iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** II **Month of publication:** February 2025

DOI: https://doi.org/10.22214/ijraset.2025.66711

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue II Feb 2025- Available at www.ijraset.com

Experimental Design and Analysis of an Enhanced SDN-Based Model for Intrusion Detection

Ahmad Ajiya Ahmad[1], Prof. Souley Boukari[2], Abdullahi Musa Bello[3], Abubakar M. Bichi[4], Sa'adatu Gimba[5], Mustapha A. Muhammad[6]

[1, 3, 4, 5]*Department of Computer Science, Faculty of Science, Federal University Gashua, Yobe State, Nigeria*
[2, 6]*Mathematical Science Department, Abubakar Tafawa Balewa University, Bauchi Sate, Nigeria*

Abstract: ***The diversification and growth of cyberspace in the world always increases exponentially. This Antecedent increased in network hurdle and complexity of managing associated network traffic. Traditionally, organizations amplify their network bandwidth with centering on purchasing more hardware to patch up such difficulty. This approach is a drawback ever since it could be an expensive lapse if the supplementary network resources are not fully utilized. Furthermore, it is habitually sometimes unfeasible to manage the traditional network in a timely manner to respond to errors and malicious activities. Software-Defined Network (SDN) has been emerging technology developed to reduce network complexity and enhanced management of the entire network effectively. SDN is an efficient and robust architecture capable to control the network from centralized point unlike the traditional network architecture. SDN discrete the data and control activity of data communications devices, with an Application programming Interface (APT). The API is a set of rules and protocols that operate as intermediaries allowing communication between several applications without revealing the inner workings of the network. Nowadays, different data center's network broadly recognizes and implemented SDN, yet, it is facing many threats which are still challenging on securing the SDN architecture. Security problems related to SDN design include, securing network devices communication, controlling the capacity of applications and risk of a compromised SDN controller threat. Therefore, deploying Intrusion Detection Systems (IDSs) to observe and manage malicious activities is an essential factor of the network design. In this work we developed an enhanced, secured and robust SDN model for intrusion detection using snort protection mechanism. The proposed SDN network is tested for quality of service (QoS) under three different network scenarios and measure the performance based on bandwidth, throughput, latency, packet loss and CPU usage. The simulation involved generating normal and various attack traffic with a different packet sizes ranging from 128 bytes to 2,048 bytes. Comparative analysis with the existing model prove the improve quality of service of the proposed SDN network.***

Keywords: Software Defined Network (SDN), Intrusion Detection Systems (IDSs), Application programming Interface (APT), Cyberspace, Quality of service (QoS).

I. INTRODUCTION

Cybersecurity challenges have become increasingly complex and pervasive in today's interconnected world. The rapid expansion of digital systems and networks has opened new avenues for cybercriminals to exploit vulnerabilities and breach sensitive information [2]. This evolution has exposed sensitive data to significant risks, necessitating advanced strategies to combat increasingly sophisticated threats [28]. Traditional methods of intrusion detection, which often rely on signature-based approaches, have shown limitations in identifying and mitigating sophisticated and evolving cyber attacks [12]. The potential consequences of successful cyber intrusions range from financial losses to compromised privacy, reputational damage, business disruption, and threats to national security [29].

Successful cyber intrusions can lead to severe consequences such as data breaches, financial theft, reputational harm, and operational disruptions. These attacks may compromise critical infrastructure, disrupt essential services, and lead to prolonged risks associated with stolen data. Such threats emphasize the importance of robust and adaptive intrusion detection systems (IDS) as organizations strive to protect their networks from an increasingly diverse range of cyber threats. The financial and operational costs of addressing these challenges continue to escalate, making it essential to explore innovative solutions to enhance network security.

In response to these challenges, Software-Defined Networking (SDN) has emerged as a promising paradigm for enhancing network performance and security. SDN is a novel network architecture that separates the data and control layers, enabling centralized control and dynamic reconfiguration of network policies [15]. By decoupling these layers, SDN offers increased flexibility, scalability, and simplified network management compared to traditional networks. SDN also improves Quality of Service (QoS) by enabling intelligent resource allocation and real-time traffic monitoring [27].

Traditional network architectures rely on hardware-centric approaches, where devices like routers and switches integrate forwarding and control functions. This integration creates challenges in scaling and adapting to large networks due to processing overhead and bandwidth limitations [10]. In contrast, SDN leverages a centralized controller to manage network operations, reducing complexity and enabling more efficient resource utilization [17]. Moreover, SDN's centralized framework facilitates enhanced security by allowing for more effective monitoring, detection, and mitigation of cyber threats [14].

Despite its advantages, SDN is not immune to security vulnerabilities. The architecture's centralized nature makes the controller a high-value target for attackers, and unauthorized access to network components poses significant risks. To address these vulnerabilities, integrating SNORT IDS techniques has become an area of active research. These techniques complement SDN's centralized control, providing dynamic and adaptive intrusion detection capabilities that outperform traditional rule-based methods.

In this study, we present an enhanced SDN-based model for intrusion detection, incorporating Snort protection mechanisms to address the limitations of existing frameworks. The proposed model is tested under various scenarios to evaluate its performance using metrics such as bandwidth, throughput, latency, packet loss, and CPU usage. By leveraging the strengths of SDN and advanced intrusion detection techniques, this research aims to provide a scalable and effective solution to the pressing cybersecurity challenges faced by modern networks.

A. Objectives

The main aim of this research is to develop an enhanced and secure Software Defined Networking model for intrusion detection. To achieve the goal of the research the following objectives are followed:

1) To implement virtual SDN model for monitoring network traffic behaviour.
2) To incorporate SDN multiple controllers into the model to avoid packets traffic overload.
3) To evaluate the performance of the model against the existing works.

B. SDN Architecture

Software-Defined Networking (SDN) is an innovative network architecture that separates the control plane from the data plane in a network. SDN is a novel revolutionizes technology over the traditional network architectures [38], [19]. SDN has built-in protocol that is called OpenFlow protocol [1], [22], it enables researchers and information technology (IT) vendors to develop and implement original resourceful network systems in an extensive approach, flexible, effective and efficient. In general, OpenFlow is currently the best-deployed SDN model, which provides communication among devices such as the switches and controllers [37]. Fig. 1 shows the SDN architecture.

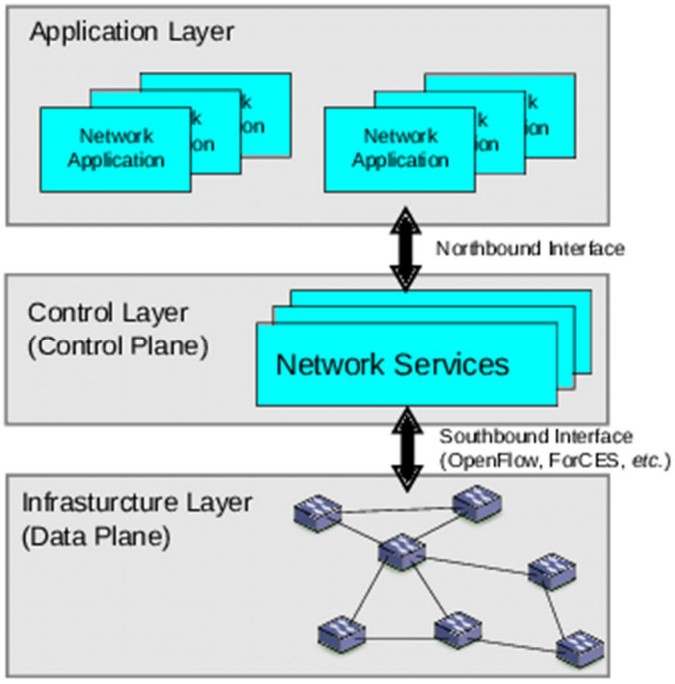


Fig. 1 SDN Architecture

1) Control Plane

The control plane is a fundamental component of networking architectures, including Software-Defined Networking (SDN), responsible for making decisions about how network traffic is managed, routed, and controlled. It is essentially the "brain" of the network, dictating how data should flow, how resources should be allocated, and how network policies should be enforced [1], [12].

In traditional networking, the control plane is integrated within individual network devices, such as switches and routers. Each device autonomously determines how to forward data packets based on protocols like Routing Information Protocol (RIP) or Open Shortest Path First (OSPF). While this decentralized approach worked well for managing relatively small and static networks, it became challenging to handle the increasing complexity and dynamic nature of modern networks [30].

In the context of SDN, the control plane takes on a new role. It is separated from the data plane, which handles the actual forwarding of data packets. In SDN, the control plane is centralized within a software-based entity known as the SDN controller. The controller holds the responsibility of configuring and managing the behavior of network devices in the data plane [34].

Key functions of the control plane include:

- *Network Topology Discovery:* The control plane is responsible for discovering the network's topology, understanding how devices are interconnected, and their capabilities.
- *Routing and Path Selection:* It determines the optimal routes for data packets to traverse the network, considering factors such as latency, bandwidth, and link availability.
- *Traffic Engineering:* The control plane manages traffic flows to ensure efficient utilization of network resources, load balancing, and avoiding congestion.
- *Policy Enforcement:* Network policies, security rules, and Quality of Service (QoS) requirements are defined and enforced through the control plane.
- *Dynamic Adaptation:* In SDN, the control plane can dynamically adjust network behavior based on changing conditions, such as traffic patterns, failures, or security threats.
- *Configuration Management:* The control plane handles the configuration of network devices, ensuring they operate according to desired settings and policies.
- *Path Computation:* For more advanced applications, such as traffic engineering and Service Level Agreement (SLA) enforcement, the control plane calculates optimal paths for traffic based on certain criteria [16].

In traditional networks, the distributed control plane can lead to configuration inconsistencies, scalability challenges, and difficulties in enforcing consistent policies. SDN's centralized control plane addresses these issues by providing a unified point of management and enabling network-wide policies to be applied consistently [13].

2) Data Plane

The data plane is a fundamental component in networking, including the context of Software-Defined Networking (SDN). It is responsible for the actual forwarding of data packets through the network, following the instructions provided by the control plane [36]. The data plane, often referred to as the "forwarding plane," handles the low-level operations involved in moving data packets from source to destination.

In traditional networking, the data plane is integrated within individual network devices such as switches, routers, and access points. These devices use forwarding tables and routing algorithms to determine the next hop for incoming data packets based on their destination addresses. This distributed forwarding process occurs in real-time and ensures that data is efficiently directed through the network. In the SDN paradigm, the data plane is distinct from the control plane. The control plane is centralized within an SDN controller [9], while the data plane remains distributed among network devices. However, the crucial distinction is that the control plane dictates how the data plane should operate, rather than each device independently making forwarding decisions.

Key functions and characteristics of the data plane include:

- *Packet Forwarding:* The primary function of the data plane is to forward incoming data packets to their appropriate next-hop destinations based on information provided by the control plane.
- *Traffic Classification:* Data plane devices can classify incoming packets based on various attributes, such as source/destination IP addresses, port numbers, and protocol types, allowing for quality of service (QoS) enforcement and traffic shaping [33].
- *Traffic Queuing and Scheduling:* In networks with varying levels of service quality requirements, the data plane may prioritize and manage the queuing and scheduling of packets to ensure timely delivery of high-priority traffic.
- *Data Packet Processing:* Network devices in the data plane can perform tasks like packet encapsulation/decapsulation, network address translation (NAT), and packet fragmentation/reassembly.

- *Load Balancing:* Data plane devices can distribute traffic across multiple paths or links to optimize resource utilization and reduce congestion.
- *Quality of Service (QoS):* Data plane mechanisms can enforce QoS policies by giving priority to certain types of traffic, such as voice or video, over less time-sensitive traffic [7].

In SDN, the data plane devices communicate with the centralized SDN controller through the southbound interface. The controller provides instructions on how the data plane should handle incoming packets, including forwarding decisions and potential modifications to packets.

The separation of the control plane and the data plane in SDN offers several advantages, including centralized policy enforcement, real-time adaptability, and simplified network management. By centralizing control and delegating forwarding decisions to network devices, SDN enables more efficient and flexible network operations.

II. RELATED WORK

Software-Defined Networking (SDN) has emerged as a transformative paradigm in network management, offering centralized control and enhanced programmability that streamline operations and facilitate innovation within modern data centers. This architectural shift decouples the control plane from the data plane, enabling more flexible and efficient network configurations. However, this centralization introduces unique security challenges that have been extensively examined in the literature. [16] Provide a comprehensive survey of SDN security, identifying potential threats such as attacks on the controller, data plane compromises, and vulnerabilities within application programming interfaces (APIs). Similarly, [28] discuss the security challenges inherent in SDN, emphasizing issues like denial-of-service (DoS) attacks and the increased attack surface due to the separation of control and data planes.

To address these security concerns, various intrusion detection systems (IDS) have been proposed. [25] Introduced an IDS tailored for OpenFlow-based SDN architectures, focusing on real-time threat detection and mitigation. Their system leverages the centralized nature of SDN to monitor network traffic and identify anomalies indicative of potential security breaches. Building upon this foundation, [5] explored the integration of machine learning techniques with IDSs to enhance detection accuracy. By employing algorithms capable of learning from network traffic patterns, their approach aims to improve the identification of both known and unknown threats, adapting to the evolving landscape of cyber threats.Hybrid detection techniques have also been investigated to bolster SDN security. [19] Examined the combination of signature-based and anomaly-based methods to detect a broader spectrum of threats. Signature-based detection is effective against known attacks, while anomaly-based detection can identify deviations from normal behavior, potentially uncovering novel threats. [13] Further advanced this approach by proposing a hybrid IDS that adapts to the dynamic nature of SDN environments, aiming to balance detection accuracy with computational efficiency. Despite these advancements, challenges remain in achieving scalable and efficient IDS solutions for SDN. Many existing systems impose significant computational overhead, which can adversely affect network performance, particularly in large-scale deployments [4]. Additionally, the rapid evolution of network traffic patterns necessitates IDSs that can adapt in real-time without compromising accuracy or speed. Addressing these limitations is crucial for the practical deployment of IDSs in SDN environments.

This study seeks to contribute to this ongoing effort by developing a lightweight, Snort-based IDS tailored specifically for SDN environments. Snort, a widely used open-source network intrusion detection system, offers a robust platform for real-time traffic analysis and packet logging. By customizing Snort to operate within the SDN architecture, this research aims to provide a scalable and efficient solution that minimizes computational overhead while maintaining high detection accuracy. The proposed system leverages the centralized control inherent in SDN to enhance monitoring capabilities and streamline threat mitigation processes.

III. METHODOLOGY

The current Software-Defined Networking (SDN) systems face significant challenges, particularly those utilizing single-controller architectures. Issues such as scalability constraints, single points of failure, increased network overhead, and susceptibility to Denial of Service (DoS) attacks severely limit their effectiveness. [2] Designed a scheme to evaluate SDN performance against DoS attacks but relied on single-controller architecture with firewall rules for mitigation. This approach lacked robustness and failed to address broader security threats comprehensively. To resolve these issues, this research proposes a distributed SDN multi-controller architecture to enhance scalability and reduce controller overload. The model incorporates SNORT Intrusion Detection System (IDS) for efficient traffic monitoring and mitigation of network vulnerabilities. SNORT, complemented with machine learning classification techniques [17], and enhances detection capabilities. The model is designed to proactively identify and neutralize network threats, with low false positive rates, improving the overall security and robustness of the SDN system.

A. *Architecture and Design of the Proposed Model*

The proposed model is developed using Mininet, an SDN emulator designed to simulate virtual networks. Mininet provides an intuitive GUI and CLI for creating complex network topologies, including hosts, switches, and controllers. The SDN architecture relies on the OpenFlow protocol, a standardized communication interface between the controller and the data plane. OpenFlow ensures seamless management of network traffic and facilitates dynamic adjustments to the forwarding plane for optimal performance [35].

The architecture integrates SNORT IDS as the central security mechanism. Operating on a rule-based approach, a SNORT monitors network traffic and detects malicious activities such as Distributed Denial of Service (DDoS) attacks. When a malicious packet is identified, SNORT alerts the SDN controller, which blocks the source IP address to prevent further damage. This event-driven approach significantly reduces latency and minimizes control plane overhead by delegating certain responsibilities to the data plane [32].

The distributed multi-controller framework ensures that traffic is balanced across multiple controllers, improving scalability and robustness. This design eliminates single points of failure and enhances the system's ability to handle increased traffic loads. Additionally, the model employs tools like Sflow-rt for real-time packet tracking, providing comprehensive insights into network behavior and performance [21]. Fig. 2 shows the Architecture of the Proposed System.

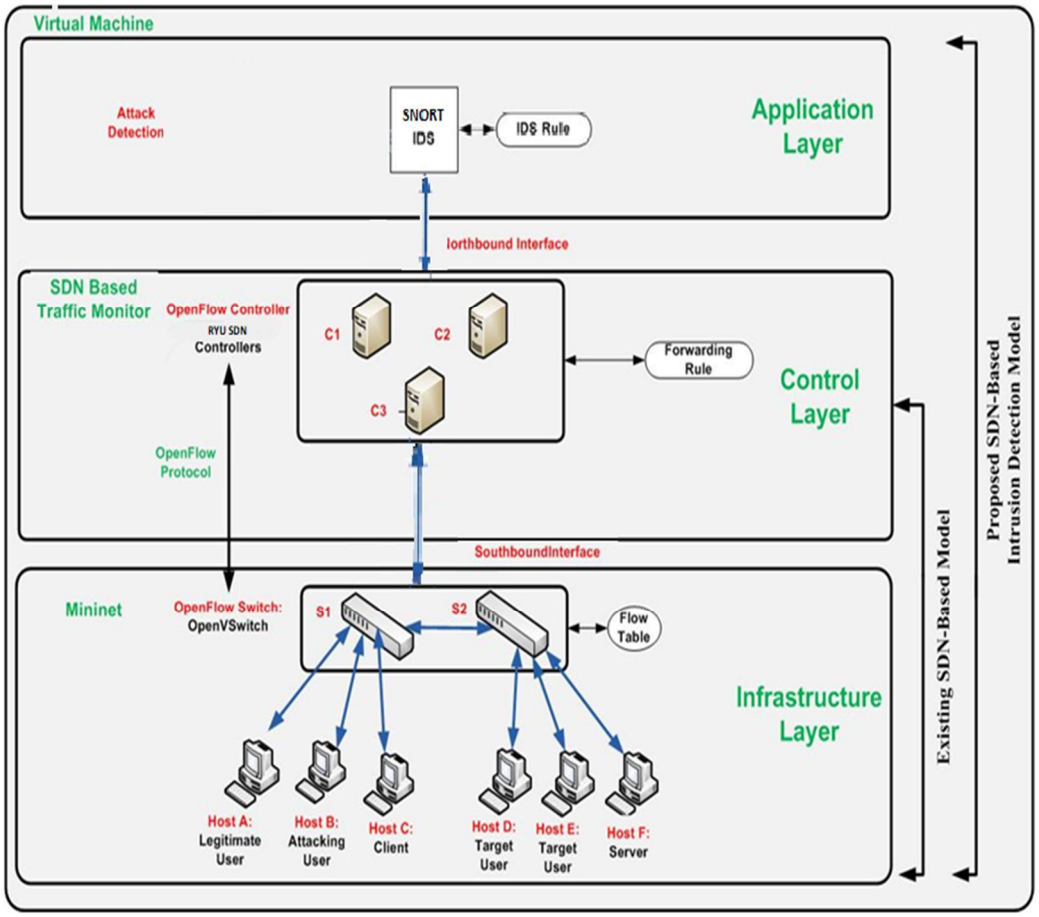


Fig. 2 Architecture of the Proposed System

B. *Implementation and Tools*

The implementation of the proposed model is carried out in a virtualized environment using VirtualBox. The system specifications include an Intel Core i5 processor, 8 GB RAM, and a 64-bit Windows 10 host operating system. The virtual environment runs Ubuntu 18.04 as the guest operating system, with Mininet version 2.3.1b1 and Ryu controller version 4.31 forming the foundation of the SDN framework.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue II Feb 2025- Available at www.ijraset.com

The network topology consists of six hosts, two switches, and three controllers. Mininet creates the topology, while OpenVSwitch is used as a virtual switch to manage packet forwarding. SNORT IDS is installed on the OpenFlow switch machine to observe and analyze network activities. Machine learning techniques, implemented using the WEKA tool, are employed for traffic classification and prediction [17]. The methodology also incorporates k-fold cross-validation to evaluate the performance of the classification models.

For traffic generation, tools like Iperf and Hping3 are utilized to simulate normal and malicious traffic, including various types of DoS and DDoS attacks (e.g., TCP-SYN flood and ICMP flood). SNORT uses predefined rules to identify these attacks and notify the SDN controller [26]. Wireshark is used for packet analysis, capturing network traffic and providing detailed insights into packet flows.

The architecture includes three links:

1) Link A, connects the SDN controllers and switches for basic network operations.
2) Link B, transmits scanned traffic from the switches to SNORT for detailed inspection.
3) Link C, relays alerts from SNORT to the SDN controllers for immediate action.

The model ensures that malicious traffic is identified and blocked efficiently. The switches maintain flow tables to store information about packets, including those flagged as malicious. Upon detection of an attack, the system dynamically updates flow rules to prevent the attacker from sending further packets [5].

C. Evaluation and Validation

The performance of the proposed model is evaluated against existing SDN-based solutions. Metrics such as system latency, throughput, packet loss, bandwidth, and CPU usage are measured. By incorporating SNORT IDS the experimental results validate the effectiveness of the proposed solution in detecting and mitigating network threats. The distributed multi-controller architecture ensures minimal latency and network overhead [25], addressing the scalability limitations of single-controller systems. The adaptation of the SNORT IDS and advanced traffic analysis tools ensures a robust and secure SDN environment, capable of handling modern network challenges [3].

IV. EXPERIMENTAL RESULT AND DISCUSSION

This section presents the experimental results and evaluates the performance of the proposed SDN network design under three different scenarios, namely Single-Controller SDN, Multiple-Controller SDN with Snort and Multiple-Controller SDN without Snort.

- Scenarios 1: single controller SDN consisting of six hosts and two switches
- Scenarios 2: SDN with 3 controllers, 2 switches, and 6 hosts with SNORT, and
- Scenarios 3: SDN with 3 controllers, 2 switches, and 6 hosts without SNORT.

The performance metrics analyzed include latency, throughput, packet loss, bandwidth, and CPU usage. The experiments were conducted using Mininet. TCP, UDP and ICMP traffic were generated by utilizing iperf and ping benchmarking tools. Results were averaged over ten trials for each test to ensure reliability. Consequently, the performance analysis for customs networks SDN scenarios is prepared by comparing all three custom scenarios on the basis of packet transmission rate. We repeatedly increase the transmitted packet size from 128 Bytes to 2,048 Bytes by a multiple of 2. The following subsections below offered the obtained results for each performance metric.

A. Latency

Latency, a critical metric, measures the time taken for packets to travel from the source to the destination. Using ICMP traffic between Host C and Host F, results showed that the Single-Controller SDN had the lowest latency across all scenarios due to centralized control. However, as the packet size increased, latency gradually rose. The Multiple-Controller SDN with Snort exhibited higher latency, attributed to the additional processing overhead of intrusion detection. Conversely, the Multiple-Controller SDN without Snort maintained latency values closer to the Single-Controller SDN, demonstrating the efficiency of distributed control in managing traffic. Fig. 3 highlights the latency distribution for various packet sizes, with the Single-Controller SDN outperforming the others.

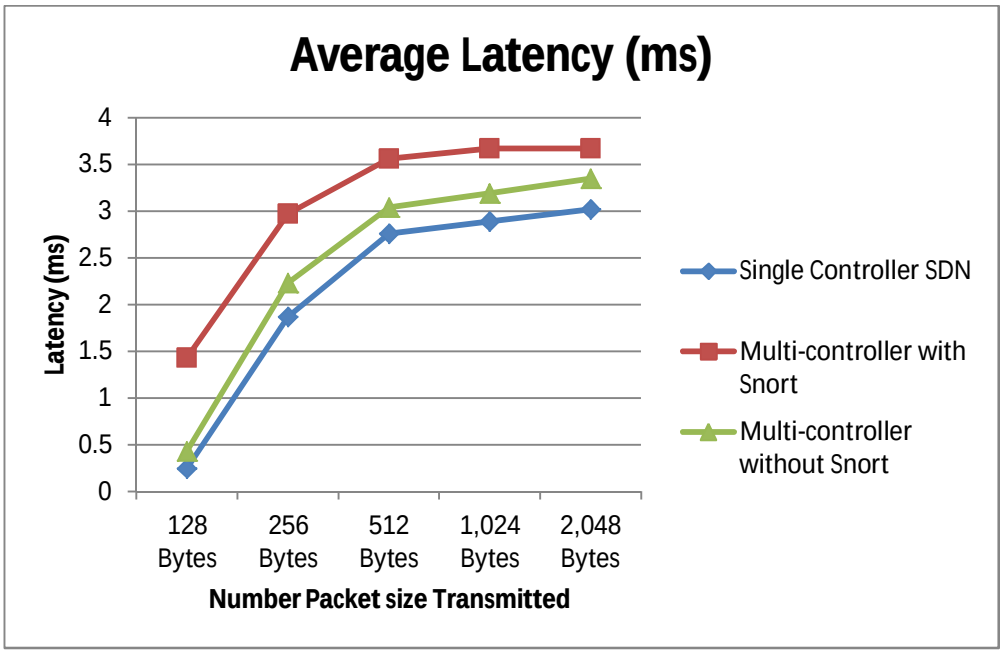


Fig. 3 Latency distribution for various packet sizes

B. Throughput

Throughput represents the data transmission rate across the network. It was evaluated using Iperf3 to measure the average data rate between Host C and Host F under TCP connections. Single-Controller SDN demonstrated the highest throughput, benefiting from streamlined communication through its centralized controller. The Multiple-Controller SDN with Snort showed slightly lower throughput due to the added intrusion detection processes, while the Multiple-Controller SDN without Snort performed comparably to the Single-Controller SDN. The results, shown in Fig. 4, indicate that the packet size did not significantly impact throughput values, maintaining stability across all scenarios.

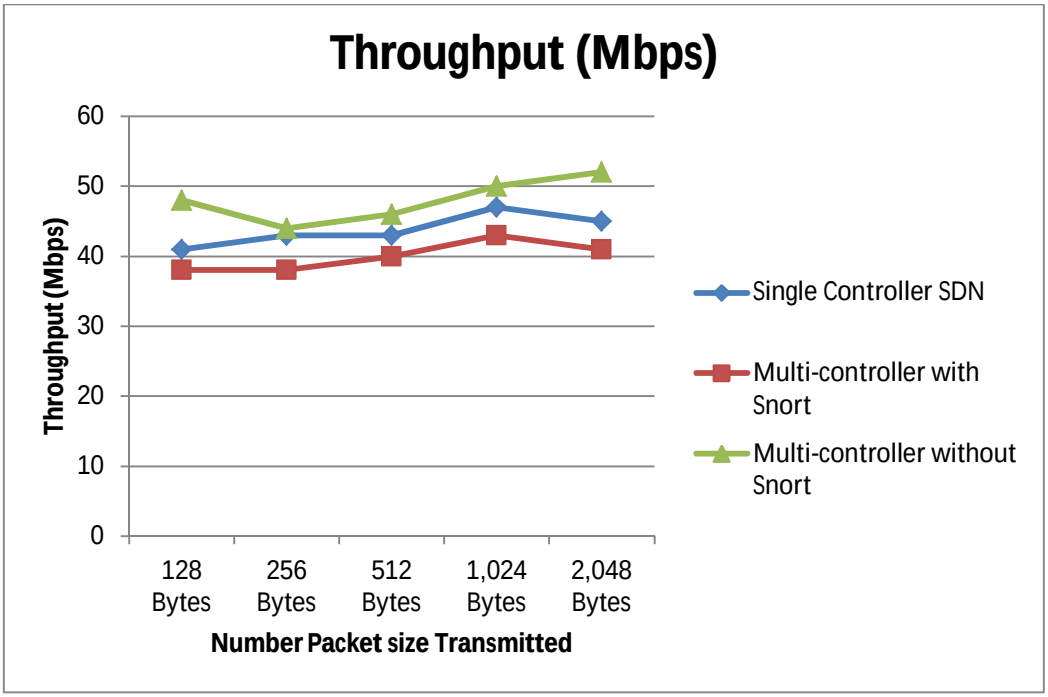


Fig. 4 Throughput distribution for various packet sizes

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue II Feb 2025- Available at www.ijraset.com

C. Packet Loss

Packet loss, measured using the ping utility, was minimal across all scenarios, signifying reliable data transmission. Single-Controller SDN experienced the least packet loss due to its centralized control, which optimized routing paths. The Multiple-Controller SDN with Snort had slightly higher packet loss percentages because of the intrusion detection overhead. However, the differences were negligible, indicating that Snort does not significantly compromise reliability. Multiple-Controller SDN without Snort maintained competitive packet loss rates, as detailed in Fig. 5.

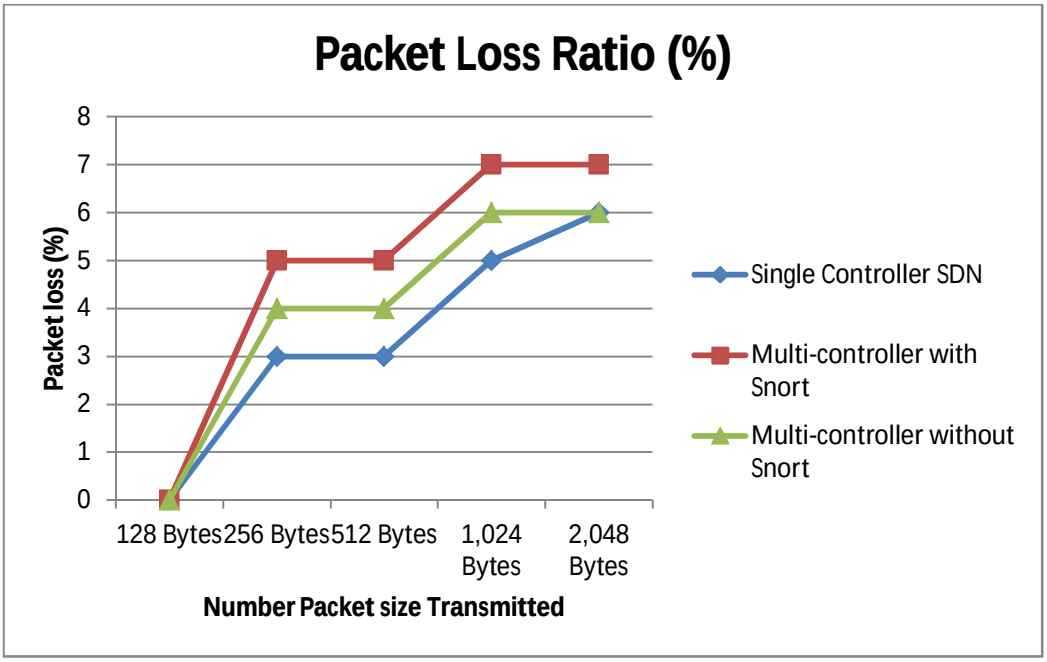


Fig. 5 Packet Loss Ratio distribution for various packet sizes

D. Bandwidth

Bandwidth utilization was measured to determine the maximum data transmission capacity in each scenario. Using Iperf, results showed that the Single-Controller SDN achieved the highest bandwidth due to its efficient data path optimization. Multiple-Controller SDN with Snort exhibited slightly reduced bandwidth, reflecting the processing overhead of Snort. However, Multiple-Controller SDN without Snort achieved bandwidth values comparable to the Single-Controller SDN, suggesting that distributed control models can handle large data transmissions effectively. Fig. 6 presents the bandwidth measurements for varying packet sizes.

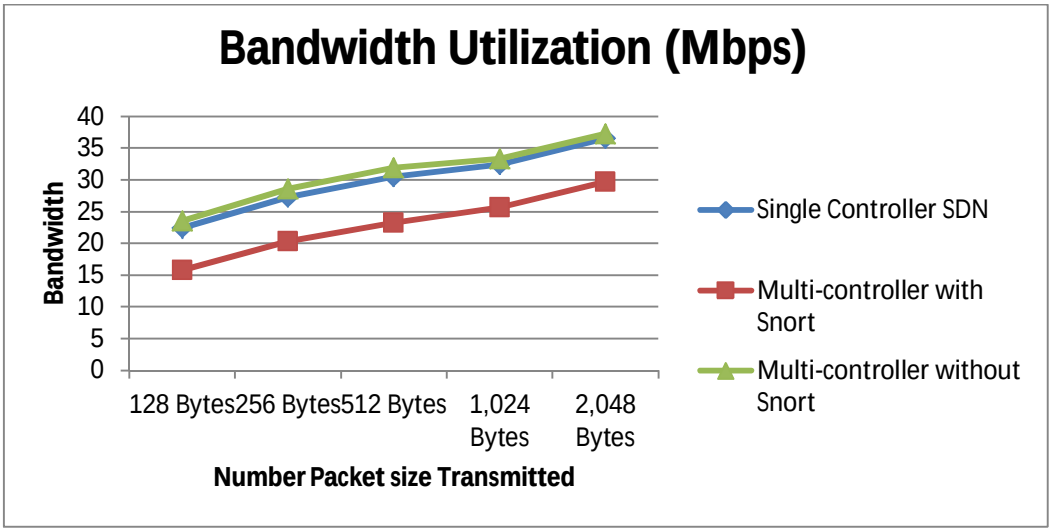


Fig. 6 Bandwidth Utilization

E. CPU Usage

CPU usage was evaluated to assess the computational demands of each scenario. A standard Linux virtualization platform for network monitoring and management is used to check the percentage CPU usage. Results revealed that Multiple-Controller SDN scenarios, especially with Snort, had higher CPU usage due to the additional processing and resource demands of managing multiple controllers and intrusion detection. The Single-Controller SDN and Multiple-Controller SDN without Snort maintained lower CPU usage, indicating efficient resource utilization. These findings suggest that scenarios prioritizing resource efficiency may benefit from simpler architectures without intrusion detection. Fig. 7 illustrates CPU usage percentages across different packet sizes.

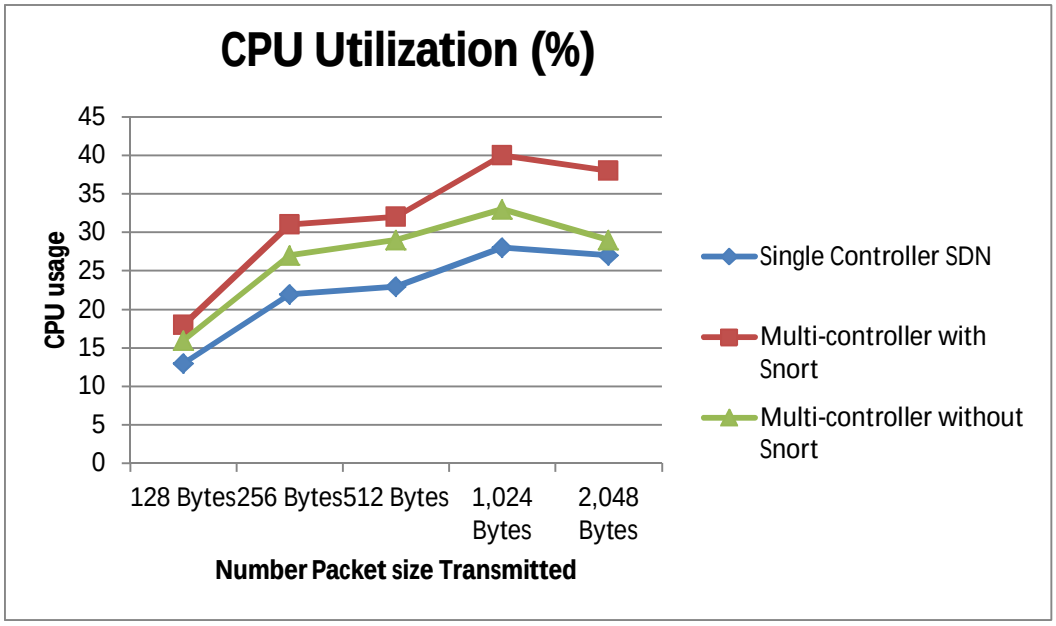


Fig. 7 percentages CPU usage

F. Discussion

The experimental findings highlight the trade-offs between security, efficiency, and scalability in SDN architectures. The Single-Controller SDN consistently excelled in latency, throughput, and CPU usage, making it suitable for applications prioritizing speed and resource efficiency. However, it lacked redundancy, making it susceptible to single points of failure. The Multiple-Controller SDN with Snort provided enhanced security but introduced performance trade-offs in terms of latency, throughput, and CPU usage. This makes it a robust choice for scenarios requiring intrusion detection and redundancy. The Multiple-Controller SDN without Snort balanced efficiency and scalability, performing comparably to the Single-Controller SDN in most metrics while addressing redundancy concerns. This configuration is ideal for scenarios requiring scalability without stringent security measures.

V. CONCLUSIONS

This research has demonstrated the potential of enhancing intrusion detection systems within Software-Defined Networking (SDN) environments through the integration of advanced technologies SNORT Intrusion Detection Systems (IDS), and network simulation tools like Mininet. The proposed model successfully combines multiple classification algorithms, feature selection methods, and SNORT IDS to develop an efficient and secure system for detecting network intrusions in SDN environments. Through comprehensive evaluation using key quality of service parameters such as bandwidth, throughput, latency, packet loss, and CPU usage—our findings confirm that the proposed SDN-based model outperforms traditional systems in terms of security and effectiveness [28]. The incorporation of SNORT IDS enhanced the model's ability to detect intrusions, providing a robust solution for SDN intrusion detection. The use of Mininet for network simulation, coupled with a virtual environment setup, further reinforced the research's practical applicability, ensuring that the model can be implemented within real-world SDN environments [11]. This research highlights the growing importance of SDN in Cybersecurity and sets the stage for future advancements. As SDN continues to evolve, it holds significant promise for transforming the way network traffic is managed and secured, making it a pivotal technology in the fight against cyber threats [31].

VI. FUTURE WORK

While this study has made significant strides in enhancing SDN-based intrusion detection, there are several key areas that warrant further exploration to further advance the field of network security:

1) Future work can focus on the real-time application of Machine Learning (ML) models within SDN controllers. This could enable dynamic reconfiguration of network traffic in response to detected threats. By automatically adjusting network policies, the system could effectively mitigate attacks, isolating the threat and minimizing its impact on the network [6].

2) The scale of data generated in SDN environments presents opportunities for leveraging big data analytics and anomaly detection techniques. Further research could explore how the volume of network data, including packet flows, payloads, and user behavior, can be analyzed to detect more subtle and advanced attack patterns such as zero-day vulnerabilities and APTs [20].

3) To maintain an up-to-date defense against evolving cyber threats, future work could involve the integration of threat intelligence feeds into SDN-based intrusion detection systems. This integration would allow for real-time threat information, enabling the system to adapt and respond to emerging threats more effectively. Establishing standardized protocols for threat intelligence sharing within SDN environments is an area for further development [8]

4) As SDN networks grow in complexity and size, ensuring that intrusion detection systems can scale accordingly becomes a critical challenge. Research focused on optimizing the scalability and performance of intrusion detection systems for large-scale SDN deployments is necessary to ensure their continued effectiveness as network environments evolve [23].

REFERENCES

[1] Ahmad A. A., Prof. Souley B., Abdullahi M. B., Muhammad A. M. amd Sa'adatu Gi. (2021). A Review on Software Defined Network (SDN) Based Network Security Enhancements. Quest Journals Journal of Software Engineering and Simulation.Volume 7, Issue 9 (2021) pp: 01-08 ISSN(Online) :2321-3795 ISSN (Print):2321-3809.

[2] Ahmed, F. A., Fatty, M. S., Ashraf, T. and Mohamed, H A. (2020). Performance Analysis and Evaluation of Software Defined Networking Controllers against Denial of Service Attacks. Journal of Physics: Conference Series.Conf. Ser. 1447 012007.

[3] Alaa T. A. (2022). Performance Evaluation of Ryu Controller in Software Defined Networks Journal of Al-Qadisiyah for Computer Science and Mathematics Vol. 14(1) 2022, pp Comp. 1–7.

[4] Barach, Jay. (2025). Towards Zero Trust Security in SDN: A Multi-Layered Defense Strategy. ICDCN '25: Proceedings of the 26th International Conference on Distributed Computing and Networking. P. 331 – 339. https://doi.org/10.1145/3700838.37036.

[5] Bhushan, B., & Jain, A. (2020).A review of machine learning methodologies for network intrusion detection systems in SDN.Journal of Network and Computer Applications, 174, 102856.https://doi.org/10.1016/j.jnca.2020.102856

[6] Brian, R. G., Bob, A. and Angelos, K. M. (2015). SDN-PANDA: Software-Defined Network platform for ANomaly Detection Applications. PhD Forum of ICNP.

[7] Brown, M., & Zhang, L. (2022). Real-time network security management with SDN: A survey of techniques and applications. Journal of Network Security, 45(3), 112-130.

[8] Chai M. C. Y. and Eswaran, S. (2025). A Survey on 5G Wireless Network Intrusion Detection Systems Using Machine Learning Techniques. Digital Forensics in the Age of AI. P: 24. DOI: 10.4018/979-8-3373-0857-9.ch008.

[9] Harris, A., & Gupta, R. (2021).Integrating threat intelligence into SDN-based intrusion detection systems.Cybersecurity Research, 39(2), 78-95.

[10] Hammad M., Nabil H. and Wael E. (2023). Enhancing Network Intrusion Recovery in SDN with machine learning: an innovative approach. Arab Journal of Basic and Applied Sciences. 30:1, 561-572, DOI: 10.1080/25765299.2023.2261219.

[11] Husham, B. A. A. (2017). Detection of DDoS Attacks against the SDN Controller using Statistical Approaches.(Master's Theses, Wright State University, 2017).Theses and dissertations CORE Scholar.

[12] Johnson, S., & Lee, K. (2021).Using Mininet for SDN-based intrusion detection modeling and testing.International Journal of Computer Networks, 37(1), 34-50.

[13] Jafarian , T., Ghaffari A., Seyfollahi A., Arasteh B. (2025). Detecting and mitigating security anomalies in Software-Defined Networking (SDN) using Gradient-Boosted Trees and Floodlight Controller characteristics. Computer Standards & Interfaces. Volume 91, 103871. https://doi.org/10.1016/j.csi.2024.103871.

[14] Jamal S. R., Ahmed M. A. and Abdul S. M. K. (2023). Performance evaluation of software-defined networking controllers in wired and wireless networks. Telecommunication Computing Electronics and Control. Vol. 21, No. 1, pp. 49~59 ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v21i1.23468.

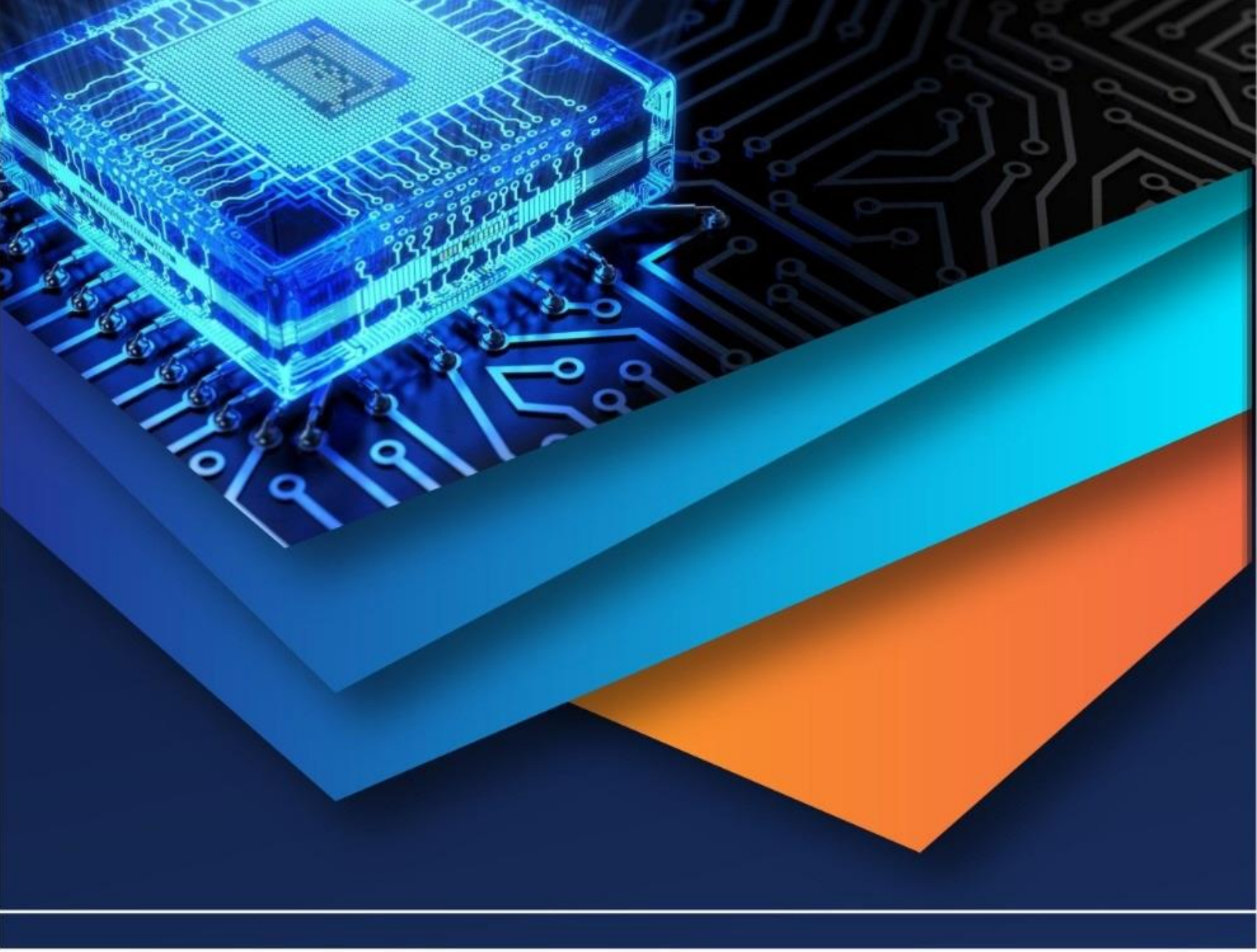
[15] Jayamagarajothi, M. and Murugeswari, P. (2015).A Survey on Data Mining and Digital Forensics Techniques for Intrusion Detection and Protection System.International Journal of Advanced Research in Computer and Communication Engineering. 4, 1, 159-164.

[16] Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., &Uhlig, S. (2015). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14-76. https://doi.org/10.1109/JPROC.2014.2371999

[17] Kulkarni M., Goswami B., Paulose J. and Malakalapalli, L. (2025). Unlocking the Power of Software-Defined Networking (SDN) in Revolutionizing Network Management. Advanced Cyber Security Techniques for Data, Blockchain, IoT, and Network Protection. P: 28. DOI: 10.4018/979-8-3693-9225-6.ch012.

[18] Kumar, S. D., Raihan, U. and Mahbubur, R. (2020).Performance Analysis of SDN-Based Intrusion Detection Model with Feature Selection Approach.International Joint Conference on Computational Intelligence, Algorithms for Intelligent.pp.483 494.

[19] Mehani, O., Kaafar, M. A., &Boreli, R. (2013). The privacy impact of network middleboxes: An SDN perspective. Security and Privacy in Communication Networks.https://doi.org/10.1007/978-3-319-04283-1_16
[20] Maddu, M. and Narasimha Y. R. (2024). Network intrusion detection and mitigation in SDN using deep learning models. International Journal of Information Security. V. 23, p. 849–862.
[21] Miller, T., & Wang, H. (2020).Big data analytics for anomaly detection in SDN environments.Journal of Big Data, 15(4), 202-220.
[22] Nalavade, K.C. and Meshram, B.B. (2011).Comparative Study of IDS and IPS.BIOINFO Computer Engineering. 1, 1, 01-04.
[23] Nunes, X., Ding, J., & Li, Y. (2020).Hybrid intrusion detection system for SDN.Journal of Information Security and Applications, 54, 102523.https://doi.org/10.1016/j.jisa.2020.102523
[24] Patel, A., Patel, S., & Sharma, R. (2019).Optimizing the scalability of SDN-based intrusion detection systems.Journal of Network Architecture, 30(6), 205-219.
[25] Porras, P., Shin, S., Yegneswaran, V., Fong, M., & Tyson, M. (2012). A security enforcement kernel for OpenFlow networks.Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN), 121-126. https://doi.org/10.1145/2342441.2342467
[26] Roesch, M., & Green, C. (2015).Snort: The open-source network intrusion detection system. Sourcefire, Inc.
[27] Said, M. E., Nhien-An, L. and Anca, J. (2020). InSDN: A Novel SDN Intrusion Dataset. DOI10.1109/ACCESS.2020.3022633, IEEE Acces
[28] Scott-Hayward, S., Natarajan, S., and Sezer, S. (2016). A Survey of Security in Software Defined Networks. IEEE Communications Surveys and Tutorials, 18(1), 623-654. https://doi.org/10.1109/COMST.2015.2453114
[29] Smith, J., Turner, C., & Chen, S. (2020). Enhancing network security with SDN-based intrusion detection: A comprehensive evaluation.Networking Journal, 21(8), 113-130.
[30] Songma, S., Sathuphan, T. and Pamutha T. (2023). Optimizing Intrusion Detection Systems in Three Phases on the CSE-CIC-IDS-2018 Dataset. Computers 2023, 12, 245. https:// doi.org/10.3390/computers12120245.
[31] Susan N. S., Muthalagu R. & Mothabhau P. P. (2024). SD-IIDS: intelligent intrusion detection system for software-defined networks. Multimedia Tools and Applications. V. 83, p. 11077–11109.
[32] Thompson, M., & Zhao, Y. (2019).Software-Defined Networking and the future of cybersecurity.Computer Networks, 118, 99-115.
[33] Vinutha, H.P. and Poornima, B. (2015).A Survey - Comparative Study on Intrusion Detection System, International Journal of Advanced Research in Computer and Communication Engineering.4, 7.410-414.
[34] Wang, Y., Zhang, Y., Singh, V., Lumezanu, C., and Jiang, G. (2013) "NetFuse: Short-circuiting traffic surges in the cloud,"IEEE International Conference on Communications (ICC). IEEE, pp. 3514–3518.
[35] Xia, W., Wen, Y., Foh, C. H., Niyato, D., &Xie, H. (2016).A survey on software-defined networking.IEEE Communications Surveys & Tutorials, 17(1), 27-51. https://doi.org/10.1109/COMST.2014.2330903
[36] Xing, T., Xiong, Z., Huang, D., and Medhi, D. (2014). "SDNIPS: Enabling Software-Defined Networking Based Intrusion Prevention System in Clouds," pp. 308–311.
[37] Zaalouk, A., Khondoker, R., Marx, R., and Bayarou, K. (2014). "OrchSec: An orchestrator-based architecture for enhancing network-security using Network Monitoring and SDN Control functions," Network Operations and Management Symposium (NOMS), IEEE. IEEE, 2014, pp. 1–9.
[38] Zawad, M. M., Rahman S. A., Islam S., Monirujjaman M. K. (2024). SDN Intrusion Detection Using Machine Learning Method. Computer Science - Cryptography and Security. https://doi.org/10.48550/arXiv.2411.05888.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY