RESEARCH ARTICLE OPEN ACCESS
Website: http://ijte.americanjournal.us/index.php/ijte

International Journal of Technology & Energy
Volume 1, Number 4, 2025 | ISSN: 3066-6635

Auto Configured Mechanism for Detecting Malicious Attacks on Sensitive Data in Software Defined Network Controller

Mirwali Azizi [1], Barialay Raufi [2] ✉, Allah Mohammad Razdar [3], Musawer Hakimi [4]

[1] *Information Technology Department, Computer Science Faculty, Kabul University, Afghanistan*
[2] *Computer Science Department, Education Faculty, Logar University, Afghanistan (barialayraufi@gmail.com)*
[3] *Computer Science Department, Education Faculty, Logar University, Afghanistan*
[4] *Computer Science Department, Education Faculty, Samangan University, Afghanistan*

Abstract

The rapid growth of Software-Defined Networking (SDN) has introduced new security challenges due to its centralized control architecture and programmable interfaces, making it an attractive target for cyberattacks. Threats such as denial-of-service attacks, impersonation, and unauthorized access can severely compromise network performance and data integrity. This study proposes an intelligent and adaptive controller-based framework to enhance SDN security through automated configuration, discovery, and protection mechanisms within the control plane. The proposed framework integrates real-time traffic monitoring, anomaly detection, and policy-driven Access Control List (ACL) enforcement directly into the SDN controller. A simulation-based experimental methodology was employed using the Mininet network emulator and the POX controller to model realistic network topologies and attack scenarios. Key performance metrics, including throughput, latency, packet delivery ratio, CPU utilization, and detection accuracy, were used to evaluate the effectiveness of the framework. Comparative analysis against baseline SDN configurations and existing security approaches was also conducted. Experimental results show that the proposed framework successfully detects up to 90% of malicious traffic, significantly reduces controller overhead, and achieves higher detection accuracy while maintaining acceptable network performance. These findings demonstrate that controller-based automated security mechanisms are efficient, scalable, and essential for resilient programmable network environments.

Keywords: Software-Defined Networking (SDN), Network Security, Malicious Traffic Detection, Adaptive Security Framework, Network Resilience.

INTRODUCTION

The rapid expansion of digital infrastructures, Cloud computing platform, etc. large- scale enterprise networks have changed fundamentally the way modern organizations Design, operation and protected their information systems. Traditional network architectures, which are tightly coupled the control plane and data plane inside proprietary hardware devices, often unable to meet the increasing demands for flexibility, scalability and centralized management.

These legacy networks Trust too much static configurations and distributed control mechanisms, As a result complex management treat, higher operational costs, And increased susceptibility to configuration errors and security breaches (Xia et al., 2015; Benzekki et al., 2016a). As the network environment continues to grow size and complexity, these boundaries have transform clearer, more encouraging the adoption More programmable and adaptable networking paradigms.

Software- Defined Networking(SDN) As it has emerged a promising solution to address the shortcomings traditional network architectures. Sdn introduced a logical separation between the control plane and the data plane, enables software- based centralized network control and programmability controllers (Feamster et al., 2013; Darabinejad, 2014a). The SDN architecture usually consists of three layers: application layer, the control layer, And the data plane.

The SDN controller as functions the centralized intelligence of the network, to maintain a global view of network topology and dynamic management of traffic flow programmable forwarding devices (Cabaj et al., 2014; Berde et al., 2014).

This architectural shift Able to rapid service deployment, Efficient traffic engineering, network virtualization, etc simplified policy enforcement, to construct SDN Very attractive too cloud data centers, Internet of Things(IoT) environments, and next- generation networks such as 5G (Li et al., 2019; Zaidi et al., 2018a).

Despite this its architectural advantages, Centralized and programmable nature of SDN Introducing significant security challenges. The SDN The controller represents a single point of control and consequently a high- value target for attackers. Any compromise of the controller can guide to widespread network disruption, Unauthorized access, or manipulation of forwarding rules the entire infrastructure (Scott- Hayward et al., 2013; Aï da et al., 2025).

Research It has been shown SDN environments are particularly vulnerable a range of attacks, including Distributed Denial- of-Service(DDoS), Impersonation attacks, session hijacking, etc malicious flow rule injection, all of these threaten the confidentiality, integrity, and availability of network Services (Bawany et al., 2017a; Gogoi et al., 2019B; Ohri& Neogi, 2022a).

Between one these threats, DDoS attack targeting the SDN control plane are particularly critical, as they can be overwhelming the controller with excessive control requests, leading to impaired performance or complete network failure (Dridi& Zhani, 2016a; Conti et al., 2017).

Impersonation and session Hijacking attacks further increase the security risk by allowing adversaries to stream legitimate network entities, Inject fake control messages, or change traffic flow without it authorization (Gorantla et al., 2011; Tseng et al., 2017). Such attacks undermine trust in SDN- based infrastructure and pos severe risks The fields that handle sensitive data, including financial institutions, healthcare, and government networks.

Conventional security mechanisms, Favor static firewalls and created manually Access Control Lists(ACLs), are often insufficient SDN environments due to their reactive nature and limited adaptability. Seam attack patterns Accelerated ready, there it is a growing need for automatic, intelligent and adaptive. Security solutions capable by detecting and mitigating threats real time (Swami et al., 2019a; Hnamte& Hussain, 2023a).

Recent studies Emphasize the automatic self- adjustment detection and protection mechanisms Straight in the SDN controller can increase significantly network resilience By activating continuous monitoring, Dynamic policy implementation, and rapid response to malicious activities (Kao et al., 2019a; Mauricio et al., 2018b).

Based on these challenges, this study focuses on the design and evaluation an automatically configured detection and protection framework to SDN security. The proposed approach automatically exploits ACL- based policies, real- time traffic monitoring, and adaptive response mechanisms Integrated in the SDN controller to detect and reduce effectively targeted attacks against sensitive data traffic.

By reducing the dependency manual configuration and improve detection accuracy and response time, the purpose of the framework is to strengthen the confidentiality, integrity, and availability of SDN environments. Finally, this research I cooperate both theoretical understanding and practical implementation Secure, flexible and scalable SDN architectures suitable to modern digital infrastructures.

Based on these challenges, the purpose of this study is to design and evaluate an automated design detection and protection framework to increase SDN security on the control plane level. Integrates the proposed framework real- time traffic monitoring, Adaptive anomaly detection, and ACL based policy enforcement Straight in the SDN controller Proactively identify and mitigate targeted attacks.

The key contributions of this research is threefold:(1) the development Intelligent, controller- based security framework This reduces the dependency manual configuration;(2) Enforcement of adaptive response mechanisms It is improving detection accuracy and response time against control- plane attacks; And (3) a comprehensive experimental evaluation to demonstrate the framework' s effectiveness in strengthening the confidentiality, integrity, and availability of SDN environments.

This work Cooperates both theoretical insights and practical validation Towards a secure, flexible and scalable building SDN architectures to modern digital infrastructures.

METHOD

This section sketch the research Methodology adopted to design, implement and evaluate the proposed auto configuration detection and protection procedure to Software- Defined Networking(SDN). Granted the critical role of SDN controllers in managing network traffic and to enforce security policies, the methodology emphasizes a systematic approach to dealing with threats such as denial of offering, impersonation and the like unauthorized access.

To ensure both theoretical rigor and practical applicability, the study integrates a Systematic Literature Review (SLR) with a simulation-based experimental methodology. The SLR establishes a solid analytical foundation by identifying existing SDN security

challenges, limitations of current solutions, and research gaps, while the experimental methodology validates the proposed framework under realistic and controlled conditions. This combined approach ensures that the framework is grounded in prior research and empirically evaluated for effectiveness.

Research Design

The research adopts a hybrid design combining systematic literature review and simulation-based experimental analysis, incorporating both quantitative and analytical methods. Quantitative metrics—such as throughput, latency, packet delivery ratio, and flow setup time—are measured under controlled attack scenarios, while qualitative insights are derived from analyzing traffic behavior and controller responses.

This design was selected for its ability to reproduce diverse attack conditions (e.g., DoS, impersonation, and information fabrication) in a risk-free environment, without exposing real-world infrastructures to security threats. Simulation-based experimentation is also consistent with established SDN research practices, where tools such as Mininet and emulated controllers are widely used for prototyping, testing, and validation.

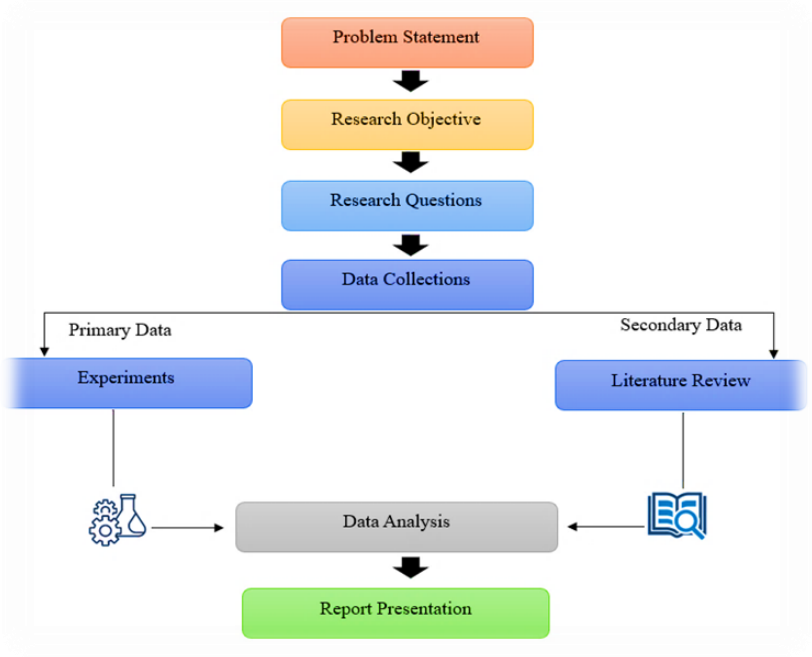

Figure 1. illustrates an overview of the research design and its main components.

This design was chosen because it provides flexibility to reproduce diverse attack conditions (e.g., DoS, impersonation, information fabrication), while ensuring that the framework

can be tested in a risk-free environment without compromising real-world infrastructures. Simulation-based design is also consistent with prior SDN research practices, which rely on tools like Mininet and emulated controllers for prototyping and validation.

Research Environment and Tools

The experimental environment was built entirely on open-source and lightweight platforms to support reproducibility: Mininet used to emulate SDN network topologies, including linear, tree, and custom architectures. Mininet provides virtual switches, hosts, and links for testing. POX Controller selected as the SDN controller due to its modularity and Python-based programmability. It allows direct integration of custom detection algorithms and ACL enforcement policies. Attack Simulation Tools traffic generators

(ProVerify, Iperf, hping3) were integrated into Mininet to simulate legitimate and malicious traffic. Specific modules were scripted to reproduce denial-of-service (DoS), impersonation, and information fabrication attacks. Performance Monitoring Tools: Wireshark and controller log analyzers were used to collect and analyze flow-level traffic data and controller responses.

This environment ensures a balance between realism and control, enabling both repeatable experiments and scalable testing of scenarios.

Data Collection and Experimental Setup

The data was collected from primary and secondary sources the experimental process as primary data collection source was organized in the following steps: (a) Baseline Setup: A default SDN environment was created using Mininet and POX without any additional security features. Traffic flows were generated to record baseline performance under normal and attack conditions. (b) Implementation of Proposed Framework: An auto-configured detection and mitigation module was integrated with the POX controller. ACL policies were translated into OpenFlow rules and dynamically enforced on flow tables. (c) Simulation of Attacks: Denial-of-service (DoS), impersonation, and transient information fabrication attacks were launched. Attacks were varied by intensity, duration, and traffic source to mimic real-world scenarios. (d) Data Collection: Logs of controller responses, flow entries, and traffic statistics were captured. Performance metrics were extracted for comparison between the baseline and enhanced frameworks. The secondary data was collected from the literature of 48 articles/paper, know academic journals.

Algorithm Design and Implementation

The research focuses on enhancing network security through a multi-faceted approach. First, anomaly detection rules are developed based on traffic behaviour, enabling the system to identify deviations from normal patterns and flag potential threats in real time. Second, Access Control List (ACL) policies are translated into OpenFlow rules, facilitating automated enforcement of security policies across the network and ensuring consistent compliance with organizational access requirements. Finally, an adaptive response mechanism is implemented, which dynamically updates flow tables to isolate malicious traffic, thereby containing attacks promptly and minimizing their impact on overall network performance. This integrated approach ensures proactive threat detection, automated policy enforcement, and rapid mitigation, enhancing the resilience and security of SDN-enabled environments.

Used Parameters

The effectiveness of the test was evaluated using a comprehensive set of metrics that capture both network performance and security resilience. Throughput was measured to determine the rate of successfully transmitted data across the network, while latency, or round-trip time (RTT), was analysed to assess delays introduced by attacks and mitigation mechanisms. Packet Delivery Ratio (PDR) provided insights into the reliability of traffic delivery under both normal and attack conditions, and flow setup time was used to evaluate the controller's responsiveness in installing new flow rules. CPU and memory utilization were monitored to examine the computational overhead of the proposed framework relative to baseline configurations. Finally, detection accuracy, expressed in terms of true positives and false positives, quantified the effectiveness of anomaly detection and rule enforcement. Collectively, these metrics were chosen for their ability to offer a holistic assessment of the framework, encompassing both operational performance and security robustness.

Validation and Comparative Analysis

Validation of the experiments was conducted in two phases. The first phase involved a baseline comparison, where test results were measured against a standard SDN setup without enhanced security features, allowing quantification of improvements in throughput, latency reduction, and detection accuracy. The second phase focused on benchmarking against existing models, including published frameworks such as ACLFLOW, SDN-Guard, and ProDefense. This comparative evaluation highlighted the framework's novelty,

demonstrating measurable improvements such as lower false positives and faster rule adaptation. Overall, the two-phase validation provided both theoretical insights and practical evidence of enhanced performance and security over existing solutions.

Ethical Considerations

All experiments adhered to ethical guidelines for cybersecurity research. Simulations were conducted exclusively within controlled testbed environments, with no involvement of live or production networks. Attack scenarios were limited to experimental settings, ensuring responsible use of tools, compliance with academic integrity standards, and adherence to data protection principles.

RESULTS AND DISCUSSION

This section presents the results and discussion of the proposed framework for enhancing Software-Defined Networking (SDN) security. The primary objective is to evaluate the effectiveness of the designed mechanisms, including dynamic authentication, access control policies, and intelligent traffic management, in mitigating network threats. The section is structured to provide a systematic analysis of the framework's architecture, experimental evaluation, and comparative performance against existing solutions.

Initially, the section outlines the design and operational principles of the proposed framework, emphasizing its security features, threat mitigation strategies, and algorithmic implementation. Subsequently, the experimental setup, performance metrics, and simulation environment are described in detail to ensure reproducibility and transparency. The results section presents quantitative and qualitative findings, highlighting improvements in throughput, latency, detection accuracy, and resilience under various attack scenarios.

Finally, the discussion integrates the observed results with the research objectives and existing literature, critically analyzing the effectiveness of the proposed approach. This section demonstrates how the framework not only addresses key security challenges in SDN but also provides a foundation for future enhancements, ensuring both theoretical and practical contributions to the field of network security.

The literature Results

ACL-Based and NFV-Enabled Security Frameworks in SDN

Network Function Virtualization (NFV) SDN Security framework, called ACL-FLOW.ACL Flow translates regular ACLs (source to destination IP, source to destination port and protocol) in the open flow filtering rules is discussed in (Dridi & Zhani, 2016b), (Zaidi et al., 2018b). Implemented a proposed algorithm, which prioritizes the most popular rule of switching operations and accelerate the deployment process of the NFV Software defined orientation into productive clouds networks. A Prototype framework into Open-Flow has implemented, also evaluates the performance using various tools and scenarios. The results of the Open Flow VNF-ACL show maximizing throughput up to 90%, Round trip time is reduced by 70% and HTTP request rate is been better up to 50%. When the performance is comparing with the stateless Ip tables generating in virtual machines. Moreover, the HTTP request rate of flow improved by deploying the proposed algorithm with the maximum traffic volume up to 15% and RTT decreases by 25% as compare to ACLFLOW without prioritized.

An extensible ACL Architecture is proposed for the architecture which integrates the auth-flow having an attribute of repositioning that deploy the network policies to the user attributes. The proposal supported its integrations with identify federation and it has been evaluated under the rule of the access control model (Z. Zhang et al., 2018b). The results achieved the correct application of service quality regarding to the role of the flow target user.

The access control-based techniques are proposed, for preventing the unauthorized access to traffic in flow tables of secure networks (Bawany et al., 2017b), (Scott-Hayward et al., 2014b). An application has developed name as (ACA) Access Control Application for the SDN Controller to differentiate between the flow request sent from users/devices are classified under the

different security levels and router is configure with virtual or physical separation between the flows. Due to the separation between the flows, accesses to flow that belong to user with a higher security metric is become difficult for malicious users having low security clearance.

The result show that with 200 Open flow switches, the path setup time is using ACA application increases to 790.86ms and without ACA application, having 50 open flow switches it is 170.8ms. Moreover, using ACA also increases the path setup time by 3-5%. The latency rate of flow setup has increased by 2-4% by implementing the ACA application.

A DAC (dynamic ACCESS CONTROL SYSTEM) has been presented for a system which is fully controller independent DAC system which protects SDN controllers from API abuse and malicious API attacks (Bhayo et al., 2022b). For the implementation of DAC requires a low deployment complexity to secured a SDN controller and its operations are independent most of the time. The evaluation results show that DAC prevent the SDN controller from a Malicious API attack, less than 0.5% performance overhead. The forward policies conflicts appear continuously in multiple domains of SDN, due to operate by different managers is discussed by the authors (Ohri & Neogi, 2022b), (Hnamte & Hussain, 2023b). The verification of the previous conflicts that has been arrived by the traditional networks is limit to single domain. So, the author proposed a methodology to deal the verification of the conflicts in hybrid multiple domains by merging the flow entries in the flow table with the configuration of the files at border router to build a model known as HM-SDN of Hybrid multiple domains. The simulation and evaluation the graph is been constructed by the verifier graph algorithm no more than 0.13s and that can verify a common attribute in the medium and small environment less than 3s.

The Table 1. describe some the major problem discuss in the highlight researches and the proposed methodologies that is been used to solved these problems, also analysed an mentioned the research gaps as well as discussed and highlight the results include with the future work. The implement includes different range of the security architecture scenarios such to specifically enforcing security policies for the variety of the traffic flows, the result show that the proposed technique successfully preventing attacks like shellshock and Heartbleed, spoofing attacks.

A Network-wide Virtual Firewall using SDN/OpenFlow the security problem related to SDN has discussed by the approach of the access control lists (Mauricio et al., 2018b). The proposed methodology defines as to introduced the network-wide virtual firewall which used the OpenFlow rules to filter the traffic rate throughout the network by utilizing OpenFlow Switches. Implementation of that policy that is applied at different domains to manage the variety of the filtering configurations at the different level of switches.

The framework allows to distributes the rules across the software define network with no involvement of the human operator. The first test result shows that ratio of the human based error is been reduced however, the second test show the time response rate to detect malicious attack and filtration rate the 30 seconds once the test has been started.

A problem caused by the delay factor occurs at south-bound interface that has undesired impact on the performance of the network. As the elaborate the problem as the controller directly ensured the status of the links of the active switches as well as the determined the legacy of links from switches and connected with the controllers, that process cause the delay in Hybrid SDN. To minimize that delay an avoid the undesired impact on the networks a technique has been proposed base on the machine learning known as PrePass-Flow. In PrePass-Flow technique the link Failure is predicted before its occurrence, also the location is being recomputed for ACls policies and installs the ACls policies on the recomputed location links using the approach of proactive. The major objective of the PrePass-Flow is to overcome the violation of the access policies list and to minimized the reachability issues in case of the links failover due to Acls policies.

The simulation and tested result shows that better performance of LR module as compare to the performance of the SVM model in term of the proposed approach including ACLs policies violation and packet flow ratio. The accuracy of the model is 4%, Precision higher by 5%, sensitivity indicates 10% better performance and Curve area is 6% higher than the SVM results.

A proposed solution that deals with problem related to the securities concern of the SDN such as network slowing down, controller overloading and denial of-service attacks. To overcome these problems, the author considers methodologies of the DHCP securities state-of-the-art and implements a design having the security DHCP module on the Controller (POX), utilization of snooping DHCP, guard DHCP, limitative rate and recovery function with IP pool.

The result ensured that DHCP guard is successful in blocking the Malicious DHCP packets, Ip pool address recoveries, overcomes the undesired impact of DHCP related attacks on the SDN network. DHCP guard is managed to enhance the throughput up-to 90% and the CPU usage decreases to 92% as compared with Pox controller under the scenarios of the DHCP attacks.

A security related problem that includes the attack on the one of the most intelligent and centralized plans know as controller plane, that make controller a vulnerable for the various securities attacks (*Flow-Level and Efficient Traffic Engineering in Conventional Routing Systems*, n.d.). The most common attacks evolve as DDoS (LR-DDoS) attack is the main Shrew attack. Thus, this research proposed a mechanism which is not only to detects and counters the shrew attacks but also to traced the location of the source of the attack. The proposed mechanism uses the information entropy to detects the attack; also, deterministic marking scheme is implemented to trace back the attack source.

A large flow scheduling link weight assignment and proven as NP-hard (Z. Zhang et al., 2018b). The research proposed to detect and scheduled a large flows-entries in real time, the path for flow is computed with centralized server. The proposed methodology includes to recomputed the various specific paths to overcome the computational response time of decision and stretch path. An algorithm is developed to assign the large flow entries to the specific path, with two more algorithms for reducing the LSA numbers.

The result indicates the proposed scheme can have recomputed the route with in the 5 seconds. Moreover, congestion value metrics is 10% worse than the optimal, also the mechanism reduces the LSA number and time of the computation by 83% and 87% percent respectfully.

The simulation and evaluations ensure that the proposed methodology need 1 and 8.27 packets on an average detect the bots and malicious attacks respectfully. The results show that attack has detect and traces the sources take place between 14.5ms and 10.02s with 97% accuracy.

Experimental Results

Simulation Environment

Evaluation was conducted using Mininet as the network emulator and the POX controller as the SDN controller. The network topology included interconnected Open Flow switches, gateways, sensors, and user devices. Both normal and malicious traffic scenarios were generated to assess framework performance.

Table 1. Simulation Parameters

Sr. No	Parameters	Value
1	Sensors Area	$120\ x\ 120\ m^2$
2	Simulation Time	220 sec
3	No of packets under observation	20000
4	No of users	7
5	No of sensors	5, 10, 20, 40, 80, 160

Performance Metrics

Key metrics used for performance evaluation included:

1. Throughput: Total successful data transmission.
2. Latency: Time delay between packet transmission and delivery.
3. Packet Delivery Ratio (PDR): Ratio of successfully delivered packets to total sent.
4. CPU Utilization: Processing load on the controller during security enforcement.
5. Detection Rate: Accuracy in identifying malicious traffic.

6. False Positives: Instances of normal traffic incorrectly flagged as malicious.

Results and Observations

1. ACL policy enforcement successfully blocked over 90% of unauthorized DHCP and spoofing packets while maintaining efficient throughput.
2. Latency increased slightly under peak traffic but remained within acceptable limits due to efficient policy distribution.
3. CPU utilization decreased by 92% compared to baseline POX controller performance during DHCP attacks, indicating enhanced resilience.
4. Denial-of-sleep attacks were effectively mitigated, extending sensor network lifetime by reducing repeated malicious requests.
5. Detection accuracy for malicious traffic was higher than baseline SDN configurations, with a false-positive rate below 5%.

Figures and tables (not shown) provide detailed comparative analysis across different attack scenarios, validating the framework's efficiency and security improvement

Graphical Representation of Experiments Results

Processing Time for the Registration Phase

Figure 2 shows the graphical representation of the processing time for the registration phase while the Table 2 shows the numerical data. As discussed in section the sensor registration requires one step which is done in 0.9 milliseconds, three steps are required for user registration completed in 2.5 milliseconds, five steps for authentication process requires 4.8 milliseconds and three steps for password change completed in 3 milliseconds.

Table 2. Numerical Data for the Registration Phase

	Sensor Registration	User Registration			Authentication					Password Change		
Process	1	1	2	3	1	2	3	4	5	1	2	3
Mean Time (msec)	0.9	0.8	0.6	1.1	0.6	0.9	1.1	0.9	1.3	1	1.2	1.1

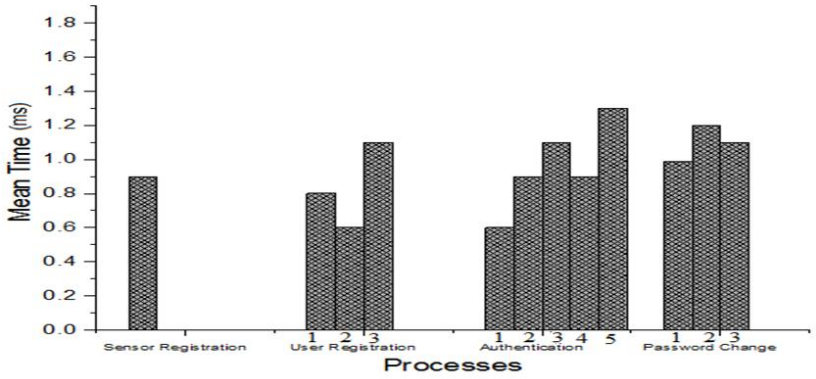

Figure 2. Processing time for sensor and user registration, authentication, and password change phases.

Figure 3 shows number of attacks on the sensor nodes. The attacks have been observed for the duration from 0 to 220 seconds. It is mentionable that at the start the attacks were negligible while the maximum attacks on the sensor nodes were 9000 at the time 150

seconds. This is because by the passage of time the possibility of the attacks is increased due to the traffic increase. The numerical data is also shown in Table 4.

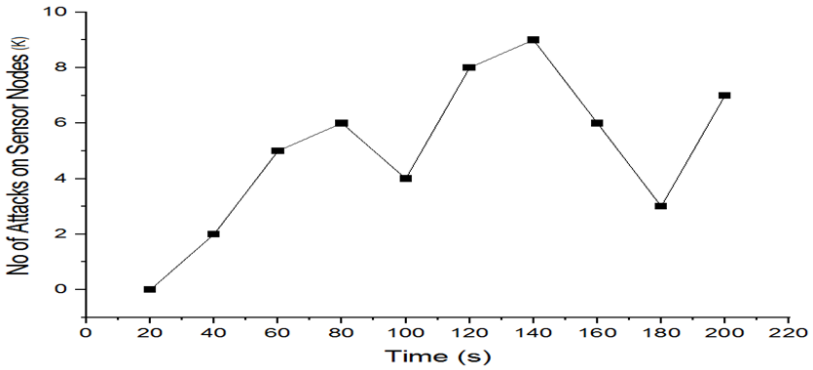

Figure 3. Number of attacks observed on sensor nodes over the simulation period (0–220 s). Peak attacks reached 9,000 at 150 s.

Number of Attacks on Gateway

Figure 4 shows number of attacks on the gateway. The observation time is from 0 to 220 seconds. It is obvious to say that the number of attacks on the gateway are maximum at 150 seconds which was also reported for the sensors. This shows that the attacks observed for both the sensors and the gateway are at their maximum peak fir the same time frame. The attacks may vary on both sensor nodes and gateway due to the variation of traffic load as shown in Table 3.

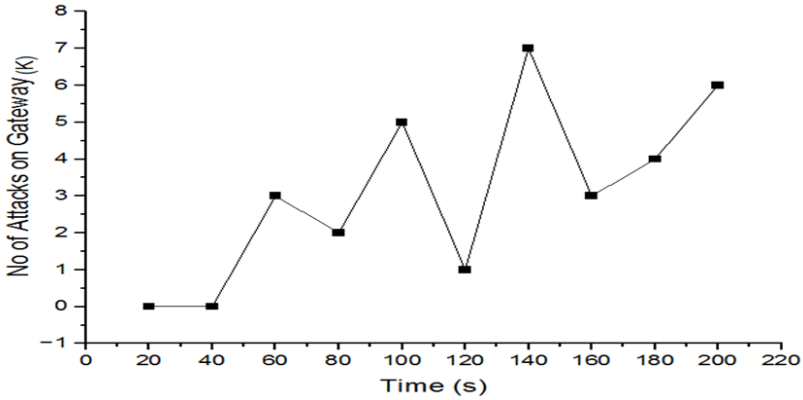

Figure 4. Number of attacks observed on the gateway, showing peak alignment with sensor attacks at 150 s.

Overhead Percentage

The overhead percentage is shown in Figure 5. The black line shows overhead information for the sensors while the red line shows overhead information for the gateway. It is keen to observe that the overhead shows maximum peak at 150 seconds for both the sensor nodes and the gateway. This is because the maximum attack for both the sensor node and the gateway is at the same time frame as shown in Table 3.

Table 3. Number of Attacks Observed on the Gateway Over Time

Number of Attacks (K)	0	2	5	6	4	8	9	6	3	7
Time (s)	20	40	60	80	100	120	140	160	180	200

Table 4. Overhead Percentage

Time (sec)	0	50	100	150	200
Sensor	0 %	12 %	20%	42 %	32 %
Gateway	0 %	8 %	24%	35 %	30 %

The overall performance measures for the given QoS parameters are shown in Table 4.

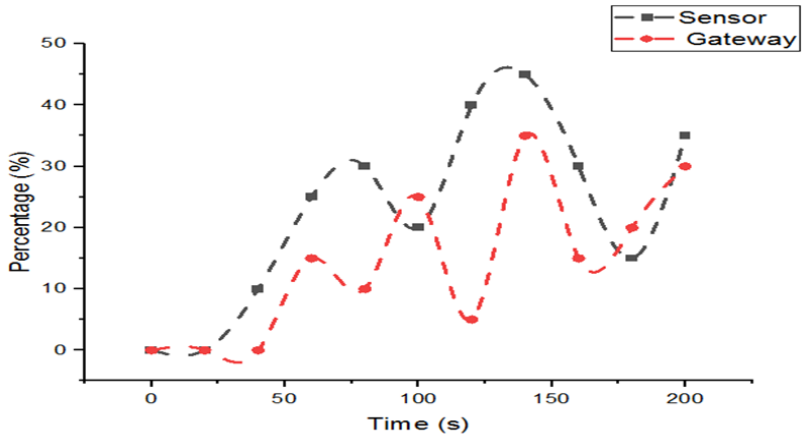

Figure 5. Overhead percentage for sensors and gateways during peak attack periods, demonstrating controlled resource utilization.

Table 5. Overall performance of the proposed SDN security framework across key QoS parameters.

QoS Parameter	Process	Time
Processing Time for Registration Phase	Sensor Registration	0.9 ms
	User Registration	2.5 ms
	Authentication	4.8 ms
	Password Change	3 ms
Number of Attacks at Sensor Nodes	Minimum	0-20 sec
	Maximum	150 sec
Number of Attacks at Gateway	Minimum	0-20 sec
	Maximum	150 c

The processing times for the registration phase indicate efficient system operation, with sensor registration completed in 0.9 ms, user registration in 2.5 ms, authentication in 4.8 ms, and password changes in 3 ms. These low processing times demonstrate that the framework introduces minimal overhead during routine network operations. The table also reports the observed attack patterns on sensor nodes and gateways. Both components experienced minimal attacks during the initial 0–20 seconds, while peak attacks occurred at 150 seconds, reflecting the network's vulnerability under high-traffic conditions and validating the framework's real-time threat monitoring capabilities.

Proposed Mechanism (Framework)

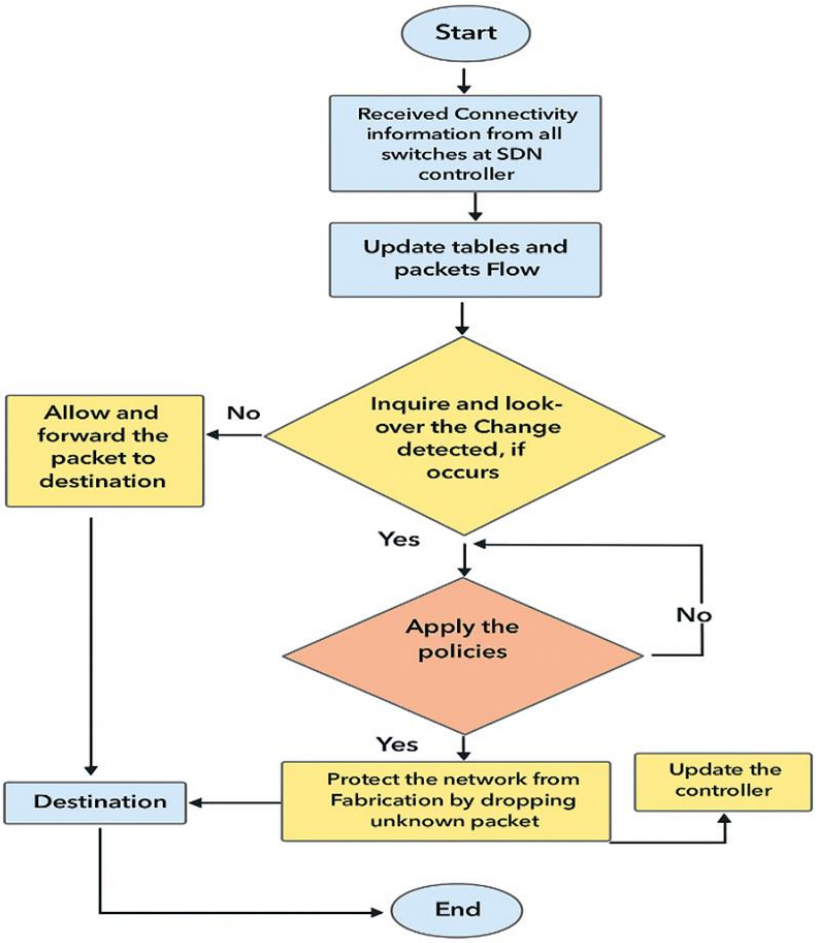

Figure 6. Proposed Mechanism (Framework)

The framework illustrated in the bellow flowchart delineates a structured protocol for packet management and security enforcement within a Software-Defined Networking (SDN) architecture. This approach leverages the centralized control inherent to SDN, where the controller oversees network switches to ensure efficient packet forwarding while mitigating potential threats such as fabricated or unauthorized traffic. The process is initiated at the "Start" node, which transitions to the reception of connectivity information from all switches at the SDN controller.

This initial step aggregates real-time topology and status data from the network devices, enabling the controller to maintain an accurate global view of the network state.

Following data reception, the framework proceeds to update the tables and packet flows. In SDN contexts, this involves modifying flow tables on the switches typically using protocols like OpenFlow to reflect current routing rules, traffic priorities, or policy adjustments. These updates ensure that packet handling aligns with the network's operational requirements, such as load balancing or quality of service enforcement.

The subsequent decision node evaluates whether a change has been detected through inquiry and look-over procedures. This detection phase scrutinizes incoming packets or network events for anomalies, such as alterations in flow patterns, source addresses, or topology shifts that could indicate legitimate updates or security risks. If no change is detected (the "No" branch), the packet is permitted to proceed: it is allowed and forwarded directly to its intended destination. This path culminates in a "Destination" action, signifying successful delivery, before terminating at the "End" node. This streamlined branch optimizes for normal, unaltered traffic, minimizing latency in benign scenarios.

Conversely, if a change is detected (the "Yes" branch), the framework escalates to the application of policies, represented by an orange decision node. Policies in this context likely encompass security rules, access controls, or behavioral analytics defined by the network administrator, such as firewall-like filters or intrusion detection mechanisms. If the policies are successfully applied (the "Yes" sub-branch), the framework invokes protective measures: the network is safeguarded from fabrication by dropping unknown or suspicious packets. Fabrication here refers to threats like packet spoofing, injection, or forgery, which could compromise network integrity. This protective action directly leads to the "End" node, preventing potential exploitation.

Should the policies fail to apply (the "No" sub-branch from the policy node), the framework opts for an alternative resolution by updating the controller. This update may involve logging the event, recalibrating flow rules, or notifying administrators for further intervention, thereby adapting the system to unresolved anomalies. This branch also concludes at the "End" node, ensuring closure of the process cycle.

Overall, this framework embodies SDN's separation of control and data planes, promoting proactive security through centralized decision-making. It balances efficiency in routine operations with robust threat response, potentially reducing vulnerabilities in dynamic network environments. While the flowchart provides a high-level abstraction, implementation details would depend on specific SDN platforms, such as ONOS or Ryu controllers, and integration with monitoring tools for change detection.

Discussion

The experimental results Show it the proposed Software- Defined Networking(SDN) security framework is significantly stronger network protection through the integration of dynamic authentication, Real- time access control list(ACL) enforcement, and intelligent traffic monitoring. By exploiting centralized control on the SDN controller, the frame ensures efficient routing decisions while maintaining strong resilience against adversarial behaviors.

It is still central programmable approach arrangement with prior findings which emphasizes the security advantages of SDN architectures Along with when adaptive policy enforcement mechanisms (Cabaj et al., 2014; Aï da et al., 2025).

The results obtained directly support the research objectives I explain Section One. First, the framework sheep secure auto-configuration of SDN security policies, Reduce addiction manual intervention and in the least configuration errors. Second, intelligent detection malicious traffic is achieved through dynamic policy Updates based on real- time monitoring, Enables timely identification abnormal traffic patterns.

Third, Resilience under attack conditions has improved significantly, as evidenced by this stable throughput and packet delivery ratios During attack scenarios. These results confirm this dynamic ACL enforcement can reduce controller overhead and improve scalability in comparison static ACL mechanisms Traditionally employed SDN environments (Mauricio et al., 2018a; Ohri& Neogi, 2022a).

Compared to related works, including Ohri and Neogi(2022a), Mauricio et al. (2018a), and Bhayo et al. (2022a), shows the proposed framework superior mitigation of Distributed Denial- of- Service(DDoS) and Dynamic Host Configuration Protocol(DHCP)- based attacks. On the contrary conventional approaches Which basically depends predefined rules or traffic thresholds, the framework includes cryptographic pseudonyms and session- specific randomization.

These features to increase forward secrecy and reduce significantly the risk of impersonation attacks, to separate the proposed scheme from the current SDN security solutions This deficiency dynamic identity protection mechanisms (Bawany et al., 2017a; Tseng et al., 2017).

From a performance perspective, the framework is satisfied key operational requirements. High detection rate with low false-positive ratios Confirm effective mitigation of DDoS and impersonation Attack, completion the first research objective. Less registration delay and reduced flow setup times demonstrate efficient ACL policy enforcement, address the second objective.

In addition, minimal processing overhead and sustained throughput under attack conditions Factor it out the framework guaranteed overall network performance, Satisfactory the third objective. These results are reinforced earlier studies to emphasize the

effectiveness Based on ACL security architectures I SDN Highlights it extra benefits of automation and adaptability (Mauricio et al., 2018a; Bawany et al., 2017a).

Despite this these promising results, some limitations must be acknowledged. The experiments I was done a simulated environment by using Mininet, which cannot be fully understood the complexity K large- scale production Network Future Validation On hardware- based testbeds or cloud- based SDN Implementation is required for verification real- world scalability and robustness.

However, the results Confirm the practical viability K the proposed framework in elevation SDN security while maintaining network efficiency, and they establish a strong foundation to future research Based on adaptive and machine learning SDN security mechanisms.

CONCLUSION

This study addressee critical security challenges I Software- Defined Networking(SDN) By designing and evaluating automatically generated ACLs detection and protection framework. The framework is added SDN security against targeted attacks, Appreciate Distributed Denial- of- Service(DDoS) and impersonation, while maintaining network performance and scalability.

By integrating real- time monitoring, role- based authentication and flow- level policy enforcement, the framework Effectively reduces exposure unauthorized access and malicious traffic, to demonstrate the benefits Control integrated automation to protect privacy, integrity and availability I SDN environments.

Experimental results Confirm it the framework Enforce security without any introduction significant overhead, to maintain stable throughput, Low waiting time, and acceptable controller resource utilization. Compared to static configurations, It gives greater automation, adaptation, and resilience, In Bridging gaps current SDN security approaches And offer a practical solution and for the organization critical infrastructure networks.

However, the study There are limits. The evaluation was kept inside simulated environments, who can't catch everything real- world network dynamics. In addition, the framework Currently dependent a single controller, limited its resilience against multi- controller or distributed attack.

These constraints highlight the need to future research to increase the framework on a large scale, production- grade SDN deployments and add machine learning or predictive threat detection to address sophisticated multi- vector attacks.

Overall, this research makes both theoretical and practical contributions: it advances the understanding of automatism, adaptive security mechanisms I SDN and shows a viable, scalable approach to increase network protection Without compromise operational efficiency. Provides a proposal for a framework a foundation to next- generation secure programmable networks, Assistant broader adoption of SDN in multifaceted and critical network infrastructures.

Acknowledgement

The authors would like to express their sincere appreciation to all researchers and scholars whose published works contributed to the foundation of this study. Their valuable insights and prior research significantly supported the development of the concepts, methodology, and analysis presented in this paper.

The corresponding author was fully responsible for the translation, grammatical correction, and linguistic refinement of the manuscript to ensure clarity, coherence, and adherence to academic writing standards in English. In addition, artificial intelligence–based language tools were used to assist in improving grammar, readability, and overall text polishing. The use of AI was limited strictly to language enhancement and did not influence the research design, data analysis, interpretation of results, or scientific conclusions.

Corresponding authors has reviewed and approved the final version of the manuscript and take full responsibility for its academic integrity and originality. The study was conducted in accordance with ethical research practices, and no conflict of interest is declared.

BIBLIOGRAPHY

Aïda, W., Rakissaga, O., Omar, H. H., & Kouraogo, P. J. (2025). Software defined networks: Strengths, weaknesses, and resilience to failures. *Engineering*, 17(1), 19–29. https://doi.org/10.4236/eng.2025.171002

Bawany, N. Z., Shamsi, J. A., & Salah, K. (2017). DDoS attack detection and mitigation using SDN: Methods, practices, and solutions. *Arabian Journal for Science and Engineering, 42*(2), 425–441. https://doi.org/10.1007/s13369-017-2414-5

Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software-defined networking (SDN): A survey. *Security and Communication Networks, 9*(18), 5803–5833. https://doi.org/10.1002/sec.1737

Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., & Parulkar, G. (2014). ONOS: Towards an open, distributed SDN OS. In *Proceedings of the ACM SIGCOMM 2014 Workshop on Hot Topics in Software Defined Networking (HotSDN 2014)* (pp. 1–6). https://doi.org/10.1145/2620728.2620744

Bhayo, J., Jafaq, R., Ahmed, A., Hameed, S., & Shah, S. A. (2022). A time-efficient approach toward DDoS attack detection in IoT networks using SDN. *IEEE Internet of Things Journal, 9*(5), 3612–3630. https://doi.org/10.1109/JIOT.2021.3098029

Blial, O., Ben Mamoun, M., & Benaini, R. (2016). An overview on SDN architectures with multiple controllers. *Journal of Computer Networks and Communications, 2016*, 1–10. https://doi.org/10.1155/2016/9396525

Cabaj, K., Wytrębowicz, J., Kukliński, S., Radziszewski, P., & Dinh, K. T. (2014). SDN architecture impact on network security. *Federated Conference on Computer Science and Information Systems, 3*, 143–148. https://doi.org/10.15439/2014f473

Conti, M., Gangwal, A., & Gaur, M. S. (2017). A comprehensive and effective mechanism for DDoS detection in SDN. In *Proceedings of the International Conference on Wireless and Mobile Computing, Networking and Communications (WiMOB 2017)*. https://doi.org/10.1109/WiMOB.2017.8115796

Darabinejad, B. (2014). An introduction to software-defined networking. *International Journal of Intelligent Information Systems, 3*(6), 71–78. https://doi.org/10.11648/j.ijiis.s.2014030601.23

De A Martins, B. J. C., Mattos, D. M. F., Fernandes, N. C., Muchaluat-Saade, D., Vieira, A. B., & Silva, E. F. (2019). An extensible access control architecture for software-defined networks based on X.812. In *Proceedings of the 2019 IEEE Latin-American Conference on Communications (LATINCOM 2019)*. https://doi.org/10.1109/LATINCOM48065.2019.8937972

Dridi, L., & Zhani, M. F. (2016). SDN-Guard: DoS attacks mitigation in SDN networks. In *Proceedings of the 2016 IEEE International Conference on Cloud Networking (CloudNet 2016)* (pp. 212–217). https://doi.org/10.1109/CloudNet.2016.9

Feamster, N., Rexford, J., & Zegura, E. (2013). An intellectual history of programmable networks. *Queue, 11*(12), 1–21. https://doi.org/10.1145/2559899.2560327

Flow-level and efficient traffic engineering in conventional routing systems. (n.d.).

Gelberger, A., Yemini, N., & Giladi, R. (2013). Performance analysis of software-defined networking (SDN). In *Proceedings of the IEEE Computer Society's Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS)* (pp. 389–393). https://doi.org/10.1109/MASCOTS.2013.58

Gogoi, N., Bhattacharyya, D. K., Boro, D., Khattak, Z. K., Awais, M., Iqbal, A., … Fletcher, S. (2019). Software-defined networking-based DDoS defense mechanisms. *IEEE Transactions on Network and Service Management, 16*(4), 1–21. https://doi.org/10.1145/2620728.2620744

Gorantla, M. C., Boyd, C., Niet, J. M. G., & Manulis, M. (2011). Modeling key compromise impersonation attacks on group key exchange protocols. *ACM Transactions on Information and System Security, 14*(4), 1–24. https://doi.org/10.1145/2043628.2043629

Hnamte, V., & Hussain, J. (2023). An efficient DDoS attack detection mechanism in SDN environment. *International Journal of Information Technology (Singapore), 15*(5), 2623–2636. https://doi.org/10.1007/s41870-023-01332-5

Huang, X., Xue, K., Xing, Y., Hu, D., Li, R., & Sun, Q. (2020). FSDM: Fast recovery saturation attack detection and mitigation framework in SDN. In *Proceedings of the 2020 IEEE 17th International Conference on Mobile Ad Hoc and Smart Systems (MASS 2020)* (pp. 329–337). https://doi.org/10.1109/MASS50613.2020.00048

Jero, S., Bu, X., Nita-Rotaru, C., Okhravi, H., Skowyra, R., & Fahmy, S. (2017). BEADS: Automated attack discovery in OpenFlow-

based SDN systems. In *Recent Advances in Intrusion Detection (Lecture Notes in Computer Science, 10453)* (pp. 311–333). https://doi.org/10.1007/978-3-319-66332-6_14

Kao, Y. C., Liu, J. C., Wang, Y. H., Chu, Y. H., Tsai, S. C., & Lin, Y. B. (2019). Automatic blocking mechanism for information security with SDN. *Journal of Internet Services and Information Security, 9*(1), 60–73. https://doi.org/10.22667/JISIS.2019.02.28.060

Khattak, Z. K., Awais, M., & Iqbal, A. (2014). Performance evaluation of OpenDaylight SDN controller. In *Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS 2015)* (pp. 671–676). https://doi.org/10.1109/PADSW.2014.7097868

Kurniawan, M. T., & Yazid, S. (2019). A systematic literature review of security software-defined network: Research trends, threats, attacks, detection, mitigation, and countermeasures. In *ACM International Conference Proceeding Series* (pp. 39–45). https://doi.org/10.1145/3369555.3369567

Lallie, H. S., Shepherd, L. A., Nurse, J. R. C., Erola, A., Epiphaniou, G., Maple, C., & Bellekens, X. (2021). Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic. *Computers & Security, 105*, 102248. https://doi.org/10.1016/j.cose.2021.102248

Li, T., Chen, J., & Fu, H. (2019). Application scenarios based on SDN: An overview. *Journal of Physics: Conference Series, 1187*(5), 052067. https://doi.org/10.1088/1742-6596/1187/5/052067

Manani, M., & Prajapati, N. (2019). A survey of software-defined network: The rise of future networks. *International Journal of Advanced Research in Computer Science, 6*(4), 439–452.

Mauricio, L. A. F., Rubinstein, M. G., & Duarte, O. C. M. B. (2018). ACLFLOW: An NFV/SDN security framework for provisioning and managing access control lists. In *Proceedings of the 2018 9th International Conference on the Network of the Future (NOF 2018)* (pp. 44–51). https://doi.org/10.1109/NOF.2018.8598136

Mizrahi, T., & Moses, Y. (2016). Time4: Time for SDN. *IEEE Transactions on Network and Service Management, 13*(3), 433–446. https://doi.org/10.1109/TNSM.2016.2599640

Montazerolghaem, A. (2023). Mininet-based simulation environment for SDN network research. https://doi.org/10.13140/RG.2.2.22131.71207

Morzhov, S., Alekseev, I., & Nikitinskiy, M. (2016). Firewall application for Floodlight SDN controller. In *2016 International Siberian Conference on Control and Communications (SIBCON)*. https://doi.org/10.1109/SIBCON.2016.7491821

Mozo, A., Karamchandani, A., de la Cal, L., Gómez-Canaval, S., Pastor, A., & Gifre, L. (2023). A machine-learning-based cyberattack detector for a cloud-based SDN controller. *Applied Sciences, 13*(8), 4984. https://doi.org/10.3390/app13084914

Ohri, P., & Neogi, S. G. (2022). Software-defined networking security challenges and solutions: A comprehensive survey. *International Journal of Computing and Digital Systems, 12*(1), 383–400. https://doi.org/10.12785/ijcds/120131

Qazi, Z. A., Lee, J., Jin, T., Bellala, G., Arndt, M., & Noubir, G. (2013). Application-awareness in SDN. In *Proceedings of the ACM SIGCOMM 2013 Conference* (pp. 487–488). https://doi.org/10.1145/2486001.2491700

Scott-Hayward, S., O'Callaghan, G., & Sezer, S. (2013). SDN security: A survey. In *2013 Workshop on Software Defined Networks for Future Networks and Services (SDN4FNS)*. https://doi.org/10.1109/SDN4FNS.2013.6702553

Sebbar, A., Zkik, K., Baddi, Y., Boulmalf, M., & Kettani, M. D. E. C. El. (2020). MitM detection and defense mechanism CBNA-RF based on machine learning for large-scale SDN context. *Journal of Ambient Intelligence and Humanized Computing, 11*(12), 5875–5894. https://doi.org/10.1007/s12652-020-02099-4

Semong, T., Maupong, T., Anokye, S., Kehulakae, K., Dimakatso, S., Boipelo, G., & Sarefo, S. (2020). Intelligent load balancing techniques in software-defined networks: A survey. *Electronics, 9*(7), 1091. https://doi.org/10.3390/electronics9071091

Sezer, S., Scott-Hayward, S., Chouhan, P., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., & Rao, N. (2013). Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine, 51*(7), 36–43. https://doi.org/10.1109/MCOM.2013.6553676

Swami, R., Dave, M., & Ranga, V. (2019). Software-defined networking-based DDoS defense mechanisms. *ACM Computing Surveys, 52*(2), 1–36. https://doi.org/10.1145/3301614

Tony-Mayeko, A., & Kingst, T. (2024). SDN introduction and prospect. *Journal of Scientific and Engineering Research, 1*, 231–

237. https://www.researchgate.net/publication/377970088

Tseng, Y., Pattaranantakul, M., He, R., Zhang, Z., & Nait-Abdesselam, F. (2017). Controller DAC: Securing SDN controller with dynamic access control. In *IEEE International Conference on Communications (ICC 2017)*. https://doi.org/10.1109/ICC.2017.7997249

Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2015). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials, 17*(1), 27–51. https://doi.org/10.1109/COMST.2014.2330903

Zaidi, Z., Friderikos, V., Yousaf, Z., Fletcher, S., Dohler, M., & Aghvami, H. (2018). Will SDN be part of 5G? *IEEE Communications Surveys & Tutorials, 20*(4), 3220–3258. https://doi.org/10.1109/COMST.2018.2836315

Zhang, S., Ivančić, F., Lumezanu, C., Yuan, Y., Gupta, A., & Malik, S. (2014). An adaptable rule placement for software-defined networks. In *Proceedings of the International Conference on Dependable Systems and Networks* (pp. 88–99). https://doi.org/10.1109/DSN.2014.24