Enhancing SDN Security and Availability with Blockchain and Dual-Layer Isolation Forest–Driven DDoS Detection

Ahmed Belkhadim
RITM – ESTC/CED, ENSEM Hassan II University, Casablanca, Morocco
ahmed.belkhadim@gmail.com (corresponding author)

Abdelilah Chahid
Laboratory of Modeling and Simulation of Intelligent Industrial Systems, ENSET of Mohammedia, Hassan II University of Casablanca, Morocco
chahidabdelillah@gmail.com

Adil Hilmani
LASTIMI, High School of Technology Sale, Mohammed V University Rabat, Morocco
adilhilmani@gmail.com

Abdelaziz Ettaoufik
LIAS, Faculty of Sciences Ben M'sick, Hassan II University of Casablanca, Morocco
aettaoufik@gmail.com

Abderrahim Maizate
RITM – ESTC/CED, ENSEM Hassan II University, Casablanca, Morocco
abderrahim.maizate@etu.univh2c.ma

Khalifa Mansouri
Laboratory of Modeling and Simulation of Intelligent Industrial Systems, ENSET of Mohammedia, Hassan II University of Casablanca, Morocco
khalifa.mansouri@enset-media.ac.ma

Received: 1 February 2026 | Revised: 18 March 2026 | Accepted: 1 April 2026

 DOI: https://doi.org/10.48084/etasr.17899

ABSTRACT

Software-Defined Networking (SDN) improves network programmability and centralized control, yet it remains vulnerable to Distributed Denial-of-Service (DDoS) attacks, particularly those targeting SDN controllers and flow-table management. This paper proposes a double-layer DDoS defense framework that integrates consortium blockchain and machine learning to enhance security and reliability in SDN environments. The architecture deploys a Financial Blockchain Shenzhen Consortium (FISCO)-BCOS consortium blockchain at the controller's northbound interface to securely store and validate flow-table information through smart contracts. To strengthen control-plane resilience, a primary–secondary controller configuration (CM/MS) is introduced, where controllers synchronize validated flow rules via blockchain consensus and support seamless failover. DDoS mitigation is performed using a two-tier strategy: (i) a time-window frequency analysis of blockchain-recorded flow data combined with a token bucket mechanism to detect and limit high-rate flooding sources, and (ii) a composite feature selection process coupled with an Isolation Forest model to detect stealthy low-rate attacks. Experiments conducted on a Mininet-based SDN testbed using the CIC-DDoS2019 dataset demonstrate that the proposed framework achieves 92.29% detection accuracy while preserving stable network transmission behavior. Results indicate that blockchain-based flow validation and controller redundancy improve SDN security and reliability without measurable degradation in Round-Trip Time (RTT) performance.

Keywords-Software-Defined Networking (SDN); blockchain; smart contracts; DDoS detection; Isolation Forest; token bucket; SDN controller

I. INTRODUCTION

Modern networking requires ever-increasing flexibility and efficiency, especially given the rapidly evolving nature of Internet technologies. As a result, Software-Defined Networking (SDN), a novel paradigm for designing, managing, and optimizing networks, has gained significant attention. A fundamental design principle of SDN is the decoupling of the control plane from the data-forwarding plane, which transforms how networks are constructed and managed.

The decoupled control plane is enabled through a single logical unit called the SDN controller, which consolidates many control plane functionalities previously spread across individual network elements. The controller provides a centralized view of the network and therefore allows for streamlined deployment of new services and/or application of management policies, as well as the dynamic reconfiguration of network operation.

Together, these interfaces enable SDN systems to create a programmable network infrastructure [1]. There are now a number of mature protocols that support SDN architectures, with OpenFlow being the most commonly deployed and accepted standard [2]. Distributed Denial-of-Service (DDoS) attacks attempt to disrupt legitimate usage of network services by flooding the targeted system with a vast volume of traffic; this can eventually exhaust the resources of the compromised machines involved in the attack [3].

Research has shown that DDoS attacks represent a very serious vulnerability within SDN architectures. Specifically, these attacks take advantage of the weaknesses in SDN systems by filling up switch flow tables, consuming all of the controller's processing capacity, and/or saturating the available bandwidth of links. As the use of SDN systems continues to increase, the damage caused by DDoS attacks is also likely to continue to rise. The current threat landscape identifies DDoS attacks as one of the largest threats to the security of computer systems today. In addition, while some mitigation techniques have been proposed to date, none have proven fully effective. In the 5G environment, with a growing number of connected devices (i.e., Internet of Things), the size and complexity of DDoS attacks will be larger than ever before, thus creating a greater need for more robust DDoS defense mechanisms specifically designed for SDN architectures. SDN systems provide high-efficiency data exchanges among multiple domains within flexible network structures; however, large-scale DDoS attacks may create a massive amount of malicious traffic, which can degrade the performance of networks and consume excessive resources when communicating across domain boundaries.

Recent advancements in SDN technology, including the use of blockchain technology, now provide the opportunity to enhance the security of SDN systems. In particular, blockchain technology provides a promising basis for enabling cost-effective, scalable, and efficient authorization among multiple domains. Blockchain enables decentralized networks and smart contracts to facilitate multi-domain information sharing while building trust among parties who do not rely on one central authority. Distributed ledger systems (Bitcoin, Ethereum) and blockchain technologies demonstrate high levels of security, transparency, and resilience in numerous areas of application. Therefore, integrating blockchain technology into SDN architectures to mitigate DDoS attacks may represent an effective method for creating next-generation decentralized defensive mechanisms.

In this context, authors in [4] completed a comprehensive review of all of the work that has been done in the area of SDN and identified the key problems that need to be solved to defend against DDoS attacks. Authors in [5] proposed a blockchain framework for collaborative DDoS mitigation in SDN, including secure storage of stream data and coordination via smart contracts in an SDN environment. Authors in [6] utilized blockchain to create global trusted relationships and developed a reputation-based evaluation system to improve both the security and routing reliability of the network. Authors in [7] created a framework called blockCSDN, which utilizes blockchain in conjunction with SDN to manage and detect intrusions. Authors in [8] created a new DDoS detection algorithm, which uses a combination of a Random Forest classifier with heterogeneous feature selection.

Recent studies show a convergence toward the use of intelligent and distributed techniques to enhance the security of SDN networks against DDoS attacks. Authors in [9], as well as authors in [5], demonstrated that machine learning models combining feature selection and ensemble learning significantly improve detection accuracy and efficiency. Authors in [10] proposed an innovative blockchain-based approach that enables collaborative, scalable, and lightweight mitigation, particularly suited to new-flow DDoS attacks in large-scale autonomous environments.

Prior research on DDoS protection systems utilizing blockchain technologies shows that nearly all studies have employed methods based on subjective decision-making (i.e., blocklists and/or whitelists) to determine what is permitted and what is blocked from accessing a system's network resources. Additionally, several studies have shown that the use of blockchain technology can increase the performance overhead associated with DDoS protection and may negatively affect the overall stability of data transmissions within SDN environments.

The main limitation in prior research on DDoS protection using blockchain technologies is, therefore, the need to develop methods that protect against DDoS attacks without incorporating subjective decision-making processes. This study proposes a double-layer DDoS defense strategy using blockchain technology and includes the following main contributions:

- The formulation of DDoS attack detection as a binary classification task in which network traffic in an SDN environment is classified as being either benign or malicious. A public collection of 77 flow-based statistical

attributes related to 11 different types of DDoS attacks, called the CIC-DDoS2019 dataset [11, 12], is used to simulate real-world operational conditions.

- An approach that combines (i) Financial Blockchain Shenzhen Consortium (FISCO)-BCOS–backed storage for SDN flow tables exposed through the POX controller northbound interface, aimed at reducing DDoS impact, and (ii) a smart-contract–enabled cross-domain traffic transmission mechanism that protects the integrity of on-chain flow rules and accelerates their retrieval by the controller.
- A hybrid detection approach that combines the Isolation Forest algorithm with time-series analysis and a token bucket mechanism. This design enables the creation of a two-level detection framework that can handle the complexity and variability of real-world DDOS attacks. Additionally, the calculation of feature weight coefficients is outlined to optimize the selection of relevant flow properties when analyzing the CIC-DDoS2019 dataset.

II. METHODOLOGY

This section describes how the proposed DDoS mitigation approach is implemented using two main components. First, at the SDN northbound interface, a FISCO-BCOS consortium blockchain is used to create a secure environment for storing and accessing flow-table data. Second, inter-domain processes use smart contracts to collect and store traffic information at predefined intervals.

The DDoS mitigation process within each domain is structured into two layers. Layer 1 focuses on identifying and eliminating flood-type, high-volume DDoS attacks that are easily detectable. Layer 2 employs the Isolation Forest algorithm and focuses on detecting subtler forms of DDoS attacks that are more difficult to identify than those addressed by Layer 1. When combined, these layers provide an effective layered protection system against both obvious and stealthy DDoS attacks.

A. Flow Table Data Processing

Within SDN topologies, the forwarding procedure closely mirrors conventional routing and forwarding mechanisms. When the data plane requires guidance, it submits a forwarding request to the SDN controller through the southbound interface. To issue precise control instructions and ensure correct packet handling, the controller must consult the flow table.

In the proposed approach, a FISCO-BCOS consortium blockchain is instantiated, and smart contracts are compiled and deployed on the blockchain. Designated SDN controllers are then enrolled on-chain and subjected to authentication and authorization. After successful verification, approved controllers use blockchain APIs, exposed via the northbound interface, to interact with a FISCO-BCOS client and record flow-table information. By embedding authorization within a blockchain-based mechanism, the process becomes more transparent and traceable.

Accordingly, the smart contracts support the following functions: (i) the smart contract owner verifies whether a given participant has the required permission to access the blockchain network, and (ii) once validated, authorized contract participants are allowed to modify and update their local flow-table records.

To implement these functionalities, this study introduces a dedicated smart-contract architecture implemented in Solidity. Table I summarizes the contract-scoped variables (i.e., variables maintained within the contract) along with their meanings.

TABLE I. VARIABLES IN THE SMART CONTRACT

Variable	Description
User	Represents various categories of external user accounts
AuthorizedAddr	Contains the blockchain addresses of participants involved in the collaborative strategy
Authorized	A mapping structure that links each collaborator's address to its associated data structure
Data	Refers to a message carrying network-related information
macExists	A mapping (src_mac → bool) indicating whether a source MAC address has already been registered
Index	A numerical identifier used to specify a particular target entity in a sequence

1) User Authorization Verification

This function verifies whether a given external account identifier (user) is a valid collaborator. It first checks whether the list of collaborator addresses is non-empty. It then verifies that the account exists in the smart contract structure and that the associated index is valid (i.e., it is an integer and lies within the bounds of the collaborator address list).

Finally, the function confirms that the address stored at the corresponding index matches the given user. The function returns true if all these conditions are satisfied; otherwise, it returns false. The verification process is detailed in Algorithm 1.

```
Algorithm 1: isAuthorizedUser(user)
Input : user, AuthorizedAddr, Authorized
Output: Boolean
1: if length(AuthorizedAddr) == 0 then
2:   return False
3: else
4:   info ← Authorized.get(user)
5:   if info == null then
6:     return False
7:   else
8:     idx ← info["index"]
9:     if idx is not an integer then
10:      return False
11:    else
12:      if idx < 0 OR idx ≥
                 length(AuthorizedAddr) then
13:       return False
14:      else
15:       return (AuthorizedAddr[idx] == user)
16:      end if
17:    end if
18:  end if
19: end if
```

2) Timestamp Retrieval Based on MAC Address

This function determines whether the information provided in the input data already exists in the stored records. It extracts the source MAC address from the input structure and verifies its validity and existence within the system. If the source MAC address is registered, the function retrieves and returns the corresponding index, which represents the timestamp assigned at the time of storage. If the MAC address is null or not found in the records, the function returns null. The detailed retrieval process is presented in Algorithm 2.

```
Algorithm 2: getTimestampByMac(data)
Input : data (struct), idxByMac (mapping
        src_mac → index), macExists
        (mapping src_mac → bool)
Output: index (timestamp) or null
1: mac ← data.src_mac
2: if mac == null then
3:   return null
4: end if
5: if macExists[mac] == false then
6:   return null
7: end if
8: return idxByMac[mac]
```

B. Detection Strategy

The proposed defense is structured into two complementary layers. Layer 1 targets high-rate flooding by monitoring blockchain-stored flow records within a sliding time window, whereas Layer 2 targets stealthy low-rate attacks by applying an Isolation Forest model on flow features after composite feature selection and by reusing rate-limiting mechanisms for sources classified as anomalous.

1) Layer 1: Time-Window Frequency Detection and Token Bucket Mitigation

In this study, the source MAC address and the timestamp stored on-chain are used as the primary monitoring variables. A sliding window of size W = 5 s is defined. For each source MAC address, we compute $N_{mac}(W)$, the number of newly recorded flow entries attributed to that MAC during the current window. A source is flagged as flooding when $N_{mac}(W) > T_{flood}$, where T_{flood} = 200 records per 5 s per MAC in the Mininet topology.

Once a MAC address is flagged, its traffic is constrained using a token bucket policer configured with a bucket capacity B_{max} = 1,000 tokens and a token generation rate R = 300 tokens/s, where one token authorizes the forwarding of one packet. The token count is updated as:

$$B(t) = \min(B_{max}, B(t - \Delta t) + R \cdot \Delta t) \quad (1)$$

Packets are forwarded only if $B(t) \geq 1$; otherwise, they are dropped or shaped.

2) Layer 2: Isolation Forest–Based Stealth Attack Detection

The second layer aims to detect stealthy and low-rate DDoS attacks using an Isolation Forest model trained on CIC-DDoS2019 flow features after data preprocessing and composite feature selection. Features with a large proportion of missing values are first removed, reducing the feature space from 77 to 63 attributes. Next, features are ranked using a composite approach that combines Mutual Information, Random Forest feature importance, and Recursive Feature Elimination, as summarized in Table II.

TABLE II. COMPOSITE FEATURE SELECTION STRATEGY

Algorithm	Description
Mutual Information	Quantifies the amount of shared information between a feature and the target variable Y, indicating how the presence or absence of a given feature contributes to accurate prediction
Random Forest	Enhances model robustness by aggregating multiple decision trees, thereby reducing variance and limiting overfitting risk
Recursive Feature Elimination	Iteratively trains the model while progressively discarding features with low relevance, leading to the optimal feature subset

Based on validation performance on the SYN traffic subset, the top seven most relevant features associated with SYN-based attack behavior are retained, as reported in Table III. These features capture key characteristics of SYN flood traffic, including packet size distribution, flow packet rates, acknowledgment flag behavior, and temporal idle patterns.

TABLE III. FEATURES SELECTED BY THE PROPOSED METHOD

Feature	Role
ACK Flag Count	Count of ACK (Acknowledgment) flags
Avg Packet Size	Average size of packets in a flow
Subflow Fwd Bytes	Number of bytes forwarded in subflows
Total Backward Packets	Total number of backward packets in the flow
Idle Mean	Average idle time between packets
Flow Packets/s	Packet transmission rate within the flow
Packet Length Mean	Mean packet length in the flow

The Isolation Forest model is trained with the following hyperparameters: n_estimators = 184, max_samples = 256 (or all samples if N < 256), max_features = 1.0, bootstrap = false. A fixed random state is used to ensure reproducibility.

Each sample x is assigned an anomaly score $s(x)$, reflecting its degree of abnormality based on the average path length across the ensemble of trees. Since anomaly detection requires a decision threshold, the threshold τ is determined using a benign validation subset by computing the 99th percentile of anomaly scores, i.e., $\tau = Q0.99(s_{benign})$. A sample is classified as malicious when $s(x) \geq \tau$ (binary decision).

The Layer 2 training and testing protocol for the SYN subset is summarized in Table IV.

TABLE IV. LAYER 2 TRAINING/TESTING PROTOCOL (SYN SUBSET)

Set	Size	Composition	Purpose
Training	70,336	BENIGN SYN only	Learn normal baseline (unsupervised Isolation Forest) and calibrate τ (Q99 of benign scores)
Test	907	BENIGN SYN + SYN anomalies	Evaluate binary detection performance using TP/TN/FP/FN and derived metrics

III. RESULTS AND DISCUSSION

In this section, an SDN topology is deployed, and basic connectivity tests, such as ICMP ping exchanges between hosts, are conducted to evaluate the effect of the proposed defense mechanism on SDN switches. Concerning DDoS detection, the first protection layer demonstrates full effectiveness in identifying flooding attacks once the predefined threshold is exceeded. Attacks that are more challenging to identify are emulated using traffic from the CIC-DDoS2019 dataset, which is recorded on the blockchain and subsequently analyzed by the second detection layer. The performance of this layer is then assessed in terms of DDoS detection accuracy.

A. Experimental Setup and Network Topology

The SDN topology was emulated using Mininet [13], whereas the control plane was orchestrated by the POX controller [14], and Scapy was employed to generate traffic at the host level. As shown in Figure 1, the experimental setup includes a FISCO-BCOS consortium blockchain [15], two SDN controllers organized in a primary–secondary (master/slave) configuration, three Open vSwitch switches, and six hosts, with each switch connected to two hosts to form an SDN subnet. Both controllers act as authorized participants in the FISCO-BCOS blockchain, which was implemented through the Python SDK and deployed smart contracts using the WeBASE framework [16].

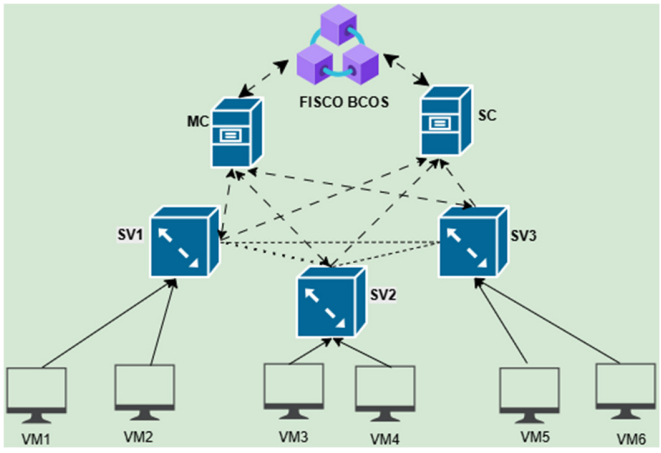

Fig. 1. Blockchain-enabled SDN architecture with primary–secondary controller for DDoS defense.

At network initialization, switches performed address discovery through broadcast mechanisms, and flow-related information processed by the primary controller, such as source MAC addresses and timestamps, was stored on the blockchain as flow-table records. During normal operation, packet forwarding relied on the primary controller to obtain validated flow rules from the blockchain, whereas the secondary controller remained synchronized via the blockchain consensus process and was capable of taking over control in the event of primary controller failure, thus improving control-plane robustness without degrading regular data forwarding.

B. Evaluation of Flow-Table Forwarding Performance Compared to the Native Controller

The proposed method for detecting and mitigating DDoS attacks influences the standard forwarding process in flow tables managed by the controller. To quantify this impact, a comparative analysis was conducted on two configurations: an improved SDN controller that utilizes blockchain-based flow-table storage based on FISCO-BCOS at the northbound interface, and a standard POX controller running without the proposed method. The goal of the comparison is to evaluate the forwarding behavior and controller stability under both configurations. The mean deviation of round-trip time (mdev) reflects network transmission stability by quantifying the average variability of Round-Trip Time (RTT) between communicating hosts.

To evaluate mdev, ICMP ping tests were conducted between two hosts in the same SDN topology (H1S1 and H2S3), with the number of requests ranging from 10 to 60. The mdev values were recorded for both the standard POX controller and the POX controller enhanced with the proposed blockchain-based strategy, and the results are presented in Figure 2. Both controllers show similar trends in mdev values as the number of ping requests increases.

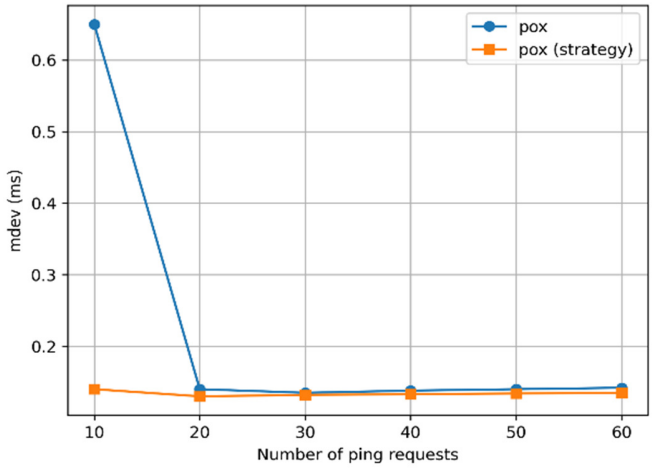

Fig. 2. Comparison of RTT stability between standard POX controller and blockchain-enhanced strategy.

An initial transient phase is observed at a small number of ping requests; after this initial phase, the mdev values quickly stabilize and remain comparable across all other ping requests (i.e., 20–60 requests). The proposed blockchain-based approach adds negligible latency compared to the standard POX controller. In some cases, the mdev values of the controller enabled by the blockchain strategy are slightly larger than those of the standard controller, whereas in other cases the mdev values for the strategy-enabled controller are slightly smaller. Thus, the stability achieved by both methods is comparable.

The findings suggest that adding blockchain-based flow-table storage and northbound smart contracts does not introduce measurable instability in RTT. Although additional

control-plane operations are needed to interact with the blockchain, the network exhibits the same stable transmission behavior as the non-blockchain controller. Therefore, the proposed approach maintains normal forwarding functionality while enhancing security capabilities against DDoS attacks.

C. Layer 1 Flooding Detection Performance

To support Layer 1 flooding detection claims with quantitative evidence, we report the True Positive Rate (TPR) (recall), False Positive Rate (FPR), and detection latency under the configured parameters. Layer 1 operates on blockchain-stored flow records using a sliding window of size W=5 s. For each source MAC address, the controller computes $N_{mac}(W)$, i.e., the number of newly recorded flow entries attributed to that MAC within the current window. A source is flagged as flooding when $N_{mac}(W) > T_{flood}$, where T_{flood} = 200 records per 5 s per MAC in the Mininet topology. Once flagged, traffic from the corresponding MAC address is rate-limited using a token bucket policer with capacity B_{max} = 1,000 tokens and token generation rate R = 300 tokens/s (1 token per forwarded packet).

The evaluation is conducted over 120 sliding windows of W = 5 s (10 min total), comprising 54 benign and 66 flooding windows generated using Scapy. A detection event is recorded when the flooding condition is satisfied for at least one source MAC address within a window.

Performance is computed using:

$$TPR = \frac{TP}{TP+FN} \quad (2)$$

$$FPR = \frac{FP}{FP+TN} \quad (3)$$

where TP and FN correspond to flooding windows, whereas FP and TN correspond to benign windows.

The results show that Layer 1 achieves TPR = 90.91% (TP = 60, FN = 6) and FPR = 7.41% (FP = 4, TN = 50). Detection latency is defined as time-to-flag, i.e., the time until the first window for which $N_{mac}(W) > T_{flood}$ becomes true. Since detection is window-based, the latency is upper-bounded by one window, resulting in time-to-flag ≤ 5 s.

D. Layer 2 DDoS Detection Performance

To evaluate the effectiveness of the proposed second-layer detection model, the classification results are analyzed using the confusion matrix shown in Figure 3. In this matrix, true positives (TP) correspond to attack flows correctly identified as attacks, whereas true negatives (TN) represent benign flows correctly classified as normal traffic. False positives (FP) denote benign flows incorrectly classified as attacks, whereas false negatives (FN) correspond to attack flows misclassified as benign.

Based on the test set of 907 flows, the confusion matrix yields 250 TP, 587 TN, 20 FP, and 50 FN. These results indicate that the proposed model correctly classifies the majority of both benign and malicious flows while maintaining a relatively low number of false alarms.

The overall classification accuracy (ACC) is computed as:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

Using (4), the proposed model achieves an accuracy of 92.29%, confirming its strong capability for DDoS attack detection.

In addition to accuracy, several complementary evaluation metrics are used to assess detection performance. The model achieves a precision of 92.59%, indicating that most flows predicted as attacks are indeed malicious. The recall (TPR) reaches 83.33%, reflecting the model's ability to correctly detect attack instances. The F1-score of 87.72% demonstrates a good trade-off between detection capability and false alarms. Furthermore, the relatively small number of false positives (20) indicates a low false alarm rate, which is essential for practical deployment in SDN environments.

These results demonstrate that the proposed composite feature selection combined with Isolation Forest provides strong binary anomaly detection performance. The improved detection capability is attributed to the selection of highly informative flow features, enabling effective characterization of abnormal traffic patterns associated with DDoS attacks.

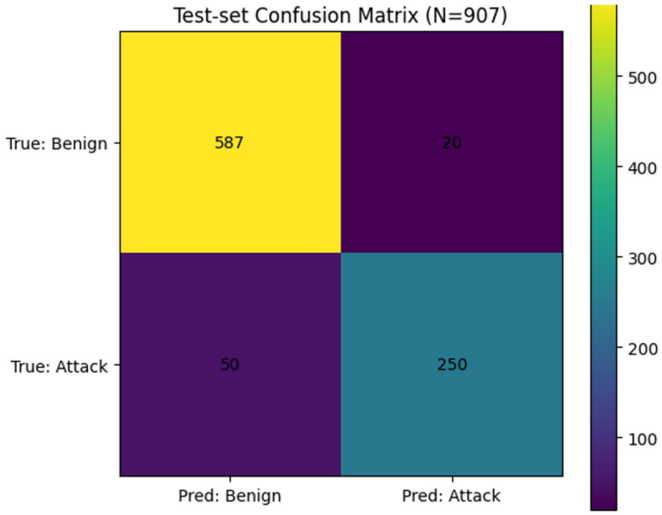

Fig. 3. Confusion matrix for Layer 2 DDoS detection using Isolation Forest.

E. Comparison With Existing Methods

Table V shows that the proposed method (Proposed + Isolation Forest) outperforms the approaches reported in studies [3, 17, 18]. The proposed approach achieves 92.29% accuracy, 92.59% precision, a ROC-AUC of 0.94, and a detection time of 3.1 s (12 ms latency). In comparison, study [3] reports approximately 90% accuracy, study [18] achieves about 91%, whereas study [17] shows significantly lower performance at 74.33% accuracy.

Beyond improved detection performance, the proposed framework introduces a two-layer detection architecture, combining rapid flooding detection with machine learning–based analysis using Isolation Forest to identify stealthy attacks. Furthermore, unlike several existing approaches, the

integration of blockchain and controller redundancy enhances system security while preserving network stability (RTT), demonstrating the robustness and practical effectiveness of the proposed framework.

TABLE V. PREDICTION ACCURACY COMPARISON ACROSS FEATURE SELECTION METHODS

Method	Layers	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ROC-AUC	Detection time (s) / latency (ms)
Proposed + IF	2	92.29	92.59	83.33	87.72	0.94	3.1 / 12
Lasso + IF	2	87.46	91.15	68.67	78.30	0.89	3.4 / 13
Pearson + IF	2	85.47	90.38	62.67	74.00	0.87	3.7 / 12
Spearman + IF	2	84.55	90.00	60.00	72.00	0.86	3.4 / 16
Kernel + IF	2	83.12	89.20	55.67	68.50	0.84	3.6 / 17
Distance correlation + IF	2	82.13	88.76	52.67	66.10	0.83	3.3 / 20
[3]	1	90.00	89.00	88.00	89.00	–	–
[17]	1	74.33	69.65	–	82.11	–	–
[18]	1	91.00	92.00	90.00	92.00	–	–

Note: IF = Isolation Forest

IV. CONCLUSIONS

The proposed framework employs blockchain technology and machine learning techniques to address security issues in Software-Defined Networking (SDN) systems. The architecture integrates smart contracts, a token bucket mechanism, an Isolation Forest classifier, and a composite feature selection method for the detection and mitigation of Distributed Denial-of-Service (DDoS) attacks. The smart contract collects and validates periodic network flow information from each domain, while the two-tier detection mechanism identifies both high-rate flooding attacks and stealthy DDoS behaviors within the network.

The SDN control plane is configured with a primary–secondary controller architecture (CM/MS), consisting of two controllers operating as authorized members of the consortium blockchain. The primary controller (CM) installs flow rules and handles normal network operations, whereas the secondary controller (MS) continuously synchronizes validated flow tables through the blockchain consensus process. This design enables real-time validation of control decisions and provides fault tolerance. In the event of primary controller failure, the secondary controller seamlessly assumes the primary role without interrupting ongoing data forwarding.

Experimental results demonstrate that the proposed solution maintains stable network transmission while accurately detecting DDoS attacks, thus confirming the effectiveness of combining blockchain-based validation with redundant controller design. However, several limitations remain, including the relatively limited size of the experimental dataset and the restricted scope of validation scenarios. Therefore, future work will evaluate the framework using larger and more diverse DDoS datasets to further assess detection robustness and will refine the composite feature selection process to improve detection performance and overall system resilience.

DECLARATION OF COMPETING INTERESTS

The authors declare no competing interests.

ACKNOWLEDGMENT

Not applicable to this work.

DATA AVAILABILITY

The CIC-DDoS2019 dataset [11, 12] was used in this study.

REFERENCES

[1] M. U. Younus, S. ul Islam, I. Ali, S. Khan, and M. K. Khan, "A survey on software defined networking enabled smart buildings: Architecture, challenges and use cases," *Journal of Network and Computer Applications*, vol. 137, pp. 62–77, July 2019, https://doi.org/10.1016/j.jnca.2019.04.002.

[2] H. Riggs, A. Khalid, and A. I. Sarwat, "An Overview of SDN Issues—A Case Study and Performance Evaluation of a Secure OpenFlow Protocol Implementation," *Electronics*, vol. 14, no. 16, Aug. 2025, Art. no. 3244, https://doi.org/10.3390/electronics14163244.

[3] R. Basfar, M. Y. Dahab, A. M. Ali, F. Eassa, and K. Bajunaied, "Enhanced Intrusion Detection in Software-Defined Networking using Advanced Feature Selection: The EMRMR Approach," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 19001–19008, Dec. 2024, https://doi.org/10.48084/etasr.9256.

[4] A. Kaur, C. Rama Krishna, and N. V. Patil, "A comprehensive review on Software-Defined Networking (SDN) and DDoS attacks: Ecosystem, taxonomy, traffic engineering, challenges and research directions," *Computer Science Review*, vol. 55, Feb. 2025, Art. no. 100692, https://doi.org/10.1016/j.cosrev.2024.100692.

[5] S. Garg, S. Goyal, and A. Bhandari, "A lightweight blockchain based scalable and collaborative mitigation framework against new flow DDoS attacks in SDN enabled autonomous systems," *Scientific Reports*, vol. 15, no. 1, Oct. 2025, Art. no. 36002, https://doi.org/10.1038/s41598-025-19989-2.

[6] Z. Zeng, X. Zhang, and Z. Xia, "Intelligent Blockchain-Based Secure Routing for Multidomain SDN-Enabled IoT Networks," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, Feb. 2022, Art. no. 5693962, https://doi.org/10.1155/2022/5693962.

[7] W. Li, Y. Wang, W. Meng, J. Li, and C. Su, "BlockCSDN: Towards Blockchain-Based Collaborative Intrusion Detection in Software Defined Networking," *IEICE Transactions on Information and Systems*, vol. E105.D, no. 2, pp. 272–279, Feb. 2022, https://doi.org/10.1587/transinf.2021BCP0013.

[8] R. Ma, Q. Wang, X. Bu, and X. Chen, "Real-Time Detection of DDoS Attacks Based on Random Forest in SDN," *Applied Sciences*, vol. 13, no. 13, July 2023, Art. no. 7872, https://doi.org/10.3390/app13137872.

[9] A. V. Kachavimath and N. D g, "An Efficient DDoS Attack Detection in SDN using Multi-Feature Selection and Ensemble Learning," *Procedia Computer Science*, vol. 252, pp. 241–250, Jan. 2025, https://doi.org/10.1016/j.procs.2024.12.026.

[10] Md. E. Haque, A. Hossain, Md. S. Alam, A. H. Siam, S. M. F. Rabbi, and Md. M. Rahman, "Optimizing DDoS Detection in SDNs Through Machine Learning Models," in *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks*, Indore, India, 2024, pp. 426–431, https://doi.org/10.1109/CICN63059.2024.10847458.

[11] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *2019 International Carnahan Conference on Security Technology*, Chennai, India, 2019, pp. 1–8, https://doi.org/10.1109/CCST.2019.8888419.

[12] "DDoS evaluation dataset (CIC-DDoS2019)." Canadian Institute for Cybersecurity (CIC), University of New Brunswick. [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html.

[13] "Mininet: An Instant Virtual Network on Your Laptop (or Other PC)." Mininet. http://mininet.org/.

[14] N. O. X. Repo, "noxrepo/pox." Apr. 27, 2026. [Online]. Available: https://github.com/noxrepo/pox.

[15] "Financial Services Blockchain Consortium." GitHub. https://github.com/FISCO-BCOS.

[16] "WeBankBlockchain/WeBASE: WeBASE（WeBank Blockchain Application Software Extension)." Gitee. https://gitee.com/WeBank/WeBASE.

[17] P. Karthika and K. Arockiasamy, "Simulation of SDN in mininet and detection of DDoS attack using machine learning," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1797–1805, June 2023, https://doi.org/10.11591/eei.v12i3.5232.

[18] H. Babbar, S. Rani, and M. Driss, "Effective DDoS attack detection in software-defined vehicular networks using statistical flow analysis and machine learning," *Plos One*, vol. 19, no. 12, Dec. 2024, Art. no. e0314695, https://doi.org/10.1371/journal.pone.0314695.