International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 4752-4759
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Securing Software-Defined Network Topology Discovery: A Comprehensive Review of Attack Detection

Jayesh Chaudhary[1]*, Jaydeep Barad[2], Bhavesh Patel[3]

[1]Sarvajanik College of Engineering and Technology, Computer Engineering Department
* **Corresponding Author Email:** jayesh.chaudhary@scet.ac.in - **ORCID:** 0000-0002-6657-773X

[2]Sarvajanik College of Engineering and Technology, Computer Engineering Department
Email: jaydeep.barad@scet.ac.in - **ORCID:** 0000-0002-6180-936X

[3]Sarvajanik College of Engineering and Technology, Computer Engineering Department
Email: bhavesh.patel@scet.ac.in - **ORCID:** 0000-0002-6540-1724

Article Info:

DOI: 10.22399/ijcesen.3194
Received : 20 April 2025
Accepted : 26 June 2025

Keywords

Deep Learning
Machine Learning
Software-Defined Network
Topology Discovery

Abstract:

Software-Defined Network (SDN) has altered interconnected system operation by dividing data layer and control layers, which allows flexible, efficient, and programmable network configurations. To protect the integrity of geospatial data, centralized control, and transformational architecture of SDN is required. SDN controller service topology discovery is important for network services which can be susceptible to malicious activities. This Paper thoroughly examines the current attack detection methods during the topology discovery process. It reviews several topology discovery threats consisting of host location hijacking, topology poisoning, and link spoofing attacks. This paper summarizes valuable scopes, challenges, and future research scope, which can be a strong foundation for the development of a strong and resilient detection system to secure SDN networks against attacks done during topology discovery process.

1. Introduction

Software-Defined Networks (SDNs) have become a transformational architecture of advanced networking by providing extreme levels of flexibility and manageability by concentrating network control. SDN has transformed network administration by dividing the data and control planes allowing for more flexible, efficient, and programmable network configurations. The main aim of "Topology Discovery Attack Detection in Software-Defined Networks" is to upgrade the security of SDN network and dependability by identifying and mitigating attacks that change network topology information. These attacks can lead to unauthorized network access, resulting in disruption of service and data integrity.Machine Learning (ML) algorithms can identify patterns and anomalies in datasets so it is considered a successful tool for improving security of SDN. This study investigates topology discovery attacks for better understanding of ML and SDN security. Volatile nature of network traffic and difficulty of SDN architecture requires smart and extensible security mechanisms. This paper focuses on how ML algorithms can be used to enhance the resilience of SDN against topology discovery attacks by properly reviewing current developments and case studies.

2. Overview of SDN Architecture

SDN is a transformational architecture for network management. Network devices make separate decisions for data forwarding in case of traditional networking while SDN uses centralized control using software-based controllers. Controller informs network devices regarding forwarding of the data traffic allowing dynamic and programmable network behavior. The software-defined networking architecture depicted in Figure 1 consists of three main parts: Application plane, Control plane, and Data plane or infrastructure plane.

2.1 Data Plane

Data plane consists of the switches and routers. SDN Controller provides the instructions which are

followed by the switches and routers. Switches and routers are used for packet forwarding.

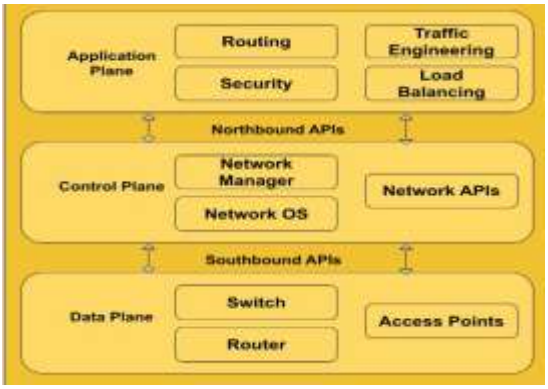

Figure 1. *SDN Architecture*

2.2 Control Plane

Control plane is the heart of the network referred to as the controller. This layer contains the SDN controller and network services. Controller makes decisions based on input received from applications. Southbound API is used to communicate between data plane and control plane.

2.3 Application Plane

SDN applications are programs that reside inside the application plane and interact with one another in an effective manner via a northbound interface. It hosts network applications like firewalls, load balancers, traffic engineering, and security.

2.4 Southbound API

The OpenFlow Protocol uses APIs, also called "southbound interfaces" to link the data plane with the control plane. Switches running OpenFlow can communicate with one or more servers. These switches can control and reroute traffic by using the flow entries from the OpenFlow controller.

2.4 Northbound API

The Application and control layer are connected through the Northbound API. For systems and applications based on the SDN architecture, it offers a network abstraction layer and facilitates communication between the control plane and the application.

3. Topology Discovery

Topology discovery is the process of mapping and understanding the structure of the network. It explains interaction between switches and how hosts connect with switches. This process involves the controller gaining information about network devices such as switches, links between switches and location of the host within the network.

In SDN Networks, topology discovery takes a centralized approach due to the programmable and unified control provided by the SDN controller. The controller gathers information about the network's topology by communicating with network devices. It maintains the global view of the network. Switch discovery, host discovery and link discovery are three methods used by SDN to discover network topology.

3.1 Switch Discovery

As a part of the switch discovery process, a handshake-based session is created between controller and network devices after switches are turned on. Features request/reply messages are exchanged between controller and switch to get features such as configuration information, active interfaces, corresponding mac addresses.

3.2 Host Discovery

Host discovery will notify the controller via packet-in messages if table fails. Consequently, the switch transmits the first packet to the controller as a packet-in message, when a host sends traffic to it and no flow rules match the incoming flow. In terms of network administration tasks, before any traffic is generated, the controller can effectively utilize host identification.

3.3 Link Discovery

The link discovery mechanism keeps track of connections between forwarding devices and is based on the OpenFlow Discovery Protocol (OFDP). These switch-to-switch connections are found by the SDN controller sending Link Layer Discovery Protocol (LLDP) advertisements to all active switch ports on a regular basis. The OFDP particularly uses LLDP packets for this.As shown in Figure 2 SDN controller sends packet-out OpenFlow message to switches on every active port which contains LLDP packet. Switch forwards connection metadata in the data plane. In figure, s1 forwards the LLDP packet to s2(p3). S2 adds its own metadata and encapsulates LLDP packets and sends a packet-in message to the controller. Controller receives a packet-in message, processes and creates a link between s1(p1) to s2(p3). This process is repeated for every link. The initial methods are inefficient in quickly identifying a change in network topology and place a burden on the SDN controller's resources especially in large

networks, due to its significant performance limitations.

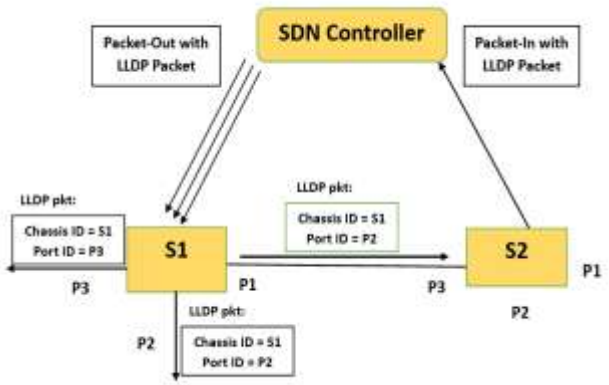

Figure 2. *Topology Discovery in SDN*

4. Attacks on Topology Discovery

Topology discovery process is vulnerable to various topology poisoning attacks which can mislead the network information. Due to these services like traffic engineering, routing, load balancing can be affected which leads to degrading the performance of the SDN network.

4.1 Link Fabrication Attack

Creating or inserting false links by manipulating LLDP packets within the SDN controller's topology display can misguide the network. As a result, routing services might be disrupted, leading to a heightened risk of DoS attacks.

4.2 Relay Link Forgery Attack

The Attacker relays genuine LLDP messages between switches. The Controller is not able to detect as LLDP messages are legitimate.

4.3 Link Latency Attack

The integrity of the network topology can be compromised through the manipulation of reported link latency values, which are falsified to suggest a reliable and high-speed connection, all without the generation of fraudulent packets. This attack capitalizes on the SDN controller's inherent reliance on its global network perspective for routing determinations, neglecting to pre-validate the veracity of the transmitted connection latency data.

4.4 Multi-Hop Link Fabrication Attack

The attacker fabricates MHL packets using protocols such as LLDP and injects them into the network. Based on the faked MHL data, the SDN controller erroneously believes there is a direct link and routes traffic appropriately. Network congestion, denial of service attacks, and even traffic eavesdropping are all consequences of improper topology.

4.5 Host Location Hijacking

Attacker intentionally misleads the SDN controller about the actual network location of a legitimate host. Due to this confidentiality of data can be compromised. Legitimate host will become unreachable.

5. Contemporary studies in SDN Topology Discovery

OpenFlow Discovery Protocol (OFDP), utilizing Link Layer Discovery Protocol (LLDP) packets, is central to the creation of the SDN topology. SDN controllers are responsible for defining the network topology via the distributed OFDP protocol, wherein they transmit an LLDP packet to every active switch port during the OFDP discovery cycle. Shrivastava and Kataoka [1] describe the multi-hop link fabrication attack in hybrid SDNs and proposed "HybridShield" as a mitigation framework. Authors focus on link verification mechanisms, demonstrating immediate detection and high accuracy to avoiding attacks. As a future work there is a need for control and data plane security in hybrid SDN architectures.

Chuang et al. (2022) [2] presents the dynamic threat environment in SDNs, using ML and deep learning (DL) models for early outlier detection. To improve the performance of the multi-class classification, the author proposed a hierarchical multiclass classification architecture which leads to strong attack detection. As a part of future work the author suggests incorporating multiple controllers to survive against a single point of failure.Smyth et al. (2023) [3] introduce the SECAP switch, a security solution implemented within the network using P4, aimed at countering topology poisoning attacks. This streamlined method specifically targets ARP cache poisoning and relay-style Link Fabrication Attacks, showing successful performance with low memory and processing demands. Future research is proposed to investigate different statistical methods for enhanced detection.Soltani et al. (2021) [4] study finds vulnerabilities in SDN controllers, with a focus on host-location hijacking and other topology poisoning techniques like Link Fabrication techniques (LFA). These assaults take advantage of flaws in network discovery protocols, such as OpenFlow Discovery Protocol (OFDP), which lets attackers add fictitious links or change the

controller's perception of the network topology. The article introduces LLA, to build a fake link between two switches in the network without directly compromising the SDN-enabled switches. Link latency is increased as the SDN controller presumes that the fake link is genuine by the attack.Zhang and Wang (2023) [5] introduced a new Transformer-based model, the Relay Link Forged Attack (RLFA) detection system. This model extracts features from network traffic to identify fake relay links in SDN environments, which are created by attackers relaying genuine LLDP packets to disrupt the SDN controller's global view and network operations.Joshep et al. (2022) [6] introduces a Lightweight Forged Attack Mitigation Algorithm (LiFAMA) that is used to protect SDN controllers against malicious nodes that try to modify the global view of the network topology by incorporating malicious LLDP packets. LiFAMA merges packet verification with delay-based analysis that consumes less resources that is useful to differentiate between legitimate and malicious LLDP packets, before updating the topology, it authenticates the LLDP packets which includes HMAC for uniqueness. The methodology is designed such that computational resources are minimized but the accuracy for attack detection is improved. LiFAMA guarantees security of the SDN topology by effectively identifying and mitigating forged attacks. It achieves a low false-positive rate during topology discovery attack detection.Wazirali et al. (2021) [7] throws light on issues associated with SDN-OpenFlow topology discovery. Authors highlight the issues of the OpenFlow Discovery Protocol (OFDP) in pervasive and large-scale networks. Study recommends for novel topology discovery methodology that spreads logic between data plane devices and control plane devices to minimize the overhead and learning time. Considering the significance of complex network environments, the author suggests periodic topology discovery.Huang et al. (2020) [8] proposes a TrustTopo which is a lightweight verification scheme for authentication based on reviewing topology poisoning attacks during SDN topology discovery. This study aims to improve network performance with attack prevention. As a part of future work, the author determines security policies during the early stage of network startup and incorporates TrustTopo with various SDN controllers for real-world network applications.Li-Der Der et al. (2020) [9], In this paper analysis of Spearman's rank correlation for topology network traffic is used to measure the latency between links that is used to detect malicious LLDP packets. Author had implemented it for anomaly detection. Authentication key is encapsulated in LLDP packets and counts for transmission of LLDP packets to switches are observed. Addition of authentication key with timestamp gives the protection against flooding or injection attacks and increases the security for the control plane.Baidya and Hewett (2020) [10] explores attacks and vulnerabilities that occur during the process of link discovery for the SDN environment. In this paper host-based and switch-based attacks are observed and to prevent against attacks an active-ports mechanism is used. Analytical impact of attack impact is measured with the routing application. Author suggests a future to do experiments with multiple controllers and diversified network scenarios.Ochoa-Aday et al. (2019) [11], In this paper implementation of proposed algorithm eTDP is done in switches with the consideration of multiple controllers. The protocol is designed in such a way that previous IP configuration is not required to distribute discovery functions to the data plane devices which facilitates the automatic link discovery. By conducting experiments with a simulation environment, the author achieves high efficiency to discover topology with minimum overhead considering the scalability of the network.Nehra et al. (2019) [12], presented a novel approach to topology discovery in SDN networks, proposing a Secure and Lightweight Link Discovery Protocol (SLDP) designed to ensure the integrity and correctness of retrieved topological data. SLDP aims to identify, detect and mitigate security risks in the process of topology discovery. The algorithm is designed to attempt to efficiently secure with less resource consumption. Creation of custom packet format of fixed length with uniqueness minimizes packet size and additional information. Due to minimization of packet size bandwidth consumptions and processing overhead are reduced. Link discovery process done only to eligible ports. After the initial iteration, LLDP packet transmission for link discovery is reduced. To evaluate the performance of SLDP, it is implemented using The Mininet emulator considering the parameters bandwidth overhead, CPU usage and time for topology discovery. It reduces topology discovery time, with less bandwidth and CPU utilization. Bui et al. (2019) [13], In this paper topology poisoning attacks are analysed in the SDN environment. Attacks are classified with impact analysis for the network topology, location of the attacker and policy for routing. Highlighting the importance of malicious switches, with the assumption of secure SDN controller and control channels, focusing on diverting traffic passes through a small number of malicious data plane devices.Xiang et al. (2020) [14], In this paper, to model and validate mechanism of topology discovery for OpenFlow controllers in SDN environment process algebra is used. Author

proposed a novel framework which captures the network traffic in the SDN environment and recognizes Link Fabrication Attack and Host Hijacking Attack which are part of the topology poisoning attacks. TopoGuard is proposed for verification of the network traffic for possible loopholes, highlighting the need for a defence mechanism for security in the SDN environment. Deep learning has been studied and reported in the literature [27-35].

Table 1. *Comparative Analysis of Topology Discovery Attack Detection*

Sr.No.	Paper Title	Method Used	Summary	Limitations	Evaluation Matrices
1	Topology Poisoning Attacks and Prevention in Hybrid Software-Defined Networks [1].	Monitoring traffic & legacy switch verification	Low-overhead detection of multi-hop link fabrication attacks in Hybrid SDN.	Limited to hybrid SDN scenario	Detection Rate, False Positive Rate, Overhead
2	RLFAT: A Transformer-Based Relay Link Forged Attack Detection Mechanism in SDN [5]	Transformer-based deep learning model	High Detection rate of Relay Link Forgery Attack.	-	Accuracy, Precision, Recall F1 score AUC
3	A Link Fabrication Attack Mitigation Approach (LiFAMA) for Software Defined Networks [6]	HMAC-based authentication	Link Fabrication Attack (LFA) prevention using secure verification of LLDP.	Shared key management needs to be secure, Computational Overhead	Topology Discovery time, Link Verification Time, CPU Utilization
4	SLDP: A secure and lightweight link discovery protocol for software defined networking [12]	Custom Packet Format, MAC based Authentication, Eligible Port List	It uses a 26-byte fixed-format packet and Uses Random MAC address and Token for Each discovery cycle.	Initial Exposure Risk, MAC Gussing attacks possible	Topology Discovery Time, Packet Size Overhead CPU Usage Detection Accuracy, Mitigation Time
5	TILAK: A token-based prevention approach for topology discovery threats in SDN [15]	Dynamic MAC-based authentication,	It safeguards against LLDP poisoning, flooding, and replay attacks.	First-cycle Exposure, Static Timing Threshold	TP, FP, TN, FN Packet Construction and Verification Time, CPU Usage, Flooding Resistance
6	Combination Attacks and Defences on SDN Topology Discovery [16]	14-phase attack sequence called Invisible Assailant Attack (IAA) introduced. And Route Path Verification method used for defence.	IAA cleverly disguises malicious activity across 14 strategic phases and Route Path Verification (RPV) validates route path integrity.	Dependency on probing packets, Limited Data Plane Visibility, Springboard dependency	Detection Time, CPU Usage, False Positive Rate, Storage Overhead, Latency and Bandwidth matrices
7	Real-Time Link Verification in Software-Defined Networks [17]	Machine Learning model	Detect fabricated links based on LLDP	-	True Positive, False Positive, Precision, ROC, F1- Score, PR

			latency dynamics		Curves and Cohen's Kappa
8	ESLD: An efficient and secure link discovery scheme for software-defined networking [19]	Port Classification, Directional LLDP Transmission and time based HMAC used for authentication.	Eliminates all superfluous LLDP packets, achieving up to 25% CPU reduction on the controller. By time based HMAC packet reuse risk eliminated.	Higher controller load, Dependency on Port classification	No. of LLDP Packets, CPU Utilization, Attack Detection Rate, Scalability

6. Conclusion

Topology Discovery is an important functionality which enables the controller to establish a complete, global perspective of the network. Due to centralized control and programmable nature SDN becomes vulnerable to attacks happening during the process of topology discovery. Link fabrication, host location hijacking and relay link forgery attacks are discussed with their defence mechanism. These attacks can affect network performance and security can be compromised.

The extensive review of the process of topology discovery with possible vulnerabilities are identified with identification and mitigation of attacks performed by malicious nodes. Policy Based, Behavioral analysis, machine learning and deep learning-based methods are used for identifying and mitigating attacks with high accuracy and minimum overhead. In most of the cases simulation environments (Mininet) are used for the implementations, but lack the real-world deployment in SDN networks.Researchers can focus on providing lightweight cryptographic solutions for security in collaboration with deep learning-based anomaly detection.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Shrivastava, P., & Kataoka, K. (2021). Topology poisoning attacks and prevention in hybrid software-defined networks. *IEEE Transactions on Network and Service Management*, *19*(1), 510–523. https://doi.org/10.1109/TNSM.2021.3122491.

[2] Chuang, H. M., Liu, F., & Tsai, C. H. (2022). Early detection of abnormal attacks in software defined networking using machine learning approaches. *Symmetry*, *14*(6), 1178. https://doi.org/10.3390/sym14061178.

[3] Smyth, D., Scott-Hayward, S., Cionca, V., McSweeney, S., & O'Shea, D. (2023). SECAP switch—Defeating topology poisoning attacks using P4 data planes. *Journal of Network and Systems Management*,*31*(1),28. https://doi.org/10.1007/s10922-023-09663-5.

[4] Soltani, S., Shojafar, M., Mostafaei, H., Pooranian, Z., & Tafazolli, R. (2021, October). Link latency attack in software-defined networks. In *2021 17th International Conference on Network and Service Management (CNSM)* (pp. 187–193). IEEE. https://doi.org/10.23919/CNSM52442.2021.9615533.

[5] Zhang, T., & Wang, Y. (2023). RLFAT: A transformer-based relay link forged attack detection mechanism in SDN. *Electronics*, *12*(10), 2247. https://doi.org/10.3390/electronics12102247.

[6] Joseph, K., Eyobu, O. S., Kasyoka, P., & Oyana, T. J. (2022). A Link Fabrication Attack Mitigation Approach (LiFAMA) for Software Defined Networks. *Electronics*, *11*(10), 1581. https://doi.org/10.3390/electronics11101581.

[7] Wazirali, R., Ahmad, R., & Alhiyari, S. (2021). SDN-OpenFlow topology discovery: An overview of performance issues. *Applied Sciences*, *11*(15), 6999. https://doi.org/10.3390/app11156999.

[8] Huang, X., Shi, P., Liu, Y., & Xu, F. (2020). Towards trusted and efficient SDN topology discovery: A

lightweight topology verification scheme. *Computer Networks, 170*, 107119. https://doi.org/10.1016/j.comnet.2020.107119.

[9] Li-Der, C., Chien-Chang, L., Meng-Sheng, L., Kai-Cheng, C., Tu, H. H., Sen, S., & Tsai, W. H. (2020). Behavior anomaly detection in SDN control plane: A case study of topology discovery attacks. *Wireless Communications and Mobile Computing, 2020*, 1–13. https://doi.org/10.1155/2020/8898738.

[10] Baidya, S. S., & Hewett, R. (2020). Link discovery attacks in software-defined networks: Topology poisoning and impact analysis. *Journal of Communications, 15*(8), 596–606. https://doi.org/10.12720/jcm.15.8.596-606.

[11] Ochoa-Aday, L., Cervelló-Pastor, C., & Fernández-Fernández, A. (2019). eTDP: Enhanced topology discovery protocol for software-defined networks. *IEEE Access, 7*, 23471–23487. https://doi.org/10.1109/ACCESS.2019.2899653.

[12] Nehra, A., Tripathi, M., Gaur, M., Babu, B., & Lal, C. (2018). SLDP: A secure and lightweight link discovery protocol for software defined networking. *Computer Networks, 150*, 225–239. https://doi.org/10.1016/j.comnet.2018.12.014.

[13]Bui, T., Antikainen, M., & Aura, T. (2019). Analysis of topology poisoning attacks in software-defined networking. In *Secure IT Systems: 24th Nordic Conference, NordSec 2019, Aalborg, Denmark, November 18–20, 2019, Proceedings* (Vol. 11875, pp. 87–102). Springer. https://doi.org/10.1007/978-3-030-35055-0_6.

[14] Xiang, S., Zhu, H., Wu, X., Xiao, L., Bonsangue, M., Xie, W., & Zhang, L. (2020). Modeling and verifying the topology discovery mechanism of OpenFlow controllers in software-defined networks using process algebra. *Science of Computer Programming, 187*, 102343. https://doi.org/10.1016/j.scico.2019.102343

[15] Nehra, A., Tripathi, M., Gaur, M., Babu, B., & Lal, C. (2018). TILAK: A token-based prevention approach for topology discovery threats in SDN. *International Journal of Communication Systems, 32*(1), e3781. https://doi.org/10.1002/dac.3781

[16] Kong, D., Li, Q., Chen, J., Wang, J., Wang, Y., & Liu, Y. (2023). Combination attacks and defenses on SDN topology discovery. *IEEE/ACM Transactions on Networking, 31*(2), 904–919. https://doi.org/10.1109/TNET.2022.3203561

[17] Soltani, S., Shojafar, M., Mostafaei, H., & Tafazolli, R. (2023). Real-time link verification in software-defined networks. *IEEE Transactions on Network and Service Management, 20*(3), 3596–3611. https://doi.org/10.1109/TNSM.2023.3238691.

[18] Alharbi, T., Portmann, M., & Pakzad, F. (2015). The (in)security of topology discovery in software defined networks. In *2015 IEEE 40th Conference on Local Computer Networks (LCN)* (pp. 502–505). IEEE. https://doi.org/10.1109/LCN.2015.7366363.

[19] Zhao, X., Yao, L., & Wu, G. (2017). ESLD: An efficient and secure link discovery scheme for software-defined networking. *International Journal of Communication Systems, 31*(3), e3552. https://doi.org/10.1002/dac.3552.

[20] Ravi, N., Shalinie, S. M., & Jose Theres, D. D. (2020). BALANCE: Link flooding attack detection and mitigation via hybrid-SDN. *IEEE Transactions on Network and Service Management, 17*(3), 1715–1729. https://doi.org/10.1109/TNSM.2020.2997734.

[21] Dhawan, M., Poddar, R., Mahajan, K., & Mann, V. (2015). SPHINX: Detecting security attacks in software-defined networks. In *Proceedings of the 2015 Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2015.23064.

[22] Nguyen, T.-H., & Yoo, M. (2017). A hybrid prevention method for eavesdropping attack by link spoofing in software-defined Internet of Things controllers. *International Journal of Distributed Sensor Networks, 13*(1), 1–11. https://doi.org/10.1177/1550147716682731.

[23] Bereziński, P., Jasiul, B., & Szpyrka, M. (2015). An entropy-based network anomaly detection method. *Entropy, 17*(4), 2367–2408. https://doi.org/10.3390/e17042367.

[24] Varadharajan, V., Karmakar, K., Tupakula, U., & Hitchens, M. (2019). A policy-based security architecture for software-defined networks. *IEEE Transactions on Information Forensics and Security, 14*(4), 897–912. https://doi.org/10.1109/TIFS.2018.2868220.

[25] Ali, S. T., Sivaraman, V., Radford, A., & Jha, S. (2015). A survey of securing networks using software defined networking. *IEEE Transactions on Reliability, 64*(3), 1086–1097. https://doi.org/10.1109/TR.2015.2421391.

[26] Deng, S., Gao, X., Lu, Z., & Gao, X. (2018). Packet injection attack and its defense in software-defined networks. *IEEE Transactions on Information Forensics and Security, 13*(3), 695–705. https://doi.org/10.1109/TIFS.2017.2765506.

[27]G Nithya, R, P. K., V. Dineshbabu, P. Umamaheswari, & T, K. (2025). Exploring the Synergy Between Neuro-Inspired Algorithms and Quantum Computing in Machine Learning. *International Journal of Computational and Experimental Science and Engineering*, 11(3). https://doi.org/10.22399/ijcesen.2484

[28]Kumari, S. (2025). Machine Learning Applications in Cryptocurrency: Detection, Prediction, and Behavioral Analysis of Bitcoin Market and Scam Activities in the USA. *International Journal of Sustainable Science and Technology*, 1(1). https://doi.org/10.22399/ijsusat.8

[29]Sivananda Hanumanthu, & Gaddikoppula Anil Kumar. (2025). Deep Learning Models with Transfer Learning and Ensemble for Enhancing Cybersecurity in IoT Use Cases. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.1037

[30]Chui, K. T. (2025). Artificial Intelligence in Energy Sustainability: Predicting, Analyzing, and Optimizing Consumption Trends. *International Journal of Sustainable Science and Technology*, 1(1). https://doi.org/10.22399/ijsusat.1

[31]Johnsymol Joy, & Mercy Paul Selvan. (2025). An efficient hybrid Deep Learning-Machine Learning method for diagnosing neurodegenerative disorders. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.701

[32]Olola, T. M., & Olatunde, T. I. (2025). Artificial Intelligence in Financial and Supply Chain Optimization: Predictive Analytics for Business Growth and Market Stability in The USA. *International Journal of Applied Sciences and Radiation Research*, 2(1). https://doi.org/10.22399/ijasrar.18

[33]R. Sundar, M. Ganesan, M.A. Anju, M. Ishwarya Niranjana, & T. Surya. (2025). A Context-Aware Content Recommendation Engine for Personalized Learning using Hybrid Reinforcement Learning Technique. *International Journal of Computational and Experimental Science and Engineering*, 11(1). https://doi.org/10.22399/ijcesen.912

[34]A, N. N., G. Siva, K. Kasiniya, S. Uma, & T, K. (2025). Optimizing Hybrid AI Models with Reinforcement Learning for Complex Problem Solving. *International Journal of Computational and Experimental Science and Engineering*, 11(3). https://doi.org/10.22399/ijcesen.2483

[35]Ibeh, C. V., & Adegbola, A. (2025). AI and Machine Learning for Sustainable Energy: Predictive Modelling, Optimization and Socioeconomic Impact In The USA. *International Journal of Applied Sciences and Radiation Research*, 2(1). https://doi.org/10.22399/ijasrar.19