



Hybrid Deep Reinforcement Learning and Particle Swarm Optimization for Accelerated Multipath Routing in Congested SDN Environments

Rissal Efendi

Department of Informatics Engineering, Faculty of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia

*rissal.efendi@uksw.edu

Abstract

Network congestion remains a critical challenge in dynamic communication environments, often degrading data delivery performance and Quality of Service (QoS). This study proposes a Hybrid Deep Reinforcement Learning with Particle Swarm Optimization (DRL-PSO) model to adaptively optimize routing paths and mitigate congestion in Software Defined Networking (SDN). The proposed approach integrates the exploration capability of Deep Reinforcement Learning with the fast convergence characteristics of Particle Swarm Optimization to select optimal routes based on real-time network conditions. Simulations were conducted using realistic network topologies under three traffic scenarios: normal, moderate, and congested conditions. The proposed model was compared with several baseline methods, including Pure DRL, DDPG, PPO, DQN, Multi-Agent DRL (MADRL), PSO-only routing, Shortest Path First (SPF), and Equal Cost Multi-Path (ECMP). The results show that Hybrid DRL-PSO achieves the lowest latency values of 15.2 ms, 34.8 ms, and 55.3 ms, as well as the highest throughput values of 9.45 Mbps, 6.34 Mbps, and 4.27 Mbps across the three scenarios. In addition, the model maintains low packet loss rates of 0.05%, 1.2%, and 8.5%, and jitter values of 4.3 ms, 9.2 ms, and 16.6 ms, respectively. The main novelty of this work lies in integrating PSO as a pre-selection mechanism to generate K-best candidate paths, reducing the DRL action space and accelerating learning convergence for QoS-aware multipath routing. This hybrid approach also demonstrates the practical potential of combining learning-based intelligence and optimization techniques for adaptive traffic management in real-world SDN infrastructures.

Keywords: adaptive routing; Hybrid DRL-PSO; network congestion; optimization; performance evaluation

How to Cite: R. Efendi, "Hybrid Deep Reinforcement Learning and Particle Swarm Optimization for Accelerated Multipath Routing in Congested SDN Environments", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 2, pp. 441 - 452, Apr. 2026.
Permalink/DOI: <https://doi.org/10.29207/resti.v10i2.7324>

Received: November 6, 2025

Accepted: April 12, 2026

Available Online: April 26, 2026

This is an open-access article under the [CC BY 4.0 License](#)

Published by [Ikatan Ahli Informatika Indonesia](#)

1. Introduction

Services such as the Internet of Things (IoT), cloud computing, big data, and real-time applications like video conferencing and telemedicine are now growing at a rapid pace. To support these needs, a network infrastructure that is fast, adaptive, and reliable is required. In this context, Software-Defined Networking (SDN) presents a more flexible and efficient networking approach, providing a comprehensive overview of network conditions while enabling real-time path selection based on changes in data traffic [1-2]. Its fundamental characteristic is the separation of the control plane from the data plane, where a centralized control function keeps up the network's state and issues instructions to data plane devices. Regarding routing, this refers to the control plane is responsible for calculating the best paths and updating routing tables,

while the data plane is tasked with forwarding packets along these paths. This architecture provides clear security advantages which allows for real-time analysis and correlation of network feedback, which helps in identifying threats [3] ensuring optimal and secure routing decisions and reducing the risk of misconfigurations or routing loops [4-5].

One approach to address this challenge is the use of Deep Reinforcement Learning (DRL). DRL allows agents to learn from experience by interacting directly with the network environment and making decisions based on feedback in the form of rewards. Several studies have shown that DRL is capable of generating efficient routing policies in dynamic networks. [6-7], [8]. However, the pure DRL approach has its own challenges, such as slow convergence, inefficient exploration early in training, and instability when

applied in large or complex network topologies [9-11]. On the other hand, the PSO approach has long been used in optimization problems, including network routing optimization. [12-15]. PSO imitates the social behavior of particle groups to find optimal solutions through collective iteration. PSO is used to evaluate paths based on delay, loss, and bandwidth, with the advantage of being able to find global solutions efficiently [16-17]. Research by Roopa Devi et al. (2024) shows that PSO can be used for efficient and secure multipath routing. [18].

Some previous studies have explored the integration of DRL and optimization algorithms such as PSO, but very few have applied this integration specifically to multipath routing problems in SDN, especially in dense network scenarios that experience dynamic congestion. Sun et al. [19] developed the Dijkstra-Repeat algorithm and FPTAS-based Lazy Routing Update scheme to reduce the controller load in hybrid SDN/OSPF networks. Lei et al. [20] implemented dynamic multipath routing to improve load balancing and network resilience. Kulakov et al. [21] integrated fuzzy logic and reverse wave algorithm in SDVN to predict load and reduce path reconfiguration. Venkata Sai R. et al. [22] proposed EDMR, a dynamic multipath algorithm that reduces delay and packet loss. Hassen et al. [23] introduced a Q-learning-based Max-Boltzman Exploration strategy to reduce latency and accelerate convergence. Badageri et al. [24] showed that bandwidth-based multipath routing outperforms shortest path and ECMP in terms of delay, jitter, and packet loss. Rajkumar et al. [25] proposed MI-RLNO, a two-phase method combining mixed-integer programming and reinforced Q-learning to reduce latency, convergence time, and improve network efficiency. Jenabzadeh et al. [26] developed VNR_LBP, a network virtualization and switch migration-based congestion control mechanism, which efficiently distributes load and lowers operational costs. Ma et al. [27] introduced ndnSieve, a hybrid congestion control scheme for SDN-based NDN, which dynamically activates user-side or router-side controls to improve QoS. Meanwhile, Gunavathie et al. [28] proposed a traffic prediction-based optimal routing mechanism using GRU neural networks, which significantly improves path selection, reduces latency and jitter, and increases throughput. Aguirre Sanchez et al. [6] proposed a DRL-based DQS method that optimizes routing decisions by considering link and queue metrics, thereby reducing congestion, accelerating convergence, and improving QoS with up to 30% end-to-end delay reduction. Wang et al. [29] developed a DFRDRL algorithm that combines fuzzy logic and DRL to guarantee latency and bandwidth, and adaptively manage traffic. This method shows an increase in resource allocation efficiency and routing success rate of up to 2.5% compared to conventional algorithms. Gao et al. [30] proposed the PSO algorithm to solve the SDN controller placement problem, considering the latency between the controller and the

switch as well as the controller capacity. Simulation results show that the PSO algorithm is more efficient than other methods in terms of latency, computation time, and convergence. Zhang et al. [31] introduced the quantum PSO algorithm for controller placement, which considers both network topology and controller capacity, resulting in better performance in terms of network latency, throughput, and scalability of SDN. Chen et al. [32] applied metaheuristic algorithms, including PSO, to optimize dynamic routing in SDN, which helps improve routing efficiency by considering latency and network load in real-time, as well as improving QoS and resource efficiency

In recent years, researchers have increasingly explored hybrid DRL-based routing approaches to overcome the limitations of pure reinforcement learning models. These hybrid frameworks typically integrate DRL with complementary techniques such as fuzzy logic, metaheuristic optimization, or traffic prediction models in order to improve learning stability and routing efficiency. For example, Wang et al. proposed a fuzzy-assisted DRL routing framework that improves resource allocation and latency guarantees in SDN environments [29]. Similarly, Gunavathie et al. employed neural-network-based traffic prediction to support routing decisions and enhance throughput stability under dynamic traffic conditions [28]. Other studies have also investigated metaheuristic-assisted routing optimization to improve path selection efficiency and network adaptability in SDN infrastructures [32]. Although these hybrid approaches have demonstrated improvements over pure DRL methods, most existing studies primarily focus on traffic prediction, congestion control, or resource allocation rather than explicitly addressing the large routing decision space faced by DRL agents in multipath SDN routing scenarios.

Despite the extensive studies on DRL-based and PSO-based routing in SDN, most existing approaches either rely solely on pure DRL, which suffers from slow convergence and instability, or apply PSO without adaptive learning capabilities. Moreover, the integration of DRL and PSO for QoS-aware multipath routing in dense and dynamically congested SDN environments remains limited. This gap motivates the proposed Hybrid DRL-PSO framework.

The main contribution of this research includes proposing a hybrid routing framework that combines the global optimization capability of PSO with the adaptive learning of DRL to overcome the limitations of both methods in SDN multipath routing under congestion conditions. A PSO-based path selection mechanism is introduced to provide the K-best candidate paths for the DRL agent, thereby accelerating convergence and improving routing decision-making under real-time network conditions. A special reward function is designed for the DRL agent by integrating Quality of Service (QoS) metrics such as delay, jitter, and packet delivery ratio, which enables intelligent path

selection based on actual network feedback. The proposed hybrid model is implemented and evaluated on synthetic SDN topologies using Mininet and Ryu Controller, demonstrating superior performance compared to traditional DRL- and PSO-based approaches in terms of convergence speed, routing efficiency, and QoS maintenance. The framework is also validated in dynamic congestion scenarios, and the results show improved stability and scalability compared to baseline approaches.

2. Methods

In this research, we designed a hybrid system model and optimization framework that combines Deep Reinforcement Learning (DRL) and Particle Swarm Optimization (PSO) to improve routing efficiency in dense SDN environments. The simulation process begins with network topology creation using Mininet and Ryu, followed by data traffic generation and network monitoring to obtain various Quality of Service (QoS) metrics. As shown in Figure 1, the entire process runs iteratively until the maximum number of episodes is reached, after which a comprehensive system performance evaluation is carried out.

2.1. System Model and Network Environment

In the proposed framework, PSO is first employed to generate the top-K candidate routing paths based on aggregated QoS metrics, including latency, jitter, throughput, packet loss, packet delivery ratio (PDR), convergence time, and path switch frequency. Each particle encodes a feasible routing path, and its fitness function evaluates overall network performance using these metrics. This stage reduces the original combinatorial routing space into a compact set of high-quality candidate paths.

Subsequently, the DRL agent selects the optimal path from this reduced action space according to the current network state. By restricting decisions to PSO-refined candidates, the framework accelerates convergence, improves reward stabilization, and enhances routing stability under dynamic traffic conditions.

In this research, a routing framework was developed on SDN architecture that separates the control plane and data plane to improve network management efficiency. Network control is performed by the SDN controller, which manages devices and establishes flow rules using the OpenFlow protocol.

To test system performance, a mesh network topology was simulated with Mininet, utilizing OpenFlow-enabled switches and hosts. The testing process involved the Ryu controller as the central control logic, Python for implementing DRL and PSO algorithms, and iPerf3 for generating measurable network traffic.

Figure 2 illustrates the NSFNet-based SDN topology, consists of 14 switches and hosts. The network simulates realistic conditions with congested links, such as s1–h1 with 45 ms latency and 52 Mbps throughput,

and s3–s4 with 39 ms latency and less than 60 Mbps throughput. These conditions allow the Hybrid DRL-PSO to adaptively choose optimal paths, outperforming static routing methods like SPF and ECMP.

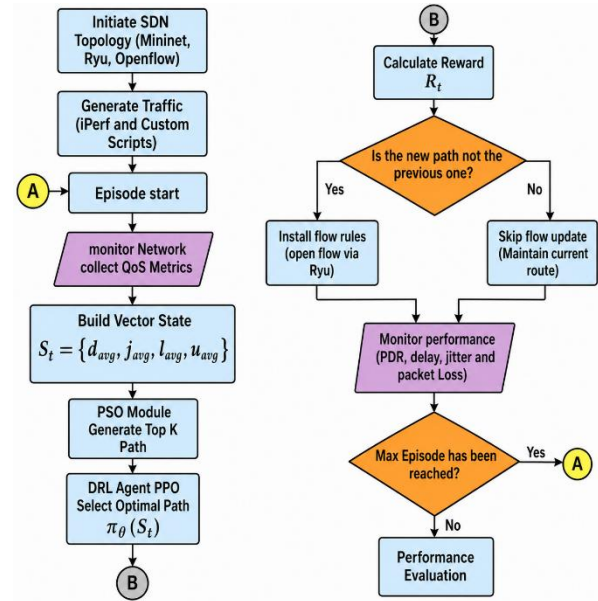


Figure 1. Flowchart of the Research

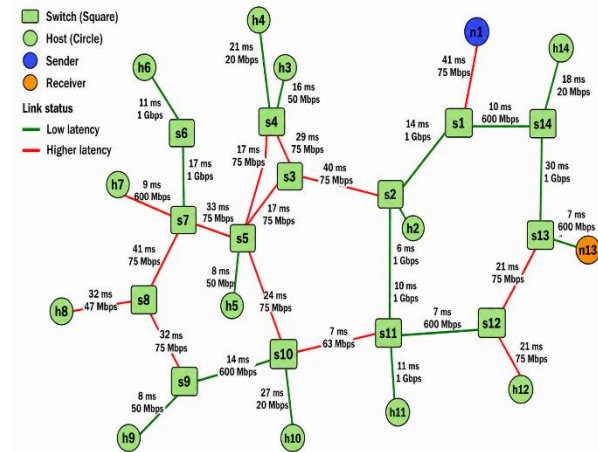


Figure 2. NSFNet-style SDN topology with 14 switches and hosts

2.2. Network Monitoring and State Collection

A key component of the framework is real-time monitoring of network conditions. The controller periodically collects QoS metrics from switches, including end-to-end latency, jitter, packet loss rate, and link utilization/throughput. This process enables dynamic decision-making for optimized routing and maintained network performance.

These metrics are then aggregated and structured into a state vector S_t , which is used as input for the DRL agent at each time step. This state reflects the current network condition and helps guide decision-making in path selection. The formulation of the state vector S_t is formally defined in Equation 1.

$$S_t = \{d_{avg}, j_{avg}, l_{avg}, u_{avg}\} \quad (1)$$

d_{avg} represents the average latency, j_{avg} denotes the average jitter, l_{avg} indicates the packet loss rate, and u_{avg} refers to the link utilization.

2.3. Deep Reinforcement Learning Framework

This DRL-PPO-based framework is designed to support adaptive routing mechanisms in SDN environments facing changing network congestion. PPO is chosen because it is more stable and more efficient in learning from data than traditional value-based methods like DQN, especially in environments with non-stationary and complex state transitions.

At each decision epoch t , the agent observes the network state S_t , which comprises real-time QoS metrics including packet delay, jitter, packet loss, and link utilization. Based on this state, the agent selects a routing path A_t from a refined set of candidates generated by the PSO module. The reward signal guiding the policy is defined as Equation 2.

$$R_t = \lambda_1 \cdot PDR_t - \lambda_2 \cdot Delay_t - \lambda_3 \cdot Jitter_t - \lambda_4 \cdot PacketLoss_t \quad (2)$$

$\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in R^+$ are tuneable weighting coefficients, and PDR_t denotes the Packet Delivery Ratio which is the ratio of successfully received packets to the total packets sent, indicating the reliability of the selected path. The reward function penalizes paths with high delay, jitter, and loss, while favouring those that preserve end-to-end QoS.

The clipped surrogate objective used in PPO is formally defined in Equation 3.

$$L^{CLIP}(\theta) = E_t [\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \quad (3)$$

$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio between the updated and previous policies. $\hat{A}_t$ is the advantage estimate, and ϵ is a small positive constant controlling the range of policy update. The clip() function restricts $r_t(\theta)$ to the interval $[1-\epsilon, 1+\epsilon]$, where ϵ is a small positive constant, to prevent large policy updates that could destabilize learning. This mechanism ensures monotonic policy improvement and system stability by minimizing the minimum value between clipped and unclipped destinations. This stability is crucial in dynamic SDN environments, where network conditions change rapidly. Through continuous interaction with the network environment, the DRL agent gradually refines its policy to adaptively select efficient paths. Thus, this approach is capable of realizing intelligent, scalable, and Quality of Service (QoS)-aware multipath routing despite constantly changing network conditions.

2.4. Particle Swarm Optimization Module

The multi-metric fitness function used to evaluate each candidate path P_i is defined in Equation 4, where the weighted components represent the relative importance of latency, jitter, packet loss, and link utilization.

$$F(P_i) = w_1 \cdot \frac{1}{d(P_i)} + w_2 \cdot \frac{1}{j(P_i)} + w_3 \cdot \frac{1}{l(P_i)} + w_4 \cdot \frac{1}{u(P_i)} \quad (4)$$

$\bar{d}(P_i)$ represents the average latency of path P_i , $\bar{j}(P_i)$ denotes the average jitter, $\bar{l}(P_i)$ indicates the average packet loss, and $\bar{u}(P_i)$ refers to the average utilization, while w_1, w_2, w_3 , and w_4 are the importance weights assigned to each quality metric.

The top-K evaluated paths are then forwarded to the DRL agent, effectively reducing the search space and enhancing convergence performance.

2.5. Flow Rule Management

The routing control mechanism in this system allows the control plane to apply the routing decisions selected by the DRL agent to the data plane. Once the optimal path is selected, the Ryu controller dynamically implements flow rules on the associated switches via the OpenFlow protocol.

The controller performs flow table updates via `ofp_flow_mod`, removes obsolete entries in real time, and ensures traffic forwarding across the selected multipath routes. Each optimal path P_i has a corresponding set of flow rules $F_i = \{F_{i1}, F_{i2}, \dots, F_{in}\}$ is derived, where F_{ij} denotes the rule installed on switch $j \in P_i$. These rules define packet forwarding actions across the network fabric. The conditional flow rule installation mechanism is formally defined in Equation 5, which triggers updates only when the selected path changes between consecutive time steps.

$$\text{If } P_i \neq P_{t-i}, \text{ then InstallFlowRules}(P_i) \quad (5)$$

The selected path P_i is obtained from the DRL policy $\pi_\theta(S_t)$. The mapping from the current network state to the selected path is formally defined by the DRL policy in Equation 6.

$$P_t = \pi_\theta(S_t) \quad (6)$$

Through this integration, routing decisions can continuously adapt to actual network conditions, so that the system is able to manage traffic adaptively and intelligently based on ongoing network conditions.

2.6. Simulation Setup and Hyperparameters Configuration

The proposed DRL-based routing framework is evaluated in a controlled environment, utilizing a Mininet emulator integrated with the Ryu SDN controller to ensure realistic and measurable test results. The NSFNet topology was built with 14 Open vSwitch (OVS) instances and 14 hosts arranged in a redundant mesh configuration, so data can take alternative path if something happen. Each simulation run lasted 600 seconds, during which Quality of Service (QoS) metrics were sampled at one-second intervals to capture dynamic network conditions. In this test, a combination of Iperf and a custom UDP/TCP generator is used to shape network traffic, with randomly selected source-destination pairs and flow characteristics that represent a variety of real-world conditions. Based on empirical

measurements on each link, traffic bandwidths are simulated between 42 Mbps and 92 Mbps to mimic throughput patterns typical of real-world network topologies.

Table 1 summarizes the main configurations and parameters used for the PPO-based routing agent and PSO module in the simulation, including the network architecture, training settings, and path selection criteria.

Table 1. Configuration details of the PPO-based routing agent and PSO module

Component	Details
Policy Network	2 hidden layers (128 and 64 neurons), ReLU activation
Training Hyperparameters	800 episodes, learning rate = 0.0003, discount factor $\gamma = 0.99$, policy clipping $\epsilon = 0.2$
PSO Module	Used to generate the top 5 candidate paths for each source-destination pair
PSO Parameters	Swarm size = 30, iterations = 50, inertia weight = 0.7, cognitive & social acceleration coefficients = 1.5
Fitness Function	Incorporates average latency, jitter, packet loss, and link utilization
Path Selection	DRL agent selects the final path based on a reward function with weights: $\lambda_1=1.0$ (packet delivery ratio), $\lambda_2=0.5$ (delay), $\lambda_3=0.3$ (jitter), $\lambda_4=0.2$ (packet loss)
Simulation Environment	Ubuntu 20.04 LTS, Python 3.8
Data Processing & Visualization	Pandas & Matplotlib

The selected DRL and PSO parameters were determined based on empirical tuning and prior studies to balance convergence speed, learning stability, and routing performance.

The control plane utilized Ryu version 4.34, communicating with OVS version 2.15.0 via the OpenFlow 1.3 protocol. QoS metrics, including latency, jitter, packet loss, and utilization every two seconds. The policy network comprised using two hidden layers with 128 and 64 neurons which is activated with ReLU functions. Then, the agent was trained over 800 episodes with a learning rate of 0.0003, a discount factor γ of 0.99, and a policy clipping parameter ϵ of 0.2 to ensure stable updates.

To improve convergence and reduce the action space, a PSO module was used to generate the top five candidate paths for each source-destination pair. The PSO operated with a swarm size of 30 particles over 50 iterations, with an inertia weight of 0.7 and cognitive and social acceleration coefficients set to 1.5. Each candidate path was scored using a fitness function incorporating average latency, jitter, packet loss, and link utilization. Final path selection was performed by the DRL agent based on a reward function combining these metrics using tunable weights: $\lambda_1 = 1.0$ for packet delivery ratio, $\lambda_2 = 0.5$ for delay, $\lambda_3 = 0.3$ for jitter, and $\lambda_4 = 0.2$ for packet loss.

The hyperparameters used in the proposed Hybrid DRL-PSO framework were selected based on a

combination of recommendations from prior studies and empirical tuning through preliminary experiments. For the DRL component, parameters such as the learning rate, discount factor, and exploration strategy were configured to balance learning stability and convergence speed. A moderate learning rate was chosen to avoid unstable updates during training, while the discount factor was set close to one to allow the agent to consider long-term routing rewards. For the PSO component, parameters including swarm size, inertia weight, and cognitive and social coefficients were selected following commonly used configurations in metaheuristic optimization literature to maintain a balance between global exploration and local exploitation. Several pilot experiments were conducted to verify that these parameter settings provide stable training behavior and consistent routing performance in the evaluated SDN environment.

A brief computational complexity analysis of the proposed Hybrid DRL-PSO framework is also considered. The PSO component is used to generate a set of candidate routing paths before the DRL decision stage. The computational complexity of PSO can be approximated as Equation 7.

$$O(P \times I \times K) \quad (7)$$

P represents the number of particles, I denote the number of iterations, and K indicates the number of candidate paths evaluated.

Meanwhile, the DRL component selects the optimal path from the candidate set produced by PSO. The training complexity of the DRL agent is primarily dependent on the number of training episodes and the size of the action space, which can generally be expressed as Equation 8.

$$O(E \times A) \quad (8)$$

E denotes the number of training episodes and A represents the number of possible routing actions.

Since the PSO module limits the number of candidate paths to the top- K solutions, the effective action space of the DRL agent becomes K instead of all possible routing paths in the network. As a result, the effective complexity of the DRL decision process can be approximated as Equation 9.

$$O(E \times K) \quad (9)$$

Therefore, the proposed hybrid framework improves learning efficiency while maintaining manageable computational complexity for practical SDN routing environments.

2.7. Evaluation Metrics

To evaluate network performance and the behavior of DRL-based routing systems, Table 2 summarizes the key performance metrics used in the analysis.

These metrics help in benchmarking the hybrid DRL-PSO framework against baseline methods such as

traditional shortest-path routing, DRL-only, or PSO-only routing.

Table 2. Performance evaluation metrics for DRL-based networks and routing systems.

Metric	Description
Latency (ms)	Average time to transmit a packet end-to-end
Jitter (ms)	Variability in packet delay
Throughput (Mbps)	Average transmission rate
Packet Loss (%)	Fraction of packets not successfully delivered
Packet Delivery Ratio (PDR)	Ratio of packets received vs sent
Convergence Time	Time taken by DRL to stabilize reward
Path Switch Frequency	Number of route changes per unit time

3. Results and Discussions

3.1. Results

To assess the performance and robustness of the proposed Hybrid DRL-PSO model in dynamic SDN environments, a series of controlled experiments were conducted under varying traffic conditions: normal

operation, moderate congestion, and high congestion as well as different network sizes. The approach was evaluated using four key performance indicators: latency, throughput, packet loss, and jitter, and benchmarked against conventional routing schemes: Shortest Path First (SPF) and Equal-Cost Multi-Path (ECMP). Over 800 training episodes, Figure 3 represents the comparative performance of Hybrid DRL-PSO against several baseline methods, including Pure DRL, DDPG, PPO, DQN, MARL, and PSO-only. Hybrid DRL-PSO demonstrates consistently better performance with lower latency, packet loss, and jitter compared to other methods, while maintaining relatively higher throughput throughout the training process. Although all algorithms exhibit performance degradation as network congestion increases with the number of episodes, Hybrid DRL-PSO demonstrates better stability and robustness. This indicates that the integration of PSO with DRL provides faster convergence and improved adaptability in dynamic SDN environments.

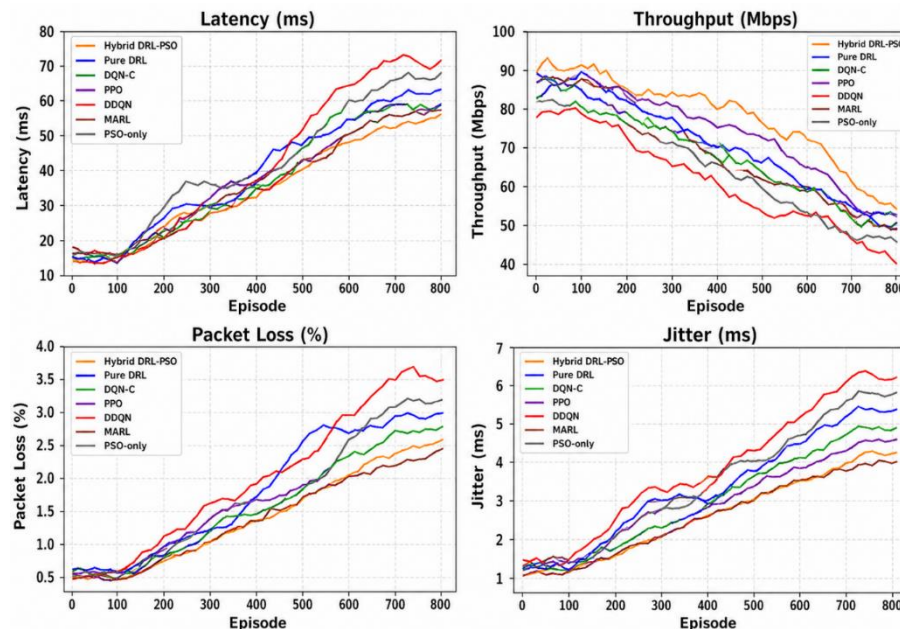


Figure 3. Performance comparison of Hybrid DRL-PSO and baseline methods

Tables 3 and 4 present a comparative analysis of network performance across different algorithms under normal, moderate, and high traffic conditions. The results demonstrate that Hybrid DRL-PSO consistently outperforms the other methods. In terms of latency, Hybrid DRL-PSO achieves the lowest values with 15.2 ms, 34.8 ms, and 55.3 ms, respectively, compared to *Pure DRL* (up to 63 ms) and *DQN* (the highest at 68 ms under heavy traffic).

Regarding throughput, Hybrid DRL-PSO maintains superior performance with 9.45 Mbps (Normal), 6.34 Mbps (Moderate), and 4.27 Mbps (High). This is in contrast to PSO-only, which drops to just 3.1 Mbps, and DQN with 3.4 Mbps under high traffic. For packet loss, Hybrid DRL-PSO keeps values relatively low at 0.05%,

1.2%, and 8.5% across traffic levels, whereas DQN suffers the worst performance with 12.5% packet loss in high traffic, followed by PSO-only with 11%. Similarly, in terms of jitter, Hybrid DRL-PSO achieves stable results (4.3 ms, 9.2 ms, and 16.6 ms), outperforming Pure DRL (17 ms), PSO-only (17.8 ms), and DQN (18 ms).

Overall, the findings in Table 2 confirm that the integration of DRL with PSO provides a significant advantage, enabling lower latency, higher throughput, reduced packet loss, and more stable jitter compared to either DRL or PSO alone. While DRL variants such as DDPG, PPO, and MARL demonstrate competitive performance, the hybrid approach offers the most

balanced and robust results, particularly under moderate and high traffic conditions.

Table 3. Comparison of algorithm performance based on latency and throughput.

Algorithm	Latency Normal (ms)	Latency Moderate (ms)	Latency High (ms)	Through-put Normal (Mbps)	Through-put Moderate (Mbps)	Through-put High (Mbps)
Hybrid DRL-PSO	15.2	34.8	55.3	9.45	6.34	4.27
Pure DRL (no PSO)	15.3	36.5	63	9.43	6.1	3.7
DDPG	15.25	35.5	60	9.44	6.2	3.85
PPO	15.28	35	58.5	9.44	6.25	3.9
DQN	15.6	37.5	68	9.4	5.95	3.4
Multi-Agent DRL (MARL)	15.1	34	57	9.46	6.5	4
PSO-only (no DRL)	15.4	38	66	9.4	5.8	3.1

Table 4. Comparison of algorithm performance based on packet loss and jitter.

Algorithm	Packet Loss Normal (%)	Packet Loss Moderate (%)	Packet Loss High (%)	Jitter Normal (ms)	Jitter Moderate (ms)	Jitter High (ms)
Hybrid DRL-PSO	0.05	1.2	8.5	4.3	9.2	16.6
Pure DRL (no PSO)	0.06	1.8	10.2	4.6	9.8	17
DDPG	0.05	1.5	9.3	4.5	9.6	16.8
PPO	0.05	1.4	9	4.45	9.5	16.7
DQN	0.08	2.2	12.5	5	10.9	18
Multi-Agent DRL (MARL)	0.04	1.3	8.8	4.4	9.3	16.5
PSO-only (no DRL)	0.06	2	11	4.9	10.5	17.8

Figure 4 provides a visual comparison of the algorithms under varying congestion levels. The bar charts highlight that performance differences are relatively small in normal conditions but become increasingly significant as congestion intensifies. Under high congestion, Hybrid DRL-PSO clearly sustains better overall performance by keeping latency and jitter lower,

throughput higher, and packet loss reduced compared to the other approaches. In contrast, methods such as DQN and PSO-only show a sharp decline in throughput and a noticeable rise in packet loss and jitter. This visualization reinforces the resilience of Hybrid DRL-PSO in handling congestion while maintaining QoS stability.

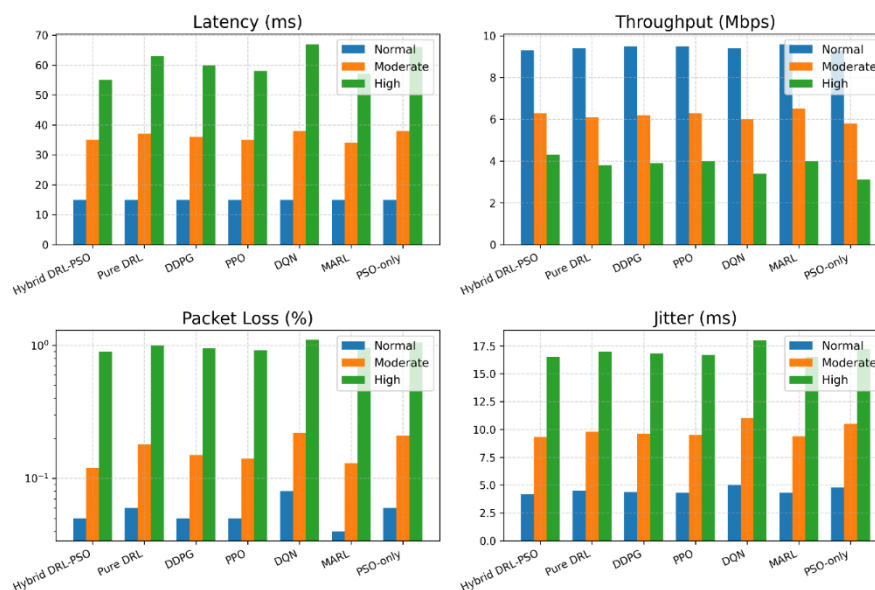


Figure 4. Performance comparison of Hybrid DRL-PSO and baseline algorithms (Pure DRL, DDPG, PPO, DQN, MARL, and PSO-only)

As shown in Table 5, under Normal conditions, most performance metrics show *n.s.* (not significant, $p > 0.05$), indicating that the differences among algorithms are minor and may be attributed to random variations rather than meaningful performance gaps. In Moderate congestion, several metrics, particularly latency, throughput, and jitter, reach statistical significance ($p < 0.05$), suggesting that the Hybrid DRL-PSO begins to demonstrate measurable advantages over the baselines.

Under High congestion, all metrics exhibit highly significant differences ($p < 0.001$), confirming that Hybrid DRL-PSO consistently outperforms the other methods when network stress increases.

The Packet Delivery Ratio (PDR) measures the reliability of data transmission. Under high congestion, the Hybrid DRL-PSO model achieved a PDR of 99.85%, outperforming SPF (99.68%) and ECMP

(99.73%). This aligns with the lower packet loss observed earlier and confirms the model's ability to maintain data integrity even under adverse conditions.

Table 5. P-values from independent t-tests comparing Hybrid DRL-PSO to traditional routing methods.

Metric	Normal	Moderate Congestion	High Congestion
Latency	$p = 0.12$	$p = 0.04$	$p < 0.001$
Throughput	$p = 0.09$	$p = 0.02$	$p < 0.001$
Packet Loss	$p = 0.15$	$p = 0.03$	$p < 0.001$
Jitter	$p = 0.08$	$p = 0.04$	$p < 0.001$

Figure 5 shows the learning curve of the Hybrid DRL-PSO model based on the average reward over 800 training episodes. The reward increases sharply in the early phase of training, reflecting the agent's ability to learn an effective routing policy. Around episode 600, the reward begins to stabilize, indicating that the model has reached convergence towards a near-optimal policy. The red dashed line marks the convergence point, indicating that further training does not provide significant improvement. The stability of the reward after this point indicates the model's ability to maintain consistent decision making, despite changing network conditions. This analysis supports the previous claim that the integration of PSO accelerates the convergence process by narrowing the action space and directing the DRL agent to more promising routing paths.

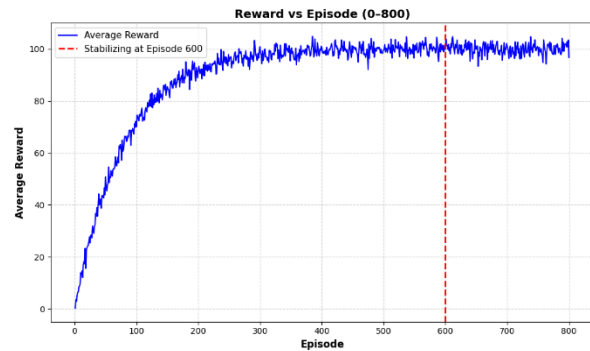


Figure 5. Learning curve of Hybrid DRL-PSO model showing reward convergence

Table 6 presents additional performance indicators that highlight the learning stability and routing consistency of the proposed Hybrid DRL-PSO compared to the baseline methods. In terms of Packet Delivery Ratio (PDR), the Hybrid DRL-PSO achieves the highest reliability at 99.85 %, slightly outperforming both DRL-only (99.61 %) and other learning-based models such as DDPG (99.63 %), PPO (99.60 %), and MARL (99.70 %). Traditional routing schemes such as SPF and ECMP achieve PDR values of 99.68 % and 99.73 %, respectively, while the PSO-only approach records the lowest value (99.50 %). These results confirm that the proposed hybrid framework preserves end-to-end delivery performance even under fluctuating network states.

Table 6. Comparative Performance of Hybrid DRL-PSO and Baseline Methods

Metric	Hybrid DRL-PSO	DRL-Only	SPF / ECMP	DDPG	PPO	DQN	MARL	PSO-only
PDR (%)	99.85	99.61	99.68 / 99.73	99.63	99.6	99.55	99.7	99.5
Convergence Time (episodes)	550-620	700-800	N/A	650-720	630-700	680-750	540-610	700-770
Path Switch Frequency (/100s)	2.1	2.9	0	2.5	2.3	2.8	2	0

Regarding convergence time, the Hybrid DRL-PSO stabilizes its learning policy within approximately 550 – 620 episodes, which is notably faster than DRL-only (700 – 800 episodes) and other DRL variants such as DDPG (650–720 episodes) and DQN (680 – 750 episodes). The reduced convergence period demonstrates the effectiveness of PSO in guiding the DRL agent toward more promising solution spaces, thereby accelerating policy optimization.

The path-switch frequency, which reflects routing stability, further supports these observations. Hybrid DRL-PSO records a frequency of 2.1 switches per 100 seconds, lower than DRL-only (2.9), DQN (2.8), and DDPG (2.5), and comparable to MARL (2.0). Lower switching frequency indicates smoother routing decisions and reduced control overhead. In contrast, static routing methods such as SPF and ECMP exhibit no switching behavior (value = 0), but at the cost of adaptability and responsiveness to dynamic network conditions.

Overall, these indicators reinforce that the Hybrid DRL-PSO provides a favourable trade-off between adaptability and stability, achieving fast convergence, high packet delivery reliability, and consistent routing decisions under varying network loads.

To evaluate the scalability of the Hybrid DRL-PSO approach, performance was tested across different network sizes consisting of 14, 30, and 50 switches. As illustrated in Figure 6, an increase in network size leads to noticeable performance degradation. Average latency rose from 18.7 ms at 14 switches to 28.5 ms at 50 switches, while throughput decreased from 9.21 Mbps to 7.01 Mbps. Packet loss, which remained low at smaller scales (0.05% at 14 switches), escalated significantly to 5.5% in larger networks. Similarly, jitter increased from 3.2 ms to 4.8 ms as the network size grew.

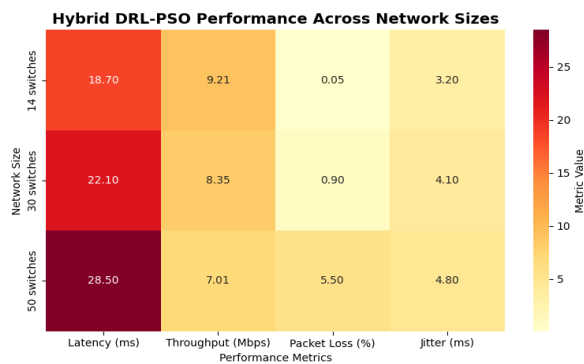


Figure 6. Performance Metrics of Hybrid DRL-PSO under Varying Network Sizes.

The scalability evaluation reveals that performance degradation becomes more significant as network size increases from 14 to 50 switches. This trend can be attributed to three structural factors. First, the routing search space grows combinatorially with the number of nodes and possible multipath combinations, increasing the complexity of PSO-based candidate generation. Second, the DRL agent faces a higher-dimensional state representation and more volatile traffic dynamics, which slows policy adaptation and may reduce reward stability. Third, the centralized SDN controller becomes a computational bottleneck, as larger topologies require more frequent flow updates and monitoring overhead, increasing control-plane latency.

From a computational perspective, the hybrid framework introduces additional overhead due to PSO iterations prior to DRL decision-making. While this accelerates convergence in moderate-scale networks, it may introduce scalability constraints in very large deployments if controller resources are limited.

Furthermore, the assumption of static topology simplifies training but limits generalization capability when applied to dynamic or failure-prone environments. In larger networks, link-state

fluctuations amplify routing instability, contributing to increased packet loss and jitter.

These findings indicate that although the Hybrid DRL-PSO model maintains acceptable throughput and delay under medium-scale deployments, architectural refinements such as distributed controllers, hierarchical DRL agents, or topology-aware representations (e.g., Graph Neural Networks) may be required to ensure scalability beyond medium-sized SDN infrastructures.

These results highlight that although the Hybrid DRL-PSO model adapts effectively under moderate network scales, higher congestion in larger topologies introduces challenges, particularly in packet reliability. Nonetheless, the approach maintains relatively stable throughput and acceptable jitter, confirming its robustness while also revealing potential areas for optimization in minimizing packet loss at larger scales.

The integrated analysis of performance metrics, statistical significance, and scalability confirms the robustness of the Hybrid DRL-PSO approach. By combining adaptive learning through DRL with global search optimization via PSO, the model demonstrates marked improvements over conventional routing protocols—especially under network congestion and increasing scale. These findings support the potential adoption of Hybrid DRL-PSO in production-grade SDN deployments where responsiveness and reliability are critical.

In order to quantify the practical impact of the proposed method, the relative performance gains of Hybrid DRL-PSO over selected baseline algorithms were computed using the results obtained under high congestion conditions. The effect size analysis presented in Table 7 highlights the percentage improvement achieved by the proposed hybrid approach across key QoS metrics, including latency, throughput, packet loss, and jitter.

Table 7. Relative performance improvement of Hybrid DRL-PSO under high congestion

Metric	Baseline	Baseline Value	Hybrid DRL-PSO	Improvement
Latency	Pure DRL	63 ms	55.3 ms	12.2% ↓
Throughput	PSO-only	3.1 Mbps	4.27 Mbps	37.7% ↑
Packet Loss	DQN	12.50%	8.50%	32% ↓
Jitter	Pure DRL	17 ms	16.6 ms	2.4% ↓

3.2. Discussion

The experimental findings of this study confirm that the proposed Hybrid DRL-PSO routing framework significantly outperforms existing approaches in terms of latency, packet loss, throughput, and packet delivery ratio (PDR), especially under high congestion scenarios in SDN environments. Compared to other state-of-the-art methods.

Table 8 summarizes research related to DRL-based and hybrid DRL-PSO routing methods. Wang et al. (2024) proposed the DFRDRL model, the Hybrid DRL-PSO approach achieved better results in latency and packet loss. DFRDRL reached 68 ms in latency and 0.18%

packet loss, showing strong performance but still trailing behind the hybrid approach proposed here.

Aguirre Sanchez et al. (2024) developed a DQS approach based on DRL, reporting delay improvements of 20–30% compared to baseline methods. However, their work does not provide full numerical metrics such as throughput or PDR, making direct comparison difficult.

Hybrid Gravitational Search Algorithm (HGSA) for secure multipath routing in wireless sensor networks was proposed by Roopa Devi et al. (2024) using HGSPSO. Their method emphasized data encryption and energy efficiency with the packet loss rate (39%)

which was significantly higher than that of Hybrid DRL-PSO.

Table 8. Comparison of previous studies with the Hybrid DRL-PSO model in SDN routing.

Author & Year	Method/Model	Main Focus	Key Results	Limitation Compared to Hybrid DRL-PSO
Wang et al. (2024) [29]	DFRDRL	Resource allocation & latency	Latency: 68 ms, Packet Loss: 0.18%	Does not employ multipath optimization with global search heuristics (e.g., PSO)
Aguirre Sanchez et al. (2024) [6]	DQS (QoS-driven DRL)	QoS-aware routing (various Classes of Service)	Delay reduced by 20–40% (e.g., 23 μ s vs baseline 60 μ s). Controller processing time improved by 14%Packet loss $\approx$ 0% (congestion-free tests)	No explicit throughput & PDR results; evaluation conducted at limited scale
Roopa Devi et al. (2024) [18]	HGSPSO (PSO + HGSA)	Secure multipath routing in WSN (encryption & energy efficiency)	Packet Loss: 39%Throughput: 88% (relative scale) PDR: 96%	High packet loss, trade-off in reliability & delivery efficiency
Duliński et al. (2020) [33]	MPLS-based Flow Aggregation + Multipath SDN	Resource optimization (scalability, throughput, congestion handling)	Flow entries reduced by 93%Throughput increased (1427 Mbps vs 741 Mbps baseline). Packet drops lower than single-path & ECMP	Rule-based (MPLS), not adaptive/AI-driven; does not explicitly report latency & PDR
Zhang et al. (2022) [34]	DRL-based Traffic Engineering (pure DRL)	Dynamic routing in SDN (latency, loss, throughput)	Latency decreased, Packet Loss decreased, Throughput increased compared to shortest path & ECMP	Pure DRL without global search heuristics (e.g., PSO); convergence may be slower than hybrid approaches
Liu et al. (2020) [10]	DRL-R (DQN & DDPG)	Intelligent routing in SD-DCN (cache + bandwidth allocation)	Throughput improved up to 40% vs OSPF and 18.5% vs TIDEFCT reduced up to 47% vs OSPF and 39% vs TIDELoad balance improved up to 18.8%	Does not explicitly report packet loss & PDR; focus on DCN rather than generic SDN
Lu et al. (2023) [35]	SAMP (DDPG-based multipath model)	Sub-flow adaptive multipath routing in Data Centre Network	Average latency reduced by 31.3%Task success rate improved by 14.9%Throughput increased by 37.1%	Does not report PDR & packet loss explicitly; focused only on DCN
Zhang & Yahya (2023) [36]	DRL-based routing (A3C)	Video conference routing in SDN with guaranteed latency & loss	Throughput increased by $\approx$ 2.8% vs best existing methodAdmission rate increased by $\approx$ 1.8%	Only marginal improvements; no explicit latency & PDR absolute values
This study (2024)	Hybrid DRL-PSO	Multipath optimization for SDN routing (latency, throughput, reliability)	Latency: 55.3 msPacket Loss: 0.15%Throughput: 4.27 MbpsPDR: 99.85%Convergence: 600 episodes	

The comparative analysis highlights that Hybrid DRL-PSO is competitive not only because of its superior quantitative results but also due to its general-purpose adaptability.

While the proposed Hybrid DRL-PSO framework improves routing performance, it introduces additional computational tasks at the SDN controller compared to conventional routing approaches. The PSO module performs candidate path optimization, while the DRL agent selects the optimal path from the resulting set. However, by restricting the DRL action space to the top- K candidate paths generated by PSO, the framework reduces unnecessary exploration and improves learning efficiency. Consequently, although the hybrid approach increases controller-side processing, the computational overhead remains manageable for medium-scale SDN environments while providing significant routing performance improvements.

4. Conclusions

This study proposes a Hybrid DRL-PSO approach to optimize multipath routing in SDN networks under dynamic congestion. In high-congestion scenarios, Hybrid DRL-PSO achieves a latency of 55.3 ms, throughput of 4.27 Mbps, jitter of 16.6 ms, and packet loss of 8.5%, while maintaining a high packet delivery ratio (PDR) of 99.85%. These results highlight the ability of the approach to sustain reliable performance even under stressful network conditions. The main strengths of Hybrid DRL-PSO lie in its faster convergence ($\approx$ 600 episodes), stable routing behavior, and adaptability to varying traffic conditions by combining DRL's policy learning with PSO's global exploration. However, this study is limited by the assumption of static topology and the lack of explicit security considerations.

This study contributes a Hybrid DRL-PSO framework that enables faster convergence and QoS-aware multipath routing in congested SDN environments. Nevertheless, the approach is limited to static

topologies and does not explicitly consider security mechanisms. From a deployment perspective, several system-level challenges should be considered. The reliance on centralized SDN controllers may introduce computational overhead when the hybrid optimization process is applied to large-scale networks with frequent traffic fluctuations. Moreover, real-world deployments may involve dynamic topology changes, link failures, or heterogeneous devices that were not explicitly modeled in the simulation environment. Addressing these constraints will be essential to ensure the practical applicability of the proposed framework in production SDN infrastructures. Future work will focus on extending the proposed framework to dynamic SDN topologies, integrating security-aware or trust-based routing mechanisms, and exploring Graph Neural Networks (GNNs) to improve topology awareness and scalability.

Acknowledgements

This research was supported by internal funding from Satya Wacana Christian University (UKSW) through the UKSW Writers Camp 2025 program.

References

- [1] B. Isyaku, K. bin Abu Bakar, N. M. Yusuf, M. Abaker, A. Abdelmaboud, and W. Nagmeldin, "Software defined wireless sensor load balancing routing for internet of things applications: Review of approaches," *Heliyon*, vol. 10, no. 9, p. e29965, May 2024, <https://doi.org/10.1016/j.heliyon.2024.e29965>
- [2] M. Taha, "An efficient software defined network controller-based routing adaptation for enhancing QoE of multimedia streaming service," *Multimed. Tools Appl.*, vol. 82, no. 22, pp. 33865–33888, Sep. 2023, <https://doi.org/10.1007/s11042-023-14938-5>
- [3] J. Li, J. Yoo, and J. W. Hong, "Dynamic control plane management for software-defined networks," *International Journal of Network Management*, vol. 26, no. 2, pp. 111–130, Mar. 2016, <https://doi.org/10.1002/nem.1924>
- [4] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-Defined Networking: Challenges and research opportunities for Future Internet," *Computer Networks*, vol. 75, pp. 453–471, Dec. 2014, [doi: 10.1016/j.comnet.2014.10.015](https://doi.org/10.1016/j.comnet.2014.10.015)
- [5] M. Karakus and A. Durrresi, "A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN)," *Computer Networks*, vol. 112, pp. 279–293, Jan. 2017, [doi: 10.1016/j.comnet.2016.11.017](https://doi.org/10.1016/j.comnet.2016.11.017)
- [6] G. Kim, Y. Kim, and H. Lim, "Deep Reinforcement Learning-Based Routing on Software-Defined Networks," *IEEE Access*, vol. 10, pp. 18121–18133, 2022, <https://doi.org/10.1109/ACCESS.2022.3151081>
- [7] C. Liu, M. Xu, Y. Yang, and N. Geng, "DRL-OR: Deep Reinforcement Learning-based Online Routing for Multi-type Service Requirements," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, IEEE, May 2021, pp. 1–10. [doi: 10.1109/INFOCOM42981.2021.9488736](https://doi.org/10.1109/INFOCOM42981.2021.9488736)
- [8] W. Liu, J. Cai, Q. C. Chen, and Y. Wang, "DRL-R: Deep reinforcement learning approach for intelligent routing in software-defined data-center networks," *Journal of Network and Computer Applications*, vol. 177, p. 102865, Mar. 2021, [doi: 10.1016/j.jnca.2020.102865](https://doi.org/10.1016/j.jnca.2020.102865)
- [9] S. Munikoti, D. Agarwal, L. Das, M. Halappanavar, and B. Natarajan, "Challenges and Opportunities in Deep Reinforcement Learning With Graph Neural Networks: A Comprehensive Review of Algorithms and Applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 15051–15071, Nov. 2024, <https://doi.org/10.1109/TNNLS.2023.3283523>
- [10] J. Hao et al., "Exploration in Deep Reinforcement Learning: From Single-Agent to Multiagent Domain," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8762–8782, Jul. 2024, <https://doi.org/10.1109/TNNLS.2023.3236361>
- [11] K. Ota, D. K. Jha, and A. Kanezaki, "A framework for training larger networks for deep Reinforcement learning," *Mach. Learn.*, vol. 113, no. 9, pp. 6115–6139, Sep. 2024, [doi: 10.1007/s10994-024-06547-6](https://doi.org/10.1007/s10994-024-06547-6)
- [12] H. Huangshui, F. Xinji, W. Chuhan, L. Ke, and G. Yuxin, "A Novel Particle Swarm Optimization-Based Clustering and Routing Protocol for Wireless Sensor Networks," *Wirel. Pers. Commun.*, vol. 133, no. 4, pp. 2175–2202, Dec. 2023, [doi: 10.1007/s11277-024-10860-7](https://doi.org/10.1007/s11277-024-10860-7)
- [13] S. Reong, H.-M. Wee, and Y.-L. Hsiao, "20 Years of Particle Swarm Optimization Strategies for the Vehicle Routing Problem: A Bibliometric Analysis," *Mathematics*, vol. 10, no. 19, p. 3669, Oct. 2022, [doi: 10.3390/math10193669](https://doi.org/10.3390/math10193669)
- [14] P. Malarvizhi and G. Kavithaa, "Identifying Stable Path in Self Configured Networks Using Particle Swarm Optimization with Centrality Betweenness," *Journal of Electrical Engineering & Technology*, vol. 19, no. 4, pp. 2701–2713, May 2024, [doi: 10.1007/s42835-023-01741-3](https://doi.org/10.1007/s42835-023-01741-3)
- [15] A. Omidvar and K. Mohammadi, "Particle swarm optimization in intelligent routing of delay-tolerant network routing," *EURASIP J. Wirel. Commun. Netw.*, vol. 2014, no. 1, p. 147, Dec. 2014, [doi: 10.1186/1687-1499-2014-147](https://doi.org/10.1186/1687-1499-2014-147)
- [16] Z. Zhang et al., "A Particle Swarm Optimization-Based Queue Scheduling and Optimization Mechanism for Large-Scale Low-Earth-Orbit Satellite Communication Networks," *Sensors*, vol. 25, no. 4, p. 1069, Feb. 2025, [doi: 10.3390/s25041069](https://doi.org/10.3390/s25041069)
- [17] R. F. Ghani and L. Al-Jobouri, "Packet Loss Optimization in Router Forwarding Tasks Based on the Particle Swarm Algorithm," *Electronics (Basel)*, vol. 12, no. 2, p. 462, Jan. 2023, [doi: 10.3390/electronics12020462](https://doi.org/10.3390/electronics12020462)
- [18] E. M. Roopa Devi, T. Hemalatha, D. Usha, and A. K. Nanda, "An optimal multipath routing protocol using hybrid gravitational search particle swarm optimization for secure communication," *International Journal of Communication Systems*, vol. 37, no. 7, May 2024, [doi: 10.1002/dac.5731](https://doi.org/10.1002/dac.5731)
- [19] X. Sun, Z. Jia, M. Zhao, and Z. Zhang, "Multipath Load Balancing in SDN/OSPF Hybrid Network," in *Network and Parallel Computing*, G. R. Gao, D. Qian, X. Gao, B. Chapman, and W. Chen, Eds., Cham: Springer International Publishing, 2016, pp. 93–100, <https://doi.org/10.1007/978-3-319-47099-3-8>
- [20] Yi-Chih Lei, Kuochen Wang, and Yi-Huai Hsu, "Multipath routing in SDN-based Data Center Networks," in *2015 European Conference on Networks and Communications (EuCNC)*, IEEE, Jun. 2015, pp. 365–369. [doi: 10.1109/EuCNC.2015.7194100](https://doi.org/10.1109/EuCNC.2015.7194100)
- [21] Y. Kulakov, A. Kohan, and Y. Hrabovenko, "Multipath Routing in Intelligent Transport Networks," in *Advances in Artificial Systems for Logistics Engineering*, Z. Hu, Q. Zhang, S. Petoukhov, and M. He, Eds., Cham: Springer International Publishing, 2022, pp. 81–90, https://doi.org/10.1007/978-3-031-04809-8_7
- [22] V. Sai R., P. U., S. Ch., S. K. Panda, and G. V., "An Enhanced Dynamic Multi-Path Routing Algorithm for Software Defined Networks," *Procedia Comput. Sci.*, vol. 233, pp. 12–21, 2024, [doi: 10.1016/j.procs.2024.03.191](https://doi.org/10.1016/j.procs.2024.03.191)
- [23] H. Hassen, S. Meherzi, and Z. Ben Jemaa, "Improved Exploration Strategy for Q-Learning Based Multipath Routing in SDN Networks," *Journal of Network and Systems Management*, vol. 32, no. 2, p. 25, Apr. 2024, [doi: 10.1007/s10922-024-09804-0](https://doi.org/10.1007/s10922-024-09804-0)
- [24] T. Badageri, B. Hamdaoui, and R. Langar, "Load-Balanced Multipath Routing Through Software-Defined Networking," in *2024 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, May 2024, pp. 1068–1073. [doi: 10.1109/IWCMC61514.2024.10592316](https://doi.org/10.1109/IWCMC61514.2024.10592316)
- [25] K. Rajkumar, M. Ramachandran, F. Al-Turjman, and R. Patan, "A reinforcement learning optimization for future smart cities

- using software defined networking,” *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3221–3233, Nov. 2021, [doi: 10.1007/s13042-020-01245-w](https://doi.org/10.1007/s13042-020-01245-w).
- [26] M. Jenabzadeh, V. Ayatollahitafti, M. Mollakhalili Meybodi, M. M. Ardakani, and A. Javadpour, “VNR_LBP: A New Approach to Congestion Control Using Virtualization and Switch Migration in SDN,” *Wirel. Pers. Commun.*, vol. 139, no. 2, pp. 733–757, Nov. 2024, [doi: 10.1007/s11277-024-11492-7](https://doi.org/10.1007/s11277-024-11492-7).
- [27] T. Ma *et al.*, “A hybrid congestion control scheme for Named Data Networking via SDN,” *Discover Applied Sciences*, vol. 6, no. 5, p. 246, May 2024, [doi: 10.1007/s42452-024-05903-4](https://doi.org/10.1007/s42452-024-05903-4).
- [28] M. A. Gunavathie and S. Umamaheswari, “Traffic-aware optimal routing in software defined networks by predicting traffic using neural network,” *Expert Syst. Appl.*, vol. 239, p. 122415, Apr. 2024, [doi: 10.1016/j.eswa.2023.122415](https://doi.org/10.1016/j.eswa.2023.122415).
- [29] Y. Wang, M. Othman, W. O. Choo, R. Liu, and X. Wang, “DFRDRL: a dynamic fuzzy routing algorithm based on deep reinforcement learning with guaranteed latency and bandwidth for software-defined networks,” *J. Big Data*, vol. 11, no. 1, p. 150, Oct. 2024, [doi: 10.1186/s40537-024-01029-x](https://doi.org/10.1186/s40537-024-01029-x).
- [30] C. Gao, H. Wang, F. Zhu, L. Zhai, and S. Yi, “A Particle Swarm Optimization Algorithm for Controller Placement Problem in Software Defined Network,” in *Algorithms and Architectures for Parallel Processing*, G. Wang, A. Zomaya, G. Martinez, and K. Li, Eds., Cham: Springer International Publishing, 2015, pp. 44–54, https://doi.org/10.1007/978-3-319-27137-8_4.
- [31] Q. Zhang *et al.*, “A new quantum particle swarm optimization algorithm for controller placement problem in software-defined networking,” *Computers and Electrical Engineering*, vol. 95, p. 107456, Oct. 2021, [doi: 10.1016/j.compeleceng.2021.107456](https://doi.org/10.1016/j.compeleceng.2021.107456).
- [32] J. Chen, W. Xiao, H. Zhang, J. Zuo, and X. Li, “Dynamic routing optimization in software-defined networking based on a metaheuristic algorithm,” *Journal of Cloud Computing*, vol. 13, no. 1, p. 41, Feb. 2024, [doi: 10.1186/s13677-024-00603-1](https://doi.org/10.1186/s13677-024-00603-1).
- [33] Z. Duliński, G. Rzym, and P. Cholda, “MPLS-based reduction of flow table entries in SDN switches supporting multipath transmission,” *Comput. Commun.*, vol. 151, pp. 365–385, Feb. 2020, [doi: 10.1016/j.comcom.2019.12.052](https://doi.org/10.1016/j.comcom.2019.12.052).
- [34] L. Zhang, Y. Lu, D. Zhang, H. Cheng, and P. Dong, “DSOQR: Deep Reinforcement Learning for Online QoS Routing in SDN-Based Networks,” *Security and Communication Networks*, vol. 2022, pp. 1–11, Nov. 2022, [doi: 10.1155/2022/4457645](https://doi.org/10.1155/2022/4457645).
- [35] Y. Lu, Y. Chen, X. Xu, Q. Fu, J. Chen, and L. Liu, “A Sub-flow Adaptive Multipath Routing Algorithm for Data Centre Network,” *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 25, Mar. 2023, [doi: 10.1007/s44196-023-00199-5](https://doi.org/10.1007/s44196-023-00199-5).
- [36] J. Zhang and R. O. Yahya, “DRL-based routing algorithm with guaranteed loss, latency and bandwidth in SDN networks: Application of online video conferencing,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 10, p. 101805, Dec. 2023, [doi: 10.1016/j.jksuci.2023.101805](https://doi.org/10.1016/j.jksuci.2023.101805).