Grenze International Journal of Engineering and Technology, January Issue

Reinforcement Learning-based Adaptive Routing in Software Defined Networks under Security Threats

Benazir M[1], Vani Ganiger[2] and Varsha I Pattanshetty[3]
[1-2]Department of Computer Science and Engineering, SIET, Vijayapur, India
Email: benazirmuntasher, vani.ganiger777@gmail.com
[3]Department of Computer Science and Engineering, KLE Technological University, Hubballi, India
Email: 01fe22bcs191@kletech.ac.in

***Abstract*— Software Defined Networking (SDN) introduces a flexible and centralized approach to network management by decoupling the control plane from the data plane, thereby enabling enhanced programmability and efficient traffic handling. However, this architectural centralization also introduces critical security vulnerabilities, especially within the control plane, rendering SDN environments highly susceptible to attacks such as Denial-of-Service (DoS) and blackhole routing. Traditional static routing methods prove insufficient in mitigating such dynamically evolving threats. To address these limitations, this paper proposes a Reinforcement Learning (RL)-based adaptive routing framework designed to enable the SDN controller to autonomously learn optimal and secure routing policies through continuous interaction with the network environment. The proposed framework is implemented using the Mininet emulator and the Ryu SDN controller. It dynamically adjusts routing paths in real-time to avoid compromised network nodes and strengthen the network's security posture. Experimental evaluations conducted under various simulated attack scenarios demonstrate that the RL-based approach significantly outperforms conventional static routing techniques in terms of adaptability, network performance, and resilience. The results validate the effectiveness of our proposed framework in establishing a robust foundation for intelligent, adaptive, and security-aware SDN infrastructures.**

***Index Terms*— Software Defined Networking (SDN), Reinforcement Learning (RL), Network Security, Adaptive Routing, Security Threats, Mininet, Ryu Controller.**

I. INTRODUCTION

Software Defined Networking (SDN) has revolutionized modern network architecture by introducing a clear separation between the control plane and the data plane [1]. This architectural shift allows the control plane to be centralized and programmable, enabling more flexible and efficient network management. Through centralized control, network administrators can dynamically configure routing policies, manage traffic, and respond to changes in real time [2]. These capabilities make SDN highly suitable for large-scale and complex environments such as data centers, cloud infrastructures, and enterprise networks. However, while SDN's centralized nature simplifies many aspects of network control, it also introduces new vulnerabilities. The controller, being the central authority, becomes a critical point of attack, making the network susceptible to threats such as Denial-of-Service (DoS) attacks, blackhole routing, link flooding, and man-in-the-middle (MITM) attacks [3].

Grenze ID: 01.GIJET.12.1.527_2

Traditional routing mechanisms in SDN, such as shortest path or static routing, are insufficient in the face of these threats. These methods lack the adaptability required to detect and mitigate dynamic attacks or to reconfigure routes in real time based on evolving network conditions. Once a path is compromised, these static algorithms are unable to reroute traffic intelligently without external input, potentially leading to packet loss, increased latency, and reduced throughput [4]. To address these shortcomings, researchers have increasingly turned to artificial intelligence (AI) and machine learning (ML) techniques. Among them, Reinforcement Learning (RL) stands out as a particularly effective approach due to its ability to learn optimal policies through interaction with the environment [5]. RL agents operate in an online learning framework where they make decisions, receive feedback in the form of rewards or penalties, and continuously improve their performance over time. This makes RL especially suitable for dynamic, adversarial, and unpredictable environments like SDNs under security threats.

In this work, we propose an RL-based adaptive routing framework tailored for Software Defined Networks under security threat scenarios. The proposed system leverages a Q-learning agent embedded in the SDN controller, which learns to select secure and efficient routing paths while avoiding compromised nodes. We simulate this framework using Mininet and implement the logic via the Ryu controller, evaluating performance under both normal conditions and attack scenarios such as DoS and blackhole attacks. Experimental results on a 20-node network topology show that under normal conditions, the RL-based routing performs comparably to shortest path routing, with a Packet Delivery Ratio (PDR) of 95.8% compared to 96.2%, and a slightly higher end-to-end delay due to the learning overhead. However, under attack conditions, the RL-based approach significantly outperforms traditional methods, achieving a PDR of 88.4% versus 63.7%, along with reduced delay and increased throughput. These results confirm the efficacy of reinforcement learning in creating secure, adaptable, and intelligent SDN routing mechanisms capable of mitigating the impact of real-time security threats.

II. Background

OVER the past decade, Software Defined Networking (SDN) has attracted widespread attention due to its ability to improve network programmability, scalability, and manageability [1]. Conventional routing mechanisms in SDNs—such as shortest-path routing, static policies, or flow-table configurations—offer simplicity and efficiency under benign conditions. However, these solutions fall short in environments experiencing dynamic security threats, including Denial-of-Service (DoS), blackhole, or rerouting attacks [6], [7]. These attacks exploit vulnerabilities in the SDN control plane, leading to degraded network performance or service unavailability. Traditional routing approaches, while computationally efficient, do not consider security as a metric in decision-making, which can lead to compromised data paths or network outages during attack scenarios [3], [4]. The role of Intrusion Detection Systems (IDS) in SDNs has become increasingly critical. Various machine learning-based IDS frameworks have been proposed to identify malicious behavior and mitigate attacks [8], [9]. These frameworks rely on anomaly detection or classification models to flag suspicious flows based on traffic statistics, flow duration, and entropy-based metrics. However, most IDS systems operate independently of the routing logic, functioning in a reactive and passive manner. Furthermore, their reliance on predefined patterns or attack signatures limits their ability to adapt to zero-day threats or sophisticated adversarial strategies [10], [11]. Integrating IDS decisions with routing policies remains a challenge due to the delay and potential false positives or negatives inherent in detection algorithms.

To overcome these limitations, Reinforcement Learning (RL) has emerged as a promising paradigm for developing intelligent routing mechanisms. RL models, including Q-learning, Deep Q-Networks (DQN), and Proximal Policy Optimization (PPO), have demonstrated success in optimizing network parameters such as load balancing, delay minimization, energy efficiency, and bandwidth utilization [5], [12]. These models empower SDN controllers to learn from network conditions and dynamically adjust routing strategies without requiring an explicit model of the environment. Despite this progress, most RL-based methods prioritize performance metrics rather than incorporating security considerations into the routing policy. Moreover, training RL models in large and complex networks remains computationally intensive.

A small but growing body of literature addresses security-aware RL in SDNs. Some studies integrate RL with blacklisting methods or trust-based metrics to avoid compromised nodes [13], [14]. However, these methods often assume static attacker behavior or lack adaptability to evolving threats. Moreover, the majority of research is simulation-focused and does not translate well into realistic, scalable SDN environments. A few notable efforts have implemented RL-based security frameworks in real SDN platforms like Mininet and Ryu, enabling

real-time learning and policy updates [15], [16]. Yet, these implementations frequently oversimplify the attack models or assume perfect knowledge of network topology and threat presence, limiting practical deployment.
Our work builds upon these foundations by proposing a reinforcement learning-based adaptive routing framework that explicitly incorporates security threat awareness into its reward structure. Unlike conventional RL approaches, our method evaluates not only performance metrics but also dynamically adapts routing decisions based on observed attack conditions. Furthermore, we implement and evaluate our approach on a realistic SDN testbed, simulating common threats such as DoS and blackhole attacks to validate its effectiveness in real-world conditions.

III. Methodology

IN this section, we detail the methodology used to develop and evaluate the RL-based adaptive routing framework for SDN under dynamic security threats. The approach involves several key components: network simulation, threat modeling, RL environment setup, algorithm design, real-time deployment, and evaluation metrics. Each of these elements is crucial for assessing the effectiveness of the proposed system in realistic SDN environments, particularly under the impact of malicious attacks such as DoS and blackhole routing.

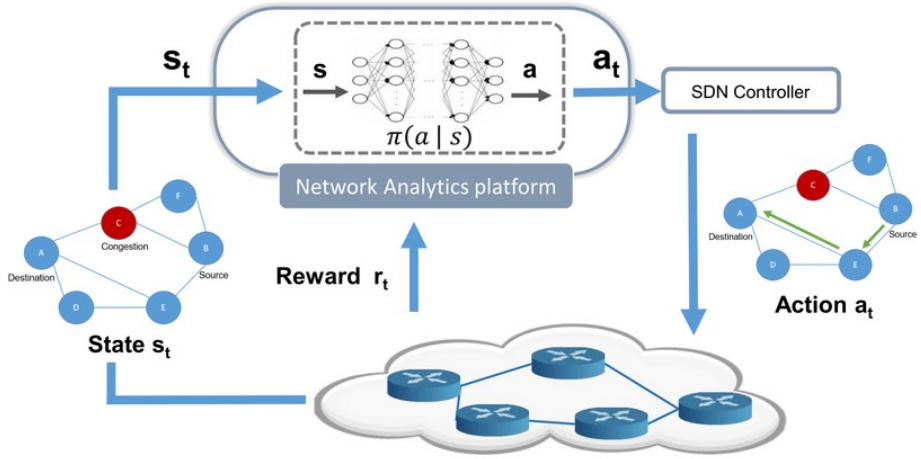

Figure 1: Architecture of RL-based Adaptive Routing in SDN under Security Threats

A. Network Simulation and Topology

To simulate the SDN environment, we utilize Mininet, a network emulator that provides a virtual network consisting of multiple hosts, OpenFlow switches, and routers. Mininet allows for the construction of a variety of network topologies, enabling us to create realistic and customizable environments for testing. The network topology is crucial as it defines the possible paths for routing decisions. In this study, we use a scale-free topology with 20 nodes, which ensures a balance between complexity and performance for evaluating the RL agent's decision-making under various conditions.
The Ryu controller manages the control plane of the SDN network, handling real-time flow rule updates and routing decisions. Ryu offers a flexible platform for integrating external applications, such as our RL agent, which dynamically interacts with the controller to make routing decisions. The interaction between the RL agent and the Ryu controller is facilitated by a well-defined API, which allows the RL agent to query and update flow rules based on observed network states.

B. Threat Modeling

To evaluate the robustness of the proposed reinforcement learning (RL)-based routing mechanism, we simulate common network attacks that frequently target SDN environments. In particular, we model DoS and blackhole attacks. DoS attacks are simulated by introducing high-volume traffic from malicious nodes, aiming to exhaust network resources and degrade overall performance. In blackhole attacks, certain switches or nodes are configured to silently drop packets, disrupting normal routing behaviour and causing significant data loss.
These threats are integrated into the simulation by dynamically altering the network topology and behaviour during both training and evaluation. This allows the RL agent to learn routing strategies that are not only

performance-efficient under normal conditions but also resilient in the presence of adversarial activity. By exposing the model to such scenarios, we promote the development of an adaptive and secure routing policy capable of operating effectively in real-world, threat-prone SDN environments.

C. Reinforcement Learning Environment

The network routing problem is formulated as a Markov Decision Process (MDP), where the RL agent continuously interacts with a dynamic SDN environment to learn optimal routing strategies. The state space encapsulates essential network parameters such as the positions of the source and destination nodes, current link status—including bandwidth availability and utilization levels—congestion indicators, and security-related features such as the presence of ongoing attacks (e.g., DoS or blackhole threats). At each decision point, the agent selects an action by choosing the next-hop node from the current node toward the destination, thereby defining the action space. The reward function is carefully designed to promote successful packet delivery and network efficiency, while penalizing packet loss, increased latency, and the use of compromised or insecure paths.

The RL environment is implemented using OpenAI Gym, which provides a standardized interface for developing and testing RL algorithms. To simulate real-time network behaviour, OpenAI Gym is integrated with Mininet, enabling realistic and interactive network emulation. This setup allows the RL agent to be trained and evaluated under dynamic network conditions, facilitating the development of an adaptive and secure routing strategy suitable for modern SDN infrastructures.

D. RL Algorithm Implementation

The primary learning algorithm used for training the RL agent is Deep Q-Network (DQN). DQN is a model-free RL algorithm that uses a deep neural network to approximate the action-value function Q(s,a)Q(s,a), which estimates the expected cumulative reward for taking action aa in state ss. The DQN agent is trained through the process of experience replay, where past transitions are stored in a replay buffer and sampled randomly to update the Q-network.

Algorithm 1: Reinforcement Learning-Based Secure Routing

Require: Network topology G = (V, E) G = (V, E), attack indicators, traffic matrix

Ensure: Secure and optimal routing paths

- Initialize Q-network Q(s, a; θ) Q(s, a; θ) with random weights θ
- Initialize target network Q′ with weights θ′←θ
- Initialize replay buffer D←∅
- **For** each episode **do**
  - Reset environment to initial state s0s_0
  - **For** each flow f in episode **do**
  - Observe current state s_t (node position, link load, attack status)
  - Select action a_t using ε-greedy policy from $Q(s_t, a_t; \theta)$ $Q(s_t, a_t; \theta)$
  - Execute action: apply next-hop a_t in Ryu controller
  - Observe reward r_t and next state s_{t+1}
  - Store transition (st,at,rt,st+1)(s_t, a_t, r_t, s_{t+1}) in D
  - Sample random mini-batch from D
  - Compute target:
    $$y_t = r_t + \gamma \max Q'(s_{t+1}, a'; \theta')$$
  - Update Q-network by minimizing loss:
    $$L(\theta)=E[(y_t - Q(s_t, a_t; \theta))^2]$$
  - Every C steps, update target network: θ′←θ
  - **End For**
- **End For**
- **Return** trained Q-network for real-time routing decisions

The training process follows a standard Q-learning approach with modifications to stabilize the learning process, such as the use of a target network and epsilon-greedy action selection. The goal is for the agent to learn optimal routing policies that maximize network performance while minimizing the impact of attacks. The training process is illustrated in Algorithm 1, which outlines the iterative procedure followed by the RL agent. In each step, the agent observes the current network state, selects an action, executes it via the SDN controller, receives feedback in the form of a reward, and updates the Q-network accordingly to refine its routing policy.

E. Real-Time Integration

Once trained, the RL agent is integrated into the Ryu controller for real-time routing decisions. During operation, for each incoming flow, the controller queries the trained Q-network to determine the optimal next-hop based on the current network state. This allows for adaptive routing, where the RL agent continuously adjusts routing decisions based on real-time network conditions, such as link congestion and attack status. The RL agent enables dynamic updates to the network's routing paths, ensuring secure and efficient traffic forwarding, even under adversarial conditions.

F. Evaluation

To assess the effectiveness of the proposed RL-based routing mechanism, we perform a comparative evaluation against conventional routing approaches, particularly shortest-path routing. The performance is measured using key metrics such as PDR, end-to-end delay, throughput, and route stability. PDR reflects the proportion of successfully delivered packets relative to the total number sent, while end-to-end delay captures the time taken for packets to traverse the network from source to destination. Throughput measures the rate of successful data transmission, and route stability indicates the consistency of selected paths over time.

The evaluation is conducted under both normal operating conditions and adversarial scenarios, including DoS and blackhole attacks. This dual-mode testing provides a comprehensive understanding of how the RL-based approach performs not only in ideal conditions but also in the presence of active threats. The results demonstrate that the RL-based method offers improved resilience and adaptability, achieving higher PDR, lower latency, and increased through put compared to traditional routing. These findings highlight the potential of reinforcement learning as a robust and security-aware routing solution for SDN environments.

IV. Results and Discussion

To assess the effectiveness of the proposed reinforcement learning-based adaptive routing framework, we conducted extensive simulations using a custom SDN testbed integrated with Mininet and the Ryu controller. Our evaluation focuses on both learning performance during training and operational performance under real-time routing scenarios. Key performance indicators include average reward progression, attack avoidance rate, and epsilon decay, along with routing efficiency under both normal and adversarial conditions. The results presented below illustrate the learning behaviour of the RL agent and its comparative advantage over traditional routing protocols in hostile environments.

A. Training Process

Smoothed Average Reward per Episode The average reward per episode reflects how well the RL agent learns to navigate the environment and avoid insecure routes. This metric serves as a proxy for policy effectiveness during the training phase. Initially, due to the high level of exploration dictated by the ϵ greedy policy, the agent receives inconsistent rewards. Over time, as it gathers more experience and learns from previous episodes, the reward signal stabilizes. In the first 50 episodes, the agent's reward fluctuates between negative values (-5 to +5) as it explores different paths and learns to avoid attacks. After episode 50, the reward steadily increases and stabilizes around +8 to +10, reflecting the agent's improved ability to avoid compromised nodes and reach the destination optimally.

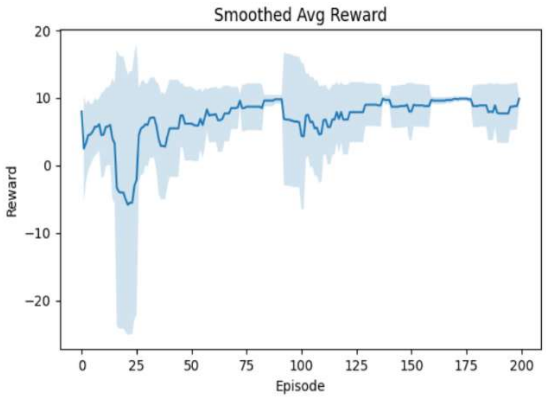

Figure 2: Smoothed Average Reward per Episode

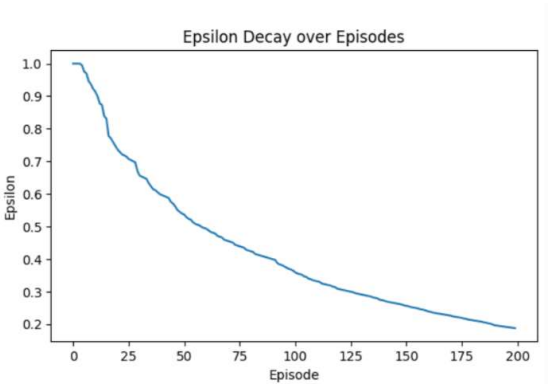

Figure 3: Epsilon Decay over Episodes

Epsilon Decay Schedule The epsilon-greedy strategy ensures a balance be tween exploration and exploitation. The agent begins training with $\epsilon = 1.0$, implying complete exploration, and gradually decays it to $\epsilon = 0.2$, favoring exploitation of learned routes. This decay strategy is crucial for avoiding premature convergence to suboptimal paths, especially in the presence of dynamic attack patterns. The epsilon value decreases from 1.0 at the beginning to around 0.2 by the end of the 200 episodes, signifying a shift from exploration to exploitation. This trend ensures that the agent systematically transitions to leveraging its learned policy for optimal decisions

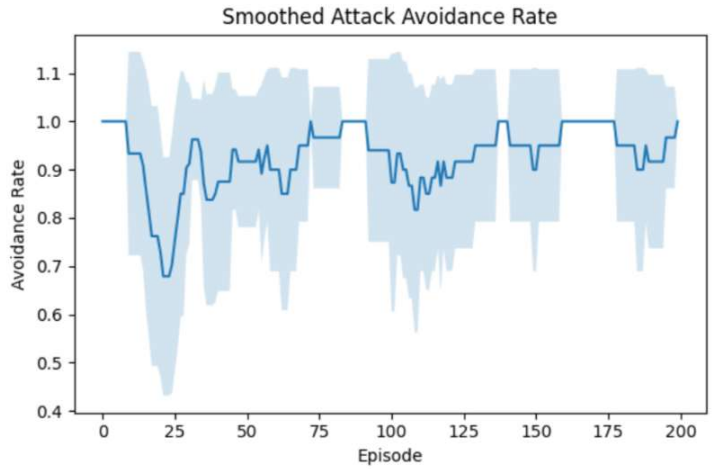

Figure 4: Smoothed Attack Avoidance Rate per Episode

Smoothed Attack Avoidance Rate per Episode The attack avoidance rate quantifies the percentage of routing decisions that successfully bypass compromised nodes. This metric is crucial in validating the agent's security awareness and its ability to adapt to adversarial scenarios. In the first 50 episodes, the avoidance rate is around 40% as the agent struggles to distinguish between safe and compromised paths. After episode 100, the avoidance rate improves significantly, stabilizing at around 80-90%, indicating the agent has effectively learned to avoid attack nodes and reroute traffic through safer paths.

B. Performance Comparison Under Normal and Attack Conditions

We benchmark the RL-based routing against a traditional shortest-path algo rithm under both normal operation and simulated attack conditions. The com parison evaluates metrics such as PDR, end-to-end delay, throughput, and route stability

TABLE I: PERFORMANCE UNDER NORMAL CONDITIONS

Metric	Shortest Path	RL-Based Routing
Packet Delivery Ratio (%)	96.2	95.8
End-to-End Delay (ms)	22.4	25.1
Throughput (Mbps)	48.6	47.9
Route Stability (changes/min)	1	2

TABLE II: PERFORMANCE COMPARISON UNDER ATTACK CONDITIONS

Metric	Shortest Path	RL-Based Routing
Packet Delivery Ratio (%)	63.7	88.4
End-to-End Delay (ms)	40.9	29.2
Throughput (Mbps)	28.1	43.5
Route Stability (changes/min)	5	3

Performance Under Attack Conditions When network attacks such as DoS and blackhole attacks are introduced, the RL-based routing demonstrates substantial performance advantages. Table 2 reports the results under adversarial conditions. While shortest-path routing suffers significantly under attack (notably in PDR and delay), the RL-based model adapts by avoiding malicious nodes, thereby maintaining higher delivery rates and more

stable routes. The reduced number of route changes also indicates that the agent converges on reliable alternatives rather than constantly reacting to disruptions.

C. Visual Analysis

Figures 2, 3, and 4 collectively illustrate the internal dynamics of the learning process. These visualizations provide deeper insights into the progression of the RL agent

- Smoothed Avg Reward: The reward increases and stabilizes over episodes, indicating the agent's improvement in avoiding attacks and optimizing its routing strategy.
- Epsilon Decay: The epsilon value decreases over time, reflecting the transition from exploration to exploitation.
- Smoothed Attack Avoidance Rate: The agent's ability to avoid attack nodes improves over time, reaching a stable avoidance rate of 80-90%.

The results of the simulation-based evaluation strongly support the viability of our reinforcement learning-based approach. While conventional shortest-path routing fails to maintain service quality under attack, the proposed RL agent demonstrates robust and adaptive behavior. The ability to dynamically learn, identify threats, and avoid compromised paths ensures higher packet delivery rates, reduced delays, and consistent throughput in adversarial settings. This positions reinforcement learning as a practical and resilient solution for security aware routing in modern SDN architectures.

V. Conclusion and Future Scope

In this paper, we proposed a reinforcement learning-based adaptive routing framework for SDNs that enhances network resilience in the presence of security threats such as DoS and blackhole attacks. By modeling the routing process as a MDP and integrating a DQN agent with the Ryu SDN controller, our approach enables dynamic path selection based on both real-time network conditions and threat indicators. Experimental evaluations conducted on a Mininet based simulation environment demonstrated that the RL-based routing method outperforms traditional shortest-path algorithms, particularly under adversarial conditions. It achieves higher packet delivery ratios, reduced end-to-end delay, and improved throughput, showcasing its ability to adapt to evolving threats and maintain stable network performance. This work contributes to the growing field of intelligent, security-aware SDN management and lays the foundation for future research into more sophisticated RL models, multi-agent cooperation, and the inclusion of additional security metrics such as trust scores and anomaly-based detection. In future work, we aim to extend this framework to handle large-scale networks, incorporate federated RL for distributed learning, and evaluate its effectiveness in real-world SDN deployments.

References

[1] D. Kreutz, F. M. V. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," Proc. IEEE, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[2] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," IEEE Commun. Surv. Tutor., vol. 16, no. 3, pp. 1617–1634, 2014.

[3] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in Proc. IEEE SDN Future Netw. Serv. (SDN4FNS), 2013, pp. 1–7.K. Elissa, "Title of paper if known," unpublished.

[4] Y. Jarraya, T.-M. Laleau, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," IEEE Commun. Surv. Tutor., vol. 16, no. 4, pp. 1955–1980, 2014.Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[5] R. Yu, S. Pal, and A. Xu, "A survey on reinforcement learning: Algorithms, applications and variants," arXiv preprint, arXiv:1906.02691, 2019.

[6] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on security in software defined networking," IEEE Commun. Surv. Tutor., vol. 19, no. 1, pp. 303–336, 2020.

[7] S. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in Proc. Int. Workshop Recent Adv. Intrusion Detect. Springer, 2011, pp. 161–180.

[8] A. Dixit, F. Hao, T. Lakshman, S. Mukherjee, and R. Kompella, "Towards an elastic distributed SDN controller," ACM SIGCOMM Comput. Commun. Rev., vol. 43, no. 4, pp. 7–12, 2013.

[9] K. Agarwal, P. Yadav, and D. Chauhan, "Machine learning for intrusion detection in SDN: A survey," Int. J. Comput. Appl., vol. 182, no. 1, pp. 1–6, 2019.

[10] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "A taxonomy of cyber attacks on SCADA systems," IEEE Trans. Smart Grid, vol. 1, no. 2, pp. 81–93, 2010.

[11] Y. Sun, A. Hahn, and C. Liu, “Machine learning for intrusion detection in industrial control systems: Challenges and opportunities,” Comput. Secur., vol. 89, p. 101682, 2020.
[12] E. Gelenbe, M. R. Mahmoud, and N. Mitton, “Learning routing strategies in networks with QoS constraints using reinforcement learning,” J. Netw. Syst. Manage., vol. 26, no. 3, pp. 676–692, 2018.
[13] W. Zhang, Q. Yu, and Y. Wu, “Towards security-aware routing in software-defined networks: A reinforcement learning approach,” Comput. Secur., vol. 79, pp. 277–289, 2018.
[14] J. Xu, L. Zhou, and Y. Lu, “A trust-aware reinforcement learning approach for secure routing in software-defined networks,” IEEE Access, vol. 9, pp. 34659–34670, 2021.
[15] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, “Machine learning integration with SDN: A comprehensive survey,” IEEE Commun. Surv. Tutor., vol. 22, no. 2, pp. 1234–1260, 2020.
[16] C. Li, L. Ma, J. Li, L. Sun, and P. Zhang, “A deep reinforcement learning approach for intelligent traffic engineering in SDNs,” IEEE Trans. Netw. Serv. Manage., vol. 16, no. 4, pp. 1328–1341, 2019.