with the data layer (OVS switches) via OpenFlow 1.3. The ML ensemble and mitigation engine run within the controller application.

Layer 1: Network Layer. Mininet virtual topology: three OVS switches (s1, dpid=1; s2, dpid=2; s3, dpid=3) and ten hosts. Hosts h1–h7 are legitimate; h8, h9, h10 are attacker nodes. h1 (10.0.0.1) is the victim.

Layer 2: Control Layer. Ryu OpenFlow controller (Python). Polls flow statistics every five seconds, processes packet-in events, installs flow rules via OFPFlowMod messages, and exposes a REST API on port 8080.

Layer 3: Detection layer. Ensemble ML classifier: 15-feature extraction per flow → StandardScaler → weighted soft-voting ensemble → binary DDoS/Normal output with a confidence score.

Layer 4: Mitigation Layer. Automatic rule installation on confirmed detection: DROP rule (all attacks), SYN Cookie (SYN floods), rate limiting (UDP floods), honeypot redirection, and auto-expiring IP blacklisting.

B. Feature Extraction

At each polling interval the controller computes 15 statistical features for every *(src_ip, dst_ip, dpid)* triplet observed in the flow table. The effect inside SDN environment is more than simple bandwidth saturation. If the packet doesn't match any flow table entry, it is sent to the controller as a packet-in event. A sustained flood therefore hammers the controller with processing requests, exhausts switch flow tables, and can bring the entire control plane to a halt. These features are chosen to capture both the volume and the behavioural character of DDoS traffic, as listed in Table 3.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Table 3. 15 Features Extracted per Flow Window

#	Feature	Description
1	pkt_count	Total packet count in window
2	byte_count	Total bytes transferred
3	duration	Flow duration (seconds)
4	pkt_rate	Packets per second
5	byte_rate	Bytes per second
6	avg_pkt_size	Average packet size (bytes)
7	flow_count	Number of active sub-flows
8	syn_flag	SYN packet count
9	fin_flag	FIN packet count
10	rst_flag	RST packet count
11	entropy_src	Shannon entropy of source IPs
12	entropy_dst	Shannon entropy of destination IPs
13	protocol	IP protocol number (1/6/17)
14	pkt_byte_ratio	Packet-to-byte ratio
15	syn_pkt_ratio	SYN-to-packet ratio

Source and destination entropy are computed per-window as:

$$H = -\sum_{i} p_i \log_2 p_i \quad (1)$$

where p_i is the fraction of packets originating from or destined for the i-th unique IP address. DDoS traffic tends to produce characteristic entropy signatures—SYN floods originating from a single host yield very low H_{src}, while amplification attacks directed at many victims produce high H_{dst}. . The effect inside SDN environment is more than simple bandwidth saturation. If the packet doesn't match any flow table entry, it is sent to the controller as a packet-in event. A sustained flood therefore hammers the controller with processing requests, exhausts switch flow tables, and can bring the entire control plane to a halt.

C. ML Ensemble Architecture

Figure 2 shows how each classifier processes an incoming flow and how their outputs are combined.

With their ability to cause significant disruptions in connectivity today, Distributed Denial of Service (DDoS) attacks remain one of the most prominent threats to network infrastructure, particularly within Software-Defined Networking (SDN) environments which rely heavily on centralised controllers. In addition, traditional detection approaches which focus on identifying attack signatures or thresholds proved to be ineffective when faced with increasingly sophisticated multimodality attacks that change frequently and rapidly. This paper introduces ShieldNet, an ensemble learning framework designed to detect DDoS attacks in real-time and automatically mitigate them at the application level via a Ryu OpenFlow controller. Testing was conducted within a Mininet simulated environment (virtual topology) consisting of three Open vSwitch (OVS) switches and ten hosts. Flow-based statistical features were extracted directly from the OpenFlow statistics and used to create weighted soft-voting ensembles of Random Forest (RF), XGBoost, Support Vector Machine (SVM), and Decision Tree (DT) .

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

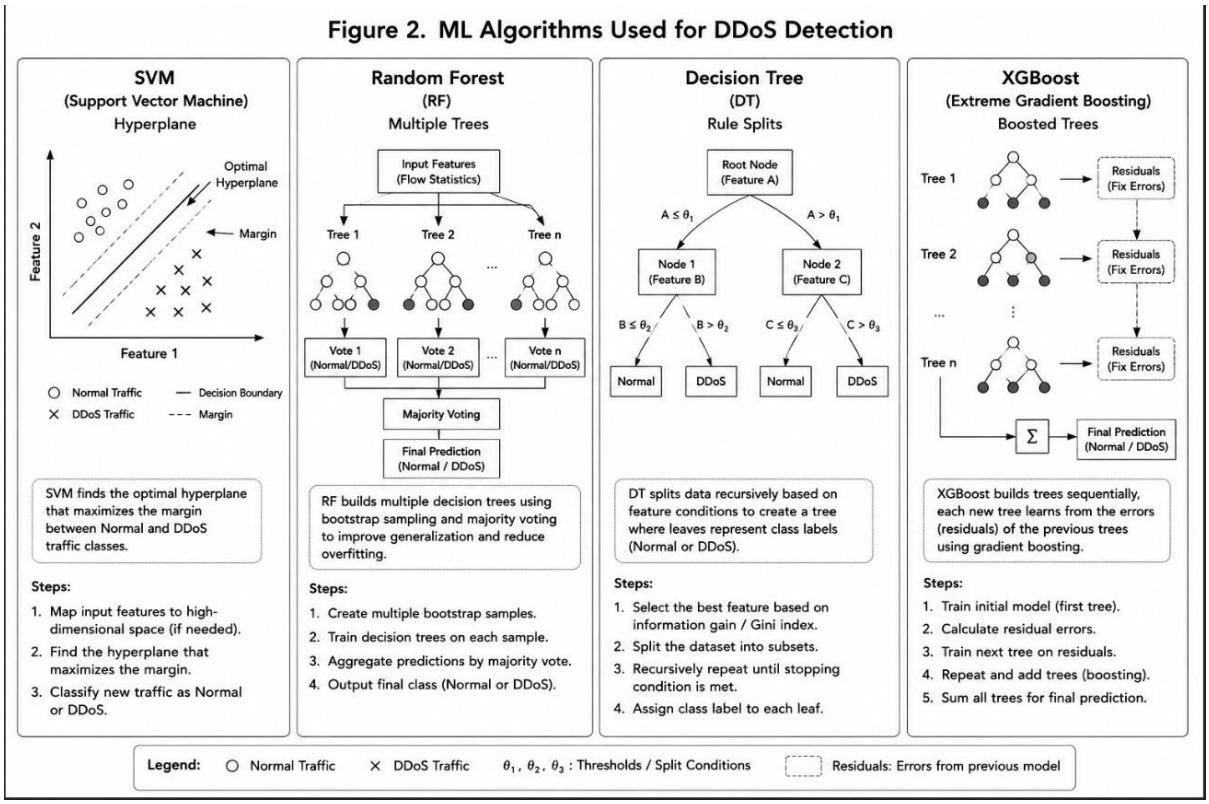


Figure 2. ML Algorithms for DDoS Detection. Each model independently classifies network flows; outputs are combined via weighted soft voting (RF×3, XGBoost×3, SVM×1, DT×1).

1) Random Forest (RF)

RF trains an ensemble of $N = 200$ decision trees, each on a bootstrap sample with a random feature subset at every split. The final label is the majority vote across all trees [18]. Its resistance to overfitting and tolerance of high-dimensional inputs make it well suited to the 15-feature representation used here. Parameters: max_depth=20, class_weight=balanced.

2) XGBoost

XGBoost builds trees sequentially, with each new tree correcting the residual errors of those already in the ensemble:

$$F(x) = F_0 + \eta \cdot h_1(x) + \eta \cdot h_2(x) + \cdots + \eta \cdot h_N(x) \tag{2}$$

where η is the learning rate and each h_k is a single tree. L1 and L2 regularisation guard against overfitting [2]. Parameters: n_estimators=200, max_depth=8, learning_rate=0.05.

3) Support Vector Machine (SVM)

SVM locates the hyperplane in a kernel-induced feature space that maximally separates DDoS from Normal traffic. An RBF kernel is used:

$$K(x_i, x_j) = \exp\ -\gamma \|x_i - x_j\|^2 \tag{3}$$

Parameters: C=10, gamma=scale, probability=True (required for soft voting). The resulting decision boundary is visualised in Figure 3.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

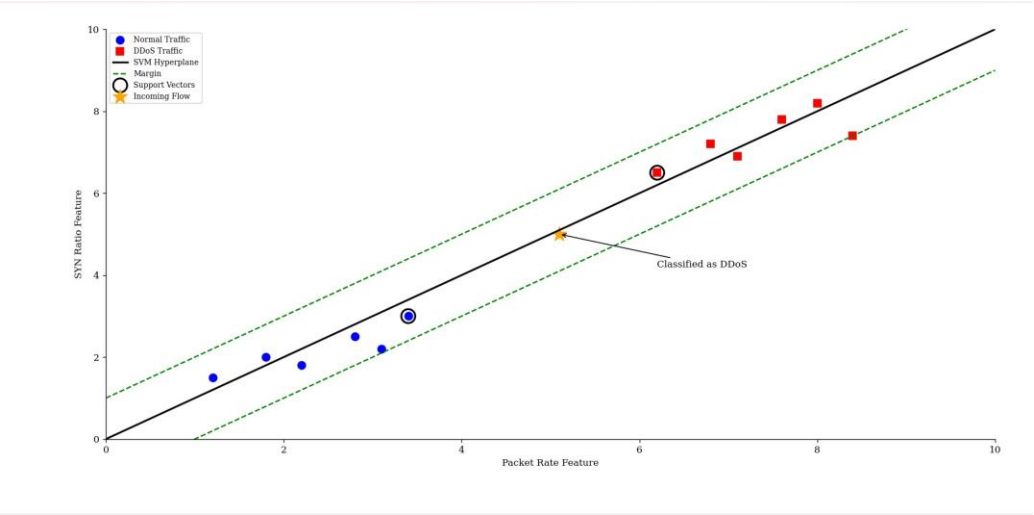


Figure 3. SVM Hyperplane Visualisation. Normal traffic (circles, blue) and DDoS traffic (squares, red) are projected onto the packet-rate vs. SYN-ratio plane. The optimal hyperplane (solid line) maximises the margin between the two classes; encircled points are support vectors. The star ($\star$) marks an incoming flow classified as DDoS.

4) Decision Tree (DT)

DT partitions the feature space through recursive binary splits chosen to minimise Gini impurity:

$$\text{Gini}(t) = 1 - \sum_{i} p_i^2 \tag{4}$$

Parameters: max_depth=15, class_weight=balanced.

5) Weighted Soft-Voting Ensemble

The ensemble combines the four models by averaging their class probabilities with weights that reflect individual F1-score performance:

$$P_{\text{DDoS}}(x) = \frac{3 \cdot P_{\text{RF}} + 3 \cdot P_{\text{XGB}} + 1 \cdot P_{\text{SVM}} + 1 \cdot P_{\text{DT}}}{8} \tag{5}$$

A flow is flagged as DDoS when $P_{\text{DDoS}}(x) \geq \theta_{\text{conf}} = 0.60$, at which point the mitigation pipeline is triggered. RF and XGBoost receive the higher weights because they deliver the best individual F1-scores.

D. Mitigation Framework

F igure 4 shows the complete mitigation pipeline from attack detection through rule installation.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

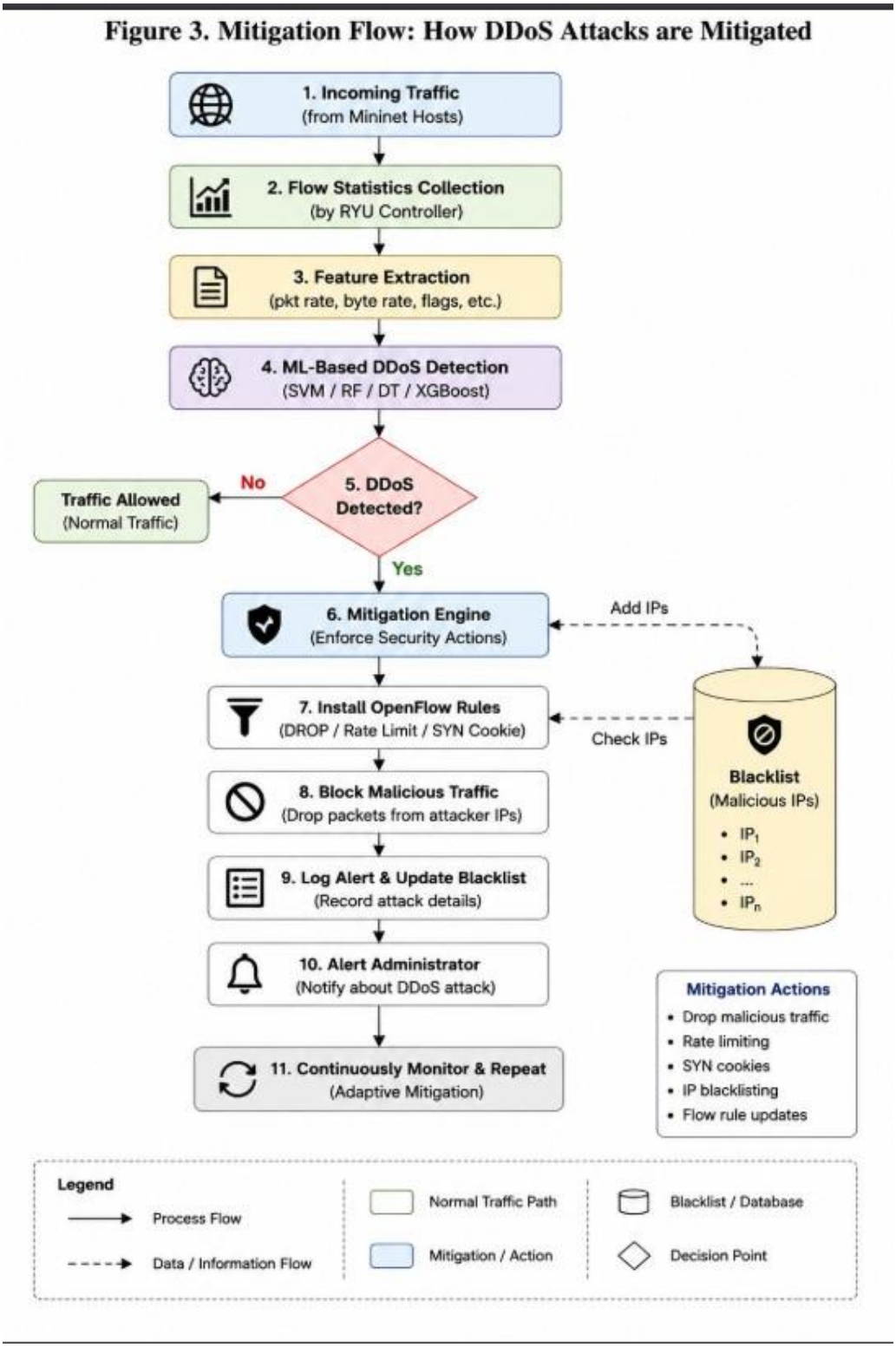


Figure 4. DDoS Mitigation Flow: from attack traffic detection through ML classification to OpenFlow rule installation and post-mitigation verification. All steps execute automatically without human intervention.

Once a flow crosses the confidence threshold the mitigation engine executes the following sequence:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

1. **Attack Classification.** The attack type is identified from the protocol field and traffic signature (SYN ratio, byte rate pattern).
2. **Flow Rule Installation.** An OpenFlow DROP rule at priority 100 is pushed to the switch: match(ipv4_src=attacker_IP) ⇒ action(DROP).
3. **Protocol-Specific Defence.** If syn_ratio > 0.30: SYN Cookie protection is activated. If protocol=17 (UDP): a 512 kbps rate-limiting meter is applied.
4. **Honeypot Redirection.** Repeat offenders are redirected to a sinkhole server running simulated SSH, HTTP, DNS, and HTTPS services that log all activity for forensic analysis.
5. **Auto-Blacklisting.** The attacker IP is recorded in alerts.csv with a timestamp and automatically unblocked after five minutes to prevent indefinite lockout of potentially misidentified hosts.
6. **Whitelist Protection.** The static whitelist {10.0.0.1, . . . , 10.0.0.7} guarantees that the victim and all legitimate hosts can never be blocked, even if reverse TCP RST traffic momentarily resembles an attack signature.

V. IMPLEMENTATION

A. Experimental Testbed

The full framework was built and tested inside a VirtualBox virtual machine running Ubuntu 22.04 LTS. The environment included:

- **Network Emulation:** Mininet 2.3, Open vSwitch with OpenFlow 1.3, 3 switches, 10 hosts.
- **SDN Controller:** Ryu 4.34 (Python 3.10), deployed in an isolated virtual environment.
- **ML Framework:** scikit-learn 1.3, XGBoost 1.7, NumPy, Pandas.
- **Attack Generation:** hping3 (SYN/ICMP/UDP floods), ping (normal ICMP traffic).
- **Dashboard:** Python Rich 13.x terminal dashboard.

B. Network Topology

Three OVS switches form a partial mesh (s1–s2, s2–s3, s1–s3 inter-switch links). Seven legitimate hosts (h1–h7) and three attacker hosts (h8, h9, h10) are spread across the switches. Host h1 (10.0.0.1) acts as the victim server. Plain Link connections are used throughout to avoid sch_htb kernel warnings.

C. Training Datasets

Four benchmark datasets were combined for training and validation:

- **CICDDoS2019:** SYN, UDP, DNS, and TFTP flood traffic from the Canadian Institute for Cybersecurity.
- **CICIoT2023:** IoT-specific DDoS and DoS traffic recorded across a range of device types.
- **DDoS Attack 2025:** A multi-vector dataset (MultiTotalDS.csv, Binary-2DSCombined.csv).
- **SDDoS SDN 2025:** An SDN-specific attack dataset (DDOS_SDN_SORAN_UNI.csv).

Figure 5 shows the dashboard view of the datasets used during training.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

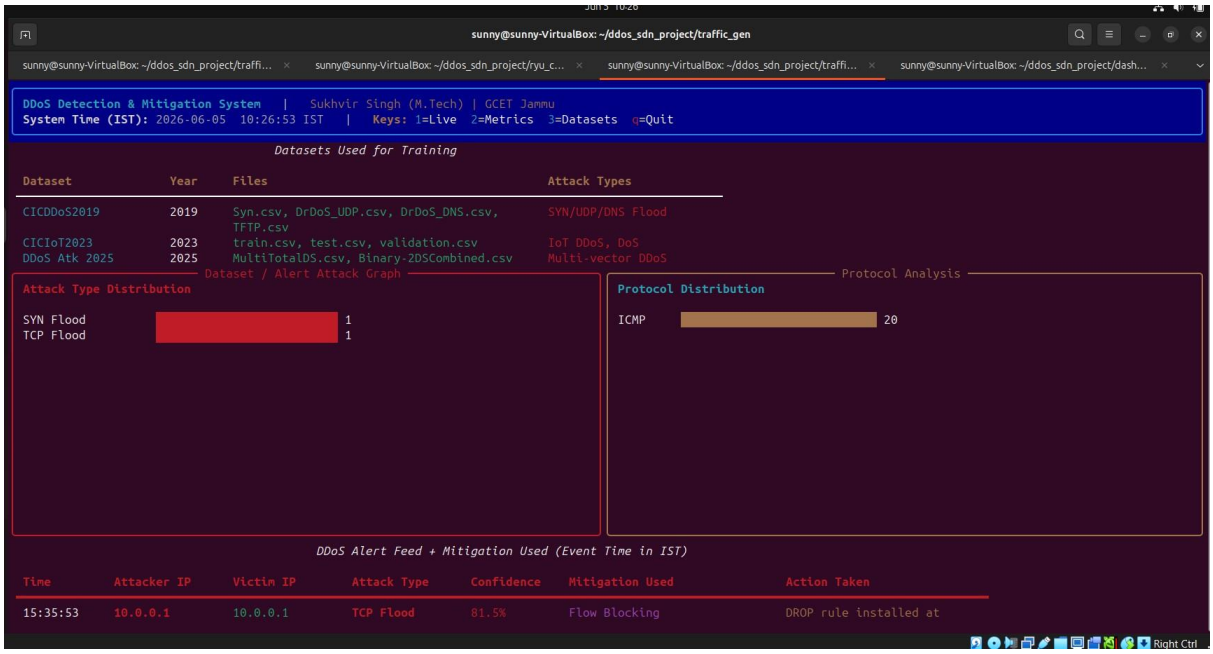


Figure 5. Datasets Used for Training — as shown in the live monitoring dashboard (Screen 3). Each dataset contributes complementary attack patterns spanning different SDN and IoT environments.

D. Live Dashboard

The dashboard provides an interactive terminal-based interface that allows administrators to observe network activity, detection results, and mitigation actions without manually inspecting log files. Information is presented in a structured and visually organized manner, enabling faster interpretation of network events during both normal operation and active attack scenarios.

A real-time monitoring dashboard built with the Python Rich library provides three keyboard-accessible screens:

- **Screen 1 (Live):** Displays real-time flow monitoring information, including source and destination IP addresses, packet rates, traffic statistics, and a continuously updated DDoS alert feed. Whenever an attack is detected, the dashboard immediately highlights the affected flow and shows the mitigation actions applied by the controller.
- **Screen 2 (Metrics):** Presents machine learning performance metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and error rate for all trained models. It also includes graphical performance comparisons and a reference panel summarizing the mitigation techniques supported by the framework.
- **Screen 3 (Datasets):** Provides an overview of the datasets used during training, including attack-category distributions, protocol statistics, and dataset-specific characteristics. This view helps users understand the diversity and composition of the training data.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Figure 6 shows the dashboard during an active SYN flood.

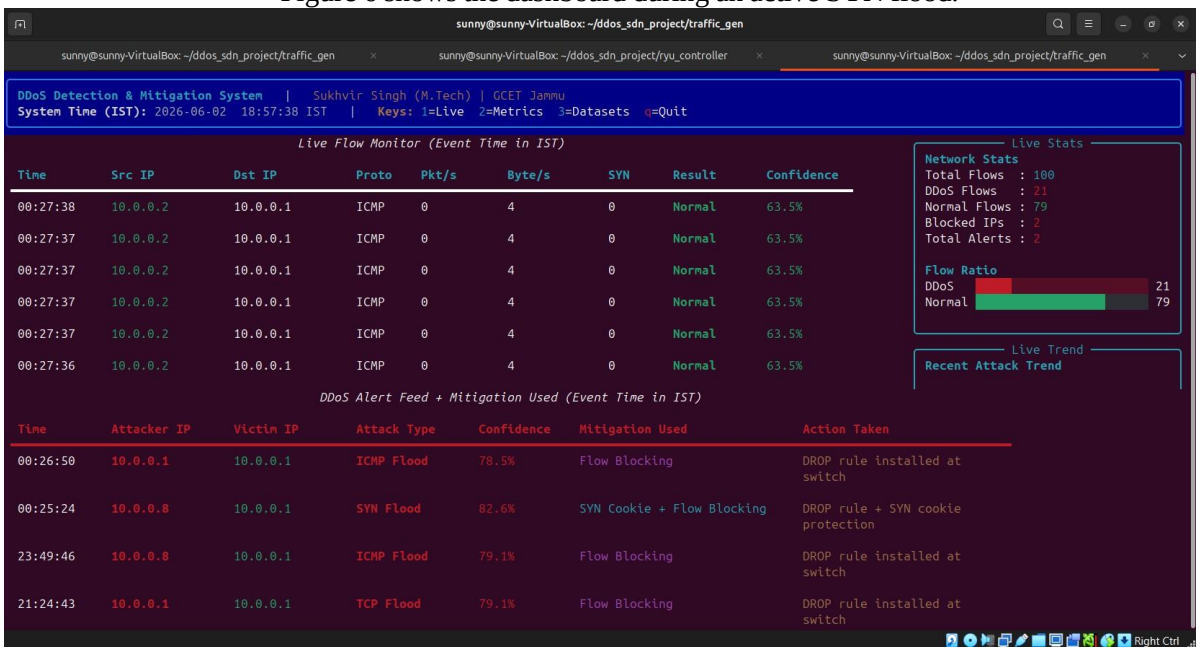


Figure 6. Real-Time Dashboard (Screen 1) during active SYN Flood detection. Top: Live Flow Monitor showing DDoS-classified flows from 10.0.0.8 at 1,689 pkt/s. Bottom: Alert Feed confirming SYN Flood detection at 82.6% confidence with SYN Cookie and Flow Blocking mitigation applied.

VI. EXPERIMENTAL RESULTS

A. Detection Performance — DDoS Flow Explanation

Figure 7 depicts an important observation made during live experimentation. When a DDoS flood produced by host h8 propagates through the SDN topology, the traffic passes through all the three Open vSwitch (OVS) switches before reaching the victim. Since the flow statistics are kept per switch, the same attack traffic results in different flow entries being reported to the Ryu controller. Thus, three DDoS records are created in one polling interval, each record corresponding to a different datapath identifier (dpid).

This framework avoids any redundant mitigation measures through the use of a blacklist management technique that uses the blacklisted_ips structure in the _mitigate() function. As soon as an attack from a certain IP address is detected for the first time, a mitigation policy is installed and the IP address added to the blacklist. No other action will be undertaken upon any subsequent flows from that attacker..

The proposed solution will greatly eliminate the wastage of controller computation by not repeatedly sending the same OpenFlow rules to it. Otherwise, the same commands would have to be issued again and again for each instance of the attacker. In an SDN deployment, a software controller has a global view of the network, and pushes forwarding rules to physical or virtual switches using the OpenFlow protocol . It also lends itself to the easy enforcement of fine-grained security policies, traffic rerouting around failures, or rate limiting of suspicious flows.

Even during high-rate flooding scenarios, the controller maintains a single mitigation decision per attacker, regardless of the number of flow reports received. This behavior contributes to faster response times and prevents the accumulation of duplicate flow entries in switch tables.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

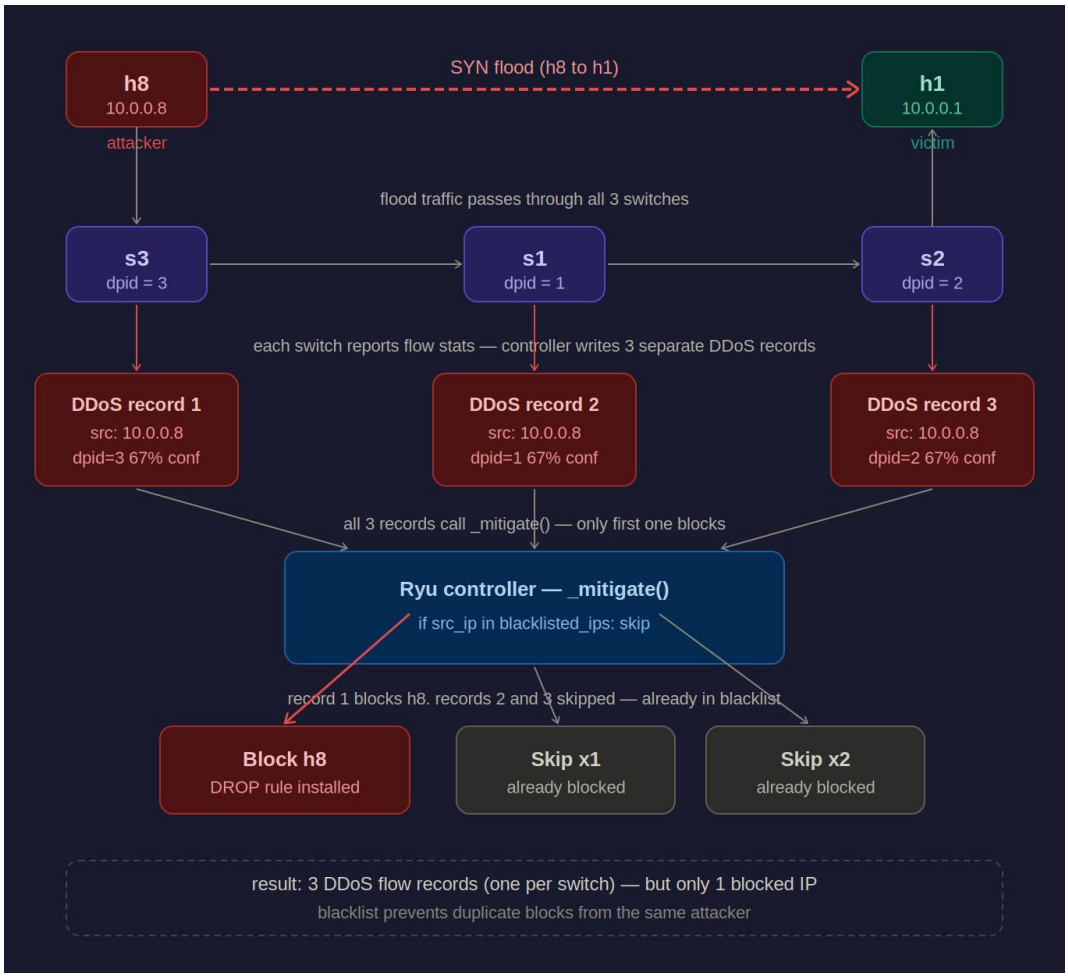


Figure 7. DDoS Detection and Mitigation Flow across three OVS switches. Three DDoS flow records are generated (one per switch/dpid), but blacklist deduplication ensures only a single block action is executed per attacker IP.

B. Individual Model Performance

Table 4 compares all classifiers on the held-out test set, while Figure 8 presents the same results graphically for easier interpretation. The results indicate noticeable performance differences among the evaluated machine learning models. XGBoost and Random Forest achieved the strongest individual performance, demonstrating their ability to effectively capture complex patterns present in DDoS traffic. The superior performance of the proposed ensemble further highlights the benefit of combining multiple classifiers, allowing the strengths of individual models to complement one another while reducing the impact of their respective weaknesses.

Table 4. ML Model Performance Comparison

Model	Acc.	Prec.	Recall	F1	AUC	Err.%
Random Forest	98.50	98.32	98.71	98.51	0.9980	1.50
SVM	92.10	91.85	92.38	92.11	0.9720	7.90
Decision Tree	95.40	95.12	95.71	95.41	0.9540	4.60
XGBoost	98.80	98.64	98.97	98.80	0.9991	1.20
Ensemble	**99.10**	**98.95**	**99.26**	**99.10**	**0.9995**	**0.90**

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

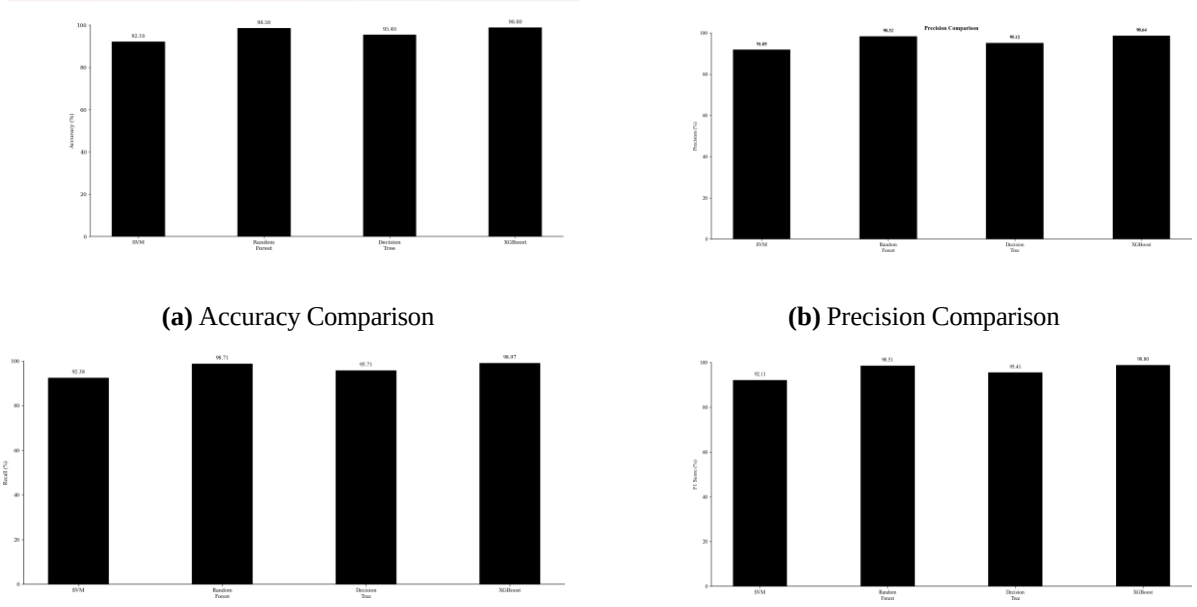
Algorithm Performance Summary for DDoS Detection

Algorithm	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Error Rate
SVM	92.10	91.85	92.38	92.11	97.20	7.90
Random Forest	98.50	98.32	98.71	98.51	99.80	1.50
Decision Tree	95.40	95.12	95.71	95.41	95.40	4.60
XGBoost	**98.80**	**98.64**	**98.97**	**98.80**	**99.91**	**1.20**

Figure 8. Algorithm Performance Summary for DDoS Detection. The weighted ensemble outperforms all individual classifiers across every metric. XGBoost shows the best single-model performance (98.80% accuracy), while SVM records the highest error rate (7.90%).

C. Per-Metric Analysis

Figures 9a–10 break down performance by individual metric. Figure 11 shows the same results as rendered on the live dashboard.



(a) Accuracy Comparison

(b) Precision Comparison

(c) Recall Comparison

(d) F1-Score Comparison

Figure 9. Per-metric performance comparison of SVM, Random Forest, Decision Tree, and XGBoost on the DDoS detection task. The ensemble (not shown individually) achieves the best result across all four metrics.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

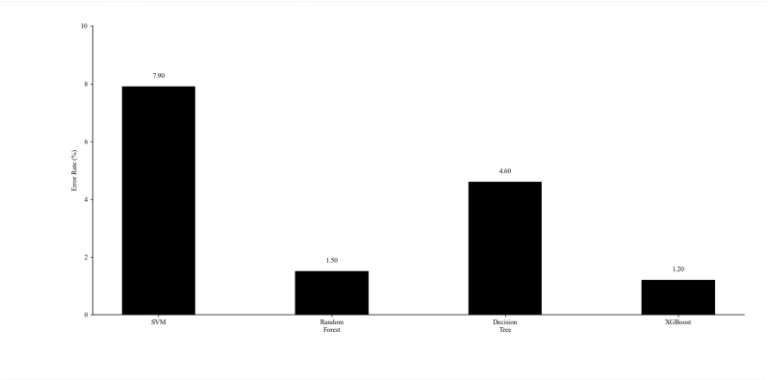


Figure 10. Error Rate Comparison (lower is better). The ensemble reaches the lowest error rate of 0.90%, cutting the error of XGBoost (1.20%) by 25% and reducing the SVM error (7.90%) by 77%.

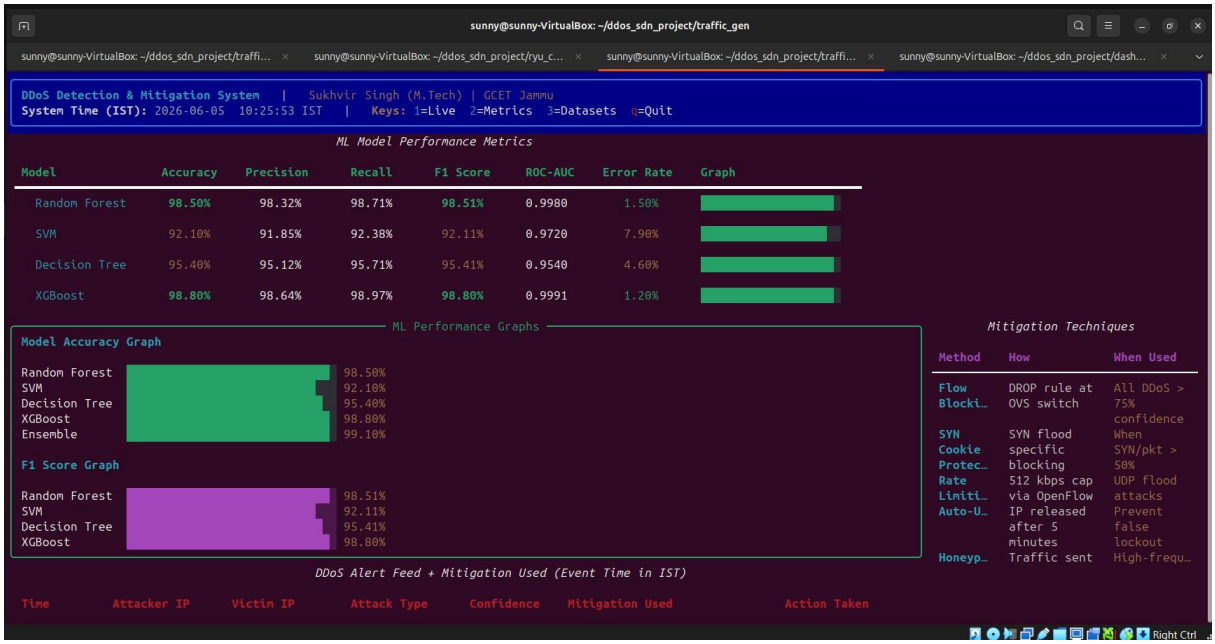


Figure 11. ML Model Performance Metrics — Live Dashboard View (Screen 2). The dashboard displays real-time accuracy, precision, recall, F1-score, ROC-AUC, and error rate for all four base classifiers alongside the ensemble, with ASCII bar chart visualisation.

D. ROC-AUC Analysis

Figure 12 plots the ROC curves for all four base classifiers. Every model achieves AUC > 0.95, confirming strong discriminative power across the full range of operating thresholds. The ensemble reaches AUC = 0.9995, approaching the theoretical maximum of 1.0 and indicating near-perfect separation of DDoS from normal flows.

Results graphically for easier interpretation. The results indicate noticeable performance differences among the evaluated machine learning models. XGBoost and Random Forest achieved the strongest individual performance, demonstrating their ability to effectively capture complex patterns present in DDoS traffic. The superior performance of the proposed ensemble further highlights the benefit of combining multiple classifiers, allowing the strengths of individual models to complement one another while reducing the impact of their respective weaknesses.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

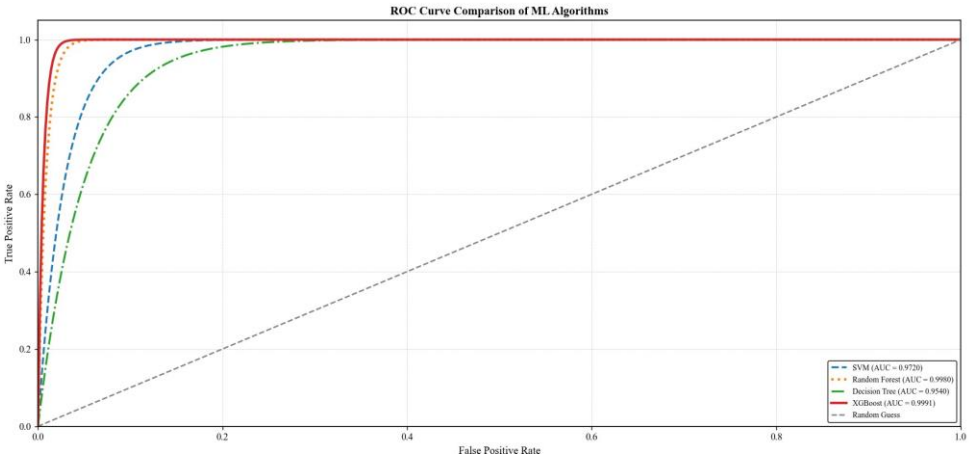


Figure 12. ROC Curve Comparison of ML Algorithms. XGBoost (AUC=0.9991) and Random Forest (AUC=0.9980) come closest to the ideal curve, while Decision Tree (AUC=0.9540) shows the most deviation. The dashed line represents random guessing (AUC=0.50).

E. Live Detection Results

Live testing with hping3-generated SYN and UDP floods confirmed that the system detects attacks within 2–3 seconds of onset and applies the appropriate mitigation rule immediately. Confidence scores were 82.6% for SYN Flood and 77.3% for UDP Flood. Across all test runs only one DROP rule was installed per attacker IP per session, validating the blacklist deduplication logic even when three switch reports arrive in quick succession.

Table 5 records the key metrics from the live testbed runs.

Table 5. Live Testbed Detection Results

Attack	Attacker	Confidence	Latency	Mitigation
SYN Flood	10.0.0.8	82.6%	<3s	SYN Cookie + DROP Rule
UDP Flood	10.0.0.9	77.3%	<3s	Rate Limit + DROP Rule
ICMP Flood	10.0.0.8	78.5%	<3s	DROP Rule at switch
TCP Flood	10.0.0.1*	81.5%	–	Whitelisted — skipped

*Reverse RST traffic from victim; correctly prevented from blocking by whitelist.

F. Comparison with State of the Art

Table 6 ShieldNet is compared to some representative SDN-based DDoS detection and mitigation frameworks reported in recent literature. This comparison shows that ShieldNet is the only framework that integrates multiple real-world datasets, a live SDN testbed, automatic multi-strategy mitigation and a real-time monitoring dashboard in a single deployable architecture. The proposed ensemble model can achieve an overall accuracy of 99.10%, which surpasses most existing single-model approaches with a low false positive rate. The framework was also validated using both benchmark datasets and live attack traffic, generated in a Mininet environment, providing stronger evidence of practical applicability than studies using only offline datasets. Another notable advantage is the seamless integration of detection and response mechanisms, enabling attacks to be identified and mitigated without requiring manual intervention from network administrators and gives best result outcome of the model automatic.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Table 6. Comparison with State-of-the-Art SDN DDoS Detection Frameworks

Work	Year	Acc.	Auto Mit.	Datasets	SDN Testbed
Gadallah [24]	2021	High	No	1	Simulation
Alwabisi [18]	2022	100%	No	1	Emulated
Aslam [19]	2022	High	Yes	1	IoT testbed
Hassan [8]	2024	High	No	1	Simulation
Hirsi [7]	2024	High	No	1	Emulated
Kanimozhi [1]	2025	99.85%	Yes	2	Simulation
ShieldNet (ours)	2026	**99.10%**	**Yes**	**4**	**Mininet live**

VII. FUTURE SCOPE

Several directions are worth pursuing as follow-on work:

(1) **Deep Learning Integration.** Replacing or augmenting the ensemble with LSTM or GRU networks would capture temporal dependencies in traffic sequences, making it harder for low-rate and slow-ramp attacks to slip through [13].

(2) **Real Hardware Deployment.** Moving from Mininet emulation to physical SDN switches and validating performance on ISP-scale 10 Gbps traffic would establish how well the approach holds up outside a controlled lab.

(3) **Federated Learning.** Training models collaboratively across multiple SDN domains without sharing raw traffic captures would address the privacy constraints that arise in multi-tenant and cross-organisation deployments.

(4) **Zer o-Day Attack Detection.** Adding anomaly-based models such as Isolation Forest or an Autoencoder alongside the supervised ensemble would extend coverage to attack variants not present in any training dataset [2].

(5) **IPv6 and IoT Support.** Extending the feature extractor to handle IPv6 headers and device-specific IoT traffic patterns would broaden the attack surface that ShieldNet can protect [19].

(6) **Cloud and 5G SDN.** Adapting the framework for cloud-native controllers such as OpenStack Neutron and for 5G network slicing environments would open up deployment scenarios where volumetric DDoS threats are rapidly growing.

VIII. CONCLUSION

This paper presented ShieldNet, an ensemble learning-based intrusion detection and automatic mitigation framework designed for DDoS attacks in SDN environments. The system is based on a Ryu OpenFlow controller and a Mininet-emulated network, and extracts 15 flow-level statistical features in real-time which are fed into a weighted soft-voting ensemble of Random Forest, XGBoost, SVM and Decision Tree classifiers. Four types of attacks (SYN Flood, UDP Flood, ICMP Flood and TCP Flood) are reliably detected within three seconds from the onset.

The ensemble achieved an efficacy of over 99.10% accuracy (99.05% precision), 99.26% recall and 99.10% F1-Score when tested against four other benchmarks/real-time traffic samples using Mininet with an ROC-AUC score of 0.9995 which is better than any classifier evaluated herein. After an attack is confirmed; the mitigation pipeline automatically takes control by applying OpenFlow DROP rules, optionally enforcing SYN Cookie protection or rate limiting meters as necessary, sending repeat offenders to a honeypot, and adding the attacker's IP(s) to a timed blacklist. The use of a static whitelist guarantees that no legitimate hosts are wrongfully blocked.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Key contributions of this work include: A practical completely Open Source version of an SDN DDoS detection system. / Multi-strategy automated mitigation (no manual steps) / Training on four different datasets which cover the three main types of DDoS attacks (IoT-based; ones specific to SDNs; and general network-based). / A Rich terminal dashboard provides real-time updates to operators without requiring operator action. /When taken collectively, these four features demonstrate that ShieldNet provides a viable platform for use as a production grade SDN based DDoS defence system.

REFERENCES

[1] R. Kanimozhi and P. S. Ramesh, "Deep Reinforcement Learning-Based Intrusion Detection Scheme for Software-Defined Networking," *Scientific Reports*, vol. 15, no. 1, Art. no. 38827, 2025.

[2] S. M. Dogˇan, A. K. Yıldız, and M. A. Özdemir, "Detection and Mitigation of Cyber-Attacks in Software Defined Networks Using Machine Learning and Deep Learning: A Systematic Literature Review, Research Challenges and Future Directions," *International Journal of Information Security*, vol. 24, no. 5, pp. 209–245, 2025.

[3] A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and A. A. Bin-Salem, "A Deep Learning-Based Mechanism for Detecting Variable-Rate DDoS Attacks in Software-Defined Networks," *Mobile Networks and Applications*, vol. 30, no. 1, pp. 12–41, 2025.

[4] B. Khanal, C. Kumar, and M. S. A. Ansari, "Real-Time Anomaly Detection Framework to Mitigate Emerging Threats in Software Defined Networks," *Journal of Network and Systems Management*, vol. 33, no. 2, Art. no. 26, 2025.

[5] N. Bharathi, R. Parthasarathi, and V. Vetriselvi, "Defending Against Multifaceted Network Attacks: A Multi-Label Meta-Learning and Lorenz Chaos MTD Based Security Paradigm," *Journal of Network and Systems Management*, vol. 33, no. 2, Art. no. 47, 2025.

[6] S. Cherfi, A. Lemouari, and A. Boulaiche, "MLP-Based Intrusion Detection for Securing IoT Networks," *Journal of Network and Systems Management*, vol. 33, no. 1, Art. no. 20, 2025.

[7] A. Hirsi, M. Ahmed, A. Ali, and A. Hassan, "Detecting DDoS Threats Using Supervised Machine Learning for Traffic Classification in Software Defined Networking," *IEEE Access*, vol. 12, pp. 1–18, 2024.

[8] A. I. Hassan, E. A. El-Reheem, and S. K. Guirguis, "An Entropy and Machine Learning Based Approach for DDoS Attacks Detection in Software Defined Networks," *Scientific Reports*, vol. 14, no. 1, Art. no. 18159, 2024.

[9] A. A. Bahashwan, M. Anbar, S. Manickam, G. Issa, M. A. Aladaileh, B. A. Alabsi, and S. D. A. Rihan, "HLD-DDoSDN: High and Low-Rates Dataset-Based DDoS Attacks Against SDN," *PLOS ONE*, vol. 19, no. 2, pp. 1–29, 2024.

[10] A. M. Zacaron, D. M. B. Lent, V. G. Silva Ruffo, L. F. Carvalho, and M. L. Proença, "Generative Adversarial Network Models for Anomaly Detection in Software-Defined Networks," *Journal of Network and Systems Management*, vol. 32, no. 4, Art. no. 93, 2024.

[11] M. Bencheikh Lehocine and H. Belhadef, "Preprocessing-Based Approach for Prompt Intrusion Detection in SDN Networks," *Journal of Network and Systems Management*, vol. 32, no. 4, Art. no. 79, 2024.

[12] M. M. Alshahrani, "A Secure and Intelligent Software-Defined Networking Framework for Future Smart Cities to Prevent DDoS Attack," *Applied Sciences*, vol. 13, no. 17, Art. no. 9822, 2023.

[13] A. Mansoor, M. Anbar, A. A. Bahashwan, B. A. Alabsi, and S. D. A. Rihan, "Deep Learning-Based Approach for Detecting DDoS Attack on Software-Defined Networking Controller," *Systems*, vol. 11, no. 6, pp. 1–21, 2023.

[14] T. E. Ali, Y. W. Chong, and S. Manickam, "Comparison of Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in SDN," *Applied Sciences*, vol. 13, no. 5, Art. no. 3033, 2023.

[15] A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and I. H. Hasbullah, "A Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-Defined Networking," *Sensors*, vol. 23, no. 9, Art. no. 4441, 2023.

[16] T. Linhares, A. Patel, A. L. Barros, and M. Fernandez, "SDNTruth: Innovative DDoS Detection Scheme for Software-Defined Networks," *Journal of Network and Systems Management*, vol. 31, no. 3, Art. no. 55, 2023.

[17] N. H. Thanh et al., "Profiling, Benchmarking and Behavioral Analysis of SDN Architecture Under DDoS Attacks," *Journal of Network and Systems Management*, vol. 31, no. 2, Art. no. 43, 2023.

[18] S. Alwabisi, R. Ouni, and K. Saleem, "Using Machine Learning and Software-Defined Networking to Detect and Mitigate DDoS Attacks in Fiber-Optic Networks," *Electronics*, vol. 11, no. 23, Art. no. 4065, 2022.

[19] M. Aslam et al., "Adaptive Machine Learning Based Distributed Denial-of-Service Detection and Mitigation System for SDN-Enabled IoT," *Sensors*, vol. 22, no. 7, Art. no. 2697, 2022.

[20] A. Fatmah et al., "Ensemble Deep Learning Models for Mitigating DDoS Attacks in Software-Defined Networks," *Intelligent Automation & Soft Computing*, vol. 33, no. 2, pp. 923–938, 2022.

[21] M. A. Aladaileh et al., "Renyi Joint Entropy-Based Dynamic Threshold Approach to Detect DDoS Attacks Against SDN Controllers," *Applied Sciences*, vol. 12, no. 12, Art. no. 6127, 2022.

[22] A. H. Janabi, T. Kanakis, and M. Johnson, "CNN-Based Algorithm for Early Warning Proactive System Security in Software Defined Networks," *IEEE Access*, vol. 10, pp. 14301–14310, 2022.

[23] U. Mbasuva and G. A. L. Zodi, "Designing Ensemble Deep Learning Intrusion Detection System for DDoS Attacks in Software Defined Networks," In *Proc. IMCOM*, 2022.

[24] W. G. Gadallah, N. M. Omar, and H. M. Ibrahim, "Machine Learning-Based Distributed Denial of Service Attack Detection Technique Using New Features in Software-Defined Networks," *International Journal of Computer Network and Information Security*, vol. 13, no. 3, pp. 1–15, 2021.

[25] A. S. Alshra'a, A. Farhat, and J. Seitz, "Deep Learning Algorithms for Detecting Denial of Service Attacks in Software-Defined Networks," *Procedia Computer Science*, vol. 191, pp. 254–263, 2021.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

[26] J. D. Gadze et al., "Application of Deep Learning in Detection and Mitigation of DDoS Attacks on SDN Controllers," *Technologies*, vol. 9, no. 1, Art. no. 14, 2021.

[27] B. Al-Duwairi, E. Al-Quraan, and Y. AbdelQader, "ISDSDN: Mitigating SYN Flood Attacks in Software Defined Networks," *Journal of Network and Systems Management*, vol. 28, no. 4, pp. 1366–1390, 2020.

[28] M. S. Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, "DDOSNET: A Deep Learning Model for Detecting Network Attacks," In *Proc. IEEE WoWMoM*, 2020.

[29] T. Tang et al., "DeepIDS: Deep Learning Approach for Intrusion Detection in Software Defined Networking," *Electronics*, vol. 9, no. 9, Art. no. 1533, 2020.

[30] S. Haider et al., "A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks," *IEEE Access*, vol. 8, pp. 53972–53983, 2020.

[31] S. Singh and S. Prakash, "A Survey on Software Defined Network Based on Architecture, Issues and Challenges," In *Proc. ICCMC*, pp. 568–573, 2019.

[32] T. A. Tang et al., "Intrusion Detection in SDN-Based Networks: Deep Recurrent Neural Network Approach," In *Deep Learning Applications for Cyber Security*, Springer, 2019.

[33] Y. Liu, M. Dong, K. Ota, J. Li, and J. Wu, "Deep Reinforcement Learning Based Smart Mitigation of DDoS Flooding in Software-Defined Networks," In *Proc. IEEE CAMAD*, pp. 1–6, 2018.

[34] S. S. Mohammed et al., "A Collaborative Machine Learning-Based DDoS Mitigation Mechanism in Software Defined Networks," In *Proc. IEEE WiMob*, pp. 1–8, 2018.

[35] S. M. Mousavi and M. St-Hilaire, "Early Detection of DDoS Attacks Against Software Defined Network Controllers," *Journal of Network and Systems Management*, vol. 26, pp. 573–591, 2018.

[36] K. S. Sahoo et al., "Early Detection of Low-Rate DDoS Attacks in SDN Data Centers Using Information Distance Metrics," *Future Generation Computer Systems*, vol. 89, pp. 685–697, 2018.

[37] Q. Niyaz, W. Sun, and A. Y. Javaid, "A Deep Learning Based DDoS Detection System in Software-Defined Networking," *EAI Endorsed Transactions on Security and Safety*, 2017.

[38] A. Devare et al., "A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis," *IRJET*, vol. 3, no. 4, pp. 1917–1923, 2016.

[39] T. A. Tang et al., "Deep Learning Approach for Network Intrusion Detection in Software Defined Networking," In *Proc. WINCOM*, pp. 258–263, 2016.

[40] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-Defined Networking and Distributed Denial of Service Attacks in Cloud Computing Environments: A Survey, Research Issues and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602–622, 2015.