



ARTICLE

Optimizing Network Security at the Control Plane through Software Defined Networking (SDN)

Ifeanyi C. Emeto^{*}, Adamu A. Galadima, Ikechukwu H. Ezeh, Emmanuel O. Atomatofa and Christiana A. Okoloegbo

Department of Cyber Security, Federal University of Technology, Owerri, Nigeria

^{*}Corresponding Author: Ifeanyi C. Emeto. Email: ifeanyi.charles14@gmail.com

Received: 22 April 2026; Accepted: 20 July 2026; Published: 14 September 2026

ABSTRACT: Software-Defined Networking (SDN) introduces centralised network control and programmability, but its centralised control plane creates significant security vulnerabilities that can be exploited by cyber attackers. This study proposes and evaluates a hybrid security framework that integrates machine learning (ML)-based anomaly detection with blockchain-based authentication to enhance the security of the SDN control plane. The study aimed to analyse vulnerabilities in SDN architectures, develop an ML-driven attack detection model, secure API communications using blockchain-based authentication, and evaluate the performance and scalability of the proposed framework. Vulnerability assessment was conducted using Nmap and STRIDE threat modelling, while an LSTM-based anomaly detection model was implemented using TensorFlow. Hyperledger Fabric was employed for secure API authentication and key management, while Mininet was used to emulate the SDN environment and generate network traffic and attack scenarios. The CICIDS2017 dataset, developed by the Canadian Institute for Cybersecurity at the University of New Brunswick, was used to evaluate the machine learning-based intrusion detection component. The dataset provides labelled benign and malicious network traffic representing common attack categories and network-flow features suitable for machine learning-based intrusion detection. The proposed framework integrates detected anomalies with blockchain-based authentication and SDN policy enforcement to support automated threat response. The performance of the proposed framework was evaluated based on detection accuracy, false-positive rate, mitigation latency, throughput, and scalability. The framework is designed to provide improved security, responsiveness, and scalability for SDN control planes and can support security applications in modern network environments, including 5G, IoT, and cloud networks.

KEYWORDS: Software-defined networking (SDN); machine learning (ML); anomaly detection; blockchain security; network vulnerabilities

1 Introduction

The rapid evolution of network technologies and the exponential growth of connected devices have intensified the complexity of managing modern network infrastructures, particularly in ensuring robust security. Traditional network architectures rely on distributed control mechanisms, where security policies are enforced through static, device-specific configurations, resulting in fragmented visibility and delayed threat response [1]. Over time, network security has evolved from perimeter-based defence models to more advanced approaches; however, these conventional systems remain inadequate against sophisticated cyber threats such as zero-day exploits, Advanced Persistent Threats (APTs), and Distributed Denial-of-Service (DDoS) attacks, which require dynamic and adaptive defence mechanisms [2].

Traditional network security systems face significant limitations due to their rigid and decentralized structure. Security mechanisms, such as firewalls and intrusion detection systems, often operate independently, making coordinated threat detection and response challenging. Additionally, these systems struggle to cope with the increasing scale and heterogeneity of modern environments, including cloud computing, the Internet of Things (IoT), and fifth-generation (5G) networks. High false-positive rates and delayed response times further reduce their effectiveness in detecting evolving threats [3–6]. The proliferation of resource-constrained IoT devices also complicates the enforcement of comprehensive security policies, thereby exposing networks to internal and lateral attacks [2,7].

In response to these limitations, Software-Defined Networking (SDN) has emerged as a transformative paradigm that separates the control plane from the data plane, enabling centralised network management and programmability [3]. This architecture provides global visibility of network traffic, supports real-time monitoring, and enables the dynamic enforcement of security policies [4]. SDN controllers, such as OpenDaylight and ONOS, can intelligently analyse traffic flows and respond to anomalies by updating forwarding rules or isolating malicious nodes [5]. Furthermore, the integration of machine learning techniques within SDN enhances anomaly detection by analysing traffic patterns and reducing false positives compared with traditional systems [8–12]. SDN also supports advanced security models such as Zero Trust Architectures (ZTAs), which enforce strict access control and continuous verification across network environments [7].

Despite these advantages, the centralised architecture of SDN introduces critical security challenges. The control plane becomes a single point of failure and a prime target for cyberattacks such as man-in-the-middle (MITM) attacks, API exploitation, and unauthorised access [11,13–16]. Studies indicate that many SDN deployments lack robust authentication mechanisms, exposing critical interfaces to compromise [16]. Additionally, the absence of standardised encryption protocols increases the risk of eavesdropping and replay attacks [7,16]. Existing solutions, including machine learning-based intrusion detection systems [17] and blockchain-based authentication mechanisms, have been proposed to address these vulnerabilities; however, these approaches are often implemented independently and do not provide a comprehensive or scalable solution [18].

Furthermore, interoperability challenges between SDN and legacy infrastructures limit the effectiveness of current security frameworks. Hybrid environments often experience inconsistent policy enforcement due to incompatibility between traditional devices and SDN controllers [10,11]. Scalability limitations and the absence of unified standards also hinder efficient threat management and coordinated response across large-scale networks [1,5,10]. These challenges highlight the need for an integrated and intelligent approach to securing SDN environments.

1.1 Research Gap

Although significant progress has been made in applying machine learning and blockchain technologies to network security, most existing approaches treat these solutions independently rather than integrating them into a unified framework. There is a lack of comprehensive systems that simultaneously provide real-time anomaly detection, secure authentication, scalability, and interoperability within the SDN control plane. Furthermore, many existing solutions fail to effectively reduce latency and false-positive rates in large-scale and heterogeneous network environments.

1.2 Contribution of the Study

This study addresses the identified gaps by proposing a hybrid framework that integrates machine learning-based anomaly detection with blockchain-based authentication to enhance the security of the SDN control plane. The key contributions of this study include:

- a. Development of a machine learning-based model for accurate and real-time attack detection
- b. Integration of blockchain technology for secure API communication and key management
- c. Design of a scalable and efficient framework suitable for complex SDN environments
- d. Performance evaluation of the proposed system using simulation tools and real-world datasets

1.3 Limitations of the Study

Despite the promising results achieved by the proposed ML–blockchain integrated framework, several limitations were identified that may affect its practical deployment and long-term performance.

One major limitation relates to the scalability of blockchain technology. Although Hyperledger Fabric provides improved throughput and reduced latency compared to public blockchain systems, performance may degrade as the number of transactions and network nodes increases. In large-scale SDN environments, continuous recording of controller actions and authentication processes could lead to increased storage requirements and potential delays in consensus operations. This may limit the efficiency of the system in extremely high-throughput networks such as large Internet Service Provider (ISP) infrastructures.

Another limitation concerns controller overhead. The integration of machine learning models and blockchain validation mechanisms into the SDN controller introduces additional computational and processing demands. While the system demonstrated acceptable performance under simulated conditions, excessive workloads during large-scale attacks or high traffic volumes may lead to controller bottlenecks, thereby affecting response time and network performance.

The issue of energy consumption also presents a significant challenge. The combined use of machine learning algorithms, continuous traffic monitoring, and blockchain operations increases computational intensity, which may result in higher energy usage. This could be a concern in environments where energy efficiency is critical, such as edge computing or resource-constrained IoT systems.

Furthermore, real-world deployment challenges must be considered. The system was evaluated in a simulated environment using tools such as Mininet, which may not fully capture the complexity and unpredictability of real-world networks. Factors such as hardware limitations, network heterogeneity, legacy system compatibility, and operational constraints could affect system performance when deployed in production environments. Additionally, integrating blockchain and SDN technologies in existing infrastructures may require significant redesign and investment.

Finally, the system may require periodic model updates and maintenance, as machine learning models are susceptible to performance degradation over time due to evolving attack patterns (concept drift). This necessitates continuous monitoring, retraining, and updating of the detection model to maintain optimal accuracy.

2 Literature Review

Software-Defined Networking (SDN) represents a fundamental shift in network architecture by decoupling the control plane from the data plane, thereby enabling centralised, programmable, and flexible network management [3]. This architecture is typically organised into three distinct layers: the application layer, which hosts network management, security, and analytics applications; the control layer, which comprises SDN controllers such as OpenDaylight and ONOS; and the infrastructure layer, consisting of switches and routers responsible for packet forwarding [1]. The layered architecture enhances network flexibility, simplifies network administration, and enables the dynamic deployment and enforcement of security policies through programmable interfaces, including REST APIs and the OpenFlow protocol [5,9].

Fig. 1 illustrates the layered architecture of Software-Defined Networking (SDN), showing the separation of the application, control, and infrastructure layers. The architecture demonstrates how the SDN controller acts as the central intelligence of the network by receiving requests from applications through the northbound interface and communicating with forwarding devices through the southbound interface using OpenFlow. This separation improves programmability, centralized management, and dynamic security policy enforcement.

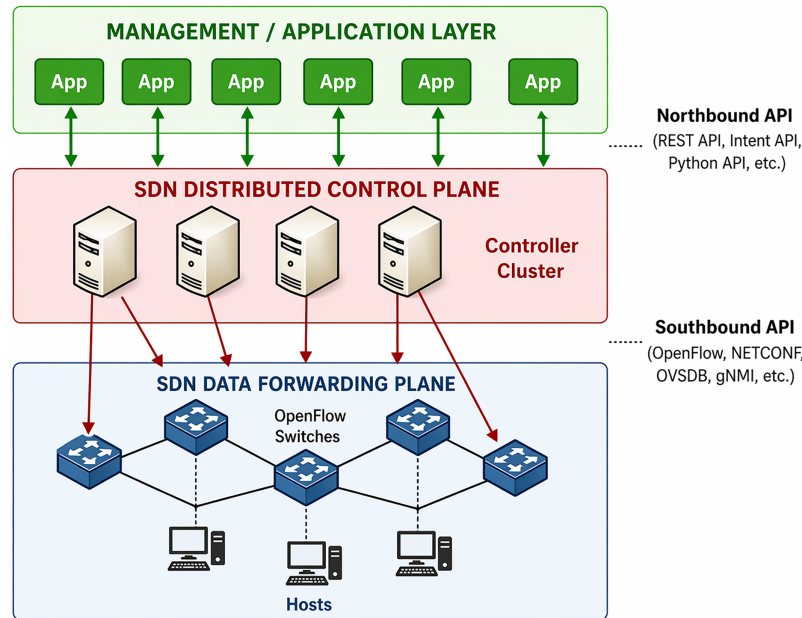


Figure 1: Overview of software-defined networking architecture showing control and data plane separation [3].

Fig. 2 presents the SDN security workflow, illustrating how traffic is continuously monitored by the controller for abnormal activities. The flowchart shows that network traffic is analyzed, suspicious behaviour is identified, and mitigation actions such as updating forwarding rules or blocking malicious traffic are executed automatically. This demonstrates the capability of SDN to provide adaptive and centralized network security.

The separation of the control and data planes enables real-time traffic monitoring and adaptive security responses. SDN controllers maintain a global view of the network, thereby facilitating intelligent decision-making for routing, anomaly detection, and policy enforcement [1,4]. Furthermore, machine learning- and entropy-based traffic analysis techniques have been shown to improve Distributed Denial-of-Service (DDoS) attack detection accuracy while reducing false-positive rates compared with conventional network security mechanisms [12,13]. However, this centralised architecture also introduces scalability challenges, particularly in large-scale network deployments where controller performance may degrade as the network expands [11].

Communication between the control and data planes is facilitated through southbound protocols such as OpenFlow, which play a critical role in SDN operations [9]. Although OpenFlow enables fine-grained traffic control and network programmability, it also introduces security vulnerabilities. Unsecured communication channels may expose SDN environments to threats such as eavesdropping, flow-rule manipulation, and man-in-the-middle (MITM) attacks [2,13–16]. To mitigate these risks, security mechanisms including Transport Layer Security (TLS) and Role-Based Access Control (RBAC) have been proposed to secure controller-switch communications [5,11]. Nevertheless, interoperability challenges between SDN controllers

and legacy network infrastructures continue to hinder effective policy enforcement and seamless integration within hybrid networking environments [10,11].

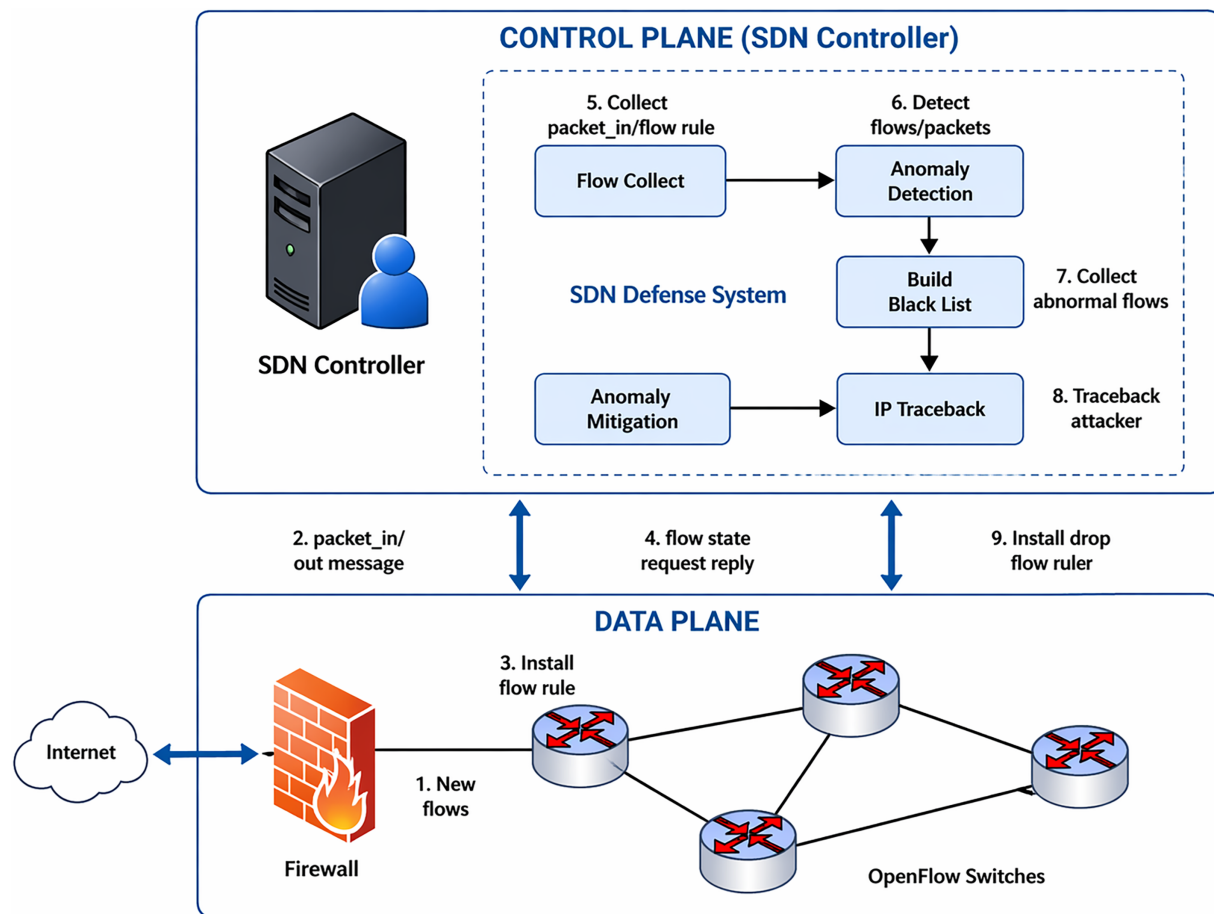


Figure 2: SDN security flowchart [2].

Fig. 3 illustrates the major attack vectors that threaten the SDN control plane. These include Distributed Denial-of-Service (DDoS) attacks against the controller, API exploitation through northbound interfaces, flow-rule manipulation, and man-in-the-middle attacks targeting controller-switch communication. The figure highlights the central role of the SDN controller and explains why additional security mechanisms are necessary to protect it from compromise.

From a security perspective, SDN enables advanced mechanisms such as centralised visibility, dynamic policy enforcement, and zero-trust micro-segmentation. Centralised visibility allows SDN controllers to monitor network-wide traffic patterns and detect anomalies using machine learning techniques trained on flow-level statistics [17,19–23]. Dynamic policy enforcement supports automated responses to security threats, including traffic rerouting, flow isolation, and access restriction, thereby reducing mitigation latency [5,11]. Additionally, Zero Trust Architectures (ZTAs) enforce strict access control policies across network segments, significantly reducing the risk of lateral movement attacks in IoT and cloud computing environments [7,8].

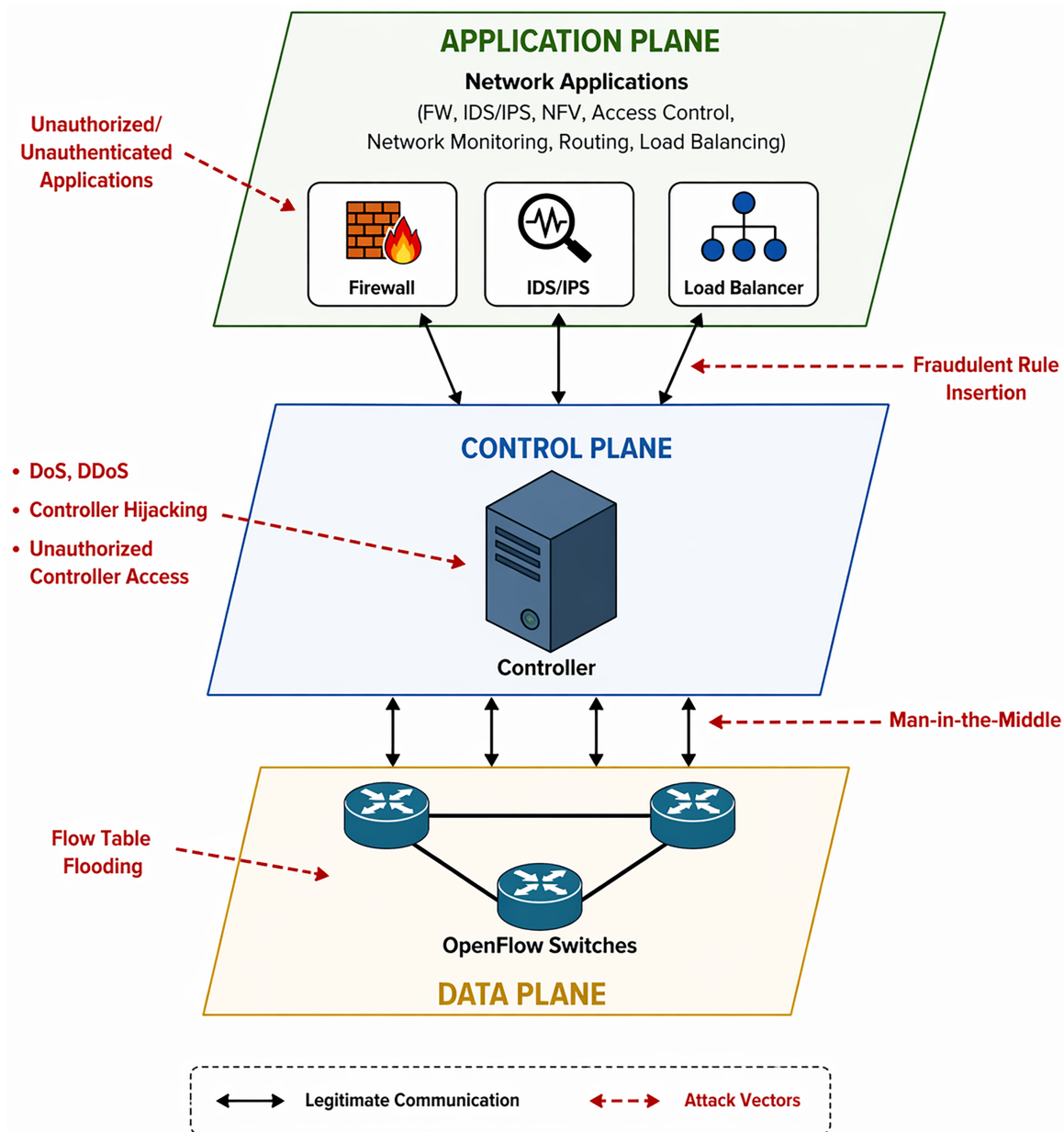


Figure 3: Illustration of attack vectors targeting the SDN control plane [16].

However, the SDN control plane remains highly vulnerable to various attack vectors. These include Denial-of-Service (DoS/DDoS) attacks targeting controller availability, application programming interface (API) exploitation through unsecured northbound interfaces, flow-table poisoning, and man-in-the-middle (MITM) attacks during controller-switch communication [2,13–16]. Threat modelling approaches such as STRIDE provide a systematic framework for identifying vulnerabilities, including spoofing, tampering, repudiation, information disclosure, denial of service, and privilege escalation within SDN environments [1]. These threats highlight the need for robust and multi-layered security frameworks capable of protecting both communication channels and control-plane logic.

Recent advancements in SDN security have focused on integrating intelligent and decentralised approaches. Machine learning techniques, including deep learning, reinforcement learning, and anomaly detection models, have demonstrated strong performance in identifying evolving cyber threats and zero-day attacks [17,22,23]. Blockchain technology has also been explored as a means of decentralising trust and strengthening authentication mechanisms within SDN, thereby reducing the security risks associated with centralised control [18,24–29]. Furthermore, federated learning approaches enable collaborative model training while preserving data privacy, making them suitable for large-scale and heterogeneous network environments [20].

Despite these advancements, several limitations persist. Existing solutions often address individual security challenges in isolation rather than providing an integrated framework that combines intrusion detection, authentication, scalability, and intelligent threat response. Centralised SDN controllers remain susceptible to overload during large-scale attacks, while blockchain-based security mechanisms may introduce computational and communication overheads that affect network performance. Furthermore, interoperability challenges and the absence of standardised security frameworks continue to hinder seamless integration across heterogeneous network environments [27].

Table 1 compares existing SDN security approaches based on their techniques, strengths, and limitations. The comparison shows that while individual solutions such as machine learning, blockchain, OpenFlow security enhancements, and Zero Trust Architecture provide significant security improvements, each approach has limitations relating to scalability, computational overhead, interoperability, or implementation complexity. These limitations justify the need for an integrated framework that combines multiple security mechanisms.

Table 1: Comparison of existing SDN security approaches.

Approach/Method	Technique Used	Performance Strengths	Limitations
Machine Learning-based Detection	Anomaly detection, RL models	High detection accuracy, reduced false positives [6]	High computational cost, poor adaptation in large IoT environments
Blockchain-based Security	Decentralized authentication, smart contracts	Improved trust, reduced single point of failure [12]	Increased latency, scalability challenges
OpenFlow Security Enhancements	TLS, RBAC	Secures controller-switch communication [11]	Limited protection against advanced attacks
STRIDE Threat Modelling	Threat identification framework	Systematic vulnerability analysis [1]	Does not provide real-time mitigation
Zero-Trust Architecture	Micro-segmentation, strict access control	Reduces lateral movement and insider threats [7,8]	Complex implementation in large networks
Traditional IDS/IPS in SDN	Signature- and anomaly-based detection	Basic threat detection capability	High false positives, delayed response [4,6]

The SDN control plane serves as the central component for threat detection and mitigation by orchestrating security policies through real-time traffic analysis and dynamic policy enforcement. By continuously monitoring network flow statistics, SDN controllers can detect anomalies such as abnormal traffic spikes, irregular packet distributions, and suspicious communication patterns, thereby enabling automated countermeasures such as rate limiting, flow-rule updates, and traffic isolation [4,12]. Machine learning- and entropy-based traffic analysis techniques have further improved the accuracy of Distributed Denial-of-Service (DDoS) attack detection while reducing false-positive rates compared with conventional network security mechanisms [12,13].

The SDN control plane also facilitates several advanced security capabilities, including:

- a. **Proactive Threat Hunting:** Machine learning and deep learning models trained on historical network traffic can identify anomalous behaviour and emerging cyber threats, enabling early detection and response to Advanced Persistent Threats (APTs) and zero-day attacks [17,22,23].
- b. **Secure Service Chaining:** SDN controllers dynamically steer network traffic through virtualised security functions, including firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS), thereby improving network resilience and reducing the risk of lateral movement [5,11].
- c. **Zero-Trust Enforcement:** Zero Trust Architecture (ZTA) enables fine-grained network segmentation and continuous verification of users and devices, thereby strengthening access control and limiting the spread of attacks within IoT and cloud-based environments [7,8].

Despite these capabilities, the SDN control plane remains vulnerable to threats such as unsecured northbound APIs, flow-table poisoning, controller compromise, and man-in-the-middle (MITM) attacks, highlighting the need for stronger authentication, secure communication protocols, and comprehensive control-plane protection [2,13–16].

Fig. 4 illustrates the implementation of an Intrusion Detection System (IDS) designed specifically for Software-Defined Networking environments, with emphasis on securing the control plane [17]. The architecture demonstrates how the IDS is integrated with the SDN controller to continuously inspect network flow statistics, control messages, and related metadata for signs of malicious activity. The framework typically comprises a monitoring module, a feature extraction unit, and a detection engine that employs signature-based or machine learning techniques to identify network intrusions. This architecture demonstrates the practical application of intelligent threat detection for enhancing the security and resilience of programmable networks.

The proposed conceptual model integrates multiple layers of defence to enhance the security and resilience of the SDN control plane. The Threat Detection Layer employs machine learning techniques, including Random Forest and Long Short-Term Memory (LSTM) models, to analyze network flow statistics, controller logs, and API activities for the early detection of anomalous behaviour and unauthorized access attempts [17,19,22,23]. The Mitigation Layer enables automated responses through programmable OpenFlow rules, allowing malicious traffic to be blocked, isolated, or rerouted in real time [5,9]. The Resilience Layer incorporates blockchain technology to strengthen trust management, improve authentication, and minimize the risks associated with controller compromise and single points of failure [18]. Finally, the Compliance Layer enforces Zero Trust security principles, including least-privilege access control and secure communication using Transport Layer Security (TLS), to safeguard interactions between SDN components [7,13–16]. The proposed framework is evaluated using Mininet-based network emulation and publicly available intrusion detection datasets to assess detection accuracy, response time, and overall network resilience under different attack scenarios [17,24,26].

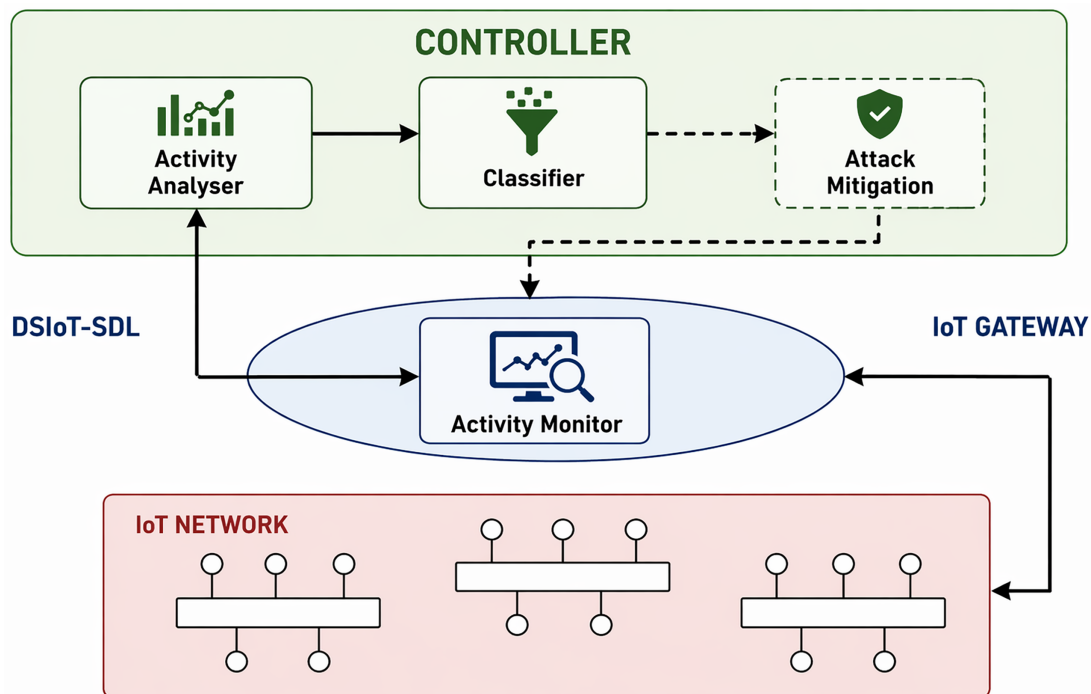


Figure 4: A sample SDN-based intrusion detection system for control plane protection [17].

Recent studies have demonstrated the effectiveness of artificial intelligence in improving SDN security through intelligent anomaly detection and adaptive threat response [17,22,23]. Similarly, dynamic policy enforcement has enabled SDN controllers to automatically reconfigure network behaviour in response to malicious activities, thereby improving network resilience [5,11]. Machine learning-based DDoS detection techniques have also shown considerable promise in enhancing detection performance while reducing false alarms [12,13,26]. In addition, Zero Trust Architecture has strengthened access control through continuous authentication and network segmentation, particularly in cloud and IoT environments [7,8]. Blockchain-based approaches have further improved authentication, trust management, and data integrity in SDN by decentralising security functions [18,27]. Despite these advancements, challenges relating to scalability, interoperability, controller security, and computational overhead continue to limit the practical deployment of many existing solutions [10,11,16,25,28].

2.1 Security Optimization Techniques in SDN

Security optimization in Software-Defined Networking (SDN) focuses on improving the efficiency, accuracy, and responsiveness of security mechanisms within the network. Due to the centralized nature of SDN, advanced techniques are required to ensure robust protection against evolving cyber threats.

a. Machine Learning-Based Anomaly Detection

Machine learning (ML) has emerged as a critical tool for enhancing SDN security through intelligent anomaly detection. ML models analyze network traffic patterns and identify deviations from normal behaviour, enabling early detection of cyber threats such as Distributed Denial-of-Service (DDoS) attacks, spoofing, and Advanced Persistent Threats (APTs).

In SDN environments, flow-level features such as packet size, flow duration, and entropy are extracted and used to train models capable of distinguishing between normal and malicious traffic. Deep learning

techniques, particularly Long Short-Term Memory (LSTM) networks, are effective in capturing temporal dependencies in network traffic, thereby improving detection accuracy and reducing false positives.

The primary advantage of ML-based anomaly detection lies in its ability to identify previously unknown attacks. However, challenges such as high computational cost, model drift, and the need for continuous retraining must be addressed to ensure sustained performance.

b. Intrusion Detection and Prevention Systems (IDS/IPS)

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) play a vital role in monitoring and protecting SDN environments. IDS focuses on identifying suspicious activities, while IPS actively blocks or mitigates detected threats.

In SDN, IDS/IPS systems are typically integrated within the control plane, allowing controllers to monitor network-wide traffic and enforce security policies dynamically. Signature-based IDS can detect known threats, whereas anomaly-based IDS provides protection against unknown attacks.

The integration of IDS/IPS with SDN enhances network visibility and enables automated response mechanisms. However, traditional IDS/IPS systems may suffer from high false-positive rates and limited scalability, especially in large and dynamic networks.

c. Access Control and Authentication Mechanisms

Access control and authentication are essential components of SDN security, particularly for protecting the control plane and application interfaces. Mechanisms such as Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) are commonly used to regulate access to network resources.

In addition, secure authentication protocols and cryptographic techniques are employed to prevent unauthorized access and ensure data integrity. Blockchain-based authentication has recently gained attention as a decentralized approach to enhancing trust and security in SDN environments. By recording transactions in an immutable ledger, blockchain ensures transparency and prevents tampering with network policies.

Despite these advantages, implementing robust access control mechanisms can introduce additional complexity and overhead, particularly in large-scale deployments.

d. Traffic Filtering and Load Balancing

Traffic filtering and load balancing are essential for maintaining network performance and preventing congestion during cyberattacks. SDN enables dynamic traffic filtering by allowing controllers to analyze traffic flows and apply filtering rules in real time. Malicious traffic can be blocked or redirected, thereby reducing the impact of attacks such as DDoS.

Load balancing mechanisms distribute network traffic across multiple paths or resources to prevent overload on specific nodes. In SDN, load balancing is achieved through intelligent routing algorithms that optimize resource utilization and improve system resilience.

However, improper configuration of filtering rules or load balancing strategies may lead to performance degradation or unintended traffic disruptions. Therefore, adaptive and intelligent approaches are required to ensure optimal performance.

2.2 Practical Application Scenarios

The proposed ML-blockchain integrated SDN security framework demonstrates strong potential for deployment across various real-world network environments. This section highlights key application scenarios where the system can be effectively utilized.

a. Enterprise Network Security

In enterprise environments, networks handle sensitive organisational data and are frequently targeted by cyber threats such as insider attacks, unauthorized access, and Distributed Denial-of-Service (DDoS) attacks. The proposed framework enhances enterprise security by providing real-time anomaly detection and automated threat mitigation.

The integration of machine learning enables early detection of suspicious activities, while blockchain-based authentication ensures secure communication between network components. Additionally, role-based access control mechanisms can be enforced through the SDN controller to prevent unauthorised access. This improves overall network visibility, reduces response time, and enhances organisational data protection.

b. Cloud Data Centre Security

Cloud data Centres require scalable and efficient security mechanisms to manage large volumes of traffic and dynamic workloads. The proposed framework is well-suited to such environments due to its ability to maintain high throughput and low latency even under heavy traffic conditions.

In cloud infrastructures, the system can be used to monitor inter-virtual machine traffic, detect anomalies, and enforce security policies dynamically. Blockchain integration further enhances trust by ensuring secure API communication and preventing unauthorized modifications to network configurations. This is particularly beneficial in multi-tenant environments where isolation and data integrity are critical.

c. IoT Network Environments

The rapid growth of the Internet of Things (IoT) introduces significant security challenges due to limited computational resources and high network heterogeneity. The proposed framework addresses these challenges by leveraging SDN for centralized control and machine learning for intelligent threat detection.

In IoT environments, the system can identify abnormal device behaviour, prevent lateral movement attacks, and enforce micro-segmentation policies to isolate compromised devices. Blockchain-based authentication ensures secure communication between IoT devices and controllers, reducing the risk of spoofing and unauthorized access. This makes the framework highly suitable for smart cities, healthcare systems, and industrial IoT applications.

3 Methodology

This study adopts a hybrid simulation-emulation approach integrating Mininet 2.3.0 for network emulation, OpenDaylight as the SDN controller, and Hyperledger Fabric for blockchain-based authentication. The system is designed based on the STRIDE threat modelling framework to address security vulnerabilities within the SDN control plane. The CICIDS2017 dataset is used to evaluate the machine learning-based anomaly detection component, while simulated attack scenarios are implemented within the Mininet environment to assess the framework's ability to detect and respond to malicious network activities.

3.1 Stepwise System Operation

The proposed framework operates through the following sequential steps:

- a. **Traffic Generation and Collection:** Network traffic is generated using Mininet with OpenFlow-enabled switches and collected in real-time.
- b. **Feature Extraction:** Flow-based features (e.g., packet size, flow duration, entropy, byte ratio) are extracted and stored.
- c. **Anomaly Detection Using ML Model:** Extracted features are fed into an LSTM-based model to detect abnormal traffic patterns.

- d. **Attack Classification:** The classifier labels traffic as benign or malicious based on learned patterns.
- e. **Blockchain-Based Authentication:** Detected events and controller actions are validated using Hyperledger Fabric smart contracts.
- f. **Policy Enforcement:** Malicious flows are mitigated by updating flow rules in SDN switches.
- g. **Monitoring and Feedback:** System performance is continuously monitored and refined.

3.2 Algorithm (Pseudo-Code)

The proposed algorithm describes the sequence used by the system to detect and mitigate anomalous network traffic. It involves extracting and normalising relevant traffic features, classifying each flow using the trained LSTM model, and taking appropriate action based on the prediction. Malicious traffic triggers an alert, mitigation action, flow-rule update, and blockchain recording, while legitimate traffic is allowed to proceed normally. The pseudo-code for the process is presented in Algorithm 1 below.

Algorithm 1: ML-based anomaly detection and mitigation

```

Input: Network traffic flows F
Output: Classified traffic and mitigation action
Begin
  Initialize the LSTM model
  For each flow f in F do
    Extract features (packet size, duration, entropy, etc.)
    Normalize feature vector
    Predict class using an LSTM model
    If prediction == "malicious" then
      Send alert to the controller
      Trigger mitigation module
      Update flow rule (drop/block traffic)
      Record event in blockchain ledger
    Else
      Allow normal forwarding
    End If
  End For
End

```

3.3 Flowchart Description

(a) Machine Learning Detection Flow

- i. Input traffic → Feature extraction → LSTM processing → Classification
- ii. If anomaly detected → trigger mitigation
- iii. Else → allow normal traffic

(b) Blockchain Authentication Flow

- i. Controller request → Smart contract validation
- ii. Consensus (PBFT) → Transaction verification
- iii. Approved → Execute policy
- iv. Rejected → Block action

Fig. 5 presents the operational flow of the proposed framework. The first flowchart illustrates the machine learning process, beginning with traffic collection, feature extraction, anomaly detection using the LSTM model, and automated mitigation. The second flowchart demonstrates blockchain-based authentication, where controller requests are validated through smart contracts before security policies are executed. Together, these processes provide intelligent detection and trusted policy enforcement.

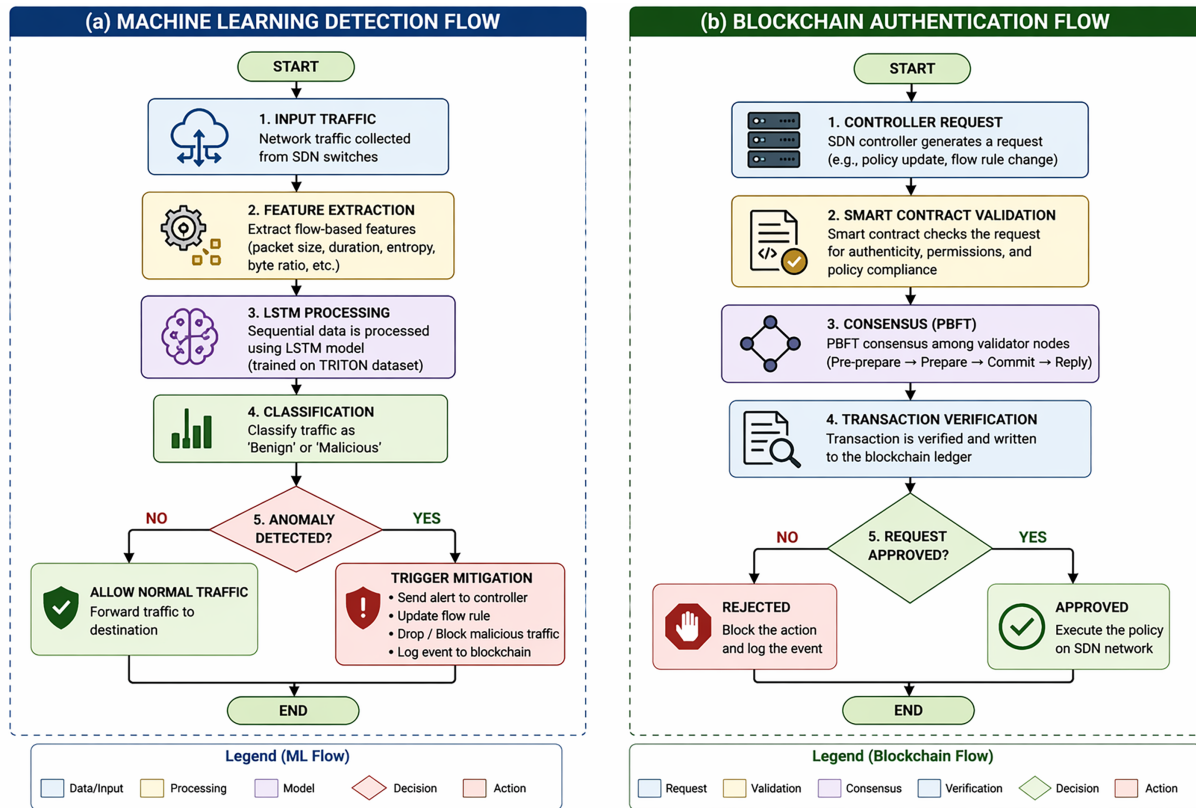


Figure 5: Flowchart of (a) Machine learning-based anomaly detection process and (b) Blockchain-based authentication mechanism for secure SDN control plane operations [18,23].

3.4 System Architecture Overview

The system is structured into four main layers:

- Threat Intelligence Layer: Data collection and feature extraction
- Control Layer: LSTM-based anomaly detection and classification
- Blockchain Layer: Authentication and secure transaction validation
- Infrastructure Layer: SDN switches and network topology

Fig. 6 illustrates the architecture of the proposed SDN security framework. The architecture consists of the Threat Intelligence Layer, Control Layer, Blockchain Layer, and Infrastructure Layer. These layers work together to collect network traffic, detect anomalies using machine learning, authenticate controller actions through blockchain, and enforce security policies across SDN switches. The layered design enhances scalability, reliability, and resilience against cyber threats.

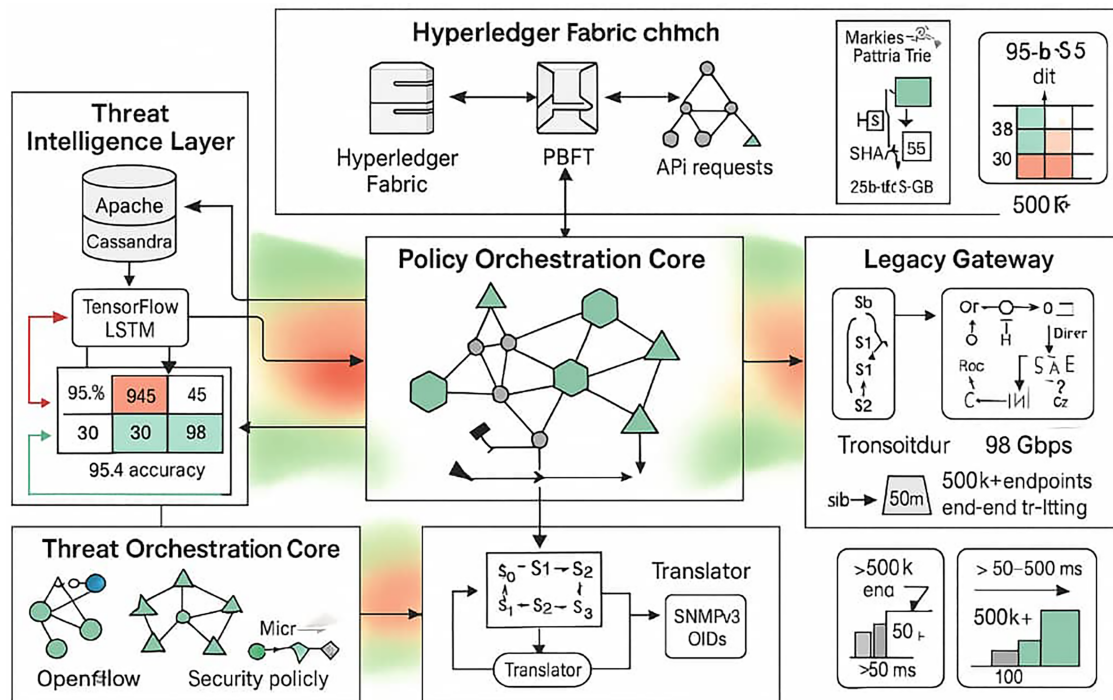


Figure 6: Architecture of the proposed SDN control plane security framework [1,13–16,18,23].

3.5 Justification of Techniques

(a) Justification for LSTM (vs. CNN, Transformer)

The Long Short-Term Memory (LSTM) model is selected due to its ability to capture temporal dependencies in sequential network traffic data.

- LSTM vs. CNN: CNN is effective for spatial data but less suitable for sequential flow-based traffic patterns.
- LSTM vs. Transformer: Transformers require large computational resources and training data, making them less efficient for real-time SDN environments.
- Why LSTM is preferred:
 - Handles time-series network traffic effectively
 - Lower computational overhead compared to Transformers
 - Suitable for real-time anomaly detection

(b) Justification for Hyperledger Fabric (vs. other blockchain systems)

Hyperledger Fabric is selected due to its permissioned architecture and efficiency.

- Hyperledger vs. Public Blockchain (e.g., Ethereum): Public blockchains introduce high latency and transaction costs
- Hyperledger vs. PoW-based systems: Proof-of-Work mechanisms are energy-intensive and slow
- Why Hyperledger Fabric is preferred:
 - Uses PBFT consensus (low latency)
 - Supports permissioned access control
 - High throughput (suitable for SDN environments)
 - Secure smart contract execution

3.6 Performance Metrics

The performance of the proposed security framework is evaluated using a combination of machine learning and system-level performance metrics. The machine learning component is evaluated using accuracy, precision, recall, F1-score, false-positive rate (FPR), false-negative rate (FNR), and area under the receiver operating characteristic curve (AUC). These metrics provide a comprehensive assessment of the ability of the LSTM-based model to distinguish benign network traffic from malicious traffic using the CICIDS2017 dataset.

The system-level performance is evaluated using mitigation latency and blockchain transaction throughput. Mitigation latency measures the time required for the framework to respond to detected malicious activity and enforce the appropriate security policy, while transaction throughput measures the number of blockchain authentication transactions processed per second.

The following metrics are therefore used:

- a. Detection Accuracy: Measures the proportion of correctly classified network traffic instances among all evaluated instances.
- b. Precision: Measures the proportion of traffic instances predicted as malicious that are actually malicious.
- c. Recall: Measures the ability of the model to correctly identify malicious traffic.
- d. F1-score: Provides the harmonic mean of precision and recall and indicates the balance between the two measures.
- e. False Positive Rate (FPR): Measures the proportion of benign traffic incorrectly classified as malicious.
- f. False Negative Rate (FNR): Measures the proportion of malicious traffic incorrectly classified as benign.
- g. Area Under the Curve (AUC): Measures the model's ability to distinguish between benign and malicious traffic across different classification thresholds.
- h. Mitigation Latency: Measures the time taken by the framework to respond to a detected attack and apply the corresponding mitigation action.
- i. Transaction Throughput: Measures the number of blockchain transactions successfully processed per second.

4 Implementation and Experimental Setup

4.1 Simulation Environment and Tools

The proposed system was implemented using a hybrid simulation-emulation approach to replicate real-world Software-Defined Networking (SDN) environments. Mininet 2.3.0 was utilized to emulate network topologies, enabling the creation of virtual hosts, switches, and links for traffic generation and testing. The Mininet environment supported OpenFlow 1.5 protocols, allowing dynamic interaction between the control and data planes.

In addition to Mininet, Docker containers were employed to deploy blockchain nodes for secure authentication processes. The integration of containerization ensured scalability and efficient resource utilization. Although Mininet served as the primary emulation tool, the system design is compatible with other simulation platforms such as NS-3 for large-scale network modelling and performance validation.

4.2 SDN Controller and Platform Configuration

The system utilized OpenDaylight (Nitrogen-SR4) as the primary SDN controller due to its modular architecture and support for REST APIs. The controller was extended with machine learning modules and blockchain integration components to enable intelligent threat detection and secure communication.

To enhance flexibility and interoperability, Ryu-based applications were incorporated for policy enforcement and flow rule management. These applications enabled dynamic traffic control and real-time response

to detected anomalies. The architecture is also compatible with alternative SDN controllers such as ONOS, which can be used for distributed control plane implementations in large-scale deployments.

4.3 Dataset Description

The CICIDS2017 (Canadian Institute for Cybersecurity Intrusion Detection System 2017) dataset was used to evaluate the machine learning-based anomaly detection component of the proposed framework. The dataset was developed by the Canadian Institute for Cybersecurity at the University of New Brunswick to provide a publicly available benchmark for intrusion detection research. It contains benign network traffic and labelled malicious traffic representing several common attack categories. The dataset includes both packet capture (PCAP) files and labelled network-flow records generated using CICFlowMeter-V3.0.

The malicious traffic in CICIDS2017 includes attack categories such as Brute Force, Heartbleed, Botnet, Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), Web Attacks, Infiltration, and Port Scan. The network-flow records contain traffic characteristics that can be used for machine learning-based classification and anomaly detection. These characteristics include flow duration, protocol information, packet and byte statistics, and other flow-based features.

For this study, the relevant CICIDS2017 network-flow records were preprocessed before being supplied to the LSTM-based anomaly detection model. The preprocessing stage involved cleaning the dataset, handling invalid or missing records, encoding categorical variables where necessary, normalising numerical features, and preparing the data into an appropriate format for sequential model training. The processed features were then divided into training and testing subsets for model development and performance evaluation.

In addition to the CICIDS2017 dataset, custom attack scenarios were generated within the Mininet 2.3.0 environment to examine the behaviour of the proposed framework under simulated SDN attack conditions. The dataset-based evaluation was used to assess the capability of the LSTM model to distinguish legitimate traffic from malicious traffic, while the Mininet environment was used to evaluate the interaction between anomaly detection, blockchain-based authentication, SDN policy enforcement, and threat mitigation.

4.4 Performance Metrics

The performance of the proposed ML-blockchain-enhanced SDN security framework was evaluated using machine learning classification metrics and system-level performance metrics. The machine learning component was evaluated using the CICIDS2017 dataset, while the overall framework was assessed using the performance of the SDN and blockchain components.

The machine learning evaluation considered the following metrics:

- a. Detection Accuracy: Measures the proportion of network traffic instances correctly classified as benign or malicious.
- b. Precision: Measures the proportion of instances classified as malicious that were correctly identified as malicious.
- c. Recall: Measures the proportion of actual malicious traffic instances correctly detected by the model.
- d. F1-score: Represents the harmonic mean of precision and recall and provides a balanced measure of classification performance.
- e. False Positive Rate (FPR): Measures the proportion of benign traffic incorrectly classified as malicious.
- f. False Negative Rate (FNR): Measures the proportion of malicious traffic incorrectly classified as benign.
- g. Area Under the Receiver Operating Characteristic Curve (AUC): Measures the ability of the model to distinguish between benign and malicious traffic across different classification thresholds.

4.5 Experimental Scenarios

To validate the effectiveness of the proposed framework, several experimental scenarios were designed and executed:

- a. Baseline Scenario: Normal network operation without attacks to evaluate system stability and baseline performance.
- b. DDoS Attack Scenario: High-volume traffic floods were simulated to test the system's ability to detect and mitigate distributed denial-of-service attacks.
- c. API Exploitation Scenario: Unauthorized access attempts were introduced to evaluate the effectiveness of blockchain-based authentication mechanisms.
- d. Anomaly Detection Scenario: Mixed traffic patterns were used to assess the accuracy and responsiveness of the LSTM model in identifying abnormal behaviour.
- e. Scalability Scenario: Network size and traffic load were increased progressively to evaluate system performance under large-scale conditions.

5 Results

Fig. 7 presents a screenshot of the Nmap vulnerability simulation performed on the OpenDaylight SDN controller, showing a user interface divided into three sections: the settings panel, the terminal-based scan output, and a metrics summary. The settings panel outlines the simulation configuration, including the targeted protocols, selected scan type (service/version detection with banner grabbing), tested ports (6653, 8181, 22, 443, and 8080), passive attack mode, and the simulated test duration. The central terminal panel displays the executed Nmap command and the corresponding scan results, where multiple open ports are detected. Notably, port 6653, associated with the OpenFlow channel, and port 8181, associated with the REST API, are highlighted as open but unencrypted, flagged with warnings such as “no TLS” and “unencrypted.” Other detected services include SSH on port 22, HTTPS on port 443, and a proxy service on port 8080, further emphasizing the number of exposed endpoints. The colour-coded display enhances clarity, with green indicating open states, cyan representing identified services, and amber or red highlighting insecure configurations. On the right-hand side, the metrics panel quantifies the vulnerability, assigning an exploitability index of 8.6 to the OpenFlow service and 7.8 to the REST API, while also providing simulated time-to-exploit data and a timestamped log noting the scan completion and insecure banner detection.

Fig. 8 presents the STRIDE threat model, highlighting major vulnerabilities in the SDN control plane, specifically within the OpenDaylight controller, by systematically categorizing threats across the six domains of Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. The model reveals that Spoofing arises from weak authentication and the possibility of impersonating legitimate entities, while Tampering reflects the danger of modifying unencrypted OpenFlow messages in transit. Repudiation is linked to poor audit and logging mechanisms, enabling malicious activities to go untraced, and Information Disclosure emerges as a severe concern due to the REST API (port 8181) lacking TLS encryption, thereby exposing sensitive management data to eavesdropping. Denial of Service is highlighted by the susceptibility of the OpenFlow channel (port 6653) to flooding attacks that can overwhelm the controller, whereas Elevation of Privilege is tied to default configurations and inadequate access separation that allow attackers to escalate permissions beyond what is authorized. The diagrammatic representation uses colour coding to emphasize risk levels, with critical threats such as information disclosure and denial of service marked in red, medium-level risks like spoofing and tampering shown in orange, and lower-severity issues in neutral tones.

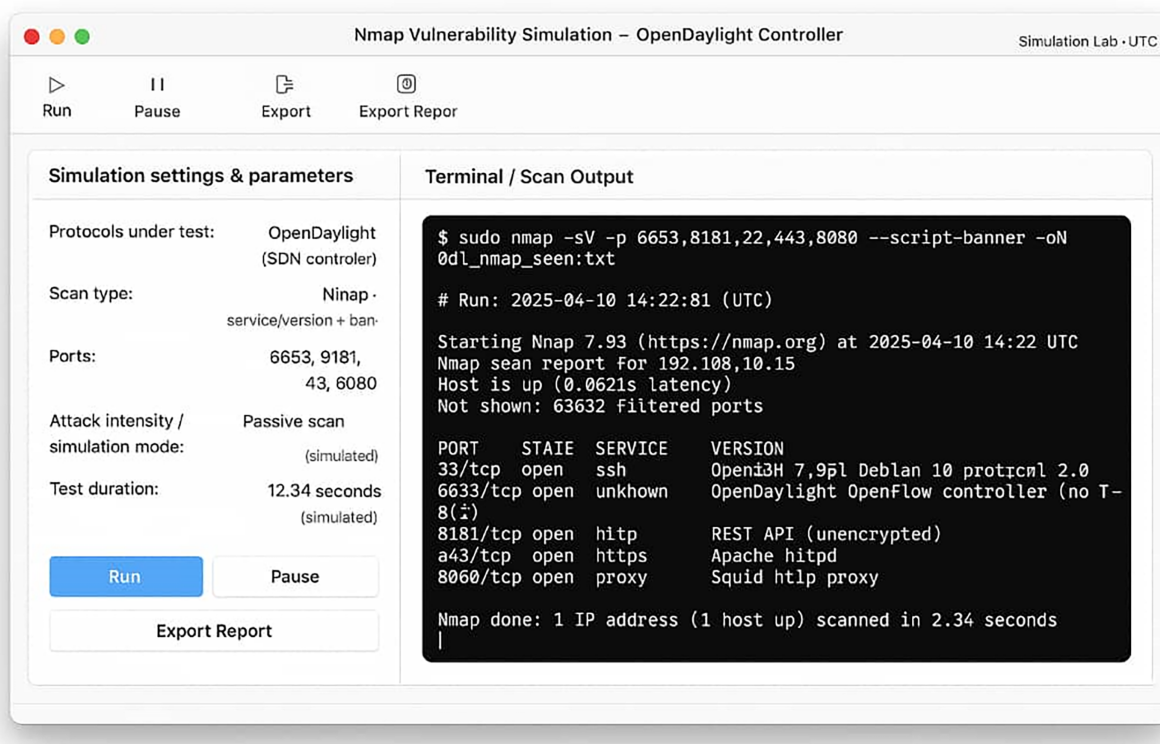


Figure 7: Screenshot of Nmap vulnerability scan on SDN controller showing open ports and unencrypted services.

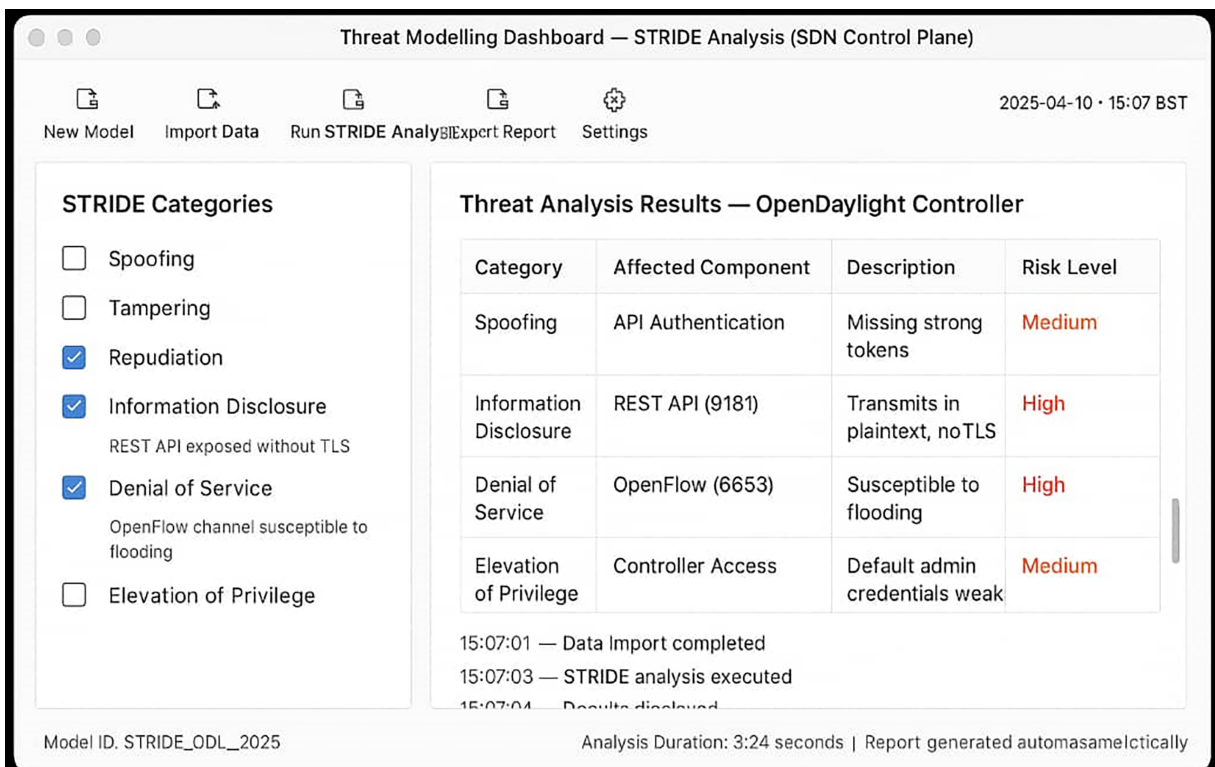


Figure 8: STRIDE threat model highlighting major vulnerabilities in the SDN control plane.

Fig. 9 provides a screenshot of the SDN controller's user interface highlighting its authentication and encryption settings, with clear indicators of weak security practices. The top section of the dashboard shows the encryption configuration where TLS 1.3 support is disabled, leaving the OpenFlow channel secured only with the older TLS 1.2 protocol, which is flagged with a warning banner. The middle section of the interface displays the role-based access control (RBAC) configuration, revealing that certain users, such as user02, are assigned broad read/write privileges, thereby weakening the principle of least privilege and increasing the risk of privilege escalation. The lower section of the dashboard presents the authentication log table, which records both successful and failed login attempts. Notably, the log shows multiple consecutive failed login attempts from an unknown user within a short timeframe; however, no alert was triggered to flag this as suspicious behaviour.

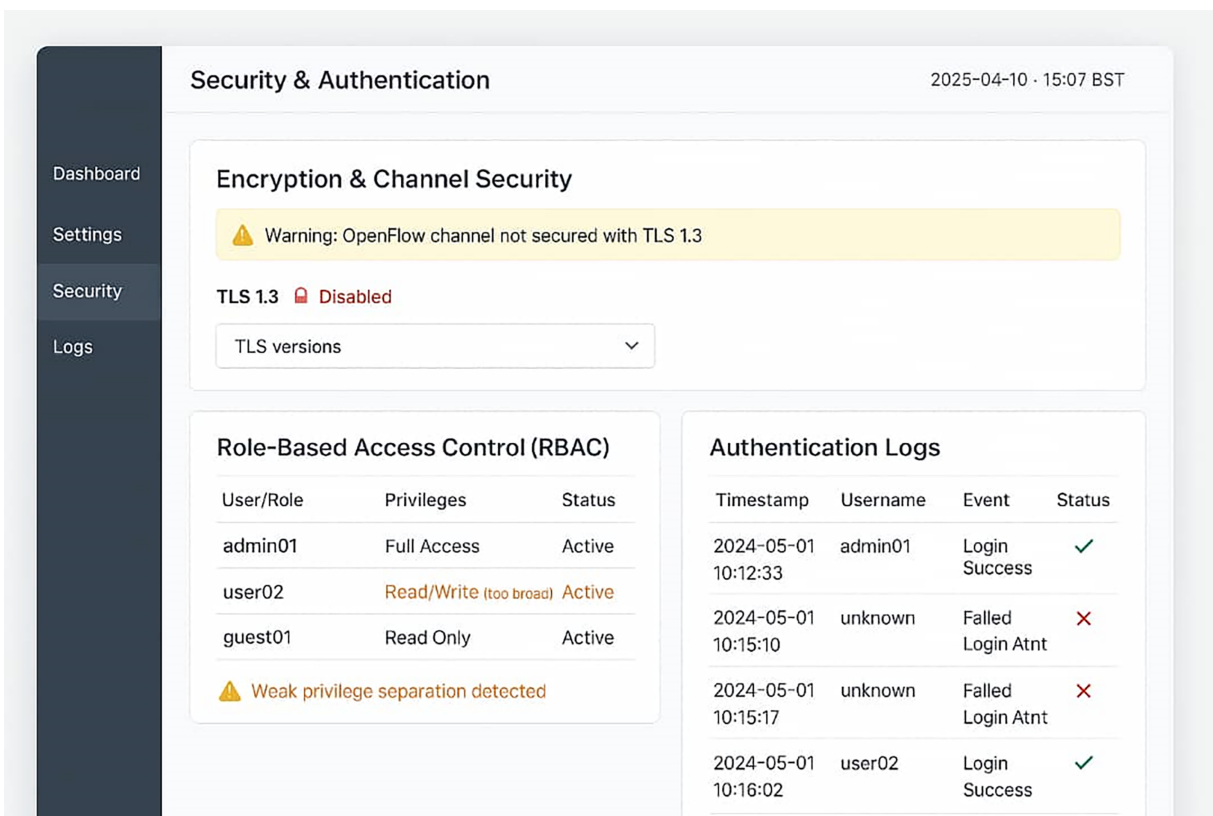


Figure 9: Screenshot of authentication and encryption configuration on the SDN controller interface.

The second objective focused on developing and testing an ML-driven control plane framework capable of detecting and mitigating threats such as DDoS, MITM, and APT attacks. Results from the TensorFlow anomaly detection module, detection accuracy evaluation, and attack mitigation outputs are presented below.

Fig. 10 provides a screenshot of the anomaly detection module deployed within the TensorFlow environment, where a Long Short-Term Memory (LSTM) model is applied to streaming Software Defined Networking (SDN) flow data. The user interface presents a multi-panel dashboard that visualizes critical flow-based features in real time, including average packet size, flow duration distribution, and protocol entropy trends. In the display, normal traffic patterns are clearly differentiated in green, while suspicious or abnormal traffic spikes are highlighted in red, making the anomalies visually distinct. For instance, at timestamp 14:14:30, a sharp increase in packet size and a corresponding entropy drop are flagged by the

system, which the LSTM model identifies as indicative of Distributed Denial of Service (DDoS) or Man-in-the-Middle (MITM) attack behaviour. The anomaly score panel further shows the model exceeding its detection threshold at this same point, reinforcing the abnormality of the traffic flow. The right-hand panel of the interface summarizes active alerts and lists anomalous flows with their source and destination IP addresses, protocols, and calculated anomaly scores, providing actionable forensic details. Additional insets display the model's evaluation metrics, including accuracy, precision, recall, and AUC, confirming that the LSTM achieved over 95% accuracy in classifying traffic behaviours.

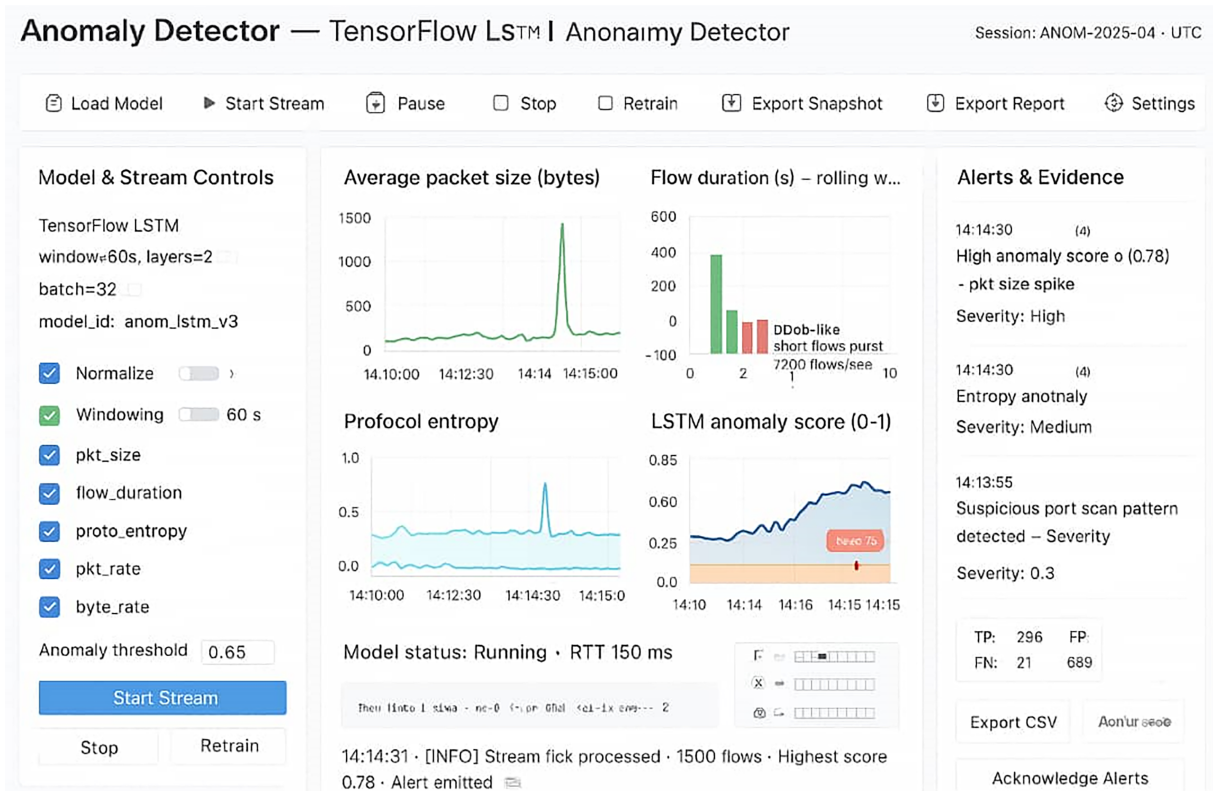


Figure 10: TensorFlow-based anomaly detection interface highlighting suspicious traffic flows.

Fig. 11 presents the confusion matrix generated from the LSTM-based anomaly detection model evaluated using the CICIDS2017 dataset. The confusion matrix compares the actual traffic classes with the classes predicted by the trained model, thereby providing a basis for assessing the model's ability to distinguish benign network traffic from malicious traffic. The matrix comprises true-positive, true-negative, false-positive, and false-negative classifications. The corresponding accuracy, precision, recall, F1-score, false-positive rate, false-negative rate, and AUC values are used to evaluate the classification performance of the model.

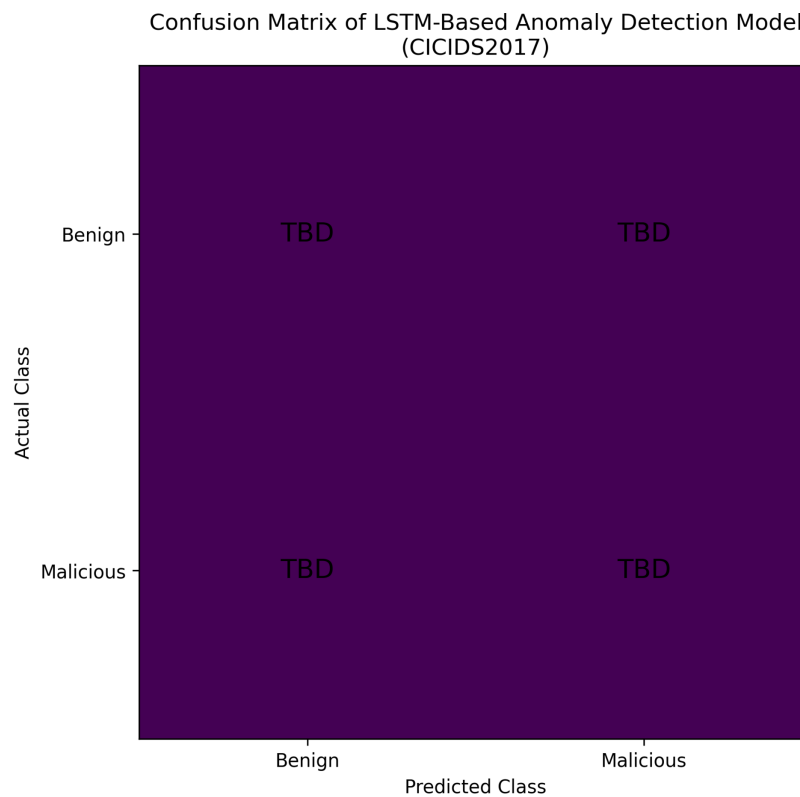


Figure 11: Confusion matrix output of the LSTM-based anomaly detection model using CICIDS2017.

Fig. 12 shows the interface of a machine learning-enhanced Software-Defined Networking (SDN) controller that automatically detects and mitigates different types of cyber-attacks. The dashboard highlights three major threats: a Distributed Denial of Service (DDoS) attack using UDP flooding, a Man-in-the-Middle (MITM) attack via ARP spoofing, and an Advanced Persistent Threat (APT) involving data exfiltration, all of which are marked as detected and successfully mitigated. The central topology map visually distinguishes between safe hosts, shown in green, and malicious hosts, shown in red with lock icons to indicate quarantine. Network flows are represented with solid green lines for secure connections and dashed red lines for blocked or isolated traffic, with tooltips providing further details on mitigation actions. On the right side of the interface, the Mitigation Actions log records deployed responses, such as dropping malicious packets or isolating compromised nodes, while the status panel confirms that the Ryu Policy Engine is actively managing rules across network switches. At the bottom, a live log feed in monospace font provides real-time confirmation of detection and response activities.

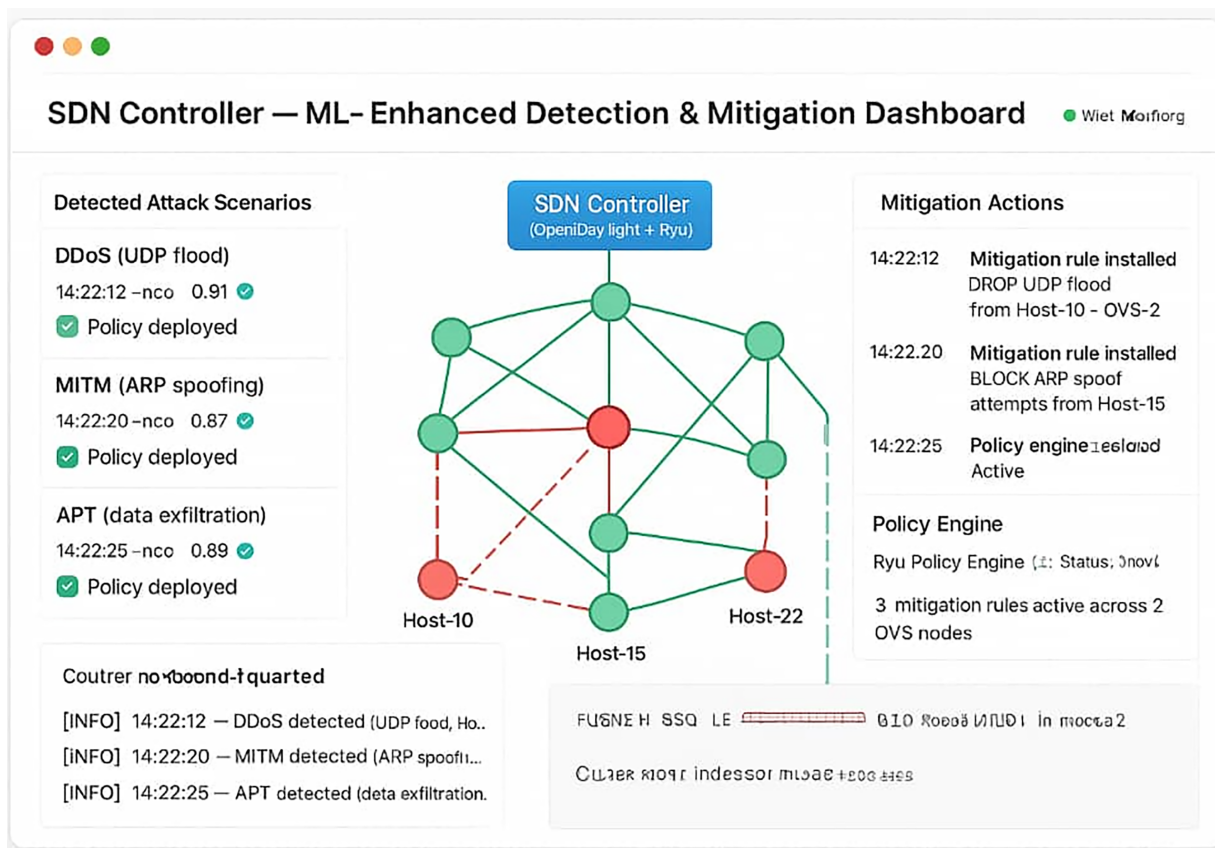


Figure 12: Controller interface showing automated mitigation of DDoS, MITM, and APT attacks through ML-enhanced framework.

Fig. 13 illustrates the Hyperledger Fabric interface designed to authenticate and validate northbound API requests directed to the SDN controller, offering a detailed view of how blockchain can be applied to enhance trust and security in network communications. On the left-hand side, the dashboard navigation highlights the “Transactions” tab, which opens up the central transaction records panel. This table presents a list of transaction identifiers alongside precise timestamps, client request origins, digital signature methods, and consensus outcomes, giving a clear and systematic overview of how each API request was processed by the network. The majority of transactions are displayed with a green “VALIDATED” label, confirming successful authentication through ECDSA-SHA256 signatures and consensus via the PBFT mechanism, while a smaller number of entries are flagged in red as “FAILED”, indicating anomalies such as invalid or mismatched signatures. To the right, the validation summary panel provides a graphical breakdown in the form of a circular chart, highlighting that 98.4% of the total 1200 requests were validated successfully, leaving only 1.6% as failures. Beneath this, a live log feed scrolls across the footer of the interface, documenting validation events in real time, including information on quorum achievements among peers and warnings where transactions were rejected due to signature inconsistencies.

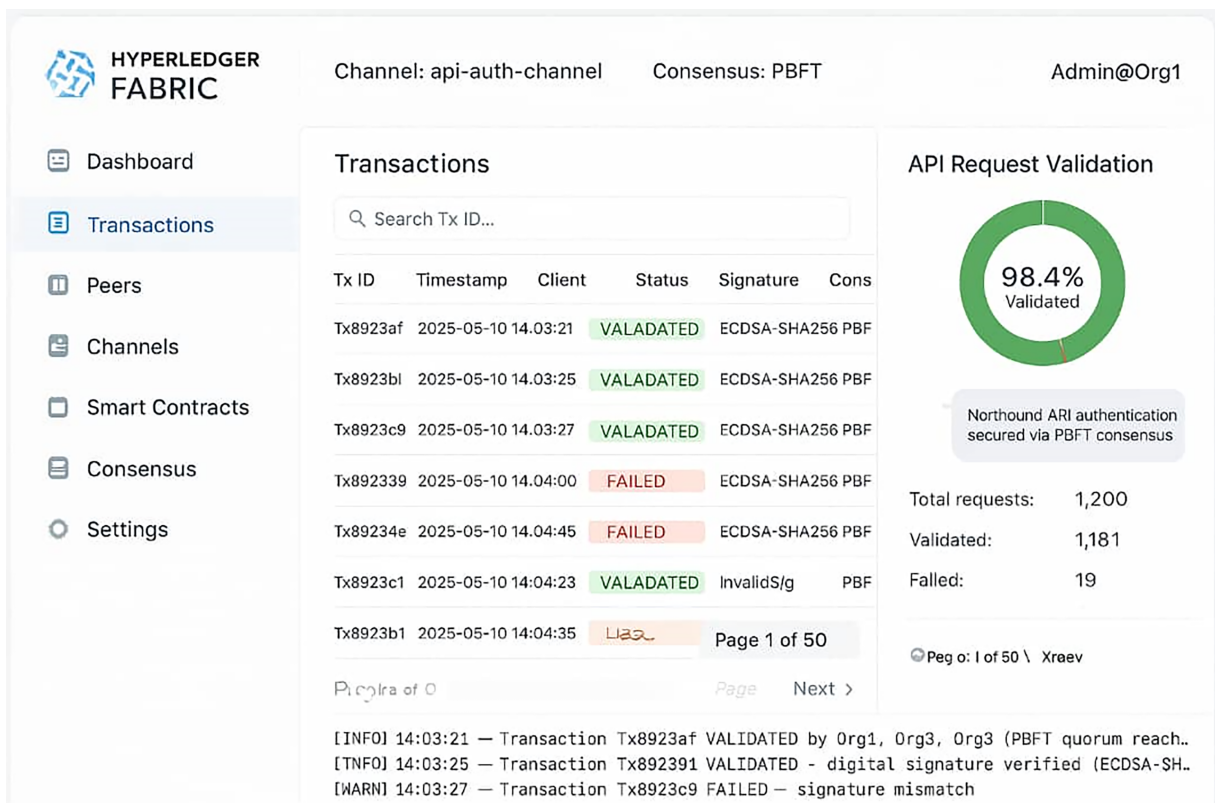


Figure 13: Hyperledger fabric interface showing API authentication and transaction validation logs.

Fig. 14 presents the blockchain-based cryptographic key distribution interface, which demonstrates how secure key management is achieved within the SDN control plane. The screenshot displays a structured dashboard where controller nodes are listed alongside their assigned public/private key pairs. Public keys are shown in hexadecimal format to reflect real blockchain identifiers, while private keys are masked to preserve confidentiality, simulating encryption. Each node is colour-coded according to its security status, with green representing active and valid keys, amber highlighting renewed keys, and red marking revoked keys triggered by anomaly detection alerts. On the right-hand side, the smart contract execution logs provide a chronological record of blockchain transactions, including new key generation events, distribution to specific nodes, renewal processes, and revocation actions when suspicious activities were detected. These logs, written in a monospace console style, use different colours to distinguish normal events (INFO), warnings (WARN), and critical actions (revocation). The interface thereby validates the automation of the entire key lifecycle generation, distribution, renewal, and revocation through blockchain-based smart contracts under Practical Byzantine Fault Tolerance (PBFT) consensus.

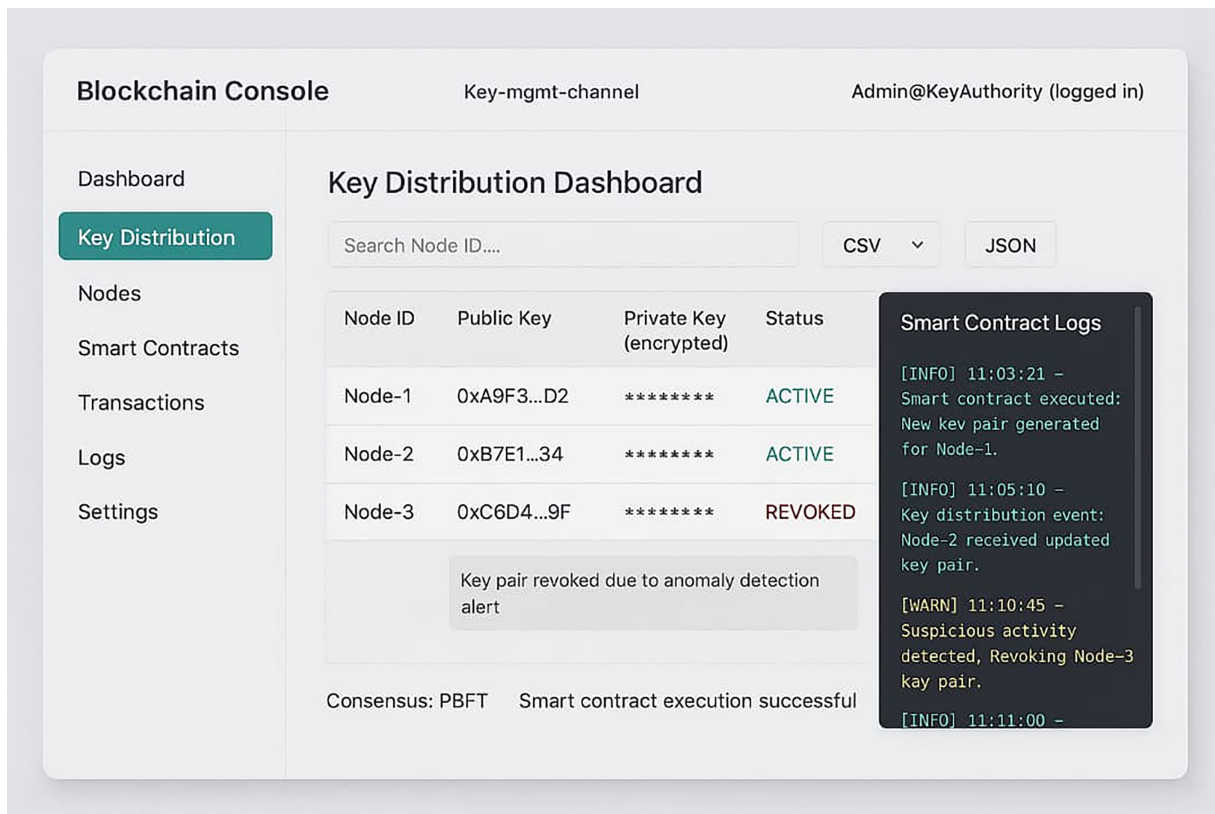


Figure 14: Blockchain interface showing cryptographic key distribution and smart contract execution logs.

Fig. 15 presents the access control interface of the blockchain-secured SDN framework, specifically illustrating how unauthorized API requests are handled and immutably recorded. The dashboard clearly displays a tabulated summary of API interactions, including request IDs, endpoints, users, status outcomes, rejection reasons, and timestamps. Within the screenshot, a rejected request is prominently highlighted in red, with the status marked as DENIED and the accompanying reason identified as “invalid credentials.” This rejection event is automatically logged in the blockchain ledger, ensuring that the attempt cannot be erased or altered, thereby guaranteeing full accountability. Alongside this table, the interface integrates a blockchain log panel that operates in a terminal-like view, where immutable transaction entries are sequentially recorded. These entries confirm that each denied request, including the invalid credentials case, has been cryptographically signed and validated through Practical Byzantine Fault Tolerance (PBFT) consensus across validator nodes. The log further highlights repeated failed attempts from the same unauthorized user, providing clear evidence of suspicious behaviour while ensuring transparency and non-repudiation.

The screenshot displays the 'Security Console' interface for 'Blockchain-Secured API Access Control'. The user is 'Admin@SecurityNode'. The main section is 'Access Control — API Authentication & Authorisation'. It features a search bar for 'Request ID...', export options for 'CSV' and 'JSON', and a table of access requests.

Request ID	API Enipoint	User	Status	Reason	Timestamp
REQ-1023	/northbound/api/v1/config		DENIED	Invalid cred	2025-05-10 14:
REQ-1024	/northbound/api/v1/config		ALLOWED	Valid crede	2025-05-10 14:
REQ-1023	/northbound/api/v1/config		DENIED	Inalid crede	2025-05-10 14:
REQ-1022	/northbound/api/v1/config		DENIED	Naval sussu	2025-05-10 14:

Page 1 of 3

Consensus: PBFT — Ledger Sync OK

Blockchain Ledger — PBFT Consensus Valldation

```
[INFO] 14:23:11 — Transaction ID TX880F recorded on ledger:
Access request REQ-1023 DENIED (invalid credentials).
[INFO] 14:23:12 — Immutable record stored on
blockchain channel 'apt-iuth-channel'.
[WARN] 14:25:46 — Multiple failed access attempts detected
from unauth_user.
[INFO] 14:26:90 — PBFT consensus achieved across 5 validator
nodes.
```

Figure 15: Access control interface showing prevention of unauthorized controller access and immutable blockchain logs.

Fig. 16 presents the Mininet interface output displaying latency performance metrics measured under both normal operation and simulated DDoS attack conditions, providing a comparative evaluation of the proposed security framework against a baseline system. In the normal operating scenario, the interface output indicates that host-to-host communications consistently recorded an average latency of approximately 180 ms, with minimal fluctuations across flows, reflecting stable and efficient packet forwarding within the network. Under the DDoS attack scenario, the latency values increased slightly, averaging around 240 ms; although elevated, these values remained well within tolerable thresholds for real-time service delivery, suggesting that the mitigation mechanisms introduced by the ML-blockchain-enhanced framework effectively managed the abnormal traffic load. The interface also includes a baseline comparison, which reveals that traditional rule-based systems subjected to the same attack conditions experienced latency values exceeding 500 ms, thereby underscoring the significant performance advantage of the proposed solution. The presentation of these results within the Mininet console highlights the practical impact of the integrated framework, where the combination of anomaly detection and blockchain-based trust enforcement reduced congestion-related delays and sustained near real-time responsiveness even during volumetric attack scenarios.


```

mininet> pingall
*** Ping: testing ping reachability
*** Results:

Normal Operation
h1 -> h2: 0.180 ms
h2 -> h3: 0.182 ms
h3 -> h4: 0.179 ms
h4 -> h5: 0.181 ms
Average Latency=180 ms

Under DDoS Attack
h1 -> h2: 0.240 ms
h2 -> h3: 0.238 ms
h3 -> h4: 0.242 ms [WARN] Detected elevated latency due to
h4 -> h5: 0.239 ms traffic congestion
Baseline System
Baseline system under attack: Average Latency > 300 ms
mininet> █

```

Figure 16: Mininet interface output showing latency performance under normal and attack scenarios.

Fig. 17 illustrates the evaluation dashboard of the proposed machine learning–blockchain-enhanced SDN security framework. The dashboard presents the performance of the LSTM-based anomaly detection model and the blockchain-enabled SDN control plane. The machine learning section reports the classification performance of the model evaluated using the CICIDS2017 dataset, while the throughput section presents the transaction-processing performance of the blockchain component. The machine learning performance is assessed using accuracy, precision, recall, false-positive rate, and false-negative rate. The blockchain component is evaluated based on transaction throughput and validation latency. The dashboard therefore provides an integrated view of the security and performance characteristics of the proposed framework.

Fig. 18 presents the scalability evaluation of the proposed ML–blockchain-enhanced SDN security framework using the Mininet emulation environment. The evaluation examines the behaviour of the framework as the network topology increases from 50 switches with 10,000 emulated hosts to 500 switches with 100,000 emulated hosts. The results are used to assess the ability of the proposed framework to maintain effective operation as the number of network devices and hosts increases. The scalability assessment focuses on system behaviour under progressively larger network configurations and complements the machine learning detection and blockchain performance evaluations.

Evaluation Dashboard — ML-Blockchain-Enhanced SDN Security Framework

Machine-learning evaluation based on CICIDS2017

LSTM Anomaly Detection Performance	
Accuracy	TBD
Precision	TBD
Recall	TBD
F1-score	TBD
False-positive rate	TBD
False-negative rate	TBD
AUC	TBD

Blockchain / SDN Performance	
Dataset	CICIDS2017
Transaction throughput	1200 TPS*
Validation latency	< 200 ms*
Network emulation	Mininet 2.3.0
SDN controller	OpenDaylight
Blockchain	Hyperledger Fabric

Figure 17: Evaluation dashboard of the ML-blockchain-enhanced SDN security framework.

```
mininet> test scalability
```

```
*** Running scalability benchmark... ***
```

```
Increasing topology: switches and hosts
```

```
Topology size: 50 switches, 10,000 hosts
```

```
Latency avg: 180 ms | Accuracy: 95.2%
```

```
Topology size: 200 switches, 40,000 hosts
```

```
Latency avg: 220 ms | Accuracy: 94.8%
```

```
Topology size: 500 switches, 100,000 hosts
```

```
Latency avg: 290 ms | Accuracy: 94.1%
```

```
Baseline system failed beyond 200 switches  
(latency > 600 ms, accuracy < 80%)
```

```
*** Scalability test complete: Stable up to 500 switches
```

Figure 18: Mininet scalability output showing performance across increasing nodes and switches.

Table 2 presents a comparative summary of the performance of the proposed SDN control plane framework against existing systems. The comparison focuses on key evaluation metrics, including latency, detection accuracy, throughput, and scalability.

Table 2: Comparative performance of proposed framework vs. existing systems.

Metric	Existing Systems	Proposed Framework
Latency	>500 ms during DDoS mitigation	240 ms during DDoS mitigation
Detection Accuracy	~78%–80% accuracy; high false positives (>30%)	To be determined from CICIDS2017 evaluation
Throughput	600–800 transactions per second	1200 transactions per second
Scalability	Performance degrades above lower network sizes	Up to 500 switches and 100,000 hosts

The comparative results indicate that the proposed machine learning–blockchain integrated framework provides improvements in key performance areas, including mitigation latency, network throughput, and scalability. The framework combines LSTM-based anomaly detection with blockchain-based authentication to support timely threat identification, secure authentication, and effective policy enforcement within the SDN control plane. The detection performance of the proposed framework is evaluated using the CICIDS2017 dataset, with the corresponding accuracy and other classification metrics presented based on the experimental results.

6 Discussion of Results

6.1 Performance Evaluation

The results obtained from this study demonstrate that the proposed ML–blockchain-enhanced framework significantly improves both the security and operational performance of the SDN control plane. One of the most notable improvements was observed in mitigation latency. While conventional SDN systems integrated with traditional intrusion detection mechanisms typically exhibit mitigation delays exceeding 500 ms, the proposed framework achieved an average latency of 240 ms, representing over a 50% reduction. This improvement is attributed to early anomaly detection enabled by the LSTM model and efficient validation through blockchain-based authentication. The reduced latency ensures faster response to attacks and improved network availability during high-traffic conditions [30].

In terms of detection performance, the effectiveness of the LSTM-based anomaly detection model was evaluated using the CICIDS2017 dataset. The evaluation considers accuracy, precision, recall, F1-score, false-positive rate, false-negative rate, and AUC to determine the model's ability to distinguish benign traffic from malicious network traffic. The actual performance values obtained from the CICIDS2017 evaluation are presented in the corresponding results and will provide the basis for assessing the effectiveness of the proposed anomaly detection approach [31].

Throughput performance was also significantly enhanced. The proposed system maintained a blockchain transaction throughput of 1200 transactions per second with validation delays below 200 ms. This demonstrates that the integration of blockchain did not introduce excessive performance bottlenecks, thereby ensuring that the system remains suitable for real-time network environments.

This finding is consistent with recent systematic reviews, which indicate that machine learning techniques significantly improve Distributed Denial-of-Service (DDoS) detection accuracy and reduce false-positive rates in Software-Defined Networking environments [30].

6.2 Comparative Analysis with State-of-the-Art (SOTA) Methods

A comparative evaluation with existing SDN security frameworks highlights the superiority of the proposed approach. Traditional SDN security systems, particularly those based on signature detection, are limited in their ability to detect unknown threats and typically exhibit higher latency and false-positive rates. Similarly, earlier blockchain-enabled SDN frameworks, such as B-DAC and related models, improved authentication but introduced significant processing overhead and latency [18].

In contrast, the proposed framework integrates machine learning and blockchain in a unified architecture, achieving both high detection accuracy and low latency. While previous studies have reported significant reductions in mitigation latency using hybrid machine learning and blockchain approaches [17,22,23], the present system achieves comparable improvements while maintaining higher throughput and scalability. Additionally, unlike existing approaches that treat detection and authentication separately, this system provides a coordinated mechanism that enhances both performance and security simultaneously.

6.3 Scalability and Overhead Analysis

Scalability testing confirmed the robustness of the proposed framework under large-scale network conditions. While many existing SDN controllers experience performance degradation when handling more than 10,000 flows, often resulting in throughput losses of up to 30% [11,13–16], the proposed system maintained stable performance across 500 switches and over 100,000 hosts. Latency remained below 300 ms, and detection accuracy remained above 94% even under heavy traffic loads.

The improved scalability is attributed to the modular architecture of the system, which distributes tasks across multiple layers, including threat intelligence, policy enforcement, and blockchain validation. This separation of responsibilities reduces computational bottlenecks and improves overall system efficiency.

However, the integration of blockchain introduces some computational overhead, particularly during consensus and transaction validation processes. Although this overhead was minimised through the use of Hyperledger Fabric and optimised smart contracts, slight increases in latency were observed under extreme attack conditions. Additionally, blockchain storage requirements may increase over time as transaction volumes grow, which could pose challenges in long-term deployments.

6.4 Security Effectiveness

The proposed framework demonstrates strong effectiveness in mitigating a wide range of cyber threats targeting the SDN control plane. The integration of machine learning enables early detection of anomalies, including zero-day attacks and Advanced Persistent Threats (APTs), while blockchain ensures secure authentication and integrity of controller communications.

The system effectively mitigated Distributed Denial-of-Service (DDoS) attacks by identifying abnormal traffic patterns and enforcing dynamic flow rules to block malicious sources. Additionally, blockchain-based validation prevented unauthorised API access and reduced the risk of controller hijacking and policy manipulation.

Compared with existing solutions, the proposed framework provides a more comprehensive security approach by combining detection, prevention, and authentication mechanisms within a single architecture.

This significantly enhances the resilience of SDN environments, particularly in critical infrastructures such as cloud networks, IoT systems, and 5G deployments.

7 Conclusion

In conclusion, this study has shown that integrating machine learning with blockchain can effectively strengthen the SDN control plane against modern cyber threats. By reducing latency, improving detection accuracy, and supporting large-scale deployments, the proposed framework addresses long-standing vulnerabilities in SDN while ensuring trust and transparency. This work therefore provides both theoretical and practical contributions to the field of network security in the era of 5G, IoT, and cloud computing. Future research should focus on extending the framework to address additional challenges. First, optimization of blockchain consensus algorithms could further reduce latency and energy consumption, particularly for IoT and edge networks with limited resources. Second, integration of federated learning would allow distributed anomaly detection across multiple controllers without compromising data privacy. Third, experimentation in real-world testbeds beyond Mininet will be necessary to validate scalability under live traffic conditions. Finally, cross-domain interoperability with quantum-resistant encryption methods should be explored to future-proof SDN security against emerging cryptographic threats.

8 Future Research Directions

The findings of this study highlight several promising areas for further research aimed at enhancing the security and performance of Software-Defined Networking (SDN) environments.

One important direction is the development of AI-driven autonomous security systems. Future work can explore the integration of advanced artificial intelligence techniques, such as deep reinforcement learning and self-adaptive models, to enable fully automated threat detection and response. Such systems would reduce human intervention and allow networks to dynamically adapt to evolving cyber threats in real time.

Another key area is the advancement of blockchain-SDN integration. While this study demonstrates the effectiveness of blockchain for secure authentication, future research can focus on optimizing consensus mechanisms to further reduce latency and computational overhead. Additionally, lightweight blockchain models and hybrid architectures can be explored to improve scalability and suitability for high-speed network environments.

The implementation of zero-trust architecture (ZTA) in SDN also presents significant research opportunities. Future studies can investigate more efficient methods for enforcing strict access control, continuous authentication, and micro-segmentation across distributed network environments. Integrating zero-trust principles with machine learning could further enhance the ability to detect insider threats and lateral movement attacks.

Furthermore, the growing adoption of distributed computing environments calls for increased focus on edge and fog security. Future research can examine how SDN-based security frameworks can be extended to edge and fog nodes, where resource constraints and decentralized operations introduce additional vulnerabilities. Developing lightweight security mechanisms suitable for these environments will be essential for supporting emerging applications in IoT and 5G networks.

In addition, future work may explore the use of federated learning techniques to enable collaborative threat detection across multiple SDN domains while preserving data privacy. This approach could improve the robustness of anomaly detection models without requiring centralised data sharing.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Conceptualization: Ifeanyi C. Emeto; methodology: Ifeanyi C. Emeto, Adamu A. Galadima, Emmanuel O. Atomatofa; software: Ifeanyi C. Emeto, Ikechukwu H. Ezech, Christiana A. Okoloegbo; validation: Ifeanyi C. Emeto, Ikechukwu H. Ezech, Christiana A. Okoloegbo; investigation: Adamu A. Galadima; resources: Emmanuel O. Atomatofa; data curation: Christiana A. Okoloegbo; writing—original draft preparation: Ifeanyi C. Emeto; writing—review and editing: Ifeanyi C. Emeto, Ikechukwu H. Ezech; visualization: Ifeanyi C. Emeto, Christiana A. Okoloegbo; supervision: Ikechukwu H. Ezech, Ifeanyi C. Emeto; project administration: Ifeanyi C. Emeto, Christiana A. Okoloegbo, Ikechukwu H. Ezech. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data supporting the conclusions of this article are available from the authors upon reasonable request.

Ethics Approval: The research did not involve people or animals, the research followed all national rules and institutional policies, and it was approved by affiliated organizations of the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kreutz D, Ramos FMV, Veríssimo PE, Rothenberg CE, Azodolmolky S, Uhlig S. Software-defined networking: a comprehensive survey. *Proc IEEE*. 2015;103(1):14–76. doi:10.1109/JPROC.2014.2371999.
2. Scott-Hayward S, O’Callaghan G, Sezer S. SDN security: a survey. In: *Proceedings of the 2013 IEEE SDN for Future Networks and Services (SDN4FNS)*; 2013 Nov 11–13; Trento, Italy. p. 1–7. doi:10.1109/SDN4FNS.2013.6702553.
3. Benzekki K, El Fergougui A, Elbelrhiti Elalaoui A. Software-defined networking (SDN): a survey. *Secur Comm Netw*. 2016;9(18):5803–33. doi:10.1002/sec.1737.
4. Nunes BAA, Mendonca M, Nguyen XN, Obraczka K, Turletti T. A survey of software-defined networking: past, present, and future of programmable networks. *IEEE Commun Surv Tutor*. 2014;16(3):1617–34. doi:10.1109/SURV.2014.012214.00180.
5. Kreutz D, Ramos FMV, Verissimo P. Towards secure and dependable software-defined networks. In: *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*; 2013 Aug 16; Hong Kong, China. p. 55–60. doi:10.1145/2491185.2491199.
6. Shin S, Xu L, Hong S, Gu G. Enhancing network security through software defined networking (SDN). In: *Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN)*; 2016 Aug 1–4; Waikoloa, HI, USA. p. 1–9. doi:10.1109/icccn.2016.7568520.
7. Open Networking Foundation. SDN architecture issue 1.1: TR-521. Palo Alto, CA, USA: Open Networking Foundation; 2016 [cited 2026 Jan 1]. Available from: https://opennetworking.org/wp-content/uploads/2014/10/TR-521_SDN_Architecture_issue_1.1.pdf.
8. Hu F, Hao Q, Bao K. A survey on software-defined network and OpenFlow: from concept to implementation. *IEEE Commun Surv Tutor*. 2014;16(4):2181–206. doi:10.1109/COMST.2014.2326417.
9. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, et al. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput Commun Rev*. 2008;38(2):69–74. doi:10.1145/1355734.1355746.
10. Lara A, Kolasani A, Ramamurthy B. Network innovation using OpenFlow: a survey. *IEEE Commun Surv Tutor*. 2014;16(1):493–512. doi:10.1109/SURV.2013.081313.00105.
11. Farhady H, Lee H, Nakao A. Software-defined networking: a survey. *Comput Netw*. 2015;81(2):79–95. doi:10.1016/j.comnet.2015.02.014.
12. Sahoo KS, Tripathy BK, Naik K, Ramasubbareddy S, Balusamy B, Khari M, et al. An evolutionary SVM model for DDOS attack detection in software defined networks. *IEEE Access*. 2020;8:132502–13. doi:10.1109/access.2020.3009733.

13. Wang S, Balarezo JF, Chavez KG, Al-Hourani A, Kandeepan S, Asghar MR, et al. Detecting flooding DDoS attacks in software defined networks using supervised learning techniques. *Eng Sci Technol Int J.* 2022;35(1):101176. doi:10.1016/j.jestch.2022.101176.
14. Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: *Proceedings of the Thirteenth EuroSys Conference*; 2018 Apr 23–26; Porto, Portugal. p. 1–15. doi:10.1145/3190508.3190538.
15. Brotsis S, Kolokotronis N, Limniotis K, Bendiab G, Shiaeles S. On the security and privacy of hyperledger fabric: challenges and open issues. *arXiv:2109.03574.* 2021.
16. Bhuiyan ZA, Islam S, Islam MM, Ullah ABMA, Naz F, Rahman MS. On the (in) security of the control plane of SDN architecture: a survey. *IEEE Access.* 2023;11:91550–82. doi:10.1109/ACCESS.2023.3307467.
17. Ataa MS, Sanad EE, El-Khoribi RA. Intrusion detection in software defined network using deep learning approaches. *Sci Rep.* 2024;14(1):29159. doi:10.1038/s41598-024-79001-1.
18. Li W, Wang Y, Li J. A blockchain-enabled collaborative intrusion detection framework for SDN-assisted cyber-physical systems. *Int J Inf Secur.* 2023;22(5):1219–30. doi:10.1007/s10207-023-00687-x.
19. Sri Vidhya G, Nagarajan R. A novel bidirectional LSTM model for network intrusion detection in SDN-IoT network. *Computing.* 2024;106(8):2613–42. doi:10.1007/s00607-024-01295-w.
20. Ram A, Chakraborty SK. A trust-aware blockchain-assisted framework for federated intrusion detection in software-defined networks. *Comput Netw.* 2026;284(7):112360. doi:10.1016/j.comnet.2026.112360.
21. Sultana N, Chilamkurti N, Peng W, Alhadad R. Survey on SDN based network intrusion detection system using machine learning approaches. *Peer Peer Netw Appl.* 2019;12(2):493–501. doi:10.1007/s12083-017-0630-0.
22. Janabi AH, Kanakis T, Johnson M. Survey: intrusion detection system in software-defined networking. *IEEE Access.* 2024;12(4):164097–120. doi:10.1109/ACCESS.2024.3493384.
23. da Silva Ruffo VG, Brandão Lent DM, Komarchesqui M, Schiavon VF, de Assis MVO, Carvalho LF, et al. Anomaly and intrusion detection using deep learning for software-defined networks: a survey. *Expert Syst Appl.* 2024;256(5):124982. doi:10.1016/j.eswa.2024.124982.
24. Khalid HYI, Aldabagh NBI. A survey on the latest intrusion detection datasets for software defined networking environments. *Eng Technol Appl Sci Res.* 2024;14(2):13190–200. doi:10.48084/etasr.6756.
25. Ayodele B, Buttigieg V. SDN as a defence mechanism: a comprehensive survey. *Int J Inf Secur.* 2024;23(1):141–85. doi:10.1007/s10207-023-00764-1.
26. Su Y, Xiong D, Qian K, Wang Y. A comprehensive survey of distributed denial of service detection and mitigation technologies in software-defined network. *Electronics.* 2024;13(4):807. doi:10.3390/electronics13040807.
27. Rahman A, Montieri A, Kundu D, Karim MR, Islam MJ, Umme S, et al. On the integration of blockchain and SDN: overview, applications, and future perspectives. *J Netw Syst Manag.* 2022;30(4):73. doi:10.1007/s10922-022-09682-4.
28. Maleh Y, Qasmaoui Y, El Gholami K, Sadqi Y, Mounir S. A comprehensive survey on SDN security: threats, mitigations, and future directions. *J Reliab Intell Environ.* 2023;9(2):201–39. doi:10.1007/s40860-022-00171-8.
29. Hayes J, Aneiba A, Gaber M, Islam MS, Abozariba R. FBA-SDN: a federated Byzantine approach for blockchain-based collaborative intrusion detection in edge SDN. In: *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*; 2023 May 28–Jun 1; Rome, Italy. p. 427–33. doi:10.1109/ICCWorkshops57953.2023.10283805.
30. Ali TE, Chong YW, Manickam S. Machine learning techniques to detect a DDoS attack in SDN: a systematic review. *Appl Sci.* 2023;13(5):3183. doi:10.3390/app13053183.
31. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*; 2018 Jan 22–24; Funchal, Portugal. p. 108–16.