Journal of Computer Science

Research Article

Prioritized Experience Replay-Based Deep Deterministic Policy Gradient for Reliable Path Selection in SDN-IoT Networks

Gaurav Kumar[1], G. S. Girisha[1] and N. Shamanth[2]

[1]*Department of Computer Science and Engineering, School of Engineering, Dayananda Sagar University, Bengaluru, India*
[2]*Department of Computer Science and Engineering, School of Engineering and Technology, Christ (Deemed to be University), Bengaluru, India*

Article history
Received: 07-08-2025
Revised: 02-04-2026
Accepted: 10-07-2026

Corresponding Author:
Gaurav Kumar
Department of Computer Science and Engineering, School of Engineering, Dayananda Sagar University, Bengaluru, India
Email: gauravkumar.res-soe-cse@dsu.edu.in

Abstract: Routing optimization is becoming prominent in Software-Defined Networks (SDN) due to the exponential growth of network traffic demands and the requirement for Quality of Service (QoS). However, reliable routing that satisfies the QoS requirements, such as end-to-end delay, packet loss, and bandwidth, remains a difficult task in SDN. To overcome this limitation, a Deep Reinforcement Learning (DRL)-based Prioritized Experience Replay-based Deep Deterministic Policy Gradient (PER-DDPG) model is proposed to enhance the routing performance in SDN with Internet of Things (SDN-IoT) with QoS requirements. Initially, requests are received and prioritized using the postponement strategy technique in the SDN controller, and the weights of the links are evaluated using the DRL method. Then, a routing path is identified by the routing algorithm, and requests in the queue are released using the time-strategy technique. Hence, reliable routing in an SDN with QoS requirements is accomplished. The proposed routing model based on DRL is evaluated by utilizing the end-to-end latency, throughput, and packet loss.

Keywords: Deep Reinforcement Learning, Quality of Service, Routing, Software-Defined Network

Introduction

The Internet of Things (IoT) significantly advances device intelligence and allows devices to exchange data automatically without human involvement. Currently, there is rapid growth in the IoT devices that are used across various applications (Paganraj et al., 2024). However, these devices often face limitations such as restricted communication range and limited processing power (Gupta et al., 2025). As IoT technology evolves, the present Internet infrastructure should satisfy a wide range of Quality of Service (QoS) demands to ensure efficient data transmission. Agricultural sensors and drones are applications that require high bandwidth and low latency for better data transmission (She et al., 2025). To achieve these constraints, Software-Defined Networking (SDN) is developed as a promising solution that comprises control and data planes. SDN enables centralized and intelligent management to monitor and control an entire network for better decision-making (Xu et al., 2024; ul Haq et al., 2025; Sharathkumar and Sreenath, 2025). The integration of SDN with IoT is known as SDN-IoT, which signifies a transformative change in the networking paradigms. The architecture of the SDN-IoT network involves three main planes: Sensing, data, and control (Arif et al., 2024; Sanchez, 2025).

In distributed SDNs, the deployment of multiple controllers provides several advantages but also introduces challenges such as scalability and reliability issues. Some of the key concerns in the SDN-IoT environment are ensuring QoS parameters and achieving an efficient network with load balancing. However, existing research utilizes conventional models that focus more on QoS parameters, and load balancing remains a challenge in SDN-IoT environments (Majdoub et al., 2024). To address these issues, Deep Reinforcement Learning (DRL) models have been utilized recently, which can learn dynamically and adapt the optimal routing techniques according to the changing network traffic patterns (Khan et al., 2025; Ye et al., 2024).

Recently, researchers have explored the combination of reinforcement learning algorithms with SDN to design energy-efficient and multi-objective routing protocols for path selection in IoT environments (Godfrey et al., 2023). However, conventional DRL-based routing approaches struggle to adapt to frequently changing network conditions, leading to overload issues and inefficient data transmissions.

Problem Statement and Objective

Conventional IoT routing protocols fail to adapt dynamically to changing network conditions, leading to compromised QoS and reliability. Current approaches do not adequately address application-specific QoS requirements and link reliability, which are essential for meeting critical IoT applications. Several existing frameworks fail to account for the dynamic network statistics and limit their adaptability. To develop an adaptive IoT routing mechanism, a Deep Deterministic Policy Gradient (DDPG) enhanced with a Prioritized Experience Replay (PER) is proposed in this research to enable intelligent routing decisions. It enhances the routing optimization in SDN based on real-time network dynamics and application-specific QoS requirements. To enhance the reliability and adaptability of routing in dynamic IoT environments, significant experiences during learning are prioritized, thereby optimizing path selection and ensuring stable QoS is essential for mission-critical IoT applications.

Contribution

The key contributions of this research are the following:

- The proposed DDPG-based model continuously learns from the environment by dynamically adjusting routes based on changing network states
- By using a PER, the model learns faster and more effectively by assigning importance to high-impact routing experiences

Literature Review

Jisi et al. (2024) introduced Reliable Routing based on Reinforcement Learning in Software-defined IoT Networks (RRSN). The introduced effective routing model controlled the SDN by employing a hybrid mechanism, which was a combination of RL and Machine Learning (ML) in the SDN controller to improve stable routing. The presented RRSN approach comprised three main modules: Support Vector Regression (SVR), a traffic classifier, and a reliable RL algorithm. Link reliability was an advantage of the introduced RL model, which assisted the agent in finding a reliable path between sender and receiver nodes in the SDN-IoT network. However, the online classifier-based flow classification method in the RRSN model was able to identify only traffic conditions based on past traffic patterns that created bottlenecks and led to unreliable routing.

Huang et al. (2024) developed a flow placement in an SDN-IoT network using a Q-learning approach based on mobility and cost-aware constraints. The developed model, namely Mobility Aware flow rule Placement with Q-learning (MAPQ), used a cost-sensitive Adaboost model to categorize the data flow into high- and low-impact for route optimization. Moreover, the Q-learning algorithm was used to predict the next possible location of end devices, which resulted in adapting the flow rule dynamically. The main advantage of the developed MAPQ model was that it efficiently handled the mobility of IoT devices. However, the MAPQ model faced difficulties adapting to sudden behavioral changes because the Q-learning model heavily relied on past flow patterns.

Prabha et al. (2024) explored a heterogeneous context-aware Graph Convolutional Network (GCN) for multipath routing with cluster head prediction in SDN-enabled IoT networks. Initially, the cluster heads were selected by the explored heterogeneous GCN model, which was combined with the Coati optimization. Then, an Eagle Swarm Optimization (ESO) algorithm was utilized for routing optimization in an SDN-IoT network based on selected cluster heads. However, the explored GCN with an ESO-based hybrid multipath routing model faced challenges in routing optimization due to its sensitivity to dynamic network changes.

Islam et al. (2023) implemented a routing strategy using a Graph Neural Network (GNN) and RL in an SDN. The queuing model was used to separate Event-Driven (ED) and Fixed Scheduling (FS) packets on Open Flow (OF) switches based on the priority of queues to reduce competition between various types of data packets in the network. The developed GNN model was used to determine the routing decision by predicting the future network conditions, which minimized the additional delays required to forward the rules on switches. However, when the nodes failed, the implemented GNN and RL models required retraining, which produced an additional delay in routing.

Kim et al. (2022) presented a DRL-based routing model for SDN. In the presented DRL model, the agent was trained to determine the best set of link weights to reduce the packet losses and network delay. To solve the degradation problem in DRL, an M/M/1/K queue-based approach was implemented to perform the learning process offline. An Aggregated Traffic Volume Matrix (ATVM) was introduced to enhance the routing performance, and a DDPG was employed to estimate link weights automatically. However, there was an increase in delay and packet loss during retraining in three topologies that affected reliable routing.

Li et al. (2023) introduced a routing algorithm that depended on DRL and a deep learning model. To enhance the various traffic conditions of SDN, a Graph Convolutional Network with a Gated Recurrent Unit (GCN-GRU) was utilized as a traffic prediction mechanism for analyzing the future traffic trends. Also, Proximal Policy Optimization (PPO) was employed to forward the data effectively in the knowledge plane. However, the presented GCN-GRU-based routing optimization method with a random threshold affected the assignment priority to the requests in the routing of SDN-IoT.

Wu and Zhu (2025) designed an intelligent routing optimization model based on a Proximal Policy Optimization-based Reinforcement learning (PPO-R) model, which was combined with a Graph Neural Network (GNN). The designed PPO-R and GNN-based routing model created several individual paths for data transmission by developing a redundant tree algorithm. Moreover, the designed routing optimization model employed in SDN improved routing generalization in high-dynamic SDN networks. However, the designed PPO-R and GNN model faced struggles in large-scale environments.

Wu et al. (2026) presented a DRL-based routing optimization with Betweenness Centrality Theory (BCT) in SDN. The presented DRL model, namely the distributed PPO algorithm, was utilized with BCT for controlled node selection, which helped in identifying reliable paths to achieve dynamic routing. The selection of controlled nodes by BCT reduced the control burden on agents and limited topology changes. However, the presented DRL-based routing optimization model with BCT faced limitations with high packet loss, which affected reliable routing.

From the above literature review, the limitations of existing routing optimization models are summarized as follows: The existing path selection and routing optimization models, such as hybrid architectures, combine DRL algorithms with GNN models to enhance QoS-based routing and link reliability. However, these models improve path selection but often struggle with scalability and high computational overhead in large-scale environments. Additionally, the model limitations lead to increased delay and subsequent packet loss that occurs during node failures and changes in network topology. Moreover, the recently developed routing optimization models heavily rely on simplistic threshold-based prioritization, which highlights a need for a robust and adaptive routing mechanism that maintains stability. Thus, a DRL model, namely DDPG enhanced with a PER, is proposed to select reliable paths with minimum delay and packet loss in the SDN-IoT network.

Materials and Methods

In this section, an overview of the proposed Reliable Routing in SDN-IoT Network (RRSIN) framework (Jisi et al., 2024) and main components is described. Fig. 1 presents the architecture of the SDN-IoT network. The proposed QoS-aware reliable path prediction is developed based on the PER-DDPG algorithm. Here, an RL-based model is proposed to identify a reliable routing path based on SDN core features such as centralized control and programmability. These features are integrated into various modules using software, which improves the effectiveness of the proposed RL-based routing algorithm.

In addition, the proposed RL algorithm uses several QoS factors such as latency, packet loss ratio, and bandwidth utilization to select an optimal route for data transmission. Also, it incorporates reliability predictions generated by an SVR model. This comprehensive state information enables the RRSIN model to dynamically adapt and modify routing paths, even under rapidly changing network conditions, which are organized into four distinct layers: Data plane, control plane, IoT device plane, and application plane.

IoT Plane

This plane forms the primary layer of the RRSIN architecture, which consists of smart devices, sensors, and actuators that are part of the network. These devices can collect data and perform specific actions in an IoT environment. Initially, the data are collected and transmitted across the network and shared through other connected devices for various applications. Smart wearable devices are an example of IoT networks that include sensors to monitor body temperature and heart rate at regular intervals. These smart devices depend on built-in sensors to capture information from the surroundings. For communication between the IoT device plane and the data plane, an edge gateway acts as a bridge between these two planes in the SDN-IoT environment.

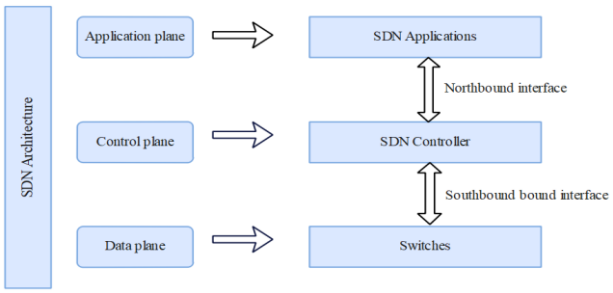

Fig. 1: SDN architecture

Data Plane

A data plane includes forwarding devices, such as OF switches, which cannot make routing decisions on their own. Instead of making a decision, these switches rely on an SDN controller to install flow rules. These OF switches forward data packets to physical ports according

to the flow tables defined by the SDN controller. In addition, when there is no action based on the predefined rule in the flow, data packets are dropped.

Control Plane

This plane is often known as the "brain of the network," which is managed by the SDN controller in this combined SDN-IoT network. The control plane interconnects with the data and application plane through southbound and northbound interfaces. By constantly monitoring network activity, the controller maintains a centralized view of the data plane topology, which allows it to generate optimal routing. The control plane adopts the dynamic nature of the IoT network and helps to generate optimal paths. This control plane comprises five key modules: Network Monitor (NM), traffic classifier, SVR, flow modification & update, and topology discovery.

SVR: This module (Liao et al., 2024) is trained on historical link failure data and predicts the reliability of the link in the data plane. During execution, it forwards the output to the data processing section.

TClassifier: In this classifier, an online framework is utilized to categorize incoming traffic in real time by efficiently extracting relevant features and minimizing delay for processing data packets.

Topology Discovery: This module utilizes the Link Layer Discovery Protocol (LLDP) to maintain a global view of network topology. This topology discovery process transmits LLDP packets to the OF switches, which forward the packets to the neighboring switches. When the neighboring switches receive the LLDP packets, they return the information as packet-in messages, which helps the module identify devices that are connected directly and build the network topology.

NM module: This module is known as an Intelligent Network Information Monitoring (INIM) module that monitors the status of the network frequently using three parameters: Topology, traffic, and performance. The NM module transmits flow_statistics_request messages to OF switches that respond to flow_statistics_reply messages, enabling regular updates of statistical data. The INIM also acquires data from the SVR module, topology discovery modules, and the classifier to provide a comprehensive view of the network. The collected data, including link reliability and performance metrics, are used to generate routing paths and define the state space for the agent of the RRSIN, with the data of paths stored in each pair of sources and destinations.

Flow modification and updating: In this process, an optimal path is created proactively using the developed agent's learned policy and installs flow entries on OF switches via southbound interfaces.

Application Plane

A Q-learning agent is integrated into the reliable path prediction model at the application plane and is used to learn optimal routing policies. The proposed model identifies the routes by interacting with the information obtained from the control plane to process it for state-space information. This plane selects actions based on state action Q-values to estimate rewards and determine optimal routing paths, which are then used to install flow entries on appropriate OF switches.

Reliability Prediction

The reliability prediction in the SDN-IoT network plays a key role in identifying optimal paths and enhancing QoS, which is closely linked to the quality of communication between nodes, i.e., OF switches. However, link reliability is often unpredictable and varies over time due to environmental factors, such as physical obstructions, node movement, electromagnetic interference, hardware noise, multipath propagation, and signal fading. To address these issues, the SVR algorithm is used to predict data reliability in SDN-IoT networks, which is mathematically represented by Eq. 1:

$$G = (V, E, g, \Re) \tag{1}$$

The graph G in Eq. 1 represents the physical topology of the SDN-IoT network, where V denotes a set of OF switches, $V = (v_1, v_2, \ldots, v_n)$, and E refers to the set of communication links. The link reliability prediction is performed using the SVR algorithm, where the graph parameters are flattened into a state-space vector for the actor-critic networks, which is mathematically expressed in Eq. 2:

$$f: e_k(v_i, v_j) \rightarrow r_k \in \Re \tag{2}$$

Where r_k indicates the reliability level of the edge within a given time slot t_i. By using a convex optimization algorithm in SVR for reliability prediction, the objective is obtained using Eqs. 3 and 4:

$$minimize \frac{1}{2}\|w^2\|,\ in\ addition \begin{cases} y_k - \langle w, x_k \rangle - b \leq \varepsilon \\ \langle w, x_k \rangle + b - y_k \leq \varepsilon \end{cases} \tag{3}$$

$$min\left[\frac{1}{2}\|w\|^2 + C\sum_1^k(\xi_k + \xi_k^*)\right] \tag{4}$$

Where C is the cost parameter with a value of 1.0 and the margin of error, which is denoted as ε with a value of 0.01, respectively. This parameter identifies the samples present in the margin that contribute to an overall error, which is expressed in Eq. 5:

$$in\ addition \begin{cases} y_k - \langle w, x_k \rangle - b \leq \varepsilon \\ \langle w, x_k \rangle + b - y_k \leq \varepsilon \\ \xi_k, \xi_k \geq 0 \end{cases} \tag{5}$$

Proposed PER-DDPG-Based Path Prediction

The proposed DDPG (Deshpande et al., 2024) is a DRL algorithm based on an Actor and Critic (AC) framework, which is utilized to address problems in continuous action spaces. DDPG combines deep neural networks with deterministic policy gradients, thereby enabling learning of continuous action policies. The complete architecture of the DDPG model is illustrated in Fig. 2. The DDPG algorithm primarily involves two networks: The actor and the critic. An actor network functions as a deterministic policy function, taking the state as the input and producing the related action as the output. A critic network serves as a Q-value function that evaluates the value of the current state and the action.

In the DRL training process, it is necessary to store the input and output data of the network, thereby requiring the establishment of an experience replay buffer to store experience data. When the data are replayed, the agent updates the network parameters according to the previously observed empirical data. The form of the data is (s, a, r, s) _ {t+1}, and all the data in the experience pool are randomly sampled during the update. The reliable routing problem based on QoS parameters is formulated as a learning process of the DDPG algorithm. To compute the optimal paths using a critic and an actor network in an SDN-IoT network is described as follows.

State Space (S): It represents the flow characteristics of an OF switch, which is characterized by the state vector s_t at time t_i, as expressed in Eqs. 6 and 7.

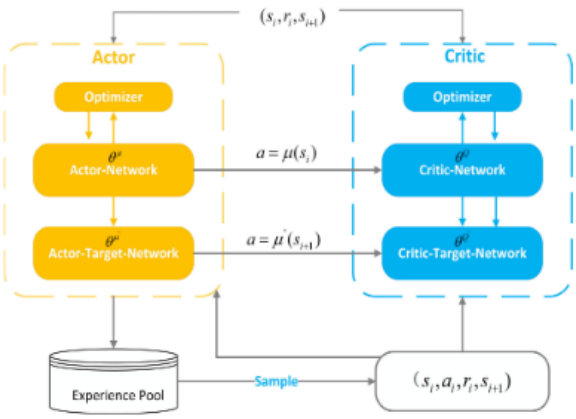

Fig. 2: Architecture of proposed DDPG algorithm

$$s_t = \{f_{ij}(t_i) | e_{ij} \in C\} \quad (6)$$

$$f_{ij}(t_i) = [d_{ij}(t_i), l_{ij}(t_i), b_{ij}(t_i), u_{ij}(t_i), R_{ij}(t_i)] \quad (7)$$

Where $d_{ij}(t_i), l_{ij}(t_i), b_{ij}(t_i), u_{ij}(t_i), R_{ij}(t_i)$ denote individual delays, packet loss ratio, bandwidth, resource utilization, and SVR predicted reliability of the link e_{ij} at the time slot t_i for all edges E. The overall state vector is denoted as $s_t \in R^M$ where M represents the number of links. Subsequently, the action space a_t time t_i is generated by the actor network and is mathematically formulated as in Eq. 8:

$$a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t \quad (8)$$

Where a_t represents continuous link weight assignments, where $a_t \in [0,1]^M$; $\mathcal{N}_t$ indicates exploration noise. These weights are then applied to compute the routing path. Subsequently, the reward function r_t is mathematically represented in Eq. 9:

$$r_t = \lambda_1 T_t - \lambda_2 d_t - \lambda_3 PLR_t + \lambda_1 R_t \quad (9)$$

Where T_t, d_t, PLR_t, and R_t indicate throughput, end-to-end delay, Packet Loss Ratio (PLR), and average path reliability, respectively. λ_i denotes weighting coefficients. The environment then transitions to the next state s_{t+1}, which forms a tuple (s_t, a_t, r_t, s_{t+1}), which is used for training the actor-critic network in the DDPG algorithm.

The replay buffer stores transitions of the form (s_t, a_t, r_t, s_{t+1}), where the state s_t is a fixed-dimensional vector constructed from link-level features. For a network with M links, each link contributes five features: Delay, packet loss ratio, bandwidth utilization, link utilization, and SVR-predicted reliability, resulting in a state dimensionality of $s_t \in R^{5M}$. For example, in a topology with 40 links, the state dimension is 200. Before training, all features are normalized using min–max normalization to the range [0,1] to ensure numerical stability and prevent feature dominance during gradient updates. The action $a_t \in [0,1]^M$ corresponds to continuous link weight assignments produced by the actor network. This explicit state formulation ensures consistency in replay buffer storage and enables efficient learning in high-dimensional network environments.

Priority Experience Replay

This experience replay is proposed to prioritize tasks to avoid overloading issues; when the value of tasks is high, then it has a greater probability of extraction, whereas when it is of lower value, the probability of extraction is low. The main objective of the prioritized experience replay mechanism is to replay less rewarding experiences at a high frequency. Moreover, the experience samples in the replay mechanism are selected using a sampling probability technique that has higher learning values. The probability of selecting an experience sample j is determined using the following Eq. 10:

$$P(j) = \frac{p_j^{\alpha}}{\sum_i p_i^{\alpha}, \alpha \in [0,1]} \quad (10)$$

Where α is a priority adjustment parameter, $\alpha \in [0,1]$, and p_j is a priority index. To maintain sample diversity, random factors are considered when selecting experience samples, which means that even experience samples with small error values have the possibility of being selected. When α is 1, the error value is directly used; if α is 0, then the actual random sampling is considered uniform. The priority indicator is based on the ranking approach given in Eqs. 11 and 12:

$$P(j) = \frac{p_j^{\alpha}}{\sum_i p_i^{\alpha}, \alpha \in [0,1]} \quad (11)$$

$$P(j) = \frac{1}{rank\ (j)}, p_j > 0 \quad (12)$$

Where $P(j)$ denotes the priority index, α represents the priority adjustment factor, and the agent is updated with experience samples that modify the original probability distribution and introduce errors (estimation bias) into the model. Consequently, the model fails to converge during the neural network training. To mitigate this issue, importance sampling weights are employed to correct the weight changes, which are mathematically represented as in Eq. (13):

$$W_j = \left(\frac{1}{m.P_j}\right)^{\beta} \quad (13)$$

From the above Eq. 9, m represents the number of experience replay pools, where the experience replay buffer is established with a total capacity of 100,000 experience samples (M = 100,000) to ensure a diverse range of historical data for the learning process. The parameter β denotes the degree of correction error, which ranges from $[0.4 \rightarrow 1.0]$. The sampling weights W_j are calculated based on experience replay pools and the correction error degree, respectively. According to the above process, the data that interacts with the environment distinguishes the importance of the experience sample and enhances the learning efficiency of the experience sample. Based on this, the priority of the tasks is considered and assigned to efficient resources, which helps the cloud system prevent overload. Utilizing this prioritizing scheme with the DDPG algorithm enhances the load-balancing efficiency in a cloud environment.

Based on the time-slot interaction, the network parameters are updated using the actor-critic framework. This framework is trained by minimizing the Temporal Difference (TD) loss function, which is mathematically represented in Eq. (14):

$$L = \frac{1}{B}\sum_{i=1}^{B}\left(r_i + \gamma Q'\left(s_{i+1}, \mu'(s_{i+1})\right) - Q(s_i, a_i)\right)^2 \quad (14)$$

Where L indicates the loss function, B represents the batch size (number of samples drawn from the replay buffer for one training iteration), i denotes the index of each sample in the mini-batch, and r_i represents an immediate reward received after taking action and a state. γ indicates a discount factor that ranges between $(0 \leq \gamma \leq 1)$, determining the importance of future rewards; s_i, a_i, and s_{i+1} represent the current state, the action state, and the next state, respectively. $Q(s_i,$) ai, and $Q'\left(s_{i+1}, \mu'(s_{i+1})\right)$ denote the predicted Q-value using the current critic network and the target Q-value at the target critic network, and $\mu'(s_{i+1})$ indicates the action generated by the target actor network for the next state. The critic parameters are updated as in Eq. (15):

$$\theta^Q \leftarrow \theta^Q - \eta_Q \nabla_{\theta Q} L \quad (15)$$

Where θ^Q represents parameters (weights and biases) of the critic network, $Q \leftarrow$ denotes the update operator, which updates the parameters iteratively, η_Q indicates the learning rate, and $\nabla_{\theta Q} L$ represents the gradient of the loss function L with respect to the critic parameters. By using this equation, the parameters of the actor network are updated by moving in the opposite direction of the gradient, which minimizes the loss. The learning rate η_Q controls the speed of convergence, which ensures stable and efficient training of the Q-function. The actor network is updated using the deterministic policy gradient in Eqs. 16 and 17:

$$\nabla_{\theta\mu} J \approx \frac{1}{B}\sum_{i=1}^{B} \nabla_a Q(s, a|\theta^Q)|_{a=\mu(s)} \nabla_{\theta\mu}\mu(s) \quad (16)$$

$$\theta^{\mu} \leftarrow \theta^{\mu} - \eta_{\mu} \nabla_{\theta^{\mu}} J \quad (17)$$

Where $\nabla_{\theta\mu} J$ denotes the gradient of the objective function J with respect to the actor parameters, $\mu(s)$ represents the actor network, and θ^{μ} represents the parameters (weights and biases) of the actor network. $\nabla_a Q(s, a|\theta^Q)|_{a=\mu(s)}$ indicates the gradient of the critical output, $\nabla_{\theta\mu}\mu(s)$ denotes the gradient of the actor output, and θ^{μ} indicates parameters of the actor network. Finally, the target networks are updated using soft updates, which is given in Eq. 18:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \quad (18)$$

Where θ' denotes parameters of the target network, θ represents parameters of the main network, τ indicates the soft update coefficient, and $(1-\tau)$ denotes the retention factor that preserves most of the previous target parameters.

The weighting coefficients λ_i in the reward function and the weight updates in the DDPG algorithm serve fundamentally different purposes and operate at different

stages of learning. The coefficients λ_1, λ_2, λ_3, and λ_4 are used to construct the scalar reward r_t as a weighted combination of multiple QoS objectives, thereby defining the optimization objective of the agent. In contrast, the weight update equations refer to the optimization of neural network parameters θ^{μ} (actor) and θ^{Q} (critic) using gradient descent. These updates depend on the reward r_t but are not directly controlled by the λ_i coefficients. Instead, the influence of λ_i is propagated indirectly through the Temporal Difference (TD)-error term$(r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1})) - Q(s_t, a_t))$, which drives the parameter updates. Thus, λ_i shapes the learning objective, while the weight update equations define how the model parameters are optimized to achieve that objective. Also, it is important to note that the loss is further weighted using importance sampling weights to correct bias introduced by prioritized sampling.

The proposed framework consists of an SVR module in the control plane and the PER-DDPG agent in the application plane in the SDN layers to ensure reliable routing. The SVR module is used to efficiently extract historical link statistics, which enhances the state representation with predictive information. Subsequently, the PER-DDPG agent utilizes the predicted information, which is combined with the current traffic matrix to create the state representation, where the actor network generates a set of continuous link weight values as an action, which denotes the routing path. Simultaneously, the critic network evaluates these actions by calculating a reward based on the throughput, latency, and packet loss ratio. This actor–critic mechanism is more suitable than discrete Q-learning for SDN-IoT routing because link weight optimization naturally lies in a continuous action space. The replay buffer stores normalized state–action–reward transitions to stabilize training and ensure convergence under dynamic network conditions. The proposed path selection framework for SDN-IoT network ensures architectural consistency while enabling adaptive and reliable routing in large-scale environments.

Results and Discussion

An experimental analysis of the PER-DDPG algorithm-based efficient routing in an SDN-IoT environment is evaluated and discussed in this section.

Experimental Setup

The proposed PER-DDPG model is implemented in Python 3.9 using TensorFlow and evaluated in a Mininet-based SDN simulation environment with an OF controller. Network topologies are generated using random and mesh configurations with node sizes varying from 20 to 100 nodes, depending on the experiment. Each link is assigned capacities between 10 and 100 Mbps with First-In-First-Out (FIFO) queue discipline and drop-tail buffering. Traffic flows follow a Poisson arrival process with exponentially distributed inter-arrival times to emulate dynamic IoT traffic. The training process is conducted for 500 episodes with 200 steps per episode, and target networks are updated using soft updates with τ = 0.005. A fixed random seed is used across all experiments to ensure reproducibility.

For a fair comparison, baseline methods such as Deep Q Network (DQN), Advantage Actor-Critic (A2C), PPO, Trust Region Policy Optimization (TRPO), A3C, and conventional DDPG are implemented under identical network conditions. The experimental configurations of baseline methods are presented in Table 1.

Exploration decay schedules (ε linearly decayed from 1.0 to 0.1 over training for value-based methods and entropy regularization for policy-based methods) and mini-batch sizes (fixed at 64 across all models) are kept consistent to ensure commensurate training effort across methods, ensuring that observed performance gains are due to the prioritized replay mechanism and actor–critic optimization rather than differences in hyperparameter tuning or training budgets. The performance of the PER-DDPG method for routing is evaluated using performance measures, namely energy consumption, delay, Packet Delivery Ratio (PDR), throughput, and PLR.

Quantitative and Qualitative Analysis

The performance evaluation of the proposed efficient routing using the PER-DDPG algorithm is presented in this section. Figs. 3 to 7 depict the performance of the PER-DDPG model, which is evaluated using state-of-the-art methods such as DRL, A3C, and conventional DDPG models. Various performance measures are used to evaluate the PER-DDPG for reliable routing in the SDN-IoT network.

Table 1: Parameter configuration of baseline models

Parameter	DQN	A2C	PPO	TRPO	A3C	DDPG	PER-DDPG (Proposed)
Exploration Strategy	ε-greedy	Entropy	Entropy	Entropy	Entropy	OU Noise	OU Noise
Initial Exploration (ε / Noise)	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Final Exploration	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Replay Buffer Size	100,000	—	—	—	—	100,000	100,000
Mini-Batch Size	64	64	64	64	64	64	64
Learning Rate	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Episodes	500	500	500	500	500	500	500
Steps per Episode	200	200	200	200	200	200	200

The performance of the PER-DDPG model is evaluated based on throughput with different metaheuristic algorithms, as shown in Fig. 3. From the graph, it is clear that PER-DDPG consistently outperforms the other methods and exhibits a steady rise in throughput, indicating its robustness. The TRPO and PPO algorithms also demonstrate good performance with increasing node counts, which suggests better adaptability to larger environments. In contrast, the A2C algorithm presents the lowest throughput owing to limited exploration or suboptimal policy updates, whereas the DQN maintains moderate performance but lacks scalability. Hence, the proposed PER-DDPG model achieves maximum throughput in high-node environments in SDN-IoT.

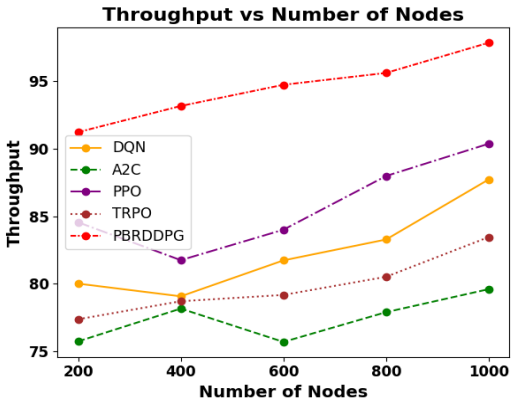

Fig. 3: Performance of the proposed PER-DDPG is based on throughput (%)

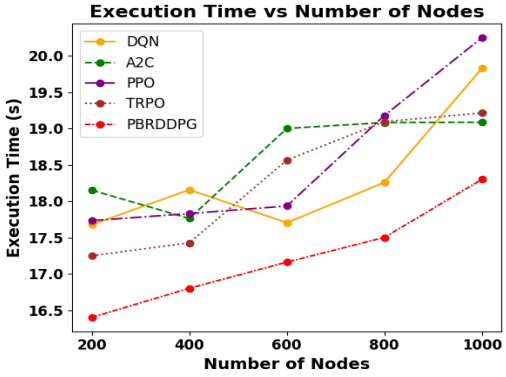

Fig. 4: Performance of the proposed PER-DDPG is based on execution time (s)

The performance analysis of the PER-DDPG model is evaluated based on the execution time with different metaheuristic algorithms, as displayed in Fig. 4.

The PER-DDPG model achieves the lowest execution time, indicating its computational efficiency and suitability for large-scale networks. In contrast, existing routing algorithms, such as TRPO and DQNs, have a higher execution time, and when the number of nodes increases, the execution time also increases, suggesting scalability issues. Moreover, the A2C and PPO algorithms maintain moderate execution times but fluctuate across different node sizes. Overall, the PER-DDPG model balances speed and scalability and makes the routing model ideal for time-sensitive and dense network environments.

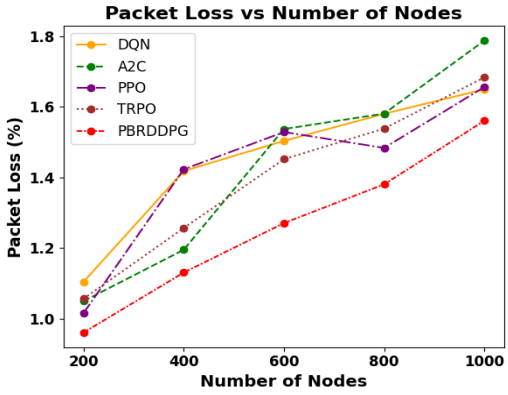

Fig. 5: Performance of the proposed PER-DDPG is based on PLR (%)

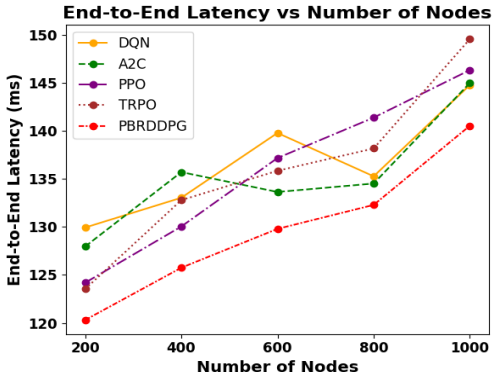

Fig. 6: Performance of the proposed PER-DDPG based on end-to-end latency (ms)

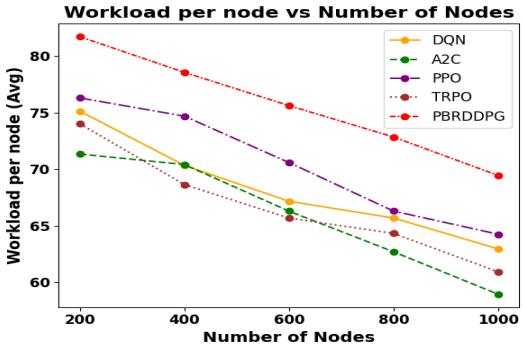

Fig. 7: Performance of proposed PER-DDPG is based on workload per node (Avg)

The performance analysis of the proposed routing model is evaluated based on the PLR with different metaheuristic algorithms, as shown in Fig. 5. The proposed routing model exhibits the lowest packet loss across all node counts, which represents superior network stability and effective congestion control in an SDN-IoT environment that enhances data transmission. Existing algorithms, such as A2C and DQN, experience higher packet loss, and TRPO and PPO models perform moderately but have high PLR owing to network load.

Fig. 6 illustrates the performance of the proposed routing model evaluated based on end-to-end latency with various optimization algorithms in the SDN-IoT environment. From the results, it is clear that the PER-DDPG model has the lowest end-to-end latency due to its efficient decision-making and optimized routing strategies. Existing algorithms, namely TRPO and DQN, depict high latencies when the number of nodes increases because of lower adaptive routing and slower convergence, which leads to inefficient data transmission paths.

Fig. 7 visualizes the performance of the proposed routing model evaluated based on the workload per node using various optimization algorithms in the SDN-IoT environment. The proposed method maintains the highest workload per node because it optimizes resource utilization by concentrating tasks on more capable nodes. The A2C and DQN algorithms indicate lower per-node workloads, possibly because of load balancing without considering the node efficiency.

Comparative Analysis

A comparative study of the proposed PER-DDPG-based path prediction in the SDN-IoT is presented in this section. For comparative analysis, the PER-DDPG model is evaluated using existing routing algorithms, such as the RRSN (Huang et al., 2024), GCN (Prabha et al., 2024), GNN (Gali and Mahamkali, 2025), and DRL (Keshari et al., 2023) models. Table 2 presents a comparative analysis of reliable path prediction in the SDN-IoT network.

Discussion

The proposed PER-DDPG model achieves better results in identifying reliable paths in the SDN-IoT networks. The proposed DDPG-based model continuously learns from the environment by dynamically adjusting routes based on changing network states. By using a PER, the model learns faster and more effectively by assigning importance to high-impact routing experiences. Some of the existing models have limitations, such as the online classifier method in the RRSN model (Jisi et al., 2024), which can only identify existing traffic conditions that create bottlenecks and lead to unreliable routing. The GCN model (Prabha et al., 2024) is insufficient for reliable routing performance requirements in complex and large-scale dynamic networks. An implemented GNN model (Islam et al., 2023) requires retraining of the model and produces an additional delay in routing when the number of nodes increases. The GCN-GRU and PPO (Li et al., 2023) models have limitations, which include high delay and packet loss during retraining of three different network topologies that affect reliable routing performance in SDN-IoT.

When compared to prior DDPG-based SDN research, the proposed PER-DDPG framework is differentiated by enhancing observability through the incorporation of predictive reliability estimates from an SVR module, rather than relying solely on partial observability. Also, the proposed model utilizes a centralized SDN controller to minimize the control latency and soft target updates to avoid significant delays, which are one of the major issues in the existing research that requires full retraining during topology changes. Moreover, by formulating the action space as continuous link-weight assignments, the proposed PER-DDPG model achieves a finer-grained routing optimization, which is more suitable for dynamic environments than the conventional discrete Q-learning approaches.

Table 2: Comparative analysis of the PER-DDPG-based reliable path prediction in the SDN-IoT network

Methods	PLR (%)	End-to-End Latency (ms)	Throughput (%)	Workload per node (Avg)	Execution time (s)
RRSN (Jisi et al., 2024)	2.73	162.84	82.37	68.25	19.57
GCN (Prabha et al., 2024)	3.19	174.91	77.42	65.93	20.18
GNN (Islam et al., 2023)	2.98	157.36	85.74	70.42	18.94
GCN-GRU and PPO (Li et al., 2023)	3.41	181.05	79.85	66.71	20.46
PPO and GNN (Wu and Zhu, 2025)	2.86	148.34	88.55	71.69	18.51
IROD – BC (Wu et al., 2026)	2.49	133.87	90.25	72.97	18.27
Proposed PER-DDPG model	1.26	129.74	94.53	75.64	17.23

Hence, the combination of reliability prediction and adaptive PER-based DDPG training ensures that the model achieves both higher reliability and faster convergence in the SDN-IoT network.

To position the proposed method within the DRL routing taxonomy, the problem is formulated as a Markov Decision Process (MDP) defined by state, action, reward, and environment components. The state

consists of fully observable, controller-level global network statistics, including delay, packet loss ratio, bandwidth utilization, and SVR-predicted link reliability for all links. The action space is continuous and corresponds to link weight assignments generated by the actor network, enabling fine-grained routing optimization. The reward is a weighted multi-objective function combining throughput, latency, packet loss, and reliability. Unlike several prior DDPG-based SDN studies that assume partial observability or rely solely on instantaneous link statistics, the proposed model enhances observability by incorporating predictive reliability estimates from SVR. Furthermore, control latency is minimized through centralized SDN control and soft target updates, whereas existing works typically retrain under topology changes or use static traffic assumptions. Thus, the proposed framework differs in its predictive state augmentation, continuous link-weight control formulation, and adaptive PER-based training strategy under dynamic traffic conditions.

However, the objective of this research is based on references to healthcare and consumer IoT scenarios, but the proposed routing framework operates strictly at the network layer. The primary function of the proposed model is to optimize the packet routing based on the QoS parameters rather than the application layer, which involves user metadata. Additionally, the traffic traces used for training and evaluation are synthetically generated based on statistical flow models and are independent of personal or demographic characteristics. It ensures that no user-level demographic attributes, such as gender, age, or regional identity, are used in this research to maintain privacy. Therefore, the routing decisions, which are made by the proposed model, are solely based on the QoS metrics, including delay, packet loss, bandwidth utilization, and predicted link reliability.

Limitations and Future Work

Moreover, real-world deployments in consumer or healthcare IoT environments provide heterogeneous usage patterns that lead to biased network behaviors by application-level activities. Therefore, the incorporation of diversity-aware traffic modeling to ensure reliable routing remains an important direction for future research to ensure fairness and generalizability in large-scale human-centric deployments.

Conclusion

A DRL-based PER-DDPG model was proposed to enhance the routing performance in SDN-IoT with QoS requirements. The major issues in reliable routing that ensured the QoS requirements were end-to-end delay, packet loss, and bandwidth, which remained a difficult task in SDN because it continuously learned from the environment and dynamically adjusted routes based on changing network states. By using a PER, the model learned faster and more effectively by assigning importance to higher-impact routing experiences. Initially, requests were received and prioritized using the postponement strategy technique in the SDN controller, and the weights of the links were evaluated using the DRL method. Subsequently, a routing path was identified by the routing algorithm, and the requests in the queue were released using the time-strategy technique. Hence, reliable routing in an SDN with QoS requirements was accomplished. Experimental results of the proposed routing model based on DRL achieved various metrics, such as a throughput of 94.53%, which was higher than the existing routing model in the SDN-IoT network. In the future, an energy-efficient routing model will be implemented to efficiently enhance reliable data transmission in the heterogeneous SDN-IoT.

Notation List

Notations	Description
G	Graph
V	OF switches, $V = (v_1, v_2, \ldots, v_n)$
E	Set of communication links
r_k	Reliability level of the edge
C	Cost parameter with a value of 1.0
ε	Margin of error with a value of 0.01
s_t	State vector
t_i	Time
$d_{ij}(t_i), l_{ij}(t_i), b_{ij}(t_i), u_{ij}(t_i), R_{ij}(t_i)$	Individual delays, packet loss ratio, bandwidth, resource utilization, and SVR predicted reliability of the link e_{ij} at a time slot t_i.
$s_t \in R^M$	Overall state vector
M	Number of links
a_t	Action space
$a_t \in [0,1]^M$	Continuous link weight assignments
$\mathcal{N}_t$	Exploration noise
r_t	Reward function
T_t, d_t, PLR_t, and R_t	Throughput, end-to-end delay, PLR, and average path reliability
λ_i	Weighting coefficients
s_{t+1}	Next state
(s_t, a_t, r_t, s_{t+1})	Tuple
j	Sample
α	Priority adjustment parameter $a \in [0,1]$
p_j	Priority index
α	Priority adjustment factor
m	Number of experience replay pools denotes the degree of correction error, which ranges from$[0.4 \rightarrow 1.0]$
W_j	Sampling weights
L	Loss function

B	Batch size
i	Index of each sample in mini-batch
r_i	Immediate reward received after taking action and state
γ	Discount factor, which ranges between $(0 \leq \gamma \leq 1)$
s_i, a_i and s_{i+1},	Current state, action, and next state
$Q(s_i, a_i)$ and $Q'(s_{i+1}, \mu'(s_{i+1}))$	Predicted Q-value using the current critic network and the target Q-value at the target critic network
$\mu'(s_{i+1})$	Action generated by the target actor network for the next state
$\nabla_{\theta\mu} J$	Gradient of the objective function J with respect to actor parameters
$\mu(s)$	Actor network
θ^{μ}	Parameters (weights and biases) of the actor network
$\nabla_a Q(s, a\|\theta^Q)\|_{a=\mu(s)}$	Gradient of critic output
$\nabla_{\theta\mu}\mu(s)$	Gradient of actor output
θ^{μ}	Parameters of actor network
θ'	Parameters of target network
θ	Parameters of main network
τ	Soft update coefficient
$1-\tau$	Retention factor
$(r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1})) - Q(s_t, a_t))$,	TD error term

Acknowledgment

The authors would like to express their sincere gratitude to Dayananda Sagar University and Christ (Deemed to be University) for providing the necessary infrastructure and support for this research. They also acknowledge the valuable guidance and encouragement received from the faculty members throughout the course of this work. Finally, thank all those who contributed directly or indirectly to the successful completion of this paper.

Funding Information

This research received no external funding.

Author's Contributions

Gaurav Kumar, **G. S. Girisha and N. Shamanth:** Methodology, validation**:** Softwareformal analysis, investigation, resources, data curation, write original draft preparation, writing eview and editing, visualization, supervision, project administration, funding acquisition.

Ethics

This research does not involve any studies with human participants or animals, nor does it include any sensitive data, personal information, or experiments requiring ethical approval. Therefore, no ethical issues are applicable to this work.

References

Arif, F., Zabeehullah, Ali Khan, N., Iqbal, J., Khalid Karim, F., 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

Deshpande, S. V., Harikrishnan, R., Ibrahim, B. S. K. K., & Ponnuru, M. D. S. (2024). Mobile robot path planning using deep deterministic policy gradient with differential gaming (DDPG-DG) exploration. *Cognitive Robotics*, *4*, 156–173. https://doi.org/10.1016/j.cogr.2024.08.002

Godfrey, D., Suh, B., Lim, B. H., Lee, K.-C., & Kim, K.-I. (2023). An Energy-Efficient Routing Protocol with Reinforcement Learning in Software-Defined Wireless Sensor Networks. *Sensors*, *23*(20), 8435. https://doi.org/10.3390/s23208435

Gupta, N. K., Yadav, R. S., & Nagaria, R. K. (2025). Energy efficient anchor zone based routing protocol for IoT networks. *Computers and Electrical Engineering*, *123*, 110144. https://doi.org/10.1016/j.compeleceng.2025.110144

Huang, G., Ullah, I., Huang, H., & Kim, K. T. (2024). Predictive mobility and cost-aware flow placement in SDN-based IoT networks: a Q-learning approach. *Journal of Cloud Computing*, *13*(1), 26. https://doi.org/10.1186/s13677-024-00589-w

Islam, M. A., Ismail, M., Atat, R., Boyaci, O., & Shannigrahi, S. (2023). Software-Defined Network-Based Proactive Routing Strategy in Smart Power Grids Using Graph Neural Network and Reinforcement Learning. *E-Prime - Advances in Electrical Engineering, Electronics and Energy*, *5*, 100187. https://doi.org/10.1016/j.prime.2023.100187

Jisi, C., Roh, B., & Ali, J. (2024). Reliable paths prediction with intelligent data plane monitoring enabled reinforcement learning in SD-IoT. *Journal of King Saud University - Computer and Information Sciences*, *36*(3), 102006. https://doi.org/10.1016/j.jksuci.2024.102006

Khan, N. A., Ud Din, I., Almogren, A., Altameem, A., & Guizani, M. (2025). Secure and Efficient AI-SDN-Based Routing for Healthcare-Consumer Internet of Things. *IEEE Transactions on Consumer Electronics*, *71*(2), 6769–6776. https://doi.org/10.1109/tce.2024.3472071

Kim, G., Kim, Y., & Lim, H. (2022). Deep Reinforcement Learning-Based Routing on Software-Defined Networks. *IEEE Access*, *10*, 18121–18133. https://doi.org/10.1109/access.2022.3151081

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

Liao, Z., Dai, S., & Kuosmanen, T. (2024). Convex support vector regression. *European Journal of Operational Research*, *313*(3), 858–870. https://doi.org/10.1016/j.ejor.2023.05.009

Majdoub, M., Kamel, A. E., & Youssef, H. (2024). QoS-Aware Cross-Domain Routing in SDN: A Comparative Study Between Competitive and Cooperative MARL Approaches. *SN Computer Science*, *5*(8), 979. https://doi.org/10.1007/s42979-024-03314-1

Paganraj, D., Tharun, A., & Mala, C. (2024). Dair-mlt: detection and avoidance of IoT routing attacks using machine learning techniques. *International Journal of Information Technology*, *16*(5), 3255–3263. https://doi.org/10.1007/s41870-024-01794-1

Prabha, R., G. A, S., Bharathi, G. P., & Sridevi, S. (2024). Hybrid Multipath Routing Cluster head prediction based on SDN-enabled IoT and Heterogeneous context-aware graph convolution network. *Peer-to-Peer Networking and Applications*, *17*(4), 2016–2030. https://doi.org/10.1007/s12083-024-01685-z

Sanchez, A., L. P., Shen, Y., & Guo, M. (2025). Mdq: A QOS-congestion-aware deep reinforcement learning approach for multi-path routing in SDN. *Journal of Network and Computer Applications*, *235*, 104082. https://doi.org/10.1016/j.jnca.2024.104082

Sharathkumar, S., & Sreenath, N. (2025). Efficient and interactive fault-tolerant, distributed SDN controller for a secured communication in a software defined network. *Multimedia Tools and Applications*, *84*(29), 36107–36143. https://doi.org/10.1007/s11042-024-20560-w

She, H., Yan, L., Mao, C., Bu, Q., & Guo, Y. (2025). Service-driven dynamic QoS on-demand routing algorithm. *Future Generation Computer Systems*, *166*, 107685. https://doi.org/10.1016/j.future.2024.107685

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

Wu, Z., Tang, Q., Cao, J., Wen, S., & Li, B. (2026). Intelligent routing optimization with deep reinforcement learning and Betweenness Centrality Theory in software-defined networks. *Computer Communications*, *247*, 108401. https://doi.org/10.1016/j.comcom.2025.108401

Xu, J., Wang, Y., Zhang, B., & Ma, J. (2024). A graph-reinforcement learning-based SDN routing path selection for optimizing long-term revenue. *Future Generation Computer Systems*, *150*, 412–423. https://doi.org/10.1016/j.future.2023.09.017

Ye, M., Zhao, C., Wen, P., Wang, Y., Wang, X., & Qiu, H. (2024). DHRL-FNMR: An Intelligent Multicast Routing Approach Based on Deep Hierarchical Reinforcement Learning in SDN. *IEEE Transactions on Network and Service Management*, *21*(5), 5733–5755. https://doi.org/10.1109/tnsm.2024.3402275

ul Haq, Q. M., Arif, F., Khan, N. A., Anwar, M. S., & Alhalabi, W. (2025). A secure AI framework for intelligent traffic prediction and routing in SDN based consumer internet of things. *IEEE Transactions on Consumer Electronics*. https://doi.org/10.1109/TCE.2025.3552609.