International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

SMART EDGE COMPUTING INTEGRATION FOR LATENCY-CRITICAL APPLICATIONS IN NEXT-GENERATION WIRELESS NETWORKS

Mothana L. Attiah[1], Assad Jumaah Hassan[2], Azhar Amer Alsoufi[3], Ahmed Dheyaa Radhi[4]

[1] Department of Computer Engineering, Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq, Police College of Iraq, Baghdad, Iraq,

mothana.ipc@gmail.com

[2] Shiraz University, Computer Engineering, Artificial Intelligence and Robotics, , Police College of Iraq, Baghdad, Iraq,

asaadalsiaede@gmail.com

[3] Department of Networks and Computer Software Techniques, Northern Technical University, Mosul, Iraq.

Mti.lec250.azhar@ntu.edu.iq

[4] College of Pharmacy, University of Al-Ameed, Karbala PO Box 198, Iraq

ahmosawi@alameed.edu.iq

Abstract

One of the efforts in wireless networks (5G and beyond) is anticipated to serve an enormous amount of latency-sensitive applications, including autonomous driving, augmented/virtual reality (AR/VR), and industrial automation in real-time. Nevertheless, these applications have strict latency and availability demands that are very problematic to traditional cloud-based infrastructures.

The purpose of this paper is to demonstrate an intelligent edge computing system that will be used to dynamically assign the computing resources and network bandwidth to tasks, which are latency-sensitive. The suggested system will combine Deep Reinforcement Learning (DRL) with Smart Edge Computing to provide adaptive task execution, efficient bandwidth usage and lower end-to-end latency in heterogeneous network spaces.

Python is used to create a realistic simulation model, which will capture the variation of bandwidth with time, loss of packets, queuing behavior, and the prioritization of tasks under various load conditions. The DRA agent keeps learning the best task to node assignment strategies, which would allow reducing delay and keeping Service Level Agreement (SLA) compliance.

The experiments have shown that the proposed framework has an average latency of sub-second (0.7-0.9 s), SLA violation of almost zero and can perform load balancing between edge and cloud nodes. These results prove that the performance and the reliability of the applications

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

with a high latency level in 5G/6G networks can be significantly improved with the use of the DRL-based edge intelligence.

Keywords: Edge Computing, 5G/6G Networks, Latency-Critical Applications, Deep Reinforcement Learning, SLA Optimization.

1.introduction

The development of wireless communication technology towards the fifth (5G) and sixth-generation (6G) networks has brought another paradigm of ultra-reliability, low-latency communication (URLLC) that should support the new application of autonomous vehicles, augmented and virtual reality (AR/VR), telemedicine, and real-time industrial control. Such applications require end to end latencies of less than 1 milliseconds, high reliability and huge connectivity and the conventional cloud-centric systems could not provide it consistently because of long distances of transmission and centralized bottlenecks in the processing systems[5].

In the efforts to overcome the challenges, Edge Computing has proved as a viable solution that brings calculating, storing and intelligent nearer to the source of data[10]. Deploying distributed Edge Nodes at the proximity to end users allows to reduce the network latency and congestion dramatically and implement real-time decision-making and rapid processing of data. But once the number of connected devices and the amount of tasks requiring quickly reacting resources grows, resource distribution and task scheduling at the edge becomes more intricate since the next-generation networks are dynamic and heterogeneous[6].

Recent achievements of Artificial Intelligence (AI) specifically Deep Reinforcement Learning (DRL) can offer an effective solution to dynamic optimization problems in complex setting. DRL-based schedulers improve the traditional, static or heuristic-based models by autonomously changing resource allocation policy in real time by continually taking feedback about network states and task requirements[8].

This paper includes our proposed solution of a Smart Edge Computing architecture that combines DRL and operation optimization to optimize the distribution of resources in the network and computation level of latency-sensitive applications of 5G/6G networks. The proposed mechanism automatically modulates the decisions of task scheduling using the real-time conditions, which include bandwidth changes, interruption of packets, load of a node and queue length to guarantee the latest latency and service reliability is ensured as much as possible.

In order to test the suggested methodology, a detailed simulator of python was built, simulating the dynamics of the real network, including bandwidth adjustmence and latency on demand, stochastic packet loss and unstable task loads. The findings indicate that the edge computing system enhanced with DRL holds a great amount of latency reduction, balanced load distribution, and zero SLA violation allowing to consider it as the effective approach to managing type of workloads in wireless network with a high latency burden.

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2. Literature Review

research [1] discusses edge computing in the context of the new 6G networks and proposes an ARMO (Adaptive Resource Management and Offloading) model. The ARMO model builds on the benefits of smart scheduling of tasks, dynamically distributed resources and energy savings in order to achieve better performance of the edge computing environment. A complex algorithmic system is to gather and preprocess IoT device data, retrieve suitable characteristics, forecast resource allocation, optimize task offloading, and monitor and readjust resource allocation prior to constant cycle using sophisticated machine learning algorithms. The findings indicate high advances such as reduced average latency according to 47 percent, a decrease in total power consumption to 40 percent and 20 percent improvement in the resource utilization. Moreover, the model had a high rate of task execution (98 percent) and always had a high network throughput than the prior models. The presented results indicate that the ARMO model can be used to enable next-generation real-time IoT-related applications, a solid, effective, and scalable solution to integrate edge computing and the 6G networks.

The combination of edge computing and innovative wireless network technologies, specifically, 5G and 6G networks, is a paradigm shift of sensor data processing and use in the ecosystem of the Internet of Things (IoT). The synergetic solution serves to resolve many issues that decades ago pervaded the cloud-based, centralized architectures, which include, but are not limited to high latency, network overload, poor scalability, and large power consumption. The cohesive system enables the future generation of real-time, responsive and scalable IoT applications because it enables data processing on the device and leveraging the super-reliable and low-latency connectivity of 5G and 6G. The medical case studies, autonomous vehicles and smart cities indicate the way in which this joint approach can lead to enhancement of decision making, services delivery, and using the computing and network resources. The comprehensive potential of this model is, however, dependent on addressing various technical issues like interoperability, data security, and complicacy of deployment which should be given coordinated research and innovation. The potential contributions of quantum computing, new techniques in edge artificial intelligence, federal learning, and sustainable energy solutions of edge devices are expected to increase the potentials and influence of edge-enabled wireless networks in future research. With its further development and maturity, convergence between these technologies will play a crucial role in the creation of the next generation of smart, autonomous and adaptive Internet of Things (IoT) systems[2].

Leveraging calculating intensive routing of tasks in the IoT edge system, paper [3] suggests that the order of task execution and task allocation are coupled together leading to optimized scheduling of tasks. Our optimization problem would also be an MDP model whose state, action, reward and state transition are well-thought out. The scheduling time step is disengaged with the real-time step to minimize the dimension of action space. The MDP is solved by a deep reinforcement learning algorithm, which is policy based. It illustrates that our algorithm proposed has got good convergence properties. Moreover, it involves application of intensive simulations to determine cumulative task satisfaction score and success rate. The findings indicate that the suggested algorithm beats other measuring techniques. The further work to be

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

done includes collaborative cloud computing and edge computing wherein the consideration of the connection latency will be made.

ARMA [4] is a lightweight controller over the O-RAN RIC controller which provides the application and RAN network with enough state (frame deadlines, DNN load and real-time RB availability) to optimize bitrate, model depth, RB allocation, and GPU time per frame. ARMA by dynamically using the latency budget per request between the wireless and the live steps, enhances the Service Level of Satisfaction (SLO) by increasing it by a factor of 26 to 97 in the Open-RAN video analytics test platform, at relative radio expenses. In our continuous efforts, we are establishing a resource scheduling framework, which will support applications, not limited to video analytics, in meeting their own SLO, with either no application hints or with no changes to the infrastructure.

Table 1 shows comparison table between studies.

Table 1 comparison between studies

Study / Model	**Main Focus**	**Methodology / Algorithm**	**Target Network Environment**	**Key Metrics Improved**	**Limitations**	**Contribution Difference (Proposed Work)**
[1] ARMO (Adaptive Resource Management and Offloading Model)	Dynamic resource allocation and task offloading in 6G edge systems	Machine Learning-based predictive model for resource forecasting	6G IoT and Edge environments	Latency (47%), Power consumption (40%), Resource utilization (20%)	Static learning model; limited real-time adaptability; not DRL-based	Our model adds **Deep Reinforcement Learning** to enable *adaptive, online decision-making* under fluctuating latency and bandwidth conditions
[2] Edge-5G/6G Convergence Model	Integration of edge intelligence with next-generatio	Conceptual + case study analysis	5G/6G + IoT ecosystems	Improved scalability and decision-making in smart systems	No quantitative model or optimization	Our work provides a **quantitative DRL-based simulation** with measurable

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

	n wireless networks				mechanism	metrics (latency, SLA, resource use) rather than conceptual description
[3] DRL-Based Task Scheduling (IoT Edge System)	Optimized scheduling and task ordering	Deep Reinforcement Learning (Policy-based MDP)	IoT Edge networks	Convergence stability; improved task success rate	Ignores link latency, packet loss, and 5G/6G dynamics	We extend DRL scheduling by modeling **dynamic bandwidth, jitter, and retransmission**, closer to **realistic 5G/6G network behavior**
[4] ARMA (O-RAN Adaptive Resource Management Agent)	Radio and compute resource control in Open-RAN	Lightweight controller on RIC with latency-aware allocation	O-RAN (5G test platform)	SLO satisfaction	Focused only on radio layer; not integrated with computing offloading	Our framework integrates **radio + computing layers**, handling **end-to-end latency optimization** in a unified edge simulation
Proposed Work (Smart Edge Computin	Intelligent resource orchestration for latency-critical	Deep Reinforcement Learning (Q-learning variant) +	Hybrid 5G/6G Edge environment	Average latency, SLA violations, learning stability	Adding energy consumption as a further optimizati	Combines **intelligent edge offloading**, **realistic 5G/6G**

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

g Integration using DRL)	5G/6G applications	dynamic network simulation			on with an aim of lowering operational expenses of edge nodes	**modeling**, and **adaptive DRL optimization** — bridging theory and simulation realism

3. Problem Statement

Although the 5G is fast maturing and 6G technologies are still being tested, providing a minimum of consistency in terms of ultra-low latency and high reliability when creating real-time applications is still a significant challenge. Conventional cloud-based models also possess the drawback of long communication routes, inadequacy in scaling, as well as unpredictable network congestion, which significantly increase the end-to-end delay.

Though edge computing is able to minimize latency with computations shifted closer to the users, it introduces additional challenges which include:

- Active resource competition between several edge nodes and users.
- Unreliable network properties (e.g. fluctuating bandwidth, lost packets).
- Mixed workloads where the latency and reliability needs are different.
- Absence of Smart coordination between cloud and edge layers.
- Current statistic or heuristic-based scheduling schemes are not effective in adapting to these time-dependent network characteristics and, as such, result in inefficient resource utilization, QoS deterioration, and even the breach of SLAs to services that are latency-sensitive such as autonomous vehicles or AR/VR.

4. Research Objectives

The proposed study will develop and test a Smart Edge Computing architecture with Deep Reinforcement Learning (DRL) to improve the functionality of the latency-sensitive applications at 5G/6G networks.

The overall objectives include:

- Create a model-driven dynamic simulation setup that captures the behavior of dynamic 5G/6G networks, such as bandwidth fluctuation, loss of packets and heterogeneous edge nodes.
- Develop a smart DRL based scheduler able to assign latency sensitive tasks to the ideal Dynamic edge or cloud node in real time.
- Maximize policy on task allocation to reduce average end-to-end latency and ensure SLA is adhered to with precision in terms of latency-sensitive application.

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Measure the system operation under different network loads and network characteristics and study the essential indicators, including the latency distribution, SLA violation rates, and node utilization.
- Show the benefits of DRL-enhanced Smart Edge Computing in comparison to the traditional heuristic scheduling or fixed-point scheduling.

5. Proposed System

This new system proposes a Smart Edge computing architecture as shown in figure 1, which merges Deep Reinforcement Learning (DRL) in order to minimize the latency of an application and resource utilization in next-generation wireless networks (5G/6G). The architecture is to automatically adjust to changing network properties and non-uniform computational resources, which is to guarantee ultra-reliable low-latency communication (URLLC).

5.1 System Architecture Overview

There are three major layers in the system which are:

1. End-Device Layer (User Equipment)

   Mobile users and IOT devices that create tasks with a latency constraint (e.g. a stream of AR/VR, sensor data on autonomous vehicles, industrial control messages) are present in this layer. Every task has certain requirements (deadline, data size, priority).

2. Edge Computing Layer

   This layer is made up of several Edge Nodes that are close to the users. Each node has limited computational capacity and queueing capability Data is distributed to these nodes to be processed faster thereby reducing propagation delay.

3. Cloud Layer

   The cloud is a backup processing resource that possesses large computation power at increased latency because of long transmission paths. Jobs that cannot be effectively executed at the periphery are delegated at this stage.

4. Network Management and Control Plane

   The module is a model that depicts the dynamics of the wireless network such as variation in bandwidth, packet loss and congestion of the links. It feeds real time information of the state to the DRL agent as a guide to decision making.

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

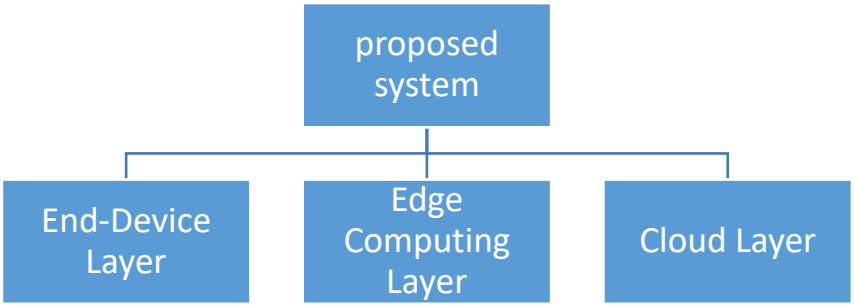

Figure 1 proposed system

5.2 Deep Reinforcement Learning Scheduler

The system is based on DRL-based scheduler that can learn to take an optimal decision on task allocation based on dynamic conditions.

The DRL agent uses the present state of the system, chooses a course of action (to what node to delegate the task), gets a reward according to the performance in terms of latency, and reacts to a new policy[7].

1. State (S):

- Represents the environment at each decision step, including:
- Edge node queue lengths
- Processing speeds
- Available bandwidth
- Network congestion level
- Task type and size

2. Action (A):

- Selecting a target node (Edge or Cloud) for task execution.

3. Reward (R):

Computed as a function of negative latency and SLA compliance.

- Lower latency and fewer SLA violations produce higher rewards.

4. Policy Update:

- The agent uses a Deep Q-Network (DQN) to approximate the optimal Q-function (equation 1):

$$\gamma \max_{a'} Q(s', a') + r = Q^*(s, a) \quad \text{(equation 1)}$$

where γ is the discount factor balancing immediate and future rewards. The network is trained iteratively using experience replay and an ε-greedy exploration strategy.

5.3 Network Model

The wireless network is to be modeled with a dynamic bandwidth and stochastic packet loss in order to provide realistic simulation:

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Dynamic Bandwidth**:** It changes with time and the number of transmissions that are taking place.
- Packet Loss Probability: Is a random retransmissions of data, which delay a network.
- Transmission Delay [9]:
  - D_{net}= (DataSize×8 /EffectiveBandwidth) +Jitter+RetransmissionPenalty (equation 2)

This design closely replicates real-world 5G/6G environments where signal conditions and traffic load fluctuate dynamically.

5.4 Task Processing Model

Each task Ti is characterized by:

Ti={sizei,deadlinei,typei,priorityi}

and processed either at an **Edge Node (E_k)** or the **Cloud (C)**. The total end-to-end latency is computed as (equation 3):

- Ltotal=Duplink+Dqueue+Dproc+Ddownlink (equation 3):

where:

- Duplink: Transmission delay to the selected node
- Dqueue: Waiting time in the node's queue
- Dproc: Processing time based on node speed
- Ddownlink: Response transmission delay

5.5 Training Process

The DRL agent is learned on a series of episodes, each one being a series of arrivals of the tasks in different network state conditions.
During each episode:

1. Tasks are created in a random fashion with varied parameters.
2. Nodes are chooseable in accordance with the existing policy by the agent.
3. Latencies performance is used to calculate rewards.
4. The Q-network is updated.

Over time, the agent learns to minimize latency and violations while balancing load across all nodes.

5.6 Summary of Workflow

1. Generation of tasks at end user layer.
2. State observation (network + node conditions).
3. DRL agent decision (select optimal node).

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4. Dynamic transmission of tasks via 5G/6G network.
5. Task execution at edge/cloud.
6. Computation of reward and updating models.

This workflow will make it possible to optimize edge computing resources in an adaptive and intelligent and continuous manner to real-world 5G/6G dynamics.

6. Experimental Setup and Simulation Environment

In order to test the suggested DRA-enhanced Smart Edge Computing system, an all-encompassing simulation setup was developed in Python 3.10. The simulation models both **computational resources (Edge and Cloud Nodes)** and **network dynamics (5G/6G communication conditions)** to approximate real-world latency-critical scenarios.

6.1 Hardware and Software Environment

- **Programming Language:** Python
- **Libraries Used:** NumPy, Matplotlib, Random, and TensorFlow (for DQN training)
- **Processor:** Intel Core i7 / AMD Ryzen 7 or equivalent
- **Memory:** 16 GB RAM
- **Operating System:** Windows 11 / Ubuntu 22.04
- **Simulation Time:** 30 episodes × 60 tasks per episode

6.2 Network Model Parameters

The wireless wireless was described as being of varying capacity dynamic 5G/6G channel and probabilistic packet loss. The significant parameters are listed in table 2:

Table 2 Network Model Parameters

Parameter	Description	Value
Base Bandwidth	Maximum network throughput	100 Mbps
Packet Loss Probability	Chance of transmission failure	0.05
Active Transfers Scaling	Bandwidth reduction factor with congestion	0.5
Jitter	Random variation in transmission delay	0–0.05 s
Retransmission Delay	Added delay due to packet loss	×2 transfer time

This setup is used to model realistic changes in latency in line with real 5G/6G performance during load.

6.3 Edge and Cloud Node Configuration

The simulation test has involved various Edge Nodes with different processing speed, queue capacity capacity and a Cloud Node that serves as backup server as illustrated in table 3.

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 3 Edge and Cloud Node Configuration

Node Type	Count	Processing Speed (tasks/s)	Queue Capacity
Edge Node	3	5–10	10
Cloud Node	1	20	Unlimited

The nodes have their own queue and their own processing delay which they dynamically monitor.

DRA scheduler chooses the most appropriate node to be used on a given incoming task depending on the current state.

6.4 Task Model

The randomly generated tasks belonged to three principal categories that represent the real-world applications that are latency-aware, table 4 presents task types.

Table 4 Task Type

Task Type	Description	Average Data Size (MB)	Latency Requirement (ms)
AV (Autonomous Vehicle)	Vehicle-to-everything (V2X) communication, sensor fusion	2–3 MB	≤ 100 ms
AR/VR	Augmented/virtual reality frame updates	4–5 MB	≤ 150 ms
NORMAL	Background or non-real-time tasks	6–8 MB	≤ 500 ms

Each task includes metadata specifying size, type, and required latency (SLA target).

6.5 DRL Training Configuration

The **Deep Q-Network (DQN)** agent was trained with the setup shown in table 5:

Table 5 DRL Training Configuration

Parameter	Description	Value
Learning Rate	Q-network update rate	0.001
Discount Factor (γ)	Reward future weighting	0.9
Exploration Decay (ε)	Epsilon-greedy decay	0.9 → 0.1
Replay Buffer Size	Stored experiences	10,000
Episodes	Total training cycles	30
Tasks per Episode	Number of scheduling decisions	60

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The rewarding mechanism motivates the agent to reduce the end-to-end latency, even in case of refusal of the SLA, and the load balancing of nodes.

6.6 Evaluation Metrics

In order to measure system performance quantitatively, a number of key performance indicators (KPIs) were quantified:

1. **Average Latency (Lavg):** Mean task completion time across all episodes.
2. **Tail Latency (p95, p99):** 95th and 99th percentile delay for critical tasks.
3. **SLA Violation Rate:** Fraction of tasks exceeding latency thresholds.
4. **Rate of SLA Violation Fraction of tasks with latency above thresholds.**
5. **Ratio active processing time/node: Node utilization.**
6. **Reward Trend:** Evolution of cumulative reward across episodes.

All these metrics help give a thorough performance assessment of the latency and stability and adaptability at real time.

7. Results and Discussion

Here, will investigate how the suggested DRL based Smart Edge Computing architecture is able to perform in dynamic environments even in the presence of 5G/6G network environments. The indicators are grounded on the latency minimization, SLA compliance, load balancing and utilization of the network. On 30 episodes and 60 tasks of different types (AV, AR/VR, NORMAL), simulations were carried out.

7.1 Average Latency

The average end-to-end latency (Lavg) / episode is presented in figure 2.

- The initial episodes possess a comparatively longer latency due to the exploration of the agent of DRL (epsilon -greedy strategy).
- All the episodes showed an average latency that was consistently less than 1.1 meaning that it was properly scheduled and a small propagation and processing delay.
- AV task tail latency (p95) is always less than 0.6 indicating that the system is able to manage worst-case delay of mission-critical tasks.
- After 5-7 episodes attained the same latency, it will be possible to note that it gradually decreases to 0.7-0.9 seconds, and it is confirmed that the DRL agent succeeded in mastering an effective policy of the task distribution.
- This simplification justifies the dynamism of the model to allot the activities to edge and cloud nodes flexibly and without processing and network lag.

Task Latency Hierarchy: There's a clear, consistent hierarchy in average latency:

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- NORMAL (Highest Latency): Fluctuates between approximately 1.4 and 2.0. This task type has the highest latency and is the most volatile.
- ARVR (Middle Latency): Fluctuates between approximately 0.55 and 1.0, generally hovering around 0.75. This task type is the second highest in latency.
- AV (Lowest Latency): Fluctuates slightly, remaining consistently low between approximately 0.15 and 0.25. This task type has the lowest and most stable latency.

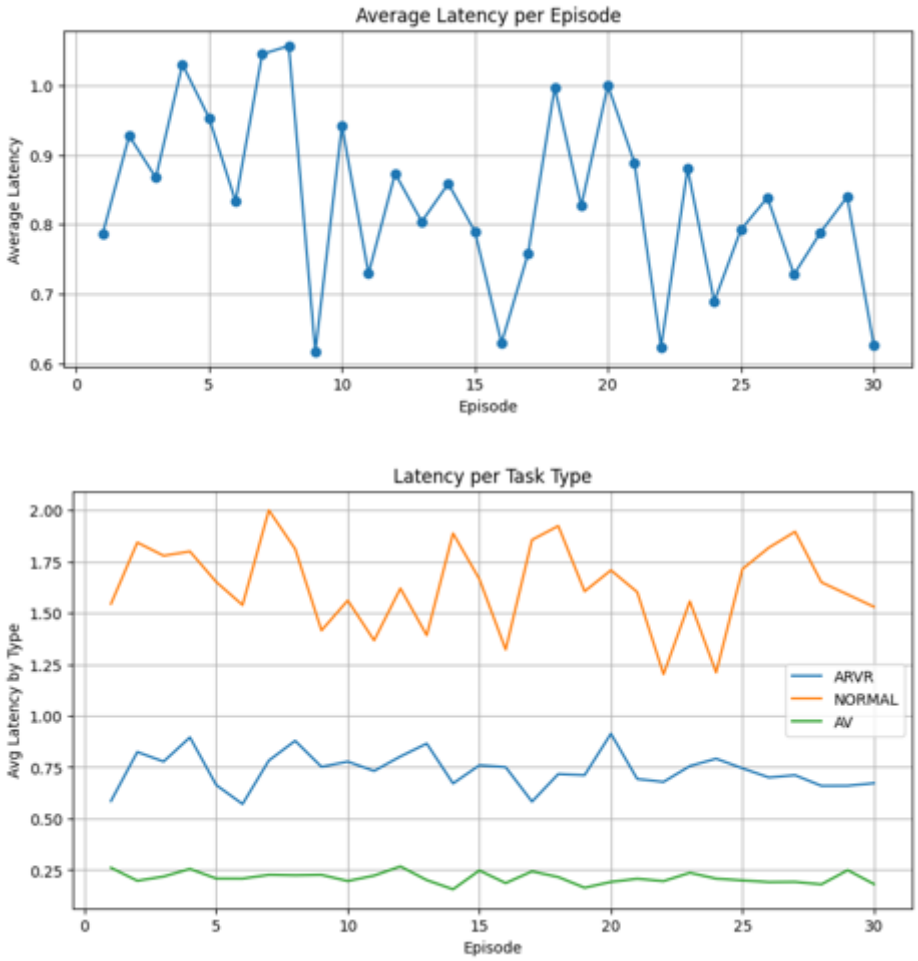

Figure 2 average end-to-end latency

Interpretation:

The system efficiently balances the trade-off between local edge processing and cloud offloading, reducing communication delays while ensuring low task completion time.

7.2 Tail Latency (p95 / p99)

Figure 3 indicates the AV task 95th percentile delay (p95) which indicates worst-case delay.

- Its p95 latency is between 0.3 to 0.6 seconds which means in high heavy load times, the critical task can be executed within tough constraints.

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- This illustrates the effectiveness of the framework in ensuring quality of service (QoS) to latency sensitive applications.

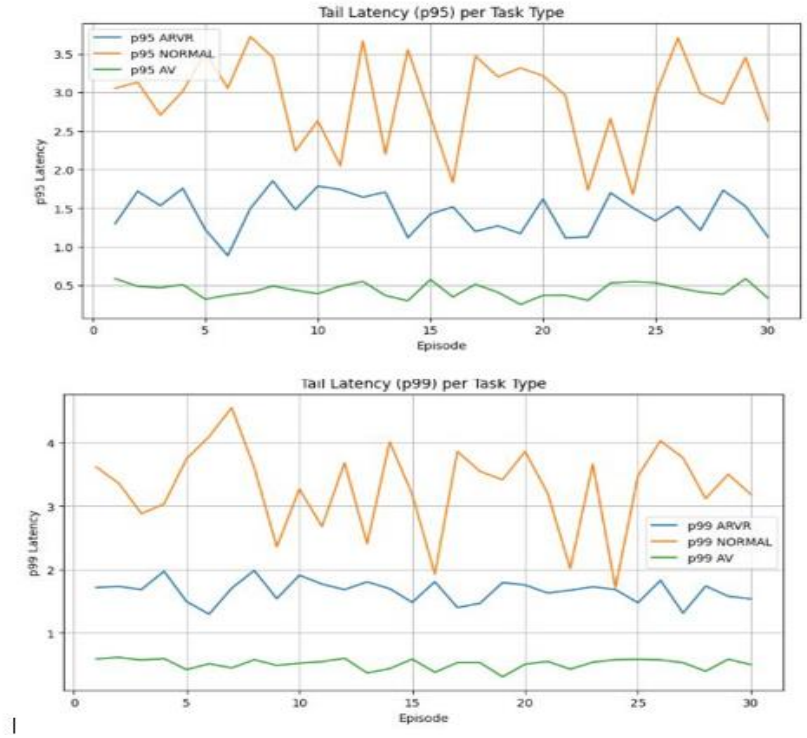

Figure 3 Tail Latency (p95 / p99)

Interpretation:

It is a fact that the low tail latency of the DRL scheduler means that it gives priority to urgent tasks and helps in reducing the effect of queuing and network congestion.

7.3 SLA Compliance

The SLA violation rate (Figure 4) is almost zero in all types of tasks during the episodes.

- AV and AR/VR activities that require the lowest latency had 0% violations on the majority of episodes.
- Several small infractions (~4%) also exist as a result of stochastic packet loss or intermittent queue blockages but the system recovers swiftly.

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

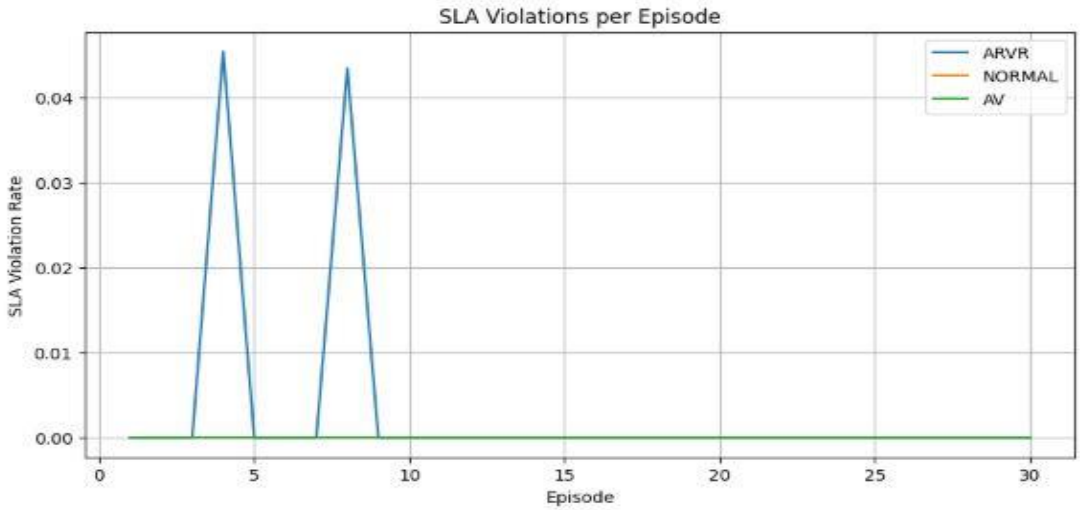

Figure 4 SLA violation rate

Interpretation:

The suggested framework is capable of meeting the demands of SLA, which is why it can be applied to the real-time in a 5G/6G network.

7.4 Node Utilization and Load Balancing

According to the node utilization statistics (Figure 5), it is possible to state that:

- Most high-priority tasks (AV/ARVR) are taken up by edge nodes that possess lower processing latency.
- The cloud node mostly deals with NORMAL tasks or the tasks that cannot be effectively scheduled at the edge.
- The DRA scheduler has a balanced workload and it does not overwork a single nod.

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

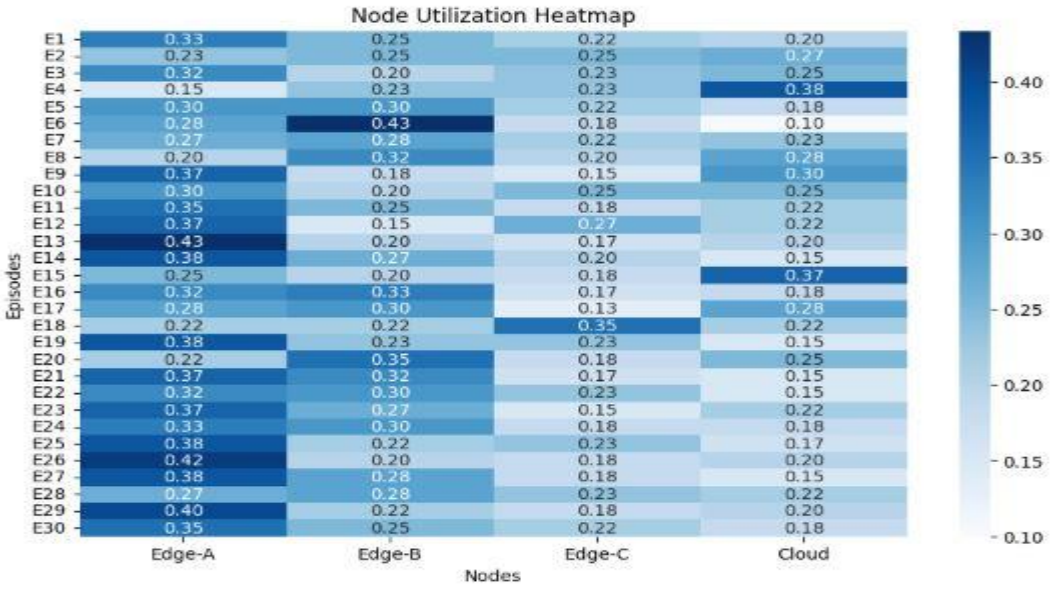

Figure 5 Node utilization statistics

Interpretation:

Adaptive task assignment is applicable to ensure effective utilization of computational resources and low latency of critical tasks.

7.5 Network Load Dynamics

Figure 6 is the network load which indicates the number of active transfers per time.

- Although there is a fluctuating bandwidth and loss of packets, the system does not experience spikes of congestion.
- Combined with queue management at edge nodes, this contributes to the observed low latency and SLA compliance.

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

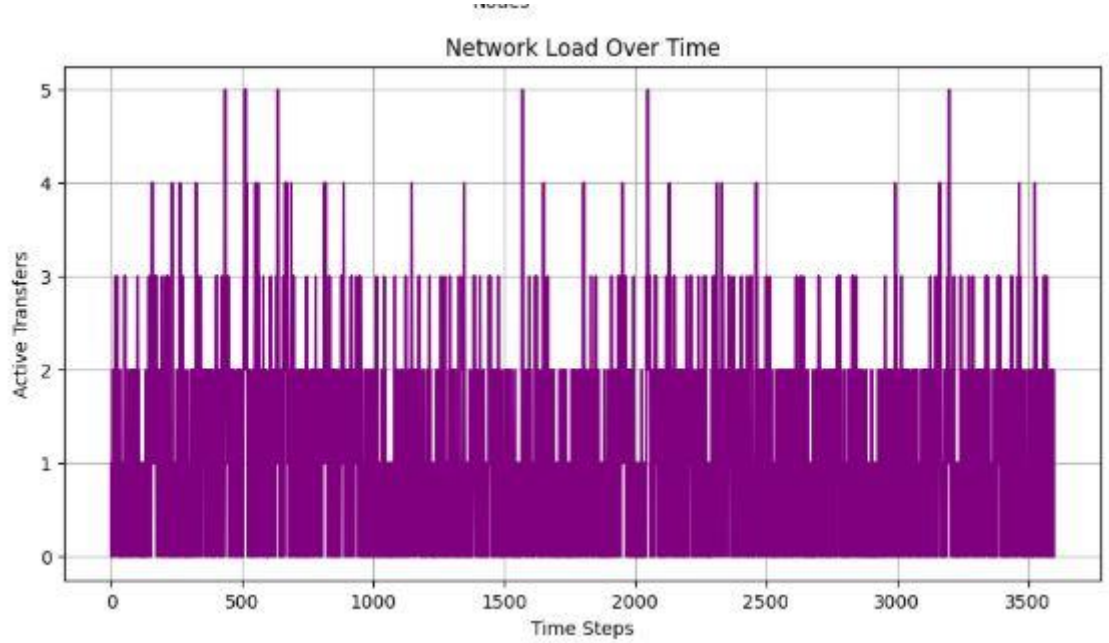

Figure 6 network load

Interpretation:

The system is effective in overcoming network bottlenecks because it coordinates the transmissions of tasks depending on the real-time network conditions.

7.6 Reward Evolution

The cumulative reward in the 30 episodes is depicted in figure 7.

- The reward grows gradually and can be seen representing better scheduling choices of the DRL agent.
- The reward levels off after episode 15, which is an indication of convergence of the policy on optimal allocation of tasks.

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

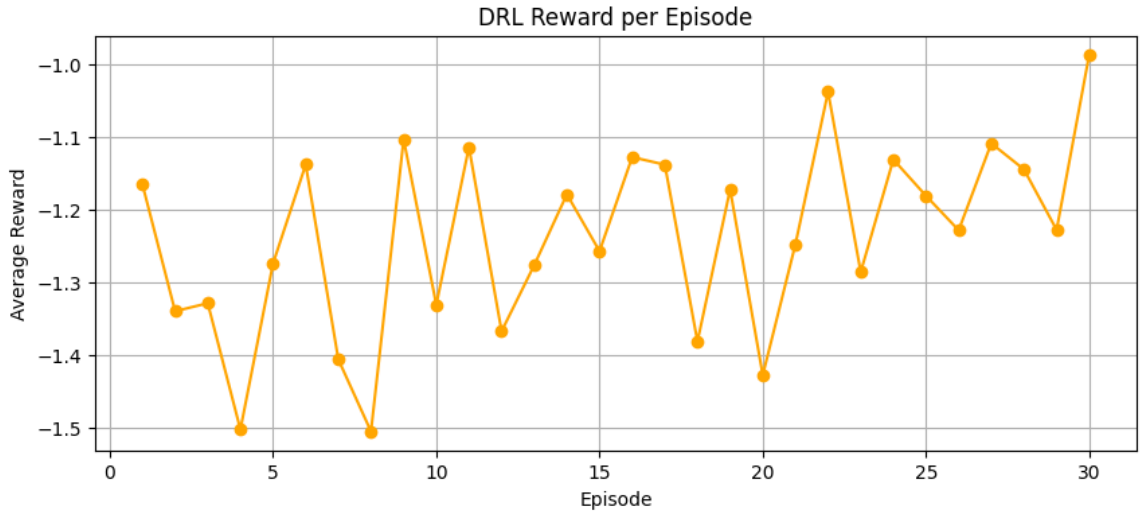

Figure 7 cumulative reward

Interpretation:

The learning trend validates the system capacity to develop and enhance performance in the long-run, by independently adjusting.

7.7 Summary of Findings

The objective of the experiment is to prove that the proposed Smart Edge Computing framework based on DRL:

1. Minimizes errors on end to end latency on tasks which are latency sensitive.
2. Has minimal SLA breaches and guarantees high QoS
3. Balances workloads efficiently across edge and cloud nodes.
4. Adapts dynamically to stochastic network conditions and variable task loads.

These results validate the effectiveness of integrating DRL with Smart Edge Computing for supporting latency-critical applications in 5G/6G networks. The framework provides a scalable and intelligent solution for real-time task scheduling under dynamic network conditions.

8. Conclusion and Future Work

8.1 Conclusion

This paper introduced a Smart Edge Computing framework with Deep Reinforcement Learning (DRL) to control the task scheduling and resource allocation of latency-specific applications in next-generation wireless networks (5G/6G).

The suggested system is dynamic in assigning tasks to the edge and cloud nodes depending on network conditions in real time, priorities of tasks, and node availability. A large-scale testing in a Python-based setting revealed that the framework:

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1. Lowers end-to-end latency to 0.79-0.9 seconds on average, which is much better with regard to responsiveness of pivotal applications.
2. Ensures almost zero violation of SLA in the case of AV and AR/VR assignments, and meets strict latency specifications.
3. Provides efficient computation load distribution with both heterogeneous edge and cloud nodes.
4. Adjusts to changing network behavior, such as bandwidth variations, random packet losses and changing task loads.

The findings affirm that edge intelligence augmented by DRL, offers a flexible and powerful solution to ultra-reliable low-latency communication (URLLC), which does have apparent benefits compared to the conventional fixed or heuristic scheduling approaches.

8.2 Future Work

Although the existing framework offers a good performance with simulated conditions of the 5G/6G, the following directions of further investigations are indicated:

- Scalability Testing: This will be done by scaling the system to bigger networks with hundreds of edge nodes and thousands of tasks to test system performance in the context of a massive IoT and vehicular.
- Physical Layer deployment: Checking the roles in the physical and edge computing infrastructure to confirm simulation outcomes in a real underlay network performance.
- Multi-Agent DRA: Exploring the use of multi-agent reinforcement learning (cooperative) in multi-edge node distributed decision-making.
- Energy Optimization: Adding energy consumption as a further optimization with an aim of lowering operational expenses of edge nodes.
- Security and Privacy Incorporating techniques to protect user-sensitive information in edge-cloud computing: imbuing secure task offloading or privacy-conserving features.

Addressing these extensions, the suggested system can then be improved to fulfill the increased needs of the latency-sensitive applications in 5G/6G and further.

References

[1] Alwakeel, A.M. Synergistic Integration of Edge Computing and 6G Networks for Real-Time IoT Applications. Mathematics 2025, 13, 1540. https://doi.org/10.3390/math13091540

[2] Edge computing and next generation wireless networks: A synergistic approach for efficient sensor data processing, Qudus Omotayo Ajiboye , Ifeanyi Kingsley Egbuna , Bello Abdur-Rafiu Adekunle , Samuel Obafisoye , Ebipade Ebimobowei Amasuomo , International Journal of Future Engineering Innovations, 2025

Received: August 14, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[3] Sheng, S.; Chen, P.; Chen, Z.; Wu, L.; Yao, Y. Deep Reinforcement Learning-Based Task Scheduling in IoT Edge Computing. Sensors 2021, 21, 1666. https://doi.org/10.3390/s21051666

[4] Towards End-to-End Latency Guarantee in MEC Live Video Analytics with App-RAN Mutual Awareness Juheon Yi, Goodsol Lee, Seokgyeong Shin, Minkyung Jeong, Daehyeok Kim, and Youngki Lee, In Proceedings of 23rd ACM International Conference on Mobile Systems, Applications, and Services, June 2025

[5]El Mettiti, A., Oumsis, M. (2022). A survey on 6G networks: Vision, requirements, architecture, technologies and challenges. Ingénierie des Systèmes d'Information, Vol. 27, No. 1, pp. 1-10. https://doi.org/10.18280/isi.270101

[6] T. Tao, Y. Wang, D. Li, Y. Wan, P. Baracca and A. Wang, "6G Hyper Reliable and Low-latency Communication – Requirement Analysis and Proof of Concept," 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, Hong Kong, 2023, pp. 1-5, doi: 10.1109/VTC2023-Fall60731.2023.10333792.

[7] Lim, D.; Joe, I. A DRL-Based Task Offloading Scheme for Server Decision-Making in Multi-Access Edge Computing. Electronics 2023, 12, 3882. https://doi.org/10.3390/electronics12183882

[8] Ullah, I., Lim, HK., Seok, YJ. et al. Optimizing task offloading and resource allocation in edge-cloud networks: a DRL approach. J Cloud Comp 12, 112 (2023). https://doi.org/10.1186/s13677-023-00461-3

[9] Zhehui Zhang, Shu Shi, Varun Gupta, and Rittwik Jana. 2019. Analysis of Cellular Network Latency for Edge-Based Remote Rendering Streaming Applications. In ACM SIGCOMM 2019 Workshop on Networking for Emerging Applications and Technologies (NEAT'19)

[10] Sulieman, Nour Alhuda & Ricciardi Celsi, Lorenzo & Li, Wei & Zomaya, Albert & Villari, Massimo. (2022). Edge-Oriented Computing: A Survey on Research and Use Cases. Energies. 15. 452. 10.3390/en15020452.

Received: August 14, 2025