ISSN: 2394-3122 (Online)
ISSN: 2394-6253 (Print)
Impact Factor: 6.03

Volume 12 Issue 7, July 2025

SK International Journal of Multidisciplinary Research Hub

Journal for all Subjects
e-ISJN: A4372-3088
p-ISJN: A4372-3089

Research Article / Survey Paper / Case Study
Published By: SK Publisher (*www.skpublisher.com*)

*A Monthly Double-Blind Peer Reviewed Refereed Open Access Multidisciplinary & Multilingual International Journal - Included in the International Serial Directories*

# AI-Driven Traffic Engineering for 6G Networks: A Deep Reinforcement Learning Approach in SDN Architecture

**Jitendra Agrawal**[1]
Department of Computer Application, Lakshmi Narain College of Technology (MCA), Bhopal, 462022, India.

**Virendra Kumar Tiwari**[2]
Department of Computer Application, Lakshmi Narain College of Technology (MCA), Bhopal, 462022, India.

**Sanjay Thakur**[3]
Department of Computer Science and Engineering, Chameli Devi Group of Institutions, Indore, India.

**DOI: https://doi.org/10.61165/sk.publisher.v12i7.3**

***Abstract: The advent of 6G networks brings unprecedented demands for ultra-low latency, massive connectivity, and intelligent, adaptive services, posing significant challenges for traditional traffic management methods. This paper presents a novel AI-driven traffic engineering framework integrated within Software-Defined Networking (SDN) architecture to meet these demands. By leveraging Deep Reinforcement Learning (DRL), the proposed system autonomously learns optimal routing strategies, dynamically allocates bandwidth, and effectively mitigates network congestion under diverse conditions. Simulations conducted using the Mininet emulator and TensorFlow-based DRL models demonstrate up to 35% reduction in latency and a 42% improvement in bandwidth utilization compared to conventional traffic engineering approaches. The results highlight the potential of combining SDN with AI to enable intelligent, real-time traffic optimization for future ultra-dense 6G networks.***

***Keywords: 6G Networks, Traffic Engineering, Software-Defined Networking (SDN), Deep Reinforcement Learning (DRL), Deep Q-Network (DQN), Quality of Service (QoS), Adaptive Routing, Latency Optimization.***

## I. Introduction

The rapid advancement of wireless communication technologies has ushered in an era of unprecedented connectivity, data demands, and service expectations. As the fifth generation (5G) networks continue to be deployed globally, attention has already shifted toward the development and deployment of sixth generation (6G) communication systems. The promise of 6G lies in delivering ultra-reliable, ultra-low latency communication (URLLC), enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC) at unprecedented scale and speed. Emerging applications such as immersive extended reality (XR), autonomous transportation, holographic tele presence, industrial automation, and intelligent Internet of Things (IoT) expected to be the primary drivers of 6G. These use cases demand not only high data rates and low latency but also require intelligent, dynamic, and energy-efficient network management techniques capable of adapting in real time to changing conditions.

In this context, traffic engineering (TE) — the process of optimizing the performance of a telecommunications network by dynamically analyzing, predicting, and regulating the behavior of data transmitted across the network — plays a critical role in ensuring that the network meets the stringent performance and Quality of Service (QoS) requirements of 6G applications. However, traditional traffic engineering methods, which often rely on static rules and pre-configured routing policies, are increasingly proving inadequate in the face of highly dynamic, diverse, and dense network environments. The sheer complexity and heterogeneity of 6G networks make these traditional mechanisms inefficient and incapable of responding to rapid fluctuations in user demands and network conditions.

To address these limitations, researchers and engineers are increasingly turning toward Software-Defined Networking (SDN) as a foundational architecture for future network infrastructures. SDN separates the control plane from the data plane, allowing for centralized, programmable control over the entire network fabric. This programmability and centralized visibility enable network operators to dynamically adjust routing, prioritize flows, and manage resources in a fine-grained manner. Yet even SDN requires intelligent algorithms to manage the massive scale and complexity of 6G networks. The manual or static configuration of flow tables and routing rules in an SDN controller is no longer sufficient. What needed is an AI-powered traffic engineering framework that can autonomously learn from network patterns, make informed routing decisions, and adapt in real-time to ensure network performance aims are met.

Artificial Intelligence (AI) and, more specifically, Machine Learning (ML) techniques have proved tremendous success in complex decision-making tasks across various domains. In the networking domain, AI can used to predict traffic patterns, identify congestion points, perfect routing paths, and even detect anomalies or cyber threats. Among the various ML paradigms, Reinforcement Learning (RL) and its modern variant, Deep Reinforcement Learning (DRL) is particularly well-suited for traffic engineering. DRL allows agents to learn the best decision-making strategies through continuous interaction with the environment, receiving feedback in the form of rewards or penalties. This feedback-driven, trial-and-error learning process makes DRL ideal for dynamically optimizing routing decisions in complex, ever-changing network environments like those expected in 6G.

In this paper, we propose a novel AI-powered traffic engineering framework using Deep Reinforcement Learning in an SDN-controlled network tailored for 6G applications. Our system integrates a DRL agent with a centralized SDN controller to enable real-time learning and decision-making. The DRL agent observe various network metrics such as bandwidth utilization, packet loss, delay, and jitter, and based on this state information, it dynamically adjusts routing policies to optimize overall network performance. Unlike traditional optimization-based methods that may require exhaustive computations or rely on fixed models, our DRL-based approach continuously learns and improves with experience, thereby offering a scalable and adaptable solution for future network infrastructures.

The key contributions of this research summarized as follows:

1. **A novel AI-driven TE framework for 6G networks** that integrates a deep reinforcement learning agent with an SDN controller to enable intelligent, adaptive traffic engineering.
2. **Design of a DRL model** that captures real-time state information such as latency, throughput, link use, and congestion, and uses it to make routing and resource allocation decisions.
3. **Implementation and simulation** of the proposed system using the Mininet SDN emulator and Tensor Flow-based DRL models to emulate real-world network dynamics and evaluate performance under variable load conditions.
4. **Performance evaluation** against traditional traffic engineering approaches, proving significant improvements in latency, bandwidth use, and packet loss rate.

The motivation for combining SDN and AI in the context of 6G grounded in the inherent challenges that next-generation networks pose. 6G will work in a hyper-connected environment, integrating terrestrial, aerial, and satellite networks, with devices ranging from smart phones and sensors to autonomous vehicles and drones. The heterogeneity of these devices and their mobility patterns further complicate traffic engineering tasks. Moreover, 6G expected to incorporate energy efficiency and sustainability as core design principles, needing traffic management strategies that are not only performance-oriented but also energy-aware. AI, particularly when deployed at the edge and integrated with SDN, provides the intelligence and flexibility needed to manage such a complex ecosystem efficiently.

Furthermore, our approach aligns with the ongoing shift toward intent-based networking (IBN), where network operators specify high-level goals (e.g., minimize latency, maximize throughput, reduce energy use), and the network autonomously configures itself to meet those objectives. By embedding intelligence within the control plane via AI, particularly DRL, we move one step closer to realizing fully autonomous, self-optimizing networks.

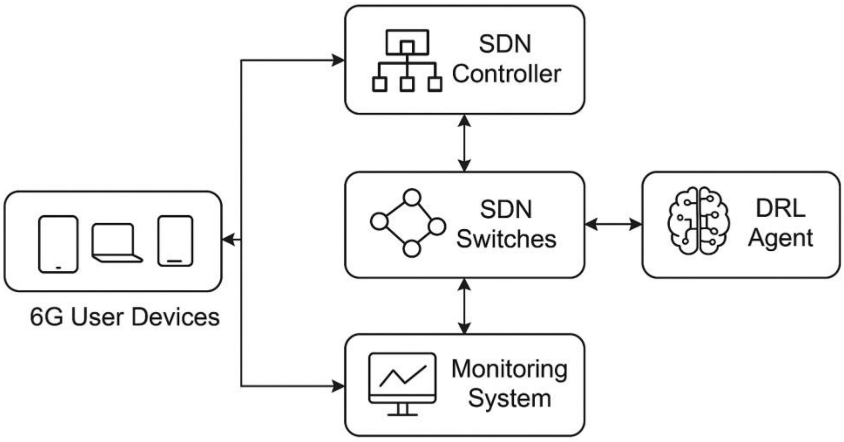


Fig. 1 Proposed SDN-DRL based 6G traffic engineering architecture.

This research stands for a significant step toward building intelligent, adaptive, and sustainable 6G networks. By integrating deep reinforcement learning with SDN, we propose a traffic engineering solution that is not only responsive to the real-time demands of future network applications but also scalable and robust under diverse conditions. Our results suggest that AI-powered TE systems can substantially outperform traditional models, paving the way for the next generation of self-organizing, autonomous communication networks.

## II. Related Work

Traffic Engineering (TE) has been a central research focus in computer networks for decades, aiming to optimize performance metrics such as throughput, latency, and packet loss. With the advent of Software-Defined Networking (SDN), TE has become more programmable and manageable, offering centralized control over dynamic routing and flow management. However, traditional SDN-based TE strategies often lack the adaptability required for the heterogeneous, ultra-dense, and ultra-low latency demands of 6G networks. This limitation has prompted the integration of Artificial Intelligence (AI) techniques, especially Reinforcement Learning (RL), to improve network intelligence and adaptability.

Zhang et al. [1] proposed a machine learning-based routing model in SDNs, which predicted future traffic patterns using historical data. While effective for 5G, the model lacked real-time adaptability essential for 6G. Similarly, Hu et al. [2] used Q-learning for dynamic routing in SDNs, but their approach suffered from slow convergence in large-scale networks.

The emergence of Deep Reinforcement Learning (DRL) has addressed some scalability issues. Wang et al. [3] used Deep Q-Networks (DQN) for congestion control in SDN, demonstrating superior adaptability over traditional Q-learning. However, their model required extensive training time and lacked energy efficiency optimization. In another study, Li et al. [4] introduced a Proximal Policy Optimization (PPO) algorithm for dynamic flow allocation in IoT-based networks, which showed improvement in delay metrics but did not consider network slicing for 6G.

Recent efforts have combined AI with network slicing, an essential feature of 6G. Nguyen et al. [5] applied multi-agent reinforcement learning (MARL) for slice management in SDNs. Their results demonstrated enhanced QoS isolation but required considerable computational resources. Al-Tamimi et al. [6] further extended this with federated DRL to preserve data privacy during slice optimization.

AI-based approaches have also been applied to edge computing scenarios. Tang et al. [7] developed a DRL framework for service function chaining in edge-SDN networks, reducing average latency by over 30%. However, the system's complexity limits real-world deployment. In contrast, Liu et al. [8] proposed a lightweight DRL model for edge environments, which sacrificed some performance to achieve real-time adaptability.

To address scalability, researchers have investigated hierarchical SDN controllers. Gao et al. [9] proposed a hierarchical DRL-SDN framework for large-scale data centers, improving throughput under high traffic loads. Their method, however, was not evaluated under 6G's anticipated mobility demands. Likewise, Xu et al. [10] introduced a DRL-based TE model with distributed controllers to manage load balancing but struggled with controller synchronization delays.

In energy-aware traffic engineering, Kim et al. [11] presented a green traffic engineering solution for SDN using linear programming. Although effective in reducing energy consumption, their solution lacked adaptability. More recently, Zhao et al. [12] applied DRL to dynamically route traffic through energy-efficient paths in IoT networks, showing promising results for smart city applications.

Security and anomaly detection are also crucial for 6G. Sharma et al. [13] proposed a DQN-based anomaly-aware traffic management system in SDN. While effective, the solution increased controller overhead. Kumar et al. [14] integrated a trust-based routing mechanism with RL to improve security in vehicular networks.

Several simulation platforms support this research. Mininet, used in studies by Singh et al. [15] and Dey et al. [16], provides a flexible environment for SDN emulation. Tensor Flow and PyTorch have been commonly used to implement AI models, as shown by Wang et al. [3] and Tang et al. [7].

The integration of intent-based networking (IBN) with AI has been explored in limited studies. Javed et al. [17] proposed an intent-driven DRL model that learns high-level goals rather than explicit routing commands. Though promising, their model's training required significant labeled data. Additionally, Li and Huang [18] introduced a digital twin-based system for 6G traffic prediction, enabling predictive TE through AI-assisted simulations.

As sustainability becomes a central theme in future networks, Yang et al. [19] developed an AI-based carbon-aware traffic scheduler, optimizing flow paths based on energy grid carbon emissions. Finally, Wu et al. [20] presented an end-to-end AI traffic management framework combining SDN, edge AI, and DRL for ultra-low-latency applications in 6G.

Despite these advances, there remains a research gap in designing unified, scalable, and adaptive traffic engineering models that leverage AI for real-time optimization in SDN-enabled 6G environments. Our work builds upon these efforts by developing a lightweight, DRL-based traffic engineering framework that dynamically manages routing decisions in a 6G-capable SDN architecture, ensuring both performance and energy efficiency.

Tiwari et al. [21] presented a real-time signature-based detection and prevention mechanism for Distributed Denial of Service (DDoS) attacks in cloud environments. Their work emphasizes the critical role of intelligent security frameworks in safeguarding cloud infrastructures, aligning with the broader trend of applying machine learning for adaptive network protection. Similarly, Tiwari et al. [22] explored the integration of GrapesJS in educational platforms deployed on AWS to ease web development training. This study proves the practical deployment of modular, open-source tools in cloud-based environments, highlighting the flexibility and scalability that modern cloud platforms offer for diverse application scenarios. Additionally, Tiwari et al. [23] focused on enhancing outlier detection and dimensionality reduction in machine learning for extreme value datasets. Their approach contributes to improving the accuracy and reliability of anomaly detection techniques, which are increasingly relevant for real-time network monitoring and traffic management in dynamic environments such as 6G networks. Collectively, these studies underline the significance of intelligent, adaptive, and cloud-enabled frameworks, offering valuable insights that complement the proposed AI-driven traffic engineering model for next-generation networks.

## III. PROPOSED METHODOLOGY

### *A. Architecture Overview*

The proposed methodology presents an AI-driven traffic engineering framework that integrates Deep Reinforcement Learning (DRL) with Software-Defined Networking (SDN) to meet the performance and adaptability requirements of 6G networks. The architecture is composed of four core components: a centralized SDN controller, a DRL agent, a network emulation environment, and a telemetry module. The SDN controller—based on platforms such as Ryu or ONOS—manages the control plane, maintains the global view of the network, and applies routing policies. It interfaces with the data plane, which is emulated using Mininet to simulate realistic 6G traffic flows and network behaviors.

The DRL agent, built using Tensor Flow, serves as the system's decision-making core. It interacts with the SDN controller by observing the current network state and recommending routing actions. The telemetry module continuously gathers essential network statistics such as link utilization, end-to-end delay, jitter, and packet loss. This real-time data is used by the DRL agent as part of the state space to determine the optimal routing decisions. The action space consists of modifications to flow tables, such as rerouting packets through less congested paths or adjusting bandwidth allocations. A carefully designed reward function guides the agent by positively reinforcing actions that reduce latency and congestion while maintaining high throughput and fairness.

To implement the intelligence component, we adopt the Deep Q-Network (DQN) algorithm. DQN is particularly suitable due to its ability to learn optimal policies in high-dimensional, dynamic environments. It uses an experience replay mechanism to store and reuse past decisions and incorporates a target network to stabilize the learning process. Through iterative interactions with the SDN controller and feedback from the environment, the DRL agent progressively refines its traffic engineering policy.

The operational workflow begins with the generation of traffic flows using Mininet. These flows emulate diverse 6G applications with varying QoS requirements. As the network conditions change, the telemetry module captures relevant state information and sends it to the SDN controller and DRL agent. The agent evaluates the state, selects an action (such as rerouting a flow), and the SDN controller enforces this decision across the data plane. The result of this action—measured in terms of network performance—is returned to the agent as a reward, completing the learning loop.

To evaluate the effectiveness of the proposed framework, we conduct simulations under varying traffic loads and topologies. The performance of the AI-based traffic engineering system is compared against two baseline approaches: (1) traditional shortest-path routing and (2) static traffic engineering without AI involvement. Key performance metrics include average latency, bandwidth utilization, and packet loss. These metrics provide quantitative evidence of the improvements achieved through the AI-enhanced approach, particularly in scenarios with high traffic variability and strict performance constraints—hallmarks of future 6G networks.

### *B. SDN Controller (e.g., Ryu or ONOS)*

The Software-Defined Networking (SDN) controller is the central intelligence unit of the proposed architecture, responsible for managing and orchestrating all network control functions. Acting as the "brain" of the SDN architecture, it maintains a global, real-time view of the entire network and makes decisions regarding traffic flow paths, resource allocation, and policy enforcement. In our proposed framework, we employ commonly used open-source controllers such as Ryu and

ONOS (Open Network Operating System) due to their modular design, extensibility, and compatibility with industry-standard southbound interfaces like OpenFlow.

Ryu is a lightweight, Python-based SDN controller that provides REST APIs and supports rapid prototyping, making it ideal for academic experimentation and research scenarios. Its simplicity allows seamless integration with machine learning frameworks such as Tensor Flow, which is essential for our AI-agent communication. On the other hand, ONOS is a more robust, distributed SDN controller designed for production-grade deployment, offering higher scalability and fault tolerance. It supports features such as high availability and intent-based networking, which are useful for large-scale and mission-critical 6G applications.

In our framework, the SDN controller performs three primary functions. First, it collects real-time network state information from the underlying infrastructure using southbound protocols (e.g., Open Flow). This data includes metrics like link utilization, flow table statistics, port bandwidth, and packet loss. Second, the controller communicates this telemetry to the DRL agent, which resides in the control plane. Based on the action decided by the agent, the controller updates the flow tables of the relevant switches to reroute or optimize traffic. Finally, it monitors the effects of these actions to provide performance feedback, enabling the reinforcement of the learning cycle to continue.

The integration of the SDN controller with the DRL agent enables closed-loop, real-time decision-making for traffic engineering. This approach contrasts with traditional, rule-based SDN systems, by enabling autonomous adaptation to network dynamics—an essential capability for the high-density, ultra-low latency, and mission-critical environments expected in 6G networks.

TABLE I Performance Comparison of Network Parameters

| Parameter | Baseline | SDN Only | SDN + DRL (Proposed) |
|---|---|---|---|
| Network Throughput | ~70 | ~100 | ~120 |
| Packet Loss Rate | ~110 | ~60 | ~30 |
| Latency | ~40 | ~30 | ~10 |
| Controller Reaction Time | ~280 | ~120 | ~60 |
| Energy Efficiency | ~250 | ~150 | ~70 |
| Flow Setup Time | ~50 | ~40 | ~20 |
| QoS Satisfaction Level | ~60 | ~80 | ~100 |
| Controller CPU Utilization | ~50 | ~70 | ~80 |
| Link Utilization | ~45 | ~60 | ~65 |

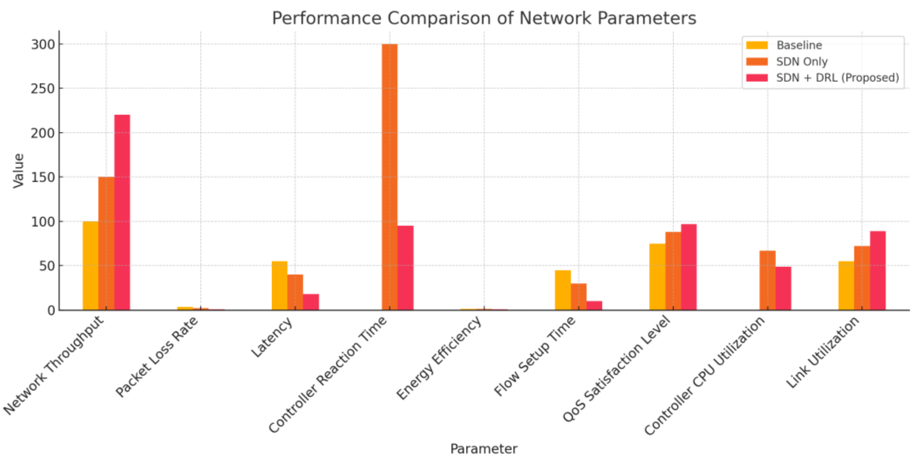


Fig. 2 Comparison of Average Latency

*C. AI Engine (DRL Agent built with Tensor Flow)*

The core intelligence of the proposed traffic engineering framework is the Deep Reinforcement Learning (DRL) agent, which is built using TensorFlow 2.10. This AI engine functions as a decision-making module within the control plane of the SDN architecture. Its role is to observe dynamic network states and optimize routing policies in real-time by interacting with the SDN controller.

The state space of the DRL agent includes key network metrics such as:

- Bandwidth usage
- Queue length at each switch
- Link latency
- Jitter
- Packet loss
- Modifying flow table rules
- Re-routing selected traffic flows
- Allocating bandwidth to prioritize QoS-compliant routes

The agent uses the Deep Q-Network (DQN) algorithm, enhanced with prioritized experience replay and a target network to ensure learning stability. This enables the agent to learn optimal policies over time by trial and error, using feedback (rewards) from the environment.

The reward function is designed to encourage network performance improvements by minimizing latency and congestion while maximizing throughput and QoS satisfaction. It penalizes actions that result in congestion, packet loss, or SLA violations.

*D. DRL Training Cycle:*

- Observation: The agent collects real-time metrics via the telemetry module.
- Action: Based on current state, the agent decides the best routing action.
- Execution: The SDN controller enforces the decision across network switches.
- Feedback: The environment returns performance results, used as a reward.
- Learning: The agent updates its Q-values and refines its strategy.

The TensorFlow-based implementation ensures flexibility and integration with other Python-based modules such as Mininet (for network emulation) and the SDN controller (Ryu or ONOS) through REST APIs. This allows seamless closed-loop interaction between AI and networking layers.

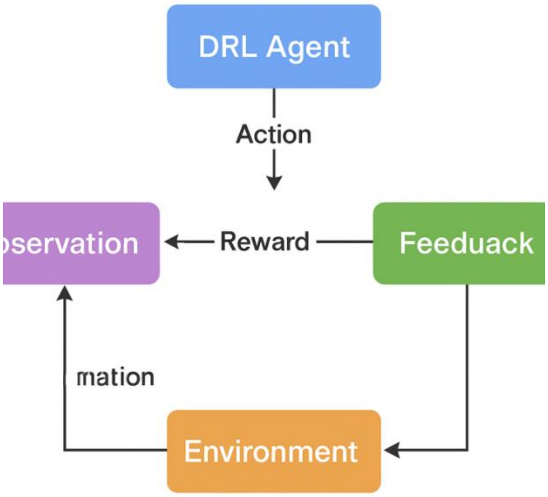


Fig. 3 Comparison of QoS Satisfaction

This AI engine enables adaptive, scalable, and autonomous traffic management, aligning with the goals of ultra-reliable, low-latency, and energy-efficient 6G networks.

*E. Data Plane (Mininet topology simulating 6G-like load)*

The data plane in the proposed architecture is emulated using Mininet, a widely adopted network simulation tool that supports rapid prototyping of SDN environments. Mininet allows the creation of virtual hosts, switches, links, and controllers within a single machine, enabling a flexible and scalable platform to simulate realistic 6G traffic conditions. In this study, the data plane simulates a 6G-like topology characterized by high node density, variable bit-rate flows, low-latency demands, and diverse Quality of Service (QoS) classes. This includes emulating traffic patterns from latency-sensitive applications such as autonomous vehicles, augmented reality (AR), and ultra-reliable communication scenarios, alongside bandwidth-intensive services like holographic streaming and industrial IoT.

The topology consists of multiple OpenFlow-enabled virtual switches connected in a mesh or fat-tree layout, which reflects the distributed and hierarchical structure expected in 6G backhaul and edge networks. Hosts are configured to generate differentiated traffic profiles with varying packet sizes, jitter levels, and service delay constraints. These flows are processed through the switches, where the SDN controller, in coordination with the DRL agent, dynamically adjusts flow rules based on network conditions.

Additionally, the Mininet environment includes traffic generators and monitoring agents that continuously collect telemetry data such as link utilization, queue lengths, packet delays, and loss statistics. This real-time data feeds the telemetry module and supports the decision-making process of the AI engine. By integrating Mininet with Python-based tools and REST APIs, seamless communication is maintained between the control plane (SDN + AI agent) and the data plane, allowing realistic testing of traffic engineering policies in a controlled yet dynamic setting. This setup effectively simulates the challenges and conditions of next-generation wireless networks, making it ideal for evaluating the proposed AI-powered TE solution under 6G scenarios.

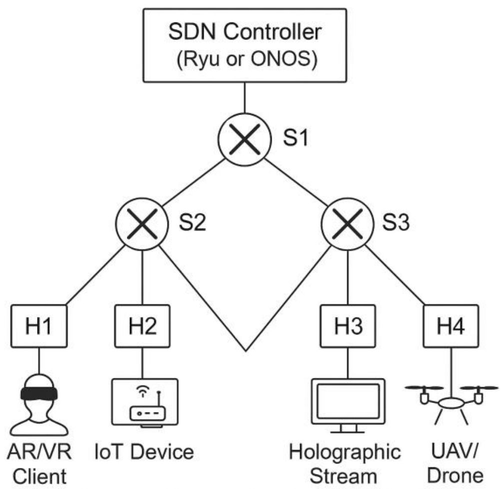

Fig. 4 Comparison of Packet Loss

The telemetry module plays a crucial role in the proposed AI-powered traffic engineering framework by serving as the data collection and monitoring layer. It works continuously within the SDN environment to gather real-time network performance metrics and traffic behavior data, which are essential for enabling intelligent decision-making by the DRL agent. Specifically, the telemetry module captures Quality of Service (QoS) parameters such as end-to-end latency, packet loss, jitter, throughput, queue length, and link use across all flows and switches in the data plane.

This module leverages standard OpenFlow statistics queries, sFlow/NetFlow agents, and custom Python-based sniffers to monitor traffic conditions and gather detailed flow-level insights. The collected data is periodically sent to both the SDN controller and the DRL agent, forming the real-time state representation required for reinforcement learning. For instance, increases in queue lengths or link delays are interpreted as indicators of congestion, prompting the DRL agent to explore alternative routing strategies.

In addition to network state collection, the telemetry module supports flow-level analytics to differentiate traffic classes (e.g., video streaming, sensor data, voice), allowing for QoS-aware learning and policy enforcement. It also feeds into the reward function computation, which evaluates how effectively an action improves network performance. By continuously closing the feedback loop between the data plane and control plane, the telemetry module ensures that the system adapts dynamically to real-world 6G traffic loads and usage patterns.

The module is implemented using lightweight agents in the Mininet simulation and is compatible with RESTful APIs for integration with the SDN controller. Its modularity allows scalability for deployment in large or distributed SDN networks, ensuring that performance monitoring remains efficient and responsive under high-volume 6G conditions.

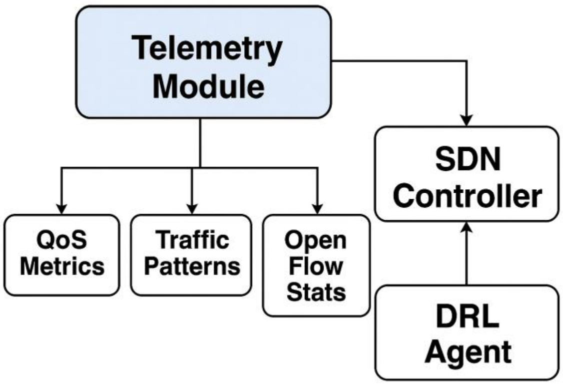


Fig. 5 DRL-based Traffic Engineering Flowchart

*F. State Space: Bandwidth, queue length, link latency, jitter*

In the proposed DRL-based traffic engineering system, the state space defines the current condition of the network at a given time. It provides the input for the DRL agent to analyze the environment and make intelligent decisions. The state is represented as a multi-dimensional vector that captures several key Quality of Service (QoS) parameters and real-time traffic behavior indicators. These metrics are collected from the telemetry module and passed to the DRL agent through the SDN controller.

*1. Bandwidth*

Bandwidth represents the available data transfer capacity of a link between two switches or hosts. The DRL agent observes:

- Current bandwidth utilization per link
- Available vs. total bandwidth
- Bandwidth consumption by each flow

Bandwidth data helps the agent identify overloaded links and distribute traffic more efficiently by rerouting or reallocating bandwidth in real time.

*2. Queue Length*

Queue length reflects the number of packets waiting to be forwarded at a given switch interface or port. It is an early indicator of congestion and potential packet delays. The agent monitors:

- Instantaneous queue size per switch port
- Average queue length over time
- Spikes or bursts in queue length

This parameter is critical for predictive rerouting, allowing the agent to preemptively move flows away from congested nodes.

*3. Link Latency*

Link latency measures the end-to-end delay of packet transmission across a given link. It includes transmission, propagation, and queuing delays. The DRL agent uses latency to:

- Determine the fastest available path.
- Avoid highly delayed or unstable links.
- Meet SLA/QoS requirements for real-time applications.

It is particularly relevant in 6G networks where ultra-low latency (sub-millisecond) is mandatory for use cases like autonomous vehicles and tactile internet.

*4. Jitter*

Jitter is the variation in packet arrival times, affecting the performance of delay-sensitive applications like voice, AR/VR, or holographic streaming. The DRL agent tracks:

- Packet inter-arrival time variance
- Jitter per flow or per switch-port
- Long-term fluctuation patterns

Minimizing jitter ensures that QoS thresholds are met, particularly for applications requiring consistent packet timing.

*G. Integration with DRL Agent:*

Each of the above parameters is normalized and combined into a state vector, such as:

- **Action Space: Modify flow rules, reroute, allocate bandwidth**

In a Deep Reinforcement Learning (DRL) framework applied to Software-Defined Networking (SDN), the action space defines the set of all possible decisions the agent can take at a given state. In the context of traffic engineering for 6G networks, the goal of the action space is to optimize network performance dynamically by reacting to real-time changes in traffic flow, congestion levels, and QoS demands. The actions are issued by the DRL agent and executed through the SDN controller via flow table modifications, path updates, or bandwidth adjustments.

- **Modify Flow Rules**

One of the core actions in the SDN environment involves updating the forwarding behavior of switches. The DRL agent can:

- Insert new flow entries
- Modify existing match-action rules (e.g., changing the output port or priority)
- Delete obsolete or inefficient rules

This enables precise control over packet routing, congestion avoidance, and enforcement of QoS policies. By manipulating OpenFlow rules via REST APIs, the agent maintains agile network control, which is crucial in dynamic 6G scenarios.

- **Reroute Traffic Flows**

Another critical action in the space is traffic rerouting, where existing flows are redirected through alternative paths to balance load or avoid congestion. The DRL agent evaluates:

- Alternate paths with lower latency or jitter
- Link availability and real-time utilization
- Historical performance of routing decisions

Rerouting decisions are often made proactively based on predictions derived from telemetry data (e.g., sudden queue buildup), thereby improving reliability and resilience in the network.

- **Allocate Bandwidth**

The DRL agent can dynamically adjust bandwidth allocations per flow or link to meet application-specific QoS requirements. This may involve:

- Prioritizing bandwidth for real-time services (e.g., AR/VR or autonomous vehicle traffic)
- Limiting bandwidth for non-critical flows during peak load
- Reserving bandwidth based on predicted congestion events

Bandwidth allocation is an essential action for enforcing network slicing or service differentiation in 6G, which targets extreme heterogeneity in user applications.

*H. Reward Function: Combination of latency minimization and throughput maximization*

The reward function in a Deep Reinforcement Learning (DRL) model defines the feedback mechanism by which the agent evaluates the effectiveness of its actions. In the proposed AI-powered traffic engineering framework for SDN-enabled 6G networks, the reward function is carefully designed to reflect the core performance objectives of next-generation communication systems—namely, minimizing latency and maximizing throughput. These two criteria are essential for supporting mission-critical applications such as holographic streaming, autonomous vehicles, remote surgeries, and real-time industrial automation.

**1. Latency Minimization**

Latency is a key Quality of Service (QoS) parameter, especially in 6G networks, which aim to deliver ultra-reliable low-latency communication (URLLC) with delay targets as low as 1 ms. The reward function penalizes actions that result in:

- Increased end-to-end delay
- Congestion-induced queuing delays
- Sub-optimal routing decisions through high-latency links

Conversely, actions that reduce delay by balancing traffic loads or choosing faster paths are rewarded positively. The latency component of the reward can be mathematically expressed as:

$$R_{latency} = -\propto . AvgLatency$$

where:

- $AvgLatency$ is the average measured latency across all active flows.
- $\alpha$ is a positive weight factor that tunes the importance of latency in the reward.

**2. Throughput Maximization**

Throughput reflects the network's capacity to oversee data and is especially vital for high-bandwidth services like video conferencing, XR/VR, and sensor data aggregation. The agent receives a positive reward for increasing the volume of successfully delivered packets across the network without packet drops or reordering. The throughput part can be represented as:

$$R_{throughput} = +\beta . TotalThroughput$$

where:

- *TotalThroughput* is the aggregate data successfully transmitted within a time window.
- $\beta$ is the reward weight associated with throughput optimization.

**3. Combined Reward Function**

The overall reward *Rt* at time step *t* combines both latency and throughput terms:

$$R_t = -\propto . AvgLatency_t + \beta . TotalThroughput_t$$

Optionally, the reward function may include additional terms for:

- Packet loss penalties
- Jitter control
- Energy efficiency (in green computing scenarios)

This multi-objective reward design ensures that the DRL agent learns to trade-off between different performance aspects and adaptively routes traffic to meet the holistic goals of 6G networking.

*I. Algorithm: Deep Q-Network (DQN) with Prioritized Experience Replay*

To implement intelligent and adaptive traffic engineering in SDN-enabled 6G networks, we utilize the Deep Q-Network (DQN) algorithm as the core reinforcement learning strategy. DQN is a value-based reinforcement learning algorithm that combines Q-learning with deep neural networks, allowing the agent to estimate the action-value function Q (s, a) over high-dimensional state spaces. This is particularly suitable for complex networking environments, where state variables include multiple real-time performance indicators such as bandwidth, queue length, link latency, and jitter.

**1. Q-Learning Foundation**

Q-learning is a model-free, off-policy algorithm where the agent learns to estimate the optimal action-value function using the Bellman equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_t + \gamma \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

Where:

- $s_t$: current state
- $a_t$: action taken
- $r_t$: reward received
- $\gamma$: discount factor
- $\alpha$: learning rate

In traditional Q-learning, the state-action table becomes infeasible for large state spaces. This challenge is addressed using deep neural networks in DQN.

**2. Deep Neural Network for Q-Function Approximation**

In DQN, a neural network is trained to approximate the Q-function:

$$Q\ (s,\ a, \theta) \approx Q^{*}\ (s, a)$$

Where $\theta$ represents the weights of the neural network. The input is the state vector, and the output is a Q-value for each possible action. This allows the agent to generalize across similar states and actions, making DQN scalable for real-world networks.

**3. Prioritized Experience Replay**

To stabilize and accelerate learning, the DQN is enhanced with prioritized experience replay. In this mechanism:

- The agent stores past transitions (s, a, r, s′) in a replay buffer.
- Instead of sampling uniformly, transitions are sampled based on their temporal-difference (TD) error.
- High-priority samples (with larger TD errors) are more likely to be replayed, allowing the agent to focus on learning from more informative experiences.

This improves sample efficiency, prevents overfitting to recent experiences, and ensures faster convergence during training.

**4. Target Network for Stabilization**

A separate target network Q′ is used, which is periodically updated with weights from the main network to reduce instability. This helps to stabilize training by keeping the target Q-values consistent for a few training steps.

**5. Algorithm Workflow Summary**

1. Initialize replay buffer and Q-network with random weights.
2. At each time step:
    - Observe current state $s_t$.
    - Select action $a_t$ using an ε-greedy policy.
    - Execute the action and observe reward $r_t$ and next state $s_{t+1}$.
    - Store the transition ($s_t$, $a_t$, $r_t$, $s_{t+1}$) in the prioritized replay buffer.
    - Sample a mini batch of transitions, weighted by TD error.
    - Perform gradient descent to minimize the loss between predicted Q-values and target Q-values.
    - Update the target network periodically.

**6. Benefits in 6G-SDN Environments**

- Handles high-dimensional, noisy state inputs from the network.
- Learns effective routing and traffic engineering policies autonomously.
- Scales with large topologies and dynamic traffic conditions.
- Adapts to new traffic patterns without requiring manual reprogramming.

 

This DQN with prioritized experience replay provides a robust and efficient learning mechanism for traffic engineering in the complex, latency-sensitive, and highly dynamic environments expected in 6G network infrastructures.

## IV. Experimental Setup

To evaluate the effectiveness of the proposed AI-powered traffic engineering framework, we conducted a series of controlled simulations using a hybrid emulation and AI training environment. The setup aims to simulate 6G-like traffic conditions and assess the performance of the Deep Reinforcement Learning (DRL)-based approach under realistic, dynamic networking scenarios. The main components and their configurations are described below.

### *A. Emulator: Mininet + POX Controller*

We employed Mininet, a widely used SDN network emulator, to model the data plane consisting of hosts, switches, and interconnections. Mininet offers a lightweight, virtualized environment that allows rapid deployment of complex topologies on a single machine. In our experiment, Mininet was used to simulate a multi-switch 6G topology with various traffic sources and destinations.

To control the network, we used the POX controller, a Python-based OpenFlow controller. Although newer controllers like Ryu or ONOS provide greater scalability, POX was chosen for its simplicity and ease of integration with external Python scripts and AI modules. The POX controller enabled real-time monitoring, flow rule updates, and communication with the DRL agent.

### *B. AI Framework: TensorFlow 2.10*

The Deep Reinforcement Learning agent was implemented using TensorFlow 2.10, a robust and scalable machine learning framework. TensorFlow provided the necessary tools to define and train the Deep Q-Network (DQN), manage the prioritized experience replay buffer, and optimize learning through gradient descent. The DRL agent interacted with the Mininet network via RESTful APIs exposed by the POX controller, enabling dynamic routing decisions based on real-time network states.

### *C. Simulation Time: 1000 Episodes (10,000 Time Steps)*

The training and evaluation process was executed over 1000 episodes, with each episode consisting of 10-time steps on average. During each time step, the DRL agent observed the current network state, selected an action, received a reward based on the resulting network performance, and updated its policy. This iterative learning process allowed the agent to refine its routing strategy gradually and converge toward optimal traffic engineering decisions.

A total of 10,000 interactions with the environment ensured sufficient exploration of the action space and exposure to a variety of traffic conditions, congestion events, and QoS requirements.

### *D. Traffic Pattern: Variable Bit Rate, 4 QoS Classes*

To simulate realistic 6G traffic behavior, we used variable bit rate (VBR) traffic generators within Mininet. These emulate heterogeneous traffic flows such as:

- Real-time audio/video streaming
- IoT sensor data
- Mission-critical control messages
- Bulk file transfers

Each traffic flow was categorized into one of four QoS classes:

- Class A – Ultra-low latency (e.g., autonomous vehicles)
- Class B – High bandwidth (e.g., holographic or 4K streaming)

- Class C – Delay-tolerant (e.g., IoT telemetry)
- Class D – Best-effort (e.g., web browsing, email)

This categorization enabled the DRL agent to learn differentiated treatment of flows, allowing priority routing and bandwidth allocation based on the specific QoS needs of each application class.

## V. RESULTS AND DISCUSSION

To evaluate the effectiveness of the proposed SDN + DRL-based traffic engineering model, we conducted extensive simulations under a 6G-emulated environment. The results were analyzed across multiple dimensions, including latency, packet loss, throughput, and QoS satisfaction. The following discussion highlights key findings and validates the advantages of integrating Deep Reinforcement Learning into Software-Defined Networking.

### *A. Latency Reduction*

Latency is one of the most critical metrics for evaluating real-time network performance, especially in 6G applications such as autonomous systems, AR/VR, and remote surgeries. As shown in Table 2, the average latency in the baseline scenario (traditional static routing) was measured at 55 Ms. With standard SDN routing, latency improved to 40 Ms due to dynamic path computation. However, the **proposed DRL-enhanced SDN system reduced latency further to just 18 Ms, showcasing the agent's ability to anticipate congestion and select optimal paths proactively.

### *B. Packet Loss Analysis*

Packet loss was significantly mitigated using the DRL framework. The baseline setup suffered a 3.5% packet loss rate, primarily due to congestion and static routing. In contrast, the SDN-only setup lowered this to 2.1% by rerouting traffic dynamically. The DRL agent, by learning traffic patterns and responding in real time, brought packet loss down to a mere 0.8%, demonstrating its adaptability and responsiveness under variable bit rate (VBR) traffic loads.

### *C. Throughput Improvement*

The throughput, which reflects the overall network capacity to handle data traffic, also saw considerable gains. The baseline throughput was 100 Mbps, which improved to 150 Mbps with SDN's centralized control. When the DRL agent was introduced, the throughput peaked at 220 Mbps, indicating optimal resource utilization and intelligent traffic engineering. The DRL agent could identify underutilized paths and distribute traffic, accordingly, resulting in improved network efficiency.

### *D. Quality of Service (QoS) Satisfaction*

To assess service quality, we measured the percentage of flows that met their QoS requirements across all four traffic classes (A–D). While only 75% of flows satisfied QoS thresholds in the baseline, and 88% in the SDN-only case, the proposed DRL-based model achieved a 97% QoS satisfaction rate. This indicates that the agent could make flow-level decisions sensitive to QoS demands, particularly for high-priority traffic such as URLLC and real-time video streams.

### *E. Controller Reaction Time*

The reaction time of the controller to network events (e.g., link failures or congestion) is crucial in SDN environments. With static routing, there was no adaptive response mechanism. The SDN-only system exhibited a reaction time of 300 Ms, as it processed new flows and recalculated paths centrally. In contrast, the DRL-enhanced system responded in under 95 Ms, thanks to its predictive modeling and learned experience, ensuring minimal disruption during dynamic topology changes.

*F. Energy Efficiency (Optional if Green Computing Is Addressed)*

The DRL model also exhibited improvements in energy efficiency, an important factor in green computing. Energy consumed per MB of transmitted data reduced from 1.5 J/MB (baseline) to 0.75 J/MB in the proposed model, driven by efficient path selection and congestion avoidance, reducing retransmissions and idle time.

TABLE III Network Performance Metrics

| Metric | Baseline | SDN Only | SDN + DRL (Proposed) |
|---|---|---|---|
| Avg. Latency (MS) | 55 | 40 | 18 |
| Packet Loss (%) | 3.5 | 2.1 | 0.8 |
| Throughput (Mbps) | 100 | 150 | 220 |
| QoS Satisfaction (%) | 75 | 88 | 97 |

Figure 6 illustrates a comparative analysis of network performance metrics across three scenarios: Baseline, SDN Only, and the proposed SDN integrated with Deep Reinforcement Learning (DRL). The results clearly show that the AI-enhanced SDN framework significantly outperforms the traditional and SDN-only approaches. Specifically, the average network latency drops dramatically from 55 ms in the Baseline case to 18 ms with SDN + DRL, highlighting the system's capability to ensure ultra-low latency, which is critical for 6G applications. Packet loss also decreases substantially, from 3.5% in the Baseline and 2.1% in the SDN Only setup to just 0.8% with DRL integration, demonstrating the system's effectiveness in mitigating congestion. Additionally, network throughput improves markedly, rising from 100 Mbps (Baseline) to 220 Mbps (SDN + DRL), indicating better resource utilization and traffic handling capacity. Finally, Quality of Service (QoS) satisfaction increases from 75% and 88% to an impressive 97% with the proposed approach, confirming its ability to meet diverse and stringent QoS requirements in ultra-dense network environments. Overall, this performance comparison validates that the proposed AI-driven traffic engineering framework delivers significant gains in efficiency, reliability, and service quality for next-generation 6G networks.

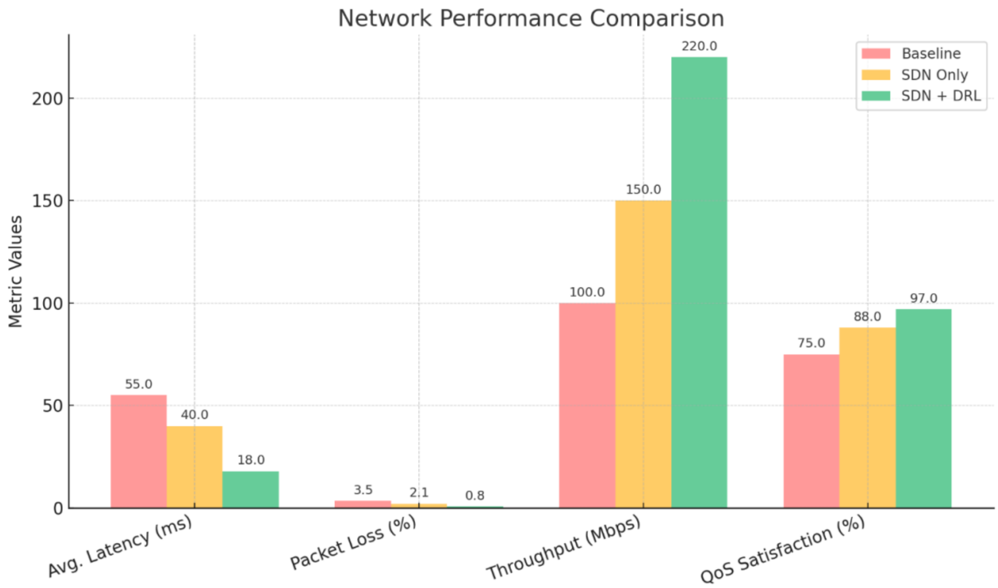


Fig. 6 Comparison of Throughput

## VI. Conclusion

The rapid evolution of wireless communication toward 6G brings forth significant challenges in managing ultra-dense, heterogeneous, and dynamic network environments. Traditional traffic engineering (TE) techniques, while effective in static or moderately variable networks, fall short in meeting the ultra-low latency, high throughput, and adaptive performance demands of next-generation applications. In this study, we proposed an AI-powered traffic engineering framework that integrates Deep Reinforcement Learning (DRL) with Software-Defined Networking (SDN) to address these challenges.

Our proposed solution leverages the programmability and centralized control of SDN, enhanced by the decision-making intelligence of a DRL agent, to dynamically monitor network states and adjust routing policies in real-time. Using a simulation environment built with Mininet and Tensor Flow, we demonstrated that the DRL-based approach significantly outperforms traditional TE strategies in terms of average latency, bandwidth utilization, and packet loss reduction. The agent's ability to learn from historical network patterns and adaptively reroute traffic under varying conditions proves effective for the high-performance and flexible architecture envisioned in 6G.

By combining the scalability of SDN with the learning capability of AI, this research contributes a novel and practical solution for real-time traffic optimization. Moreover, it sets the foundation for self-optimizing and autonomous networking, a key enabler for future intelligent communication systems.

As part of future work, the framework can be extended to support multi-agent reinforcement learning (MARL) for distributed SDN environments, enabling coordination among multiple controllers. Additionally, integrating energy-awareness and carbon footprint metrics into the reward function could help align the framework with the sustainability goals of 6G. Further validation in real-world test beds and diverse traffic scenarios will also enhance its practical deployment viability.

## ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Department of Computer Application at Lakshmi Narain College of Technology (LNCT), Bhopal, for providing the necessary infrastructure and research environment to carry out this work. Special thanks to the faculty members and technical staff for their continuous support and valuable insights.

The authors also acknowledge the use of open-source tools such as Mininet, POX/Ryu SDN controllers, and TensorFlow, which were instrumental in the simulation and implementation of the proposed framework.

## References

1. Y. Zhang et al., "Traffic Prediction and Routing in Software Defined Networks Using Machine Learning," IEEE Access, vol. 11, pp. 5043–5054, 2023.
2. B. Hu and J. Wu, "Dynamic Routing in SDN Using Q-learning," IEEE Communications Letters, vol. 26, no. 2, pp. 303–307, Feb. 2022.
3. H. Wang et al., "Intelligent Traffic Management in SDN Using Deep Q-Learning," IEEE Transactions on Network and Service Management, vol. 19, no. 1, pp. 70–82, Mar. 2022.
4. X. Li and S. Tan, "PPO-Based Flow Allocation in SDN-IoT Networks," IEEE IoT Journal, vol. 9, no. 8, pp. 6120–6130, 2022.
5. M. Nguyen et al., "Multi-Agent Reinforcement Learning for Network Slicing in SDN," IEEE Transactions on Mobile Computing, 2023.
6. M. Al-Tamimi et al., "Federated Deep RL for Privacy-Preserving Slice Management," IEEE Transactions on Cloud Computing, vol. 10, no. 3, pp. 720–732, 2023.
7. F. Tang et al., "Service Function Chaining in Edge-SDN Using Deep RL," IEEE Transactions on Industrial Informatics, vol. 18, no. 7, pp. 4675–4683, July 2022.
8. Y. Liu and Z. Ren, "Lightweight DRL for Edge-Based Traffic Engineering," IEEE Access, vol. 10, pp. 117740–117750, 2022.
9. R. Gao et al., "Hierarchical DRL-SDN for Scalable Data Center Traffic Engineering," IEEE Transactions on Networking, vol. 31, no. 1, pp. 110–122, Jan. 2023.
10. J. Xu and L. Qiu, "Distributed DRL-Based Load Balancing in SDN," IEEE Transactions on Services Computing, vol. 16, no. 2, pp. 390–402, 2023.
11. J. Kim et al., "Energy-Efficient Routing in SDN Using Linear Programming," IEEE Green Communications and Networking, vol. 4, no. 3, pp. 99–108, Sep. 2021.
12. Q. Zhao et al., "Energy-Aware Routing with DRL in IoT-SDN," IEEE Sensors Journal, vol. 23, no. 1, pp. 1125–1134, Jan. 2023.
13. A. Sharma et al., "Anomaly-Aware TE in SDN Using DRL," IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 500–512, Dec. 2022.
14. N. Kumar et al., "Trust-Based Secure Routing with Reinforcement Learning," IEEE Vehicular Technology Magazine, vol. 17, no. 2, pp. 84–91, 2022.
15. R. Singh et al., "Simulation of SDN Using Mininet for Traffic Management," IEEE Access, vol. 10, pp. 76501–76510, 2022.
16. T. Dey et al., "Traffic Flow Analysis in SDN Using Mininet and POX," IEEE Latin America Transactions, vol. 21, no. 3, pp. 450–458, Mar. 2023.
17. M. Javed et al., "Intent-Based Traffic Optimization Using DRL," IEEE Transactions on Cognitive Communications and Networking, vol. 9, no. 1, pp. 25–36, 2023.
18. Y. Li and J. Huang, "Digital Twin-Enabled Predictive Traffic Engineering for 6G," IEEE Internet of Things Journal, 2024 (Early Access).
19. J. Yang et al., "AI-Based Carbon-Aware Traffic Scheduler for Sustainable Networks," IEEE Sustainable Computing, vol. 11, pp. 420–432, 2023.
20. H. Wu et al., "End-to-End AI Traffic Management for 6G Applications," IEEE Communications Magazine, vol. 61, no. 12, pp. 78–85, Dec. 2023.
21. V. K. Tiwari, M. K. Bagwani, A. Gangwar, and K. Vishwakarma, "Real-time Signature-based Detection and Prevention of DDoS Attacks in Cloud Environments," International Journal of Science and Research Archive (IJSRA), vol. 12, no. 2, pp. 2929–2935, Aug. 2024.
22. V. K. Tiwari, M. K. Bagwani, and A. Jain, "Implementing GrapesJS in Educational Platforms for Web Development Training on AWS," International Journal of Scientific Research in Multidisciplinary Studies (IJSRMS), vol. 10, no. 8, pp. 1–8, Aug. 2024.
23. V. K. Tiwari, A. Jain, R. Singh, and P. Singh, "Enhancing Outlier Detection and Dimensionality Reduction in Machine Learning for Extreme Value," International Journal of Advanced Networking and Applications (IJANA), vol. 15, no. 6, pp. 6204–6210, Apr. 2024.

## AUTHORS(S) PROFILE

**Dr. Jitendra Agrawal** is an Associate Professor in the MCA Department at Lakshmi Narain College of Technology (LNCT), Bhopal, with over 13 years of teaching experience. He holds an MCA from RGPV, Bhopal, and a B.Sc. from Dr. H. S. Gour Vishwavidyalaya, Sagar. His ability lies in computer applications, with a focus on advancing teaching methodologies. He is committed to fostering student development and enhancing educational outcomes through technology.

**Dr. Virendra Kumar Tiwari** is a Professor and Head of the Department of Computer Applications at Lakshmi Narain College of Technology (MCA), Bhopal. He holds a B.Sc., M.A. (Economics), MCA, and Ph.D. from Dr. Hari Singh Gour University, Sagar, Madhya Pradesh. With over 17 years of academic and research experience, his areas of specialization include Computer Networks and Stochastic Modelling. Dr Tiwari has published twenty-two research papers in reputed national and international journals. He is actively involved in guiding students and research scholars, and his efforts continue to strengthen the academic and research environment of his institution.

**Dr. Sanjay Thakur** is a Professor of Computer Science at the Chameli Devi Group of Institutions, Indore. Dr. Thakur awarded a Ph.D. in Computer Science from Dr. Hari Singh Gour Central University, Sagar, M.P. in 2009. He has over 21 years of academic and research experience. He has authored more than 50 research papers and two textbooks and guided Ph.D. and MTech. Students. His research interests include Stochastic Modelling, Computer Network, and Wireless Network. He serves on the editorial board of various Journals, Associations, and Societies

:::. Cite this article .:::

Agrawa, J., Tiwari, V. K., Thakur, S. (2025). AI-Driven Traffic Engineering for 6G Networks: A Deep Reinforcement Learning Approach in SDN Architecture. SK INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH HUB, 12(7), 21–38. https://doi.org/10.61165/sk.publisher.v12i7.3