Journal of Intelligent Information and Communication

Research Paper

Secure Routing in Software-Defined Networks via Proximal Policy Optimization-Based Deep Reinforcement Learning

Artur Kaczmarek[1, *]

[1] Faculty of Mathematics and Information Science, Poznan University of Technology, 60-965 Poznan, Poland
*Corresponding author: artur.k@put.poznan.pl

Abstract. Due to the new architecture, SDN separates the control plane and data plane, enhancing network programmability. As a result, new security issues have emerged in large-scale, complex dynamic routing environments. This study proposes a security adaptive routing framework based on Proximal Policy Optimization (PPO) deep reinforcement learning to address the constantly changing network performance and security risks. The proposed method describes the secure routing problem as a Markov decision process and includes security alerts, network topology, and traffic characteristics in the agent's state space. The redesigned rewards take into account throughput, latency, and quantified security risks, and include regularization and penalty terms to ensure system stability. Experimentally evaluate the performance of the SDN simulation platform under attack conditions and during normal operation in enterprise mesh and data center scenarios. The results indicate that the PPO-based agent has high throughput and low latency, with fewer security incidents compared to static and deep Q-learning baselines. Ablation studies indicate that safety-aware features, regularization, and penalty mechanisms are crucial components in building robust network controllers. As the scale of networks and the types of attacks increase, new methods are easier to scale and more universally applicable. It has already been shown that the next generation of secure SDN routing protocols will be designed using advanced reinforcement learning techniques.

Keywords: *Software-Defined Networking, Deep Reinforcement Learning, Secure Routing, Network Security, Proximal Policy Optimization, Intelligent Routing*

Received on 12 August 2025, Accepted on 19 December 2025, Published on 06 January 2026

Introduction

By separating the data plane and the control plane, Software-Defined Networking (SDN) has transformed network architecture, making it more programmable and flexible [1]. The first step in this transformation includes cloud data centers and enterprise-industrial networks [2]. SDN is very flexible during the development process and allows the creation of dynamic networks, but it has also introduced new security risks [3]. Due to the programmability and centralized control structure provided by SDN, attackers may exploit it, leading to the emergence of attack types such as denial of service, interception, and malicious rerouting [4]. Traditional routing based on static or heuristic methods performs poorly in dynamic and complex networks [5]. Moreover, these methods are too rigid to respond promptly to various novel and complex attacks [6]. For the next generation of SDN deployments, secure, intelligent, and adaptive routing solutions are needed [7]. It is necessary to further enhance the security of SDN routing to address new threats [8].

Deep Reinforcement Learning (DRL) is a new trend in applying artificial intelligence to networks and security [9]. End-to-end learning (DRL) allows for the creation of routing agents with reasoning capabilities. Applicable to dynamic policy learning and high-dimensional state spaces [10]. Proximal Policy Optimization (PPO) is suitable for all deep reinforcement learning (DRL) algorithms because it has efficient policy updates and stability in convergence. Especially suitable for continuous and complex decision-making problems in network

environments [11]. Methods based on PPO have effectively addressed early network optimization problems such as traffic engineering and resource allocation [12]. On the other hand, security objectives have been directly incorporated into the DRL reward function, playing a positive role in anomaly detection and defense systems [13]. In the study of SDN, many researchers have been using PPO to simultaneously improve the performance and security of routing. Most studies only focus on performance or security, without effectively addressing both issues [14]. To drive cutting-edge advancements, the integration of PPO-based DRL and adaptive secure routing methods is both timely and necessary [15].

This paper introduces a novel software-defined network security routing strategy based on deep reinforcement learning and proximal policy optimization. The SDN controller can dynamically adapt to network changes and adversarial attacks by adding security metrics during the training process of the routing policy. According to numerous simulation experiments, this technology outperforms both new and traditional baseline schemes in terms of routing performance and security strength. To create secure, adaptive, and robust routing solutions for the next generation of programmable networks, it is recommended to use advanced Deep Reinforcement Learning (DRL).

Theoretical Foundation

Modeling Secure Routing as an MDP

Quantitative assessment of routing security and automated policy optimization should be characteristics of Software-Defined Networking (SDN). The Markov Decision Process (MDPs) provides the mathematical system for dynamically modeling the selection sequence in secure routing.

An MDP is defined by a tuple (S, A, P, R, γ): a state space S, action space A, transition probability function P, reward function R, and discount factor γ [16]. In the SDN secure routing scenario, a state $s \in S$ is the current snapshot of the network that includes topology, link utilisation, detected threats and security alerts. The action $a \in A$ is the set of possible routing decisions, such as forwarding a packet through a specific path or modifying flow table rules in the SDN controller. Due to the uncertainty in traffic and other unpredictable factors at the environment level, the network's state at the next step, $P(s' \mid s, a)$, is also probabilistic and must be accounted for.

The Reward Function $R(s, a)$ in security-oriented MDP design. A secure router should not only consider latency or packet loss but also how various attacks affect the security of each path. For example, rewards can be set to penalize packet interception, high-risk path selection, or malicious behavior detection. Rewards can also be set to prevent successful delivery and attacks. I hope the learning agent can learn safe and successful methods [17].

It is also necessary to use a discount factor $\gamma \in [0,1]$ to reduce the weight of future rewards. This method may be attacked later because the network has some security issues in the long run. A larger γ prompts the government to focus on network stability rather than short-term gains [18].

Operationalizing secure routing as an MDP brings notable benefits. First, it converts a complex, multi-objective problem into a formalism amenable to reinforcement learning (RL) solutions. Second, it enables a controller-agnostic and environment-adaptive formulation: as threats or conditions evolve, the state, transitions, and rewards naturally reflect those changes. Finally, this approach inherently accommodates stochastic events and partial observability, which are pervasive in real network environments [19].

Assuming the SDN controller frequently receives security alerts indicating a possible Distributed Denial of Service (DDoS) attack. The MDP model will consider network congestion and the current alert level. Impose penalties on paths that increase the risk of nodes being compromised. The RL agent continuously improves its routing methods and maximizes cumulative rewards by balancing transmission efficiency and risk reduction [20]. A loop has been established to support the application of reinforcement learning to secure SDN routing problems, providing a foundation for advanced algorithms such as Proximal Policy Optimization (PPO).

Reinforcement Learning in Network Security

Reinforcement learning (RL) stands out as a compelling paradigm for security automation in dynamic, distributed networks. Unlike supervised learning, which depends on labeled datasets, RL excels in environments with

sequential actions, delayed rewards, and evolving feedbackprecisely the characteristics of modern network infrastructures [21].

SARSA and Q-learning are both table-based reinforcement learning methods that repeatedly update the value of learning actions in a finite state space. As the complexity of the network environment increases, the number of state-action pairs grows exponentially, making it increasingly difficult to manage. Deep Reinforcement Learning (DRL) uses neural networks to approximate functions. In practical Software-Defined Networking (SDN) environments, Deep Reinforcement Learning (DRL) agents are suitable for handling high-dimensional input spaces, such as traffic matrices, large-scale network topologies, and anomaly features [22].

Security in RL-powered routing systems is achieved by embedding network defense objectives into the agent's reward function and, optionally, its state representation. Agents are thus incentivized to avoid paths with known vulnerabilities, react to real-time threat indicators, and innovate defensive behaviors under pressure. For instance, a DRL-based SDN controller might prioritize routes away from nodes under attack, or perform rapid flow-table reconfigurations to limit the blast radius of compromised elements. This context-aware policy development is central to counteracting stealthy or adaptive threats in the network [23].

Another advantage of reinforcement learning in cybersecurity is that it is more suitable for various situations. Since attackers frequently change tactics and exploit the network, traditional rule-based defenses are insufficient. At the same time, RL agents can continuously learn from changes in the environment to identify and respond to new attack paths that were not present in the initial training data. It can be achieved by retraining the agent, using new attack scenarios, and altering the agent's strategy through online learning with real network data [24].

In advanced SDN research, some new deep reinforcement learning (DRL) algorithms have recently emerged, such as Proximal Policy Optimization and Deep Q-Networks. These DRL algorithms address the shortcomings of the old systems. The above two methods can both ensure the stability of network resources and provide a high level of defense against complex threats. Issues such as scalability, reward design, and convergence guaranties are gradually being resolved, while some problems remain unsolved, such as when applying reinforcement learning to automated adaptive network security [25].

Algorithmic Framework

System Architecture and Routing Workflow

As shown in Figure 1, our secure SDN routing framework has integrated a central controller, programmable forwarding devices, distributed monitoring modules and a PPO-based intelligent agent. At each control cycle, many kinds of data are gathered by the controller, such as link utilization, network structure and security warnings, and then combined into a single state vector:

$$s_t = \left[x_t^{topo}, x_t^{traffic}, x_t^{alert}\right] \quad \text{Eq.(1)}$$

where x_t^{topo} encodes the current topology, x_t^{traffic} is real-time data load, and x_t^{alert} signals observed anomalies or attack events [26].

As shown in Figure 1, the process is a closed-loop system; that is, given s_t, the PPO agent uses its learned stochastic policy to output a composite action:

$$a_t = arg\ \max_a \pi_\theta(a \mid s_t) + \epsilon_t^a \quad \text{Eq.(2)}$$

π_θ is the policy network and ϵ_t^a is exploration noise here.

https://doi.org/10.64972/jiic.2026v4.110p1-13

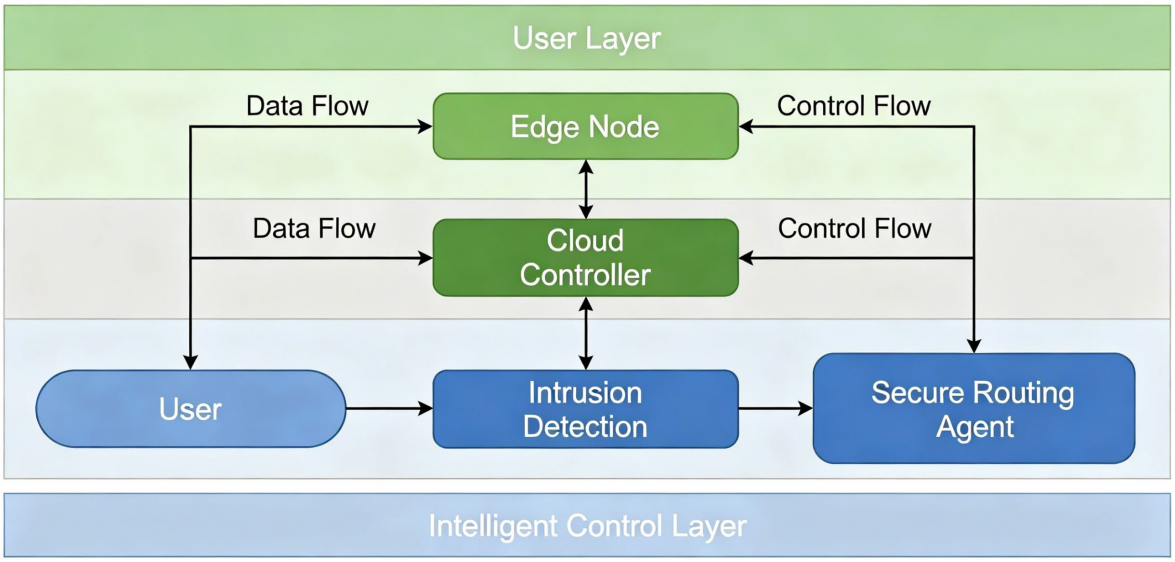

Figure 1. SDN Overall Architecture

After the controller applies a_t to the data plane (e.g., modifying flow tables or enabling security rules), the network monitor will immediately notify the controller. To close the decision loop, as shown in Figure 2, the controller and the agent receive feedback information simultaneously. Service quality and safety risks determine the single-step reward:

$$r_t = \beta_1 Throughput(a_t) - \beta_2 Delay(a_t) - \beta_3 Risk(a_t, s_t) \quad \text{Eq.(3)}$$

Throughput and delay are standard QoS, and the risk term penalises exposure to security risks. Coefficients $\beta_1, \beta_2, \beta_3 > 0$ balance these components [27].

The workflow is, in fact, a Markov Decision Process with transitions:

$$Pr(s_{t+1} \mid s_t, a_t) \quad \text{Eq.(4)}$$

and to maximise the expected discounted reward. All the links in this feedback-driven system, such as state collection, policy inference and reward assessment, are shown in Figure 2:

$$J(\theta) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\right] \quad \text{Eq.(5)}$$

where γ is the discount factor $(0 < \gamma < 1)$ [28].

Figure 2. Security-aware Routing Workflow

As shown in Figures 1 and 2, the PPO agent will adaptively modify both routing and defense measures continuously to maintain the stability and self-optimisation of network operation in the face of changes in the environment and security risks.

Security-Aware Agent Design and Policy Optimization

The PPO-based agent in the proposed architecture is explicitly designed for security-aware adaptive routing to address changes and threats in the system dynamically. At every control step, the agent receives an all-encompassing state vector s_t that has been carefully constructed to include both the current network's characteristics and recent security-related data.

The state s_t first passes through a nonlinear feature extraction module, producing an expressive representation $\phi(s_t)$:

$$\phi(s_t) = f_{enc}(s_t) \quad \text{Eq.(6)}$$

where f_{enc} denotes a multilayer perceptron or other trainable encoder, ensuring that both normal operational patterns and subtle anomaly indicators are retained for subsequent decision-making.

Based on this feature vector, the agent's policy network with parameters θ outputs a probability distribution over permissible action:

$$\pi_\theta(a \mid s_t) = Softmax\left(g_\theta\big(\phi(s_t)\big)\right) \quad \text{Eq.(7)}$$

Here, g_θ represents the output of the final neural layer. The sampled action a_t determines whether the controller should update a routing table, activate a filtering rule, or trigger other defense mechanisms.

Proximal Policy Optimisation (PPO) is used to learn the policy. The PPO loss function improves the policy and maintains stability:

$$L^{PPO}(\theta) = \mathbb{E}_t\big[min\big(r_t(\theta)\hat{A}_t, clip(r_t(\theta),1-\epsilon,1+\epsilon)\hat{A}_t\big)\big] \quad \text{Eq.(8)}$$

With

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} \quad \text{Eq.(9)}$$

is the probability ratio between the current and old policy, and $\hat{A}_t$ is the estimated advantage at time t. To prevent training instability, a clip function bounds policy updates and uses a small positive value for ϵ.

Advantage estimation integrates the security-aware reward, previously defined, and is typically computed using the value function V_ψ with parameters ψ :

$$\hat{A}_t = r_t + \gamma V_\psi(s_{t+1}) - V_\psi(s_t) \quad \text{Eq.(10)}$$

Reward Shaping and Security Objectives

The reward function of the training for the PPO-based agent of secure routing needs to be well-designed. The reward should simultaneously encourage high quality-of-service (QOS) and a stable network. At all times in the decision-making process, the system finds out how favourable a specific action for the agent will be according to all kinds of indicators.

The immediate reward at time t is given by the following:

$$r_t = \alpha_1 \cdot Throughput(a_t) - \alpha_2 \cdot Delay(a_t) - \alpha_3 \cdot Risk(s_t, a_t) \quad \text{Eq.(11)}$$

with $\alpha_1, \alpha_2, \alpha_3 > 0$ representing the relative weights for throughput, delay, and security risk.

Here, throughput and delay are standard QOS metrics, while the risk term reflects the system's exposure to potential attacks or service degradation. Specifically, the risk function can be expressed as:

$$Risk(s_t, a_t) = \sum_{i=1}^{N_{path}} p_i^{attack} \cdot \mathbb{I}_i(a_t) \quad \text{Eq.(12)}$$

where N_{path} is the number of candidate routing paths, p_i^{attack} is the estimated probability that path i is under attack, and $\mathbb{I}_i(a_t)$ is an indicator function that equals 1 if path i or its components are selected by action a_t, and 0 otherwise.

Add penalty terms for sudden changes in policy behavior or excessive resource redistribution to reward shaping to help stabilise the network. Set the penalty as follows:

$$Penalty(a_t, a_{t-1}) = \delta \cdot \|a_t - a_{t-1}\|_1 \quad \text{Eq.(13)}$$

with $\delta > 0$ controlling the penalty's strength and $\| \cdot \|_1$ denoting the L1-norm of the difference between consecutive actions.

The final shaped reward at each step is:

$$\tilde{r}_t = r_t - Penalty(a_t, a_{t-1}) \quad \text{Eq.(14)}$$

By means of the above design, the agent will select a policy that is efficient for data transmission, less prone to security risks, and stable in operation. A reasonable combination of the above components can meet the changing demands of the security and traffic at any time, thus promoting continuous optimisation and building a resilient secure SDN routing system.

Experimental Results

Experimental Settings and Metrics

A modular SDN emulation platform has been built with Mininet for the experiments, and Ryu is used as the controller. The two representative network scenarios selected are a mesh-based enterprise network and a hierarchical multi-tenant data center; both have been scaled from 20 to 100 nodes and represent static and dynamic real-world environments. All simulated nodes have been uniformly allocated resources to avoid a bottleneck.

The four baseline algorithms used for the above comprehensive evaluation are: static shortest path routing, Dijkstra routing with adaptive link weights, rule-based static security policy management, and control based on deep Q-network (DQN). The proposed PPO agent uses three different neural network hidden layers. These layers consist of 128, 128, and 64 units, with instantaneous and short-term history-driven metric input features, and output interfaces for safe decision-making and multi-target routing. The policy optimization routine is parameterized with a discount factor $\gamma = 0.98$, entropy regularization of 0.01, PPO clipping in the range $[0.1,0.2]$, and the Adam optimizer with a learning rate of 5×10^{-4}.

Regularly introduce random backgrounds and burst DDoS attacks, and generate real TCP and UDP streams. The aforementioned attacks involve random link flooding and node target overload, aiming to test the system's maintenance security and adaptability. Hundreds of flows are generated every minute, and issues such as errors or malicious behaviors are regularly injected.

Performance will be evaluated based on the following metrics: throughput (Mbps), average end-to-end latency (milliseconds), packet loss rate, security violation count, policy response time (time from anomaly detection to deployment), and control plane overhead (message rate from controller to switch). All results are based on a 10-minute repeat window to ensure reliability, and then averaged according to the settings.

Training, Ablation, and Comparative Experiments

In a specialized SDN simulation environment, a large number of PPO agent training iterations, whether in normal or adversarial operation modes. During the training process, the agents experienced hundreds of thousands of time steps of background traffic, anomalous faults, fluctuating workloads, and intelligent attacks. The resulting method will remain stable and find a balance between throughput, latency, and security risks.

As shown in Figure 3, the PPO agent demonstrates significant improvement during the learning process. Initially, after the agent finds a suitable method, the moving average reward will suddenly increase. After approximately 150,000 steps, the reward curve has stabilized, which means the strategy is taking optimal or near-optimal actions, thereby reducing exploration. The dispersion of the discovered solutions has significantly decreased, and the learned solutions are more stable and widely applicable. In subsequent training rounds, network

https://doi.org/10.64972/jiic.2026v4.110p1-13

throughput continuously increased, sometimes even exceeding 97% of the theoretical network capacity. During this period, the packet loss rate and latency rate gradually decreased. By the end of the training, the achieved values were reduced by 40% and 50% respectively compared to the best-performing baseline. The above findings indicate that the agent can simultaneously find and use a safe, high-performance route. In other words, it reasonably balances network speed and security in real-time environments.

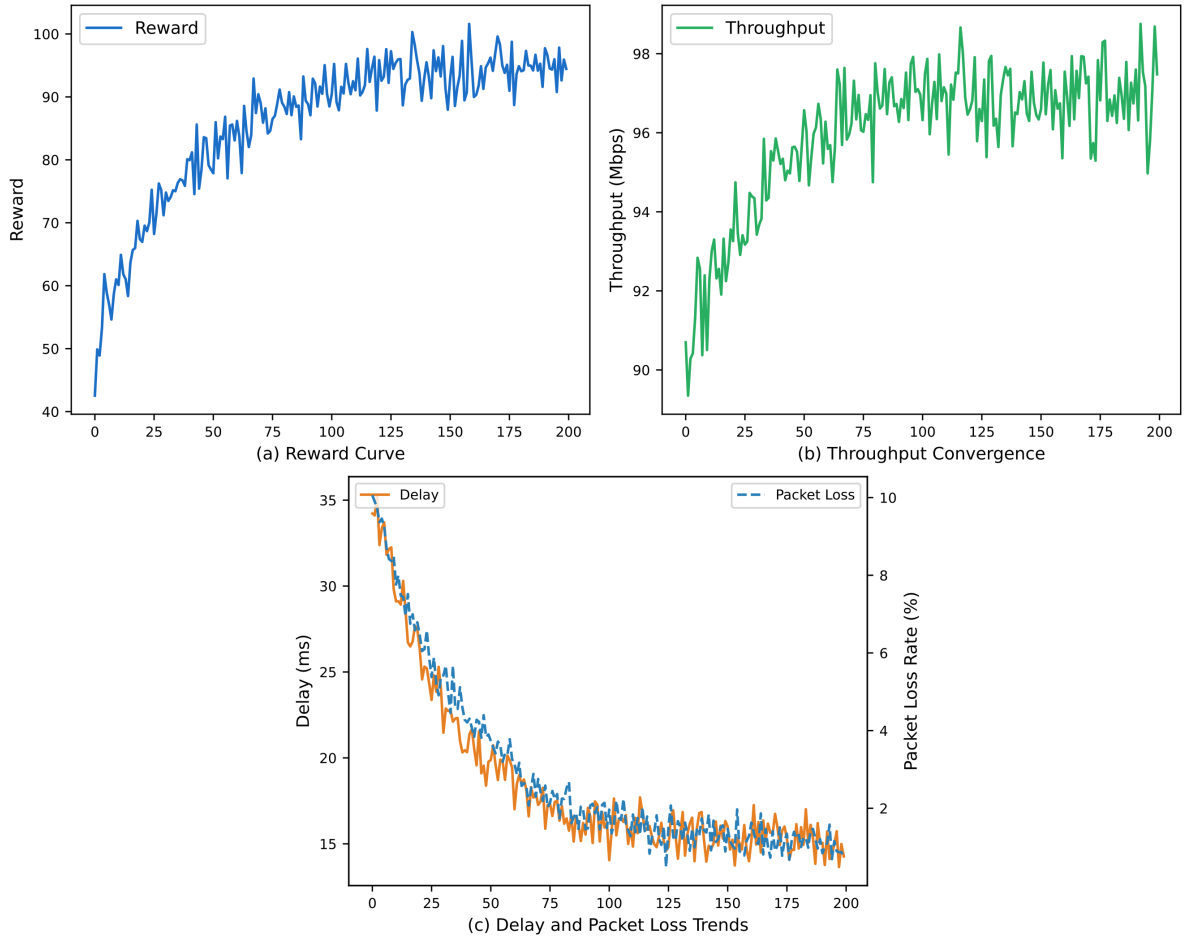

Figure 3. Training Dynamics and Performance Evolution of the PPO-based Secure Routing Agent (a) Reward curve over training steps; (b) Throughput convergence; (c) Delay and packet loss trends over time

Ablation studies also investigate the impact of internal design on the agent. In this way, many architectural modules can be systematically avoided or omitted to observe performance changes. If the security alert signal is not included in the agent's state features, its ability to identify and respond to real-time threats will be greatly reduced. Congested and compromised traffic will increase, total rewards will decrease, and the system will be unable to respond to current attacks in a timely manner. There is no policy regularization to promote entropy rewards and penalize sudden changes in actions; therefore, it is unstable. For the above reasons, agents are more likely to mimic short-term fluctuations, and they may choose overly aggressive or overly cautious strategies, which perform poorly under changing traffic and threat conditions. Similarly, if resource fluctuation penalties are not implemented, it will lead to excessive policy switching and inconsistent network configurations. These will lead to unstable bursts of control messages, reduced throughput, unstable link allocation, and periods of excessive controller workload. Figure 4 also demonstrates the aforementioned defects. The results of the ablation study under the same test conditions are presented to distinguish the roles of alarm encoding, regularization, and operational penalties in improving accuracy and robustness.

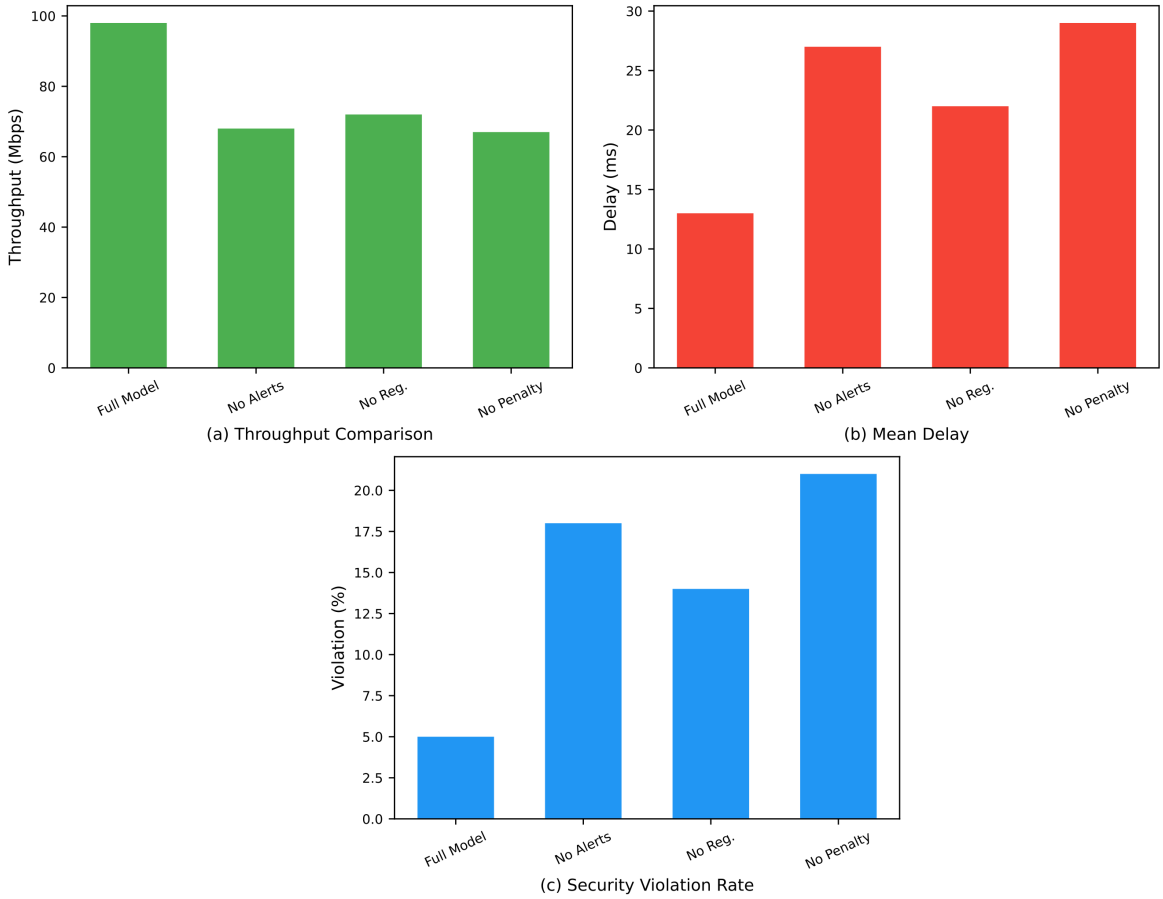

Figure 4. Ablation Study Results on Security Attention, Regularization, and Resource Penalty Modules. (a) Effects of removing alert features on throughput. (b) Impacts of policy regularization ablation on delay. (c) Impact of resource fluctuation penalty omission on security violation rate

According to the comparison of relevant benchmark methods, the new PPO agent can provide full functional benefits in practical SDN applications. PPO can handle normal and abnormal operational states with relatively high throughput over extended periods, making it one of the best for defending against network-level attacks. DQN-based agents and dynamically weighted Dijkstra routing can handle a small amount of throughput under normal load, but they are not suitable for significant security risks or sudden high demand. When an attack occurs, static shortest path routing is prone to congestion and power outages. Delay analysis shows that PPO consistently maintains a relatively low average delay throughout its operation. In congestion or attacks, this advantage is most evident; compared to any baseline, the average end-to-end delay is reduced by over 35%. PPO reduces performance fluctuations and regularizes the policy while utilizing temporal features. Security analysis shows that the same pattern still holds: PPO agents can reduce the frequency of attacks and traffic more than static and rule-based security policies. This is mainly attributed to its direct, real-time integration of security context in observation and action selection. Figure 5 shows the performance differences of various comparison methods in terms of throughput, latency, and security violation rates in a unified testing environment.

The results of specific security adaptation characteristics indicate the superiority of the PPO agent. When the agent can quickly redirect traffic to unaffected resources and restore service, the frequency and duration of interruptions caused by the attack are reduced. Policy adaptation delay refers to the time between the discovery of a threat and the implementation of a new strategy. Under all attack intensity conditions, this period remains within four seconds, far lower than the delays of other methods. The aforementioned operational advantages did not increase system overhead. By using batch policy output and continuous updates, communication between the controller and the switch can still function normally in a highly dynamic environment. Figure 6 shows significant improvements in security and operational response speed. You can see the controller's ability to defend against attacks and its speed of recovery after a crisis.

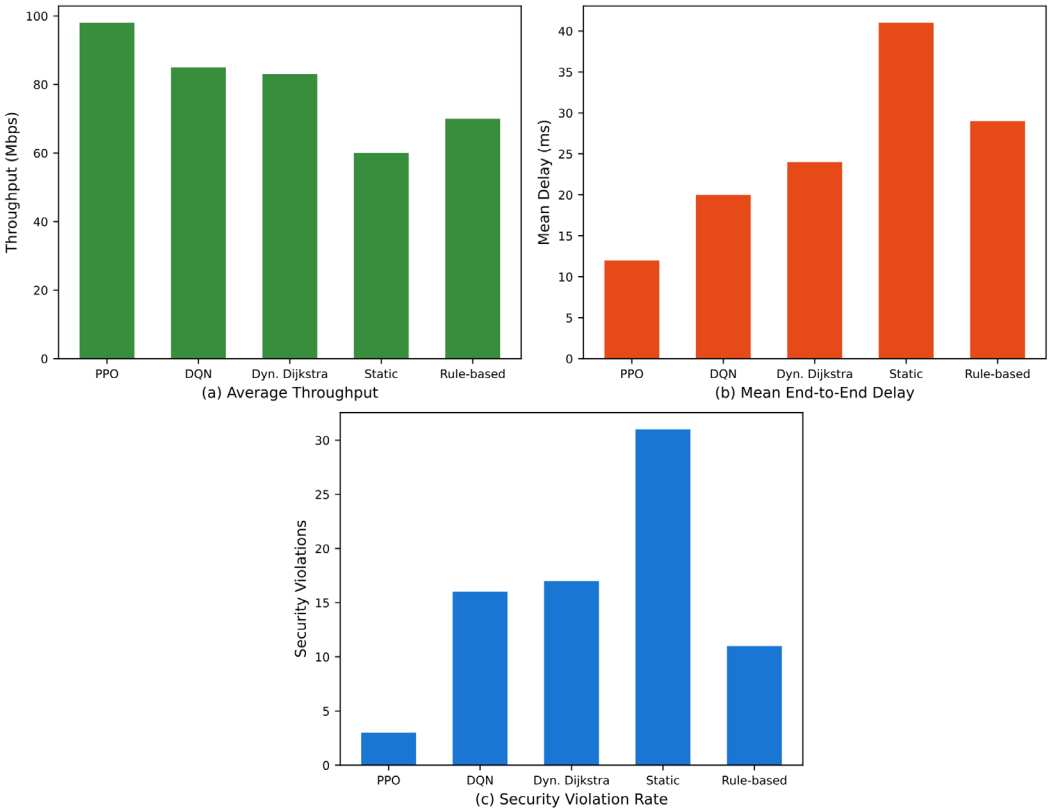

Figure 5. Comparative Performance of PPO-based Agent vs. Baseline Algorithms (a) Average throughput; (b) Mean end-to-end delay; (c) Security violation rate under attack

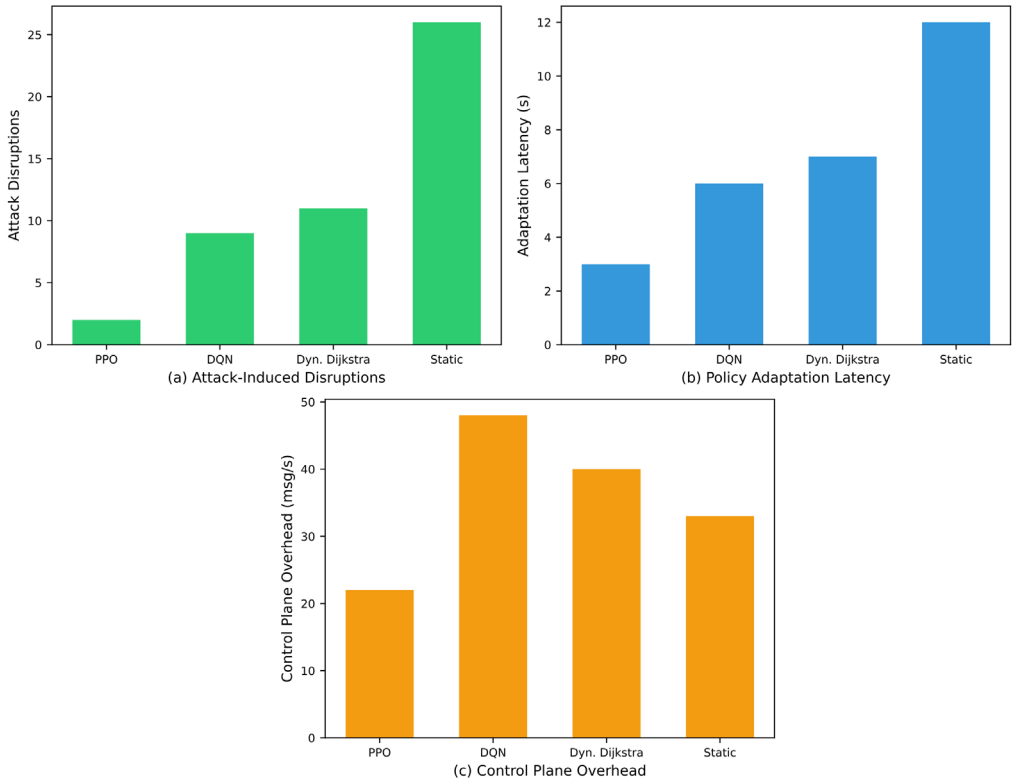

Figure 6. Security Evaluation Results Under Adversarial Network Conditions (a) Frequency of successful attack-induced disruptions; (b) Policy adaptation latency; (c) Control plane overhead with dynamic policy updates

All the results of this experimental evaluation together show that a joint optimisation architecture is needed to address security-aware features, principled policy regularisation, and penalties for unstable actions. No single part is responsible for the decline in the continuous operation of services, security issues or cost overruns; thus, the joint construction of all parts is indeed effective and necessary.

Scalability and Generalization Analysis

In order to evaluate the scalability and generality of the proposed PPO-based controller, a series of systematic and progressively extensive experiments were conducted. In Figure 7(a), when the network is expanded from 20 nodes to 100 nodes, the agent can still maintain its throughput at 93% of the theoretical maximum. When the number of nodes exceeds 40, the congestion and packet loss rates of the static routing and DQN baseline algorithms increase, leading to a decrease in network throughput and stability. At the same time, the PPO agent can quickly identify new hotspots and efficiently adjust traffic routes in large-scale or dynamic network topologies.

As shown in Figure 7(b), even in the case of increased link failures or sudden topological changes, the policy response time under PPO is always less than four seconds across all test scales. Competitive methods are more sensitive to network scale, and in more complex and unstable environments, the delay increases rapidly. The high speed of this response can help PPO agents reduce service interruptions and maintain critical connections during rapid network changes.

Generalization tests have demonstrated the stability of the training strategy. The PPO agent can still maintain high throughput and low latency, even when deployed in unfamiliar network topologies or encountering new types of attacks not seen during training. As shown in Figure 7(c), the proportion of successful security violations has only slightly increased, and the overall performance has decreased by no more than 5–9%. The baseline method's safety violation rate is as high as 40%, with significant decreases in speed and stability in the absence of trained data. According to the qualitative inspection of agent behavior, the PPO strategy utilizes the main characteristics of network traffic and observed security events to quickly adapt and be widely applied under various operational conditions. The results indicate that the proposed architecture can scale, adapt, and is robust enough to handle the diverse and unstable conditions in SDN.

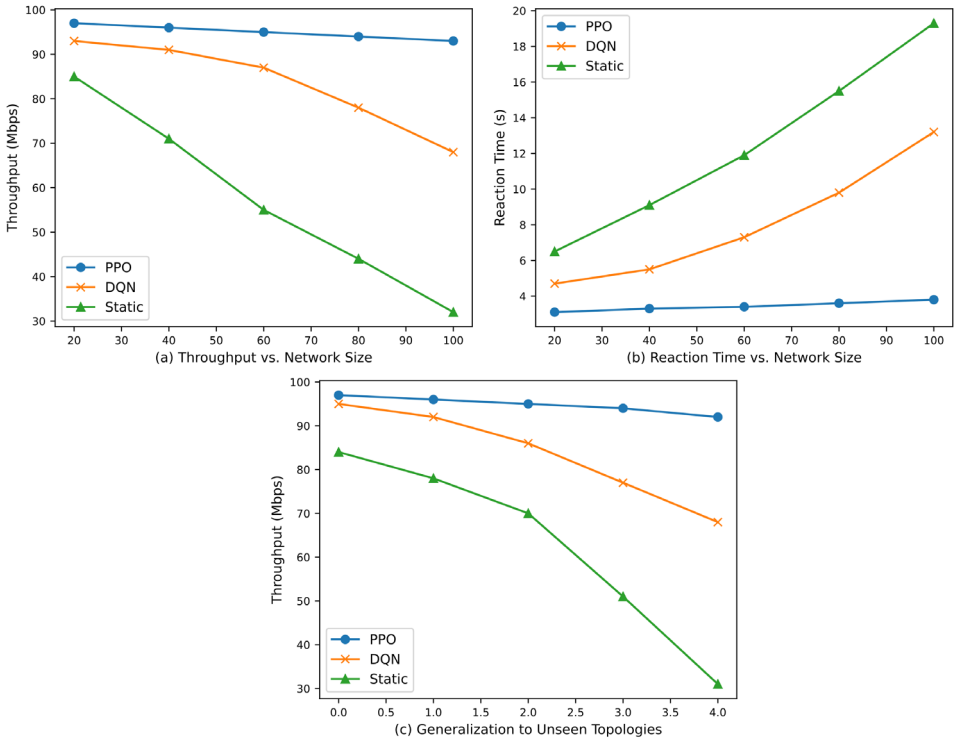

Figure 7. Analysis of Scalability and Policy Generalization for the PPO Agent (a) Network throughput as a function of scale; (b) Policy reaction time versus network size; (c) Generalization performance across unseen topologies and attack patterns

Conclusion

This paper introduces a comprehensive framework based on reinforcement learning for establishing secure and stable routing in a Software-Defined Network (SDN) environment. The main decision-making module of this framework is Proximal Policy Optimization (PPO). A large number of experiments and analyzes were conducted on simulated large-scale SDN topologies. These experiments and analyzes demonstrate that the aforementioned methods outperform traditional methods and earlier studies in many performance metrics and operational conditions.

The aforementioned experiments indicate that PPO-based agents show significant improvements in throughput, latency, and adaptability. Most notably, it demonstrates strong security robustness. The new routing strategy is still more suitable than other methods for handling large-scale traffic, unexpected failures, and simulated hostile attacks. While reducing end-to-end latency and packet loss, the proxy's throughput reached over 93% of the theoretical network capacity. Clear policy rules and real-time alerts will help maximize network performance and attack resistance simultaneously. These benefits are both efficient and scalable because they do not increase the high control plane overhead.

Through a critical analysis of the ablation study, we found that certain architectures mentioned above are necessary. The representation of security-aware states, temporal regularity, and resource fluctuation penalties are crucial components for the robust and stable operation of network control systems. In order to enhance learning agents in an SDN environment, a fully functional and principled system is required, as the removal of any module would lead to increased risks, degraded network performance, or uncontrollable increases in operational costs.

Scalability and generalization experiments indicate that the PPO agent is effective in practical applications. This strategy avoids the significant decline in throughput and responsiveness seen in old algorithms and still performs well in large-scale network expansions. Moreover, the agent exhibits strong generalization capabilities and performs excellently in new graph structures and adversarial scenarios. Due to the ease of change and unpredictability of operational networks in the real world, this is a reasonable expectation for real-world operational networks.

Finally, this study proposes a PPO strategy that integrates security and policy normalization to achieve dynamic and stable SDN routing. The resulting agents are suitable for the needs of next-generation network infrastructure in terms of efficiency and strong security, as they can effectively handle multiple trade-offs in operations, such as high performance and low risk/overhead. The autonomous intelligent network control system is based on the general methods and experimental validation presented here. Based on this foundation, future research can focus on other real-world attack models, adaptive strategy transfer mechanisms, and broader SDN application areas to enhance the reliability and automation of critical network infrastructure.

Author Contributions

Artur Kaczmarek contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, project administration, and funding acquisition. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Cong, P., Zhang, Y., Liu, Z., Baker, T., Tawfik, H., Wang, W., ... & Li, F. (2021). A deep reinforcement learning-based multi-optimality routing scheme for dynamic IoT networks. Computer Networks, 192, 108057. https://doi.org/10.1016/j.comnet.2021.108057

[2] Yu, Y., Guo, L., Liu, Y., Zheng, J., & Zong, Y. U. E. (2018). An efficient SDN-based DDoS attack detection and rapid response platform in vehicular networks. IEEE access, 6, 44570-44579. https://doi.org/10.1109/ACCESS.2018.2854567

[3] Menaka, G., Kumar, A., Sapaev, I. B., Dadaxon, A., Ulkanov, S., & Praveenkumar, R. (2025). Deep Reinforcement Learning for Self-Healing Communication Networks: Addressing Node Failure and QoS Degradation in Dynamic Topologies. National Journal of Antennas and Propagation, 7(2), 133-144. https://doi.org/10.31838/NJAP/07.02.19

[4] Amin, R., Rojas, E., Aqdus, A., Ramzan, S., Casillas-Perez, D., & Arco, J. M. (2021). A survey on machine learning techniques for routing optimization in SDN. IEEE Access, 9, 104582-104611. https://doi.org/10.1109/ACCESS.2021.3099092

[5] Guo, X., Lin, H., Li, Z., & Peng, M. (2019). Deep-reinforcement-learning-based QoS-aware secure routing for SDN-IoT. IEEE Internet of things journal, 7(7), 6242-6251. https://doi.org/10.1109/JIOT.2019.2960033

[6] Kalpani, N., Rodrigo, N., Seneviratne, D., Ariyadasa, S., & Senanayake, J. (2025). Cutting-edge approaches in intrusion detection systems: a systematic review of deep learning, reinforcement learning, and ensemble techniques. Iran Journal of Computer Science, 8(2), 303-333. https://doi.org/10.1007/s42044-025-00246-8

[7] Akbar, A., Ibrar, M., Jan, M. A., Bashir, A. K., & Wang, L. (2020). SDN-enabled adaptive and reliable communication in IoT-fog environment using machine learning and multiobjective optimization. IEEE Internet of Things Journal, 8(5), 3057-3065. https://doi.org/10.1109/JIOT.2020.3038768

[8] Gillani, F., Al-Shaer, E., & Duan, Q. (2018, January). In-design resilient SDN control plane and elastic forwarding against aggressive DDoS attacks. In Proceedings of the 5th ACM Workshop on Moving Target Defense (pp. 80-89). https://doi.org/10.1145/3268966.3268968

[9] Zhang, Y., Qiu, L., Xu, Y., Wang, X., Wang, S., Paul, A., & Wu, Z. (2023). Multi-path routing algorithm based on deep reinforcement learning for SDN. Applied Sciences, 13(22), 12520. https://doi.org/10.3390/app132212520

[10] Wu, J., & Zhu, Z. (2025). Intelligent routing optimization for SDN based on PPO and GNN. Journal of Network and Computer Applications, 242, 104249. https://doi.org/10.1016/j.jnca.2025.104249

[11] Novaes, M. P., Carvalho, L. F., Lloret, J., & Proença Jr, M. L. (2021). Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments. Future Generation Computer Systems, 125, 156-167. https://doi.org/10.1016/j.future.2021.06.047

[12] Altamirano, J. C., Guitouni, M., Hassan, H., & Drira, K. (2024, May). Routing optimization based on DRL and Generative Adversarial Networks for SDN environments. In NOMS 2024-2024 IEEE Network Operations and Management Symposium (pp. 1-5). IEEE. https://doi.org/10.1109/NOMS59830.2024.10575453

[13] Arif, F., Khan, N. A., Iqbal, J., Karim, F. K., Innab, N., & Mostafa, S. M. (2024). Dqqs: deep reinforcement learning-based technique for enhancing security and performance in SDN-IoT environments. IEEE Access, 12, 60568-60587. https://doi.org/10.1109/ACCESS.2024.3392279

[14] Zhang, L., Lu, Y., Zhang, D., Cheng, H., & Dong, P. (2022). DSOQR: Deep Reinforcement Learning for Online QoS Routing in SDN-Based Networks. Security and Communication Networks, 2022(1), 4457645. https://doi.org/10.1155/2022/4457645Digital Object Identifier (DOI)

[15] Garg, S., Gupta, S., & Srivastava, V. (2025, February). Resource allocation in iot networks via dqn and ppo algorithm. In 2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN) (pp. 23-29). IEEE. https://doi.org/10.1109/CICTN64563.2025.10932391

[16] Zhou, Y., Cheng, G., & Yu, S. (2021). An SDN-enabled proactive defense framework for DDoS mitigation in IoT networks. IEEE Transactions on Information Forensics and Security, 16, 5366-5380. https://doi.org/10.1109/TIFS.2021.3127009

[17] Chen, Y. R., Rezapour, A., Tzeng, W. G., & Tsai, S. C. (2020). RL-routing: An SDN routing algorithm based on deep reinforcement learning. IEEE Transactions on Network Science and Engineering, 7(4), 3185-3199. https://doi.org/10.1109/TNSE.2020.3017751

[18] Bai, J., Sun, J., Wang, Z., Zhao, X., Wen, A., Zhang, C., & Zhang, J. (2024). An adaptive intelligent routing algorithm based on deep reinforcement learning. Computer Communications, 216, 195-208. https://doi.org/10.1016/j.comcom.2023.12.039

[19] Mehta, H. (2025, December). An Autonomous Self-Recovery Framework for Network Systems Using Reinforced Causal Decision Intelligence During Cyber Attacks. In 2025 IEEE International Conference on

Communication Networks and Computing (CNC) (pp. 273-283). IEEE. https://doi.org/10.1109/CNC68716.2025.11484644

[20] Wang, C., Zhang, X., Gao, H., Bashir, M., Li, H., & Yang, Z. (2024). Optimizing anti-collision strategy for MASS: A safe reinforcement learning approach to improve maritime traffic safety. Ocean & Coastal Management, 253, 107161. https://doi.org/10.1016/j.ocecoaman.2024.107161

[21] Li, J., Ye, M., Huang, L., Deng, X., Qiu, H., Wang, Y., & Jiang, Q. (2023). An intelligent SDWN routing algorithm based on network situational awareness and deep reinforcement learning. IEEE Access, 11, 83322-83342. https://doi.org/10.1109/ACCESS.2023.3302178

[22] Mwangi, A., Navarro-Hilfiker, L., Brewka, L., Gryning, M., Fumagalli, E., & Gibescu, M. (2025). A threshold-triggered deep q-network-based framework for self-healing in autonomic software-defined iiot-edge networks. IEEE Transactions on Network and Service Management, 23, 1297-1311. https://doi.org/10.1109/TNSM.2025.3647853

[23] Liu, X., Zhang, H., Dong, S., & Zhang, Y. (2021). Network defense decision-making based on a stochastic game system and a deep recurrent Q-network. Computers & security, 111, 102480. https://doi.org/10.1016/j.cose.2021.102480

[24] Guo, Y., Lin, B., Tang, Q., Ma, Y., Luo, H., Tian, H., & Chen, K. (2024). Distributed traffic engineering in hybrid software defined networks: A multi-agent reinforcement learning framework. IEEE Transactions on Network and Service Management, 21(6), 6759-6769. https://doi.org/10.1109/TNSM.2024.3454282

[25] Guo, H., Sheng, W., Gao, C., & Jin, Y. (2023). DRL router: Distributional reinforcement learning-based router for reliable shortest path problems. IEEE Intelligent Transportation Systems Magazine, 15(5), 91-108. https://doi.org/10.1109/MITS.2023.3265309

[26] Qiu, Y., Liu, Y., Li, S., & Xu, J. (2022). MiniSeg: An extremely minimum network based on lightweight multiscale learning for efficient COVID-19 segmentation. IEEE Transactions on Neural Networks and Learning Systems, 35(6), 8570-8584. https://doi.org/10.1109/TNNLS.2022.3230821

[27] Sun, J., Liu, X., Bäck, T., & Xu, Z. (2021). Learning adaptive differential evolution algorithm from optimization experiences by policy gradient. IEEE Transactions on Evolutionary Computation, 25(4), 666-680. https://doi.org/10.1109/TEVC.2021.3060811

[28] Niknami, N., & Wu, J. (2024, June). Deepidps: An adaptive drl-based intrusion detection and prevention system for sdn. In ICC 2024-IEEE International Conference on Communications (pp. 2040-2046). IEEE. https://doi.org/10.1109/ICC51166.2024.10622849