Intelligent Resource Allocation Scheme for Throughput Enhancement in 6G Networks: A Smart Hospital Perspective

Team Number: C1

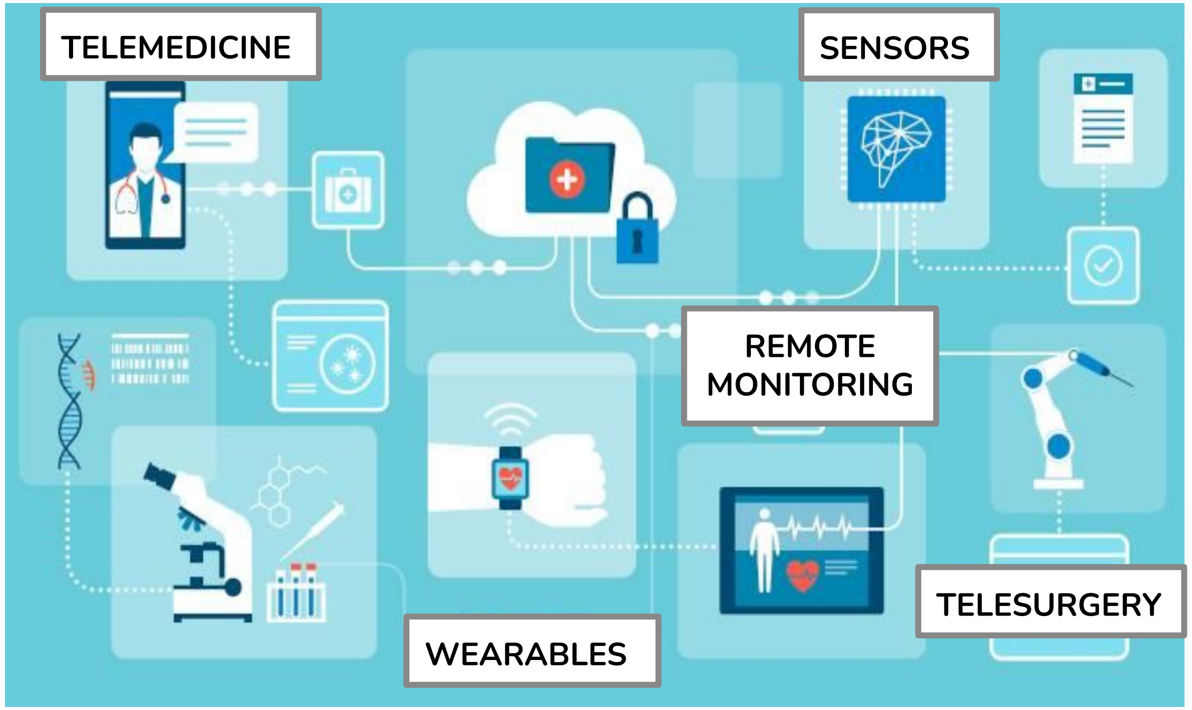
AM.EN.U4CSE20234: Hrishika Dayan
AM.EN.U4CSE20243: Krishnapriya V S
AM.EN.U4CSE20250: Namita Suresh
AM.EN.U4CSE20269: Srilakshmi S R

Guide : Dr. Simi S

Smart Hospitals

India has → **1.4 beds per 1,000 people**
1 doctor per 1,445 people
1.7 nurses per 1,000 people.

Over 75% of the healthcare infrastructure is concentrated in metro cities, where only **27% of the total population resides**



Key Limitations of Current Healthcare

SPACE
- Geographical Distance
- Capacity Constraints

TIME
- Response Time
- Appointment wait times

Increasing adoption of smart hospital technologies + Proliferation of IoT devices

Surge in demand for
- High-speed Internet access
- Efficient management of Connections
- Optimized resource allocation

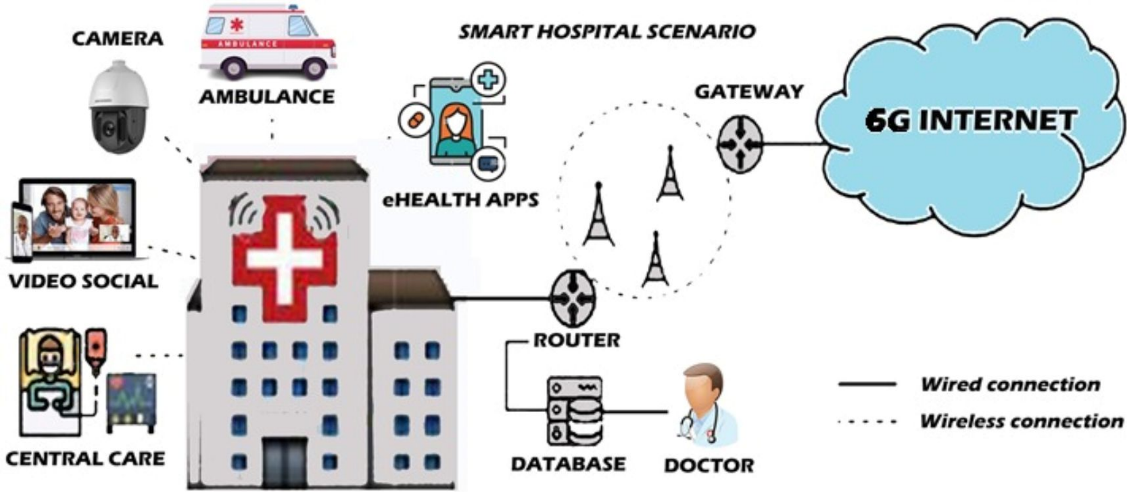
Growing Demand for
- Wireless connectivity
- Seamless networks

Need of faster and improved technologies.

Communication requirement in Smart Hospital

Addressing **bandwidth management** to prioritize critical applications and ensure smooth smart hospital operations.

1. Increased bandwidth allowing smart hospitals to efficiently handle the high-density medical data.
2. Low latency for remote healthcare such as remote surgeries, telemedicine consultations.
3. Ultra-fast data transfer facilitating real-time sharing of medical data, high-resolution images.



Smart Hospital + 6G

- Real-time remote consultations with low latency and high bandwidth
- IoT-enabled healthcare including wearable health monitors, smart medical equipment, and sensors.
- Enhanced privacy and security of sensitive patient data

Background Study

Title & Year	Problem	Contributions	Limitations	Open problems/Future work
Lu, Shuaibing, et al. "A Dynamic Service Placement Based on Deep Reinforcement Learning in Mobile Edge Computing." Network 2.1 (2022): 106-122.	Modern Mobile Edge Computing (MEC) grapples with the intricate challenge of optimizing service placement for diverse users with dynamic needs, balancing factors such as user requirements, limited edge server resources, and fluctuating workloads, often hindered by the limitations of static rules and heuristic algorithms.	1. Proposes a DRL-based framework, utilizing DDPG, for dynamic service placement in MEC. 2. The DRL agent optimally places services , considering user requirements and resource availability.	1. Single-user focus; extending the framework for multi-user interactions 2. Integrating security and privacy mechanisms into the DRL framework is crucial for secure and confidential data transmission during offloading processes..	1. Adapting the DRL framework for multi-user scenarios, introducing dynamic interactions and resource sharing mechanisms. 2. Incorporating real-time network conditions and user mobility patterns for enhanced adaptability in the decision-making process.
Chen, Zhao, and Xiaodong Wang. "Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach." EURASIP Journal on Wireless Communications and Networking 2020.1 (2020): 1-21.	Addressing the challenge of optimizing computation offloading decisions in multi-user mobile edge computing (MEC) environments by mitigating high communication overhead and potential delays associated with centralized approaches relying on a base station (BS).	. 1. Utilization of the DDPG algorithm for individual users to optimize offloading decisions. 2. Improved system delay, energy efficiency, and adaptability to dynamic changes.	1. Basic network models , not reflect real wireless channels and dynamics fully. 2. The paper lacks exploration-exploitation trade-offs in DRL.	1. Real-world network modeling: Incorporating realistic models for dynamic wireless channels and network delays. 2. Security and privacy: Integrating considerations for secure and confidential data transmission in the DRL framework.

Background Study

Title & Year	Problem	Contributions	Limitations	Open problems/Future work
Hortelano, Diego, et al. "A comprehensive survey on reinforcement-learning-based computation offloading techniques in Edge Computing Systems." Journal of Network and Computer Applications 216 (2023): 103669.	Edge computing faces challenges of constrained local resources and network delays, impacting application performance. Offloading computations to the cloud offers relief, yet complexities arise from dynamic factors like fluctuating resource availability, network bandwidth, and application requirements.	1. Presents a comprehensive survey of RL-based computation offloading techniques in edge computing systems. 2. It categorizes and analyzes existing works based on various aspects, RL algorithms employed, and considered application scenarios .	1. Limited coverage of offloading approaches , such as optimization-based methods. 2. Restricted discussion on practical implementation challenges and considerations for real-world deployment.	1.Offloading approaches integrating RL with other techniques for enhanced robustness and adaptability. 2. Application of multi-agent RL for facilitating multi-device collaboration and resource sharing in edge networks.
Chen, Yan, et al. "Dynamic task allocation and service migration in edge-cloud iot system based on deep reinforcement learning." *IEEE Internet of Things Journal* 9.18 (2022): 16742-16757.	Achieving sustained optimal system performance in edge-cloud IoT systems requires dynamic task allocation and effective service migrations, ensuring continuous service while adapting to evolving demands.	1.Formulates the dynamic task allocation and service migration (DTASM) problem as an MDP. 2.Proposes a deep reinforcement learning (DRL) based approach.	1.Limited evaluations on a small-scale simulation. 2. Lacks real-world experiments.	1.Incorporate more realistic networking characteristics into the system model. 2.Generalize the DRL model to account for diverse application types, mobility patterns and wireless environments.
Salameh, Ahmed I., and Mohamed El Tarhuni. "From 5G to 6G—challenges, technologies, and applications." *Future Internet* 14.4 (2022): 117.	The limitations of 5G networks in meeting increasing demands for higher data rates, lower latency, and ultra-high reliability have spurred the emergence of 6G research to address these shortcomings and facilitate new applications and use cases.	1.Offers a comprehensive overview of 6G, while reviewing research activities from 2015-2020. 2. Summarizes surveys on 6G technologies.	1.The discussion on 6G technologies is high-level and does not provide in-depth technical or any details 2. Does not discuss about th real world scenarios.	Detailed technical designs need to be developed for the key 6G technologies. Experimental research and trials are needed to validate the 6G concepts.

Project Contributions

- Deep Reinforcement Learning based resource allocation scheme that optimizes the allocation of bandwidth in smart hospitals.
- Introduction of Deep Q-Network (DQN) algorithm to the existing Markov Decision Process (MDP) framework in the context of smart hospital resource allocation.
- Comparison of the performance of Deep Q-Network (DQN) and Deep Deterministic Policy Gradient (DDPG) algorithms for resource allocation.

Objectives

1. **Comparative performance analysis** between DQN and DDPG algorithms in terms of resource allocation efficiency.
2. **Review Existing Research –** gather knowledge and insights from prior work.
3. **Assess the applicability and effectiveness** of different DRL algorithms in the context of smart hospital resource allocation.

Outcomes

1. **Model Learning -** Achieve successfully trained deep reinforcement learning model.
2. **Resource Optimization -** Achieve computational resource (edge capability) and communication resources (throughput) optimization.
3. **Identification of algorithmic strengths and weaknesses** in the context of smart hospital environments

Assumptions

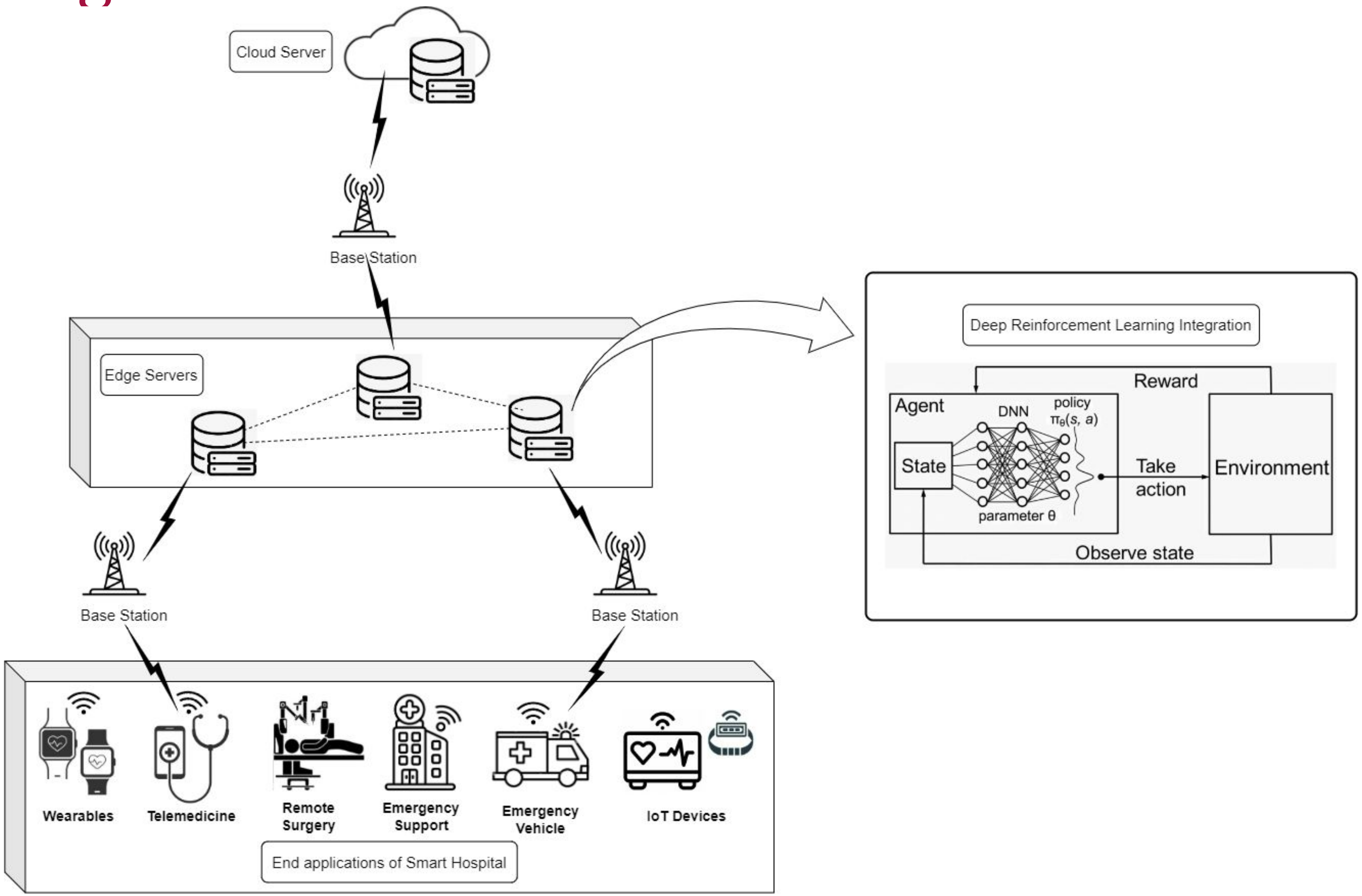
1. **6G network infrastructure** is deployed in the smart hospital.
2. **IoT devices and applications are compatible** with the 6G network

Contribution 1:

Communication Requirement Analysis of Smart Hospital

End applications of Smart Hospitals	Maximum Latency	Minimum Throughput
Telemedicine	100 milliseconds	1 Mbps
Remote Surgery	20 milliseconds	100 Mbps
Wearable Device Communication	50 milliseconds	100 Kbps
Emergency Support	100 milliseconds	1 Mbps
Emergency Vehicle	50 milliseconds	1 Mbps

High Level Design: DRL Model for Resource Allocation



Progress After First Review

1. **Revised MDP model** - Refined our Markov Decision Process formulation for improved resource allocation in smart hospitals. This included:
   - Enhanced State Space: We expanded the state space to encompass CPU utilization and network load (number of users) for a more comprehensive understanding of the environment.
   - Reward Function Update: The reward function was modified to include factor for computational resources as well as communication resources, ensuring efficient allocation.

2. **Introducing Deep Q-Network (DQN):** Implemented a new algorithm DQN within the established MDP framework. This allows us to compare the performance of DQN with DDPG (used in Phase 1) for resource allocation.

Progress After First Review

3. Experimental Design and Analysis: Conducted extensive experiments to evaluate both DQN and DDPG implementations. The experiments focused on:

- Impact of User and Edge Server Variations: We investigated how changes in user numbers and edge server availability affect the learning process and agent performance.
- Convergence of Algorithms: We ensured both DQN and DDPG models achieved convergence in the reward curve, confirming effective learning.

4. Three Evaluation Cases: To comprehensively analyze the system's behavior under various conditions, we designed three experimental cases:

Cases	Users	Edges	Evaluation criteria
Case 1 (Balanced Resource Distribution)	10	10	Baseline for balanced resource allocation
Case 2 (Higher Computing Demands)	15	10	Tests model's adaptability to increased computational needs
Case 3 (Resource Surplus)	10	15	Evaluates system efficiency with excess resources

Progress After First Review

5. Detailed Performance Analysis: For each case, we analyzed the following metrics through plots:

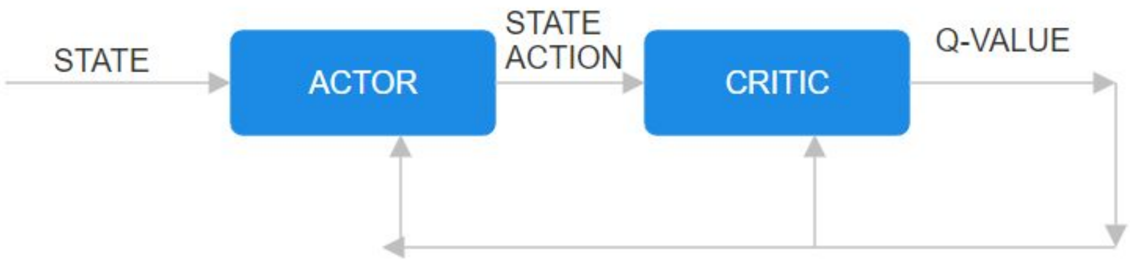
- Reward vs. Episode: Evaluates the model's learning progress.
- Throughput vs. Episode: Measures the system's ability to deliver data efficiently.
- Edge Capability vs. Episode: Assesses the utilization of computational resources at edge servers.

6. Further Analysis (Plots): Created additional plots to delve deeper into how throughput and edge capability behave under changing user and edge server configurations. These plots specifically examine:

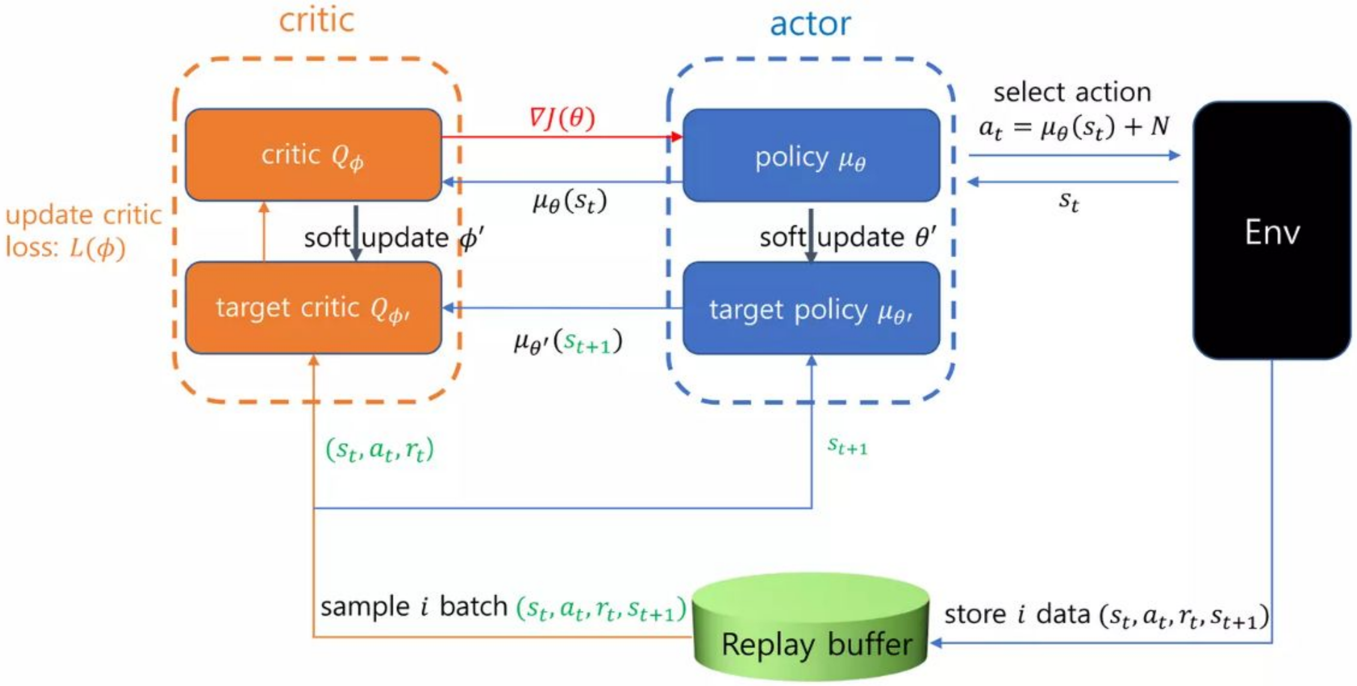
- Throughput vs. Number of Edges (constant users)
- Throughput vs. Number of Users (constant edges)
- Edge Capability vs. Number of Edges (constant users)
- Edge Capability vs. Number of Users (constant edges)

Algorithms

DDPG



1. Initialization
2. Interaction
3. Experience Replay
4. Critic Network Update:
   - Calculate Target Q-value
   - Minimize Loss
5. Actor Network Update:
   - Maximize Q-value
   - Update Policy
6. Target Network Update (Soft Update)
7. Repeat



Algorithms

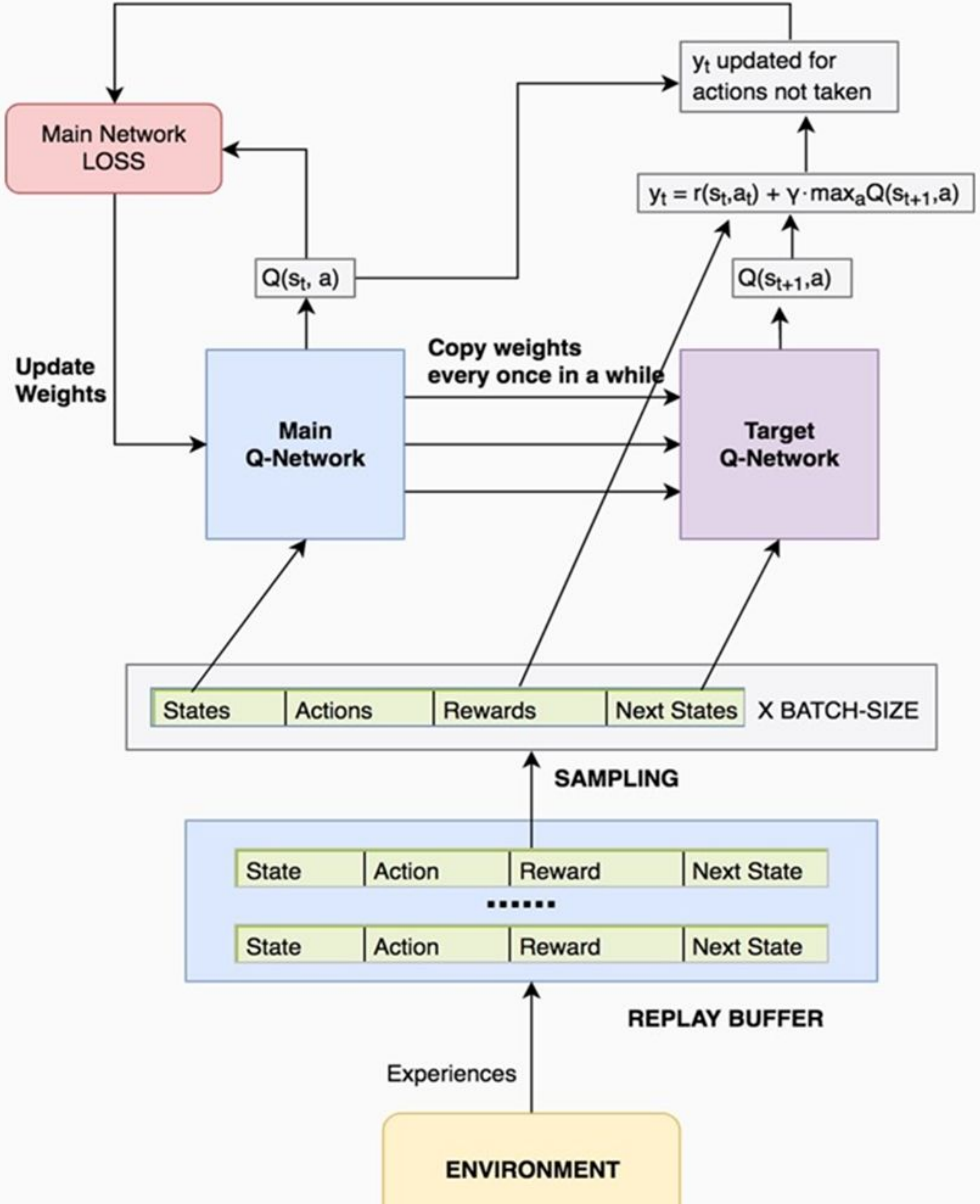
Deep Q-Network

1. Initialization

2. Interaction
   a. Epsilon-Greedy Policy

3. Experience Replay
   a. Target Q-value
   b. Minimise Loss

4. Repeat



Methodology: Design of MDP

Optimization Objective - Efficiently allocate resources to users and edge servers.

State Space(S): $\{ <CPU_{util}, P_{avail}, B_{avail}, D_{trans}, N_{users}, T_{off}> \}$

CPU_{util} : CPU utilization of each edge server

P_{avail} : Available computing resources of each edge server

B_{avail} : Available migration bandwidth of each connection between edge servers

D_{trans} : Data Transmitted

N_{users} : Current load of the network

T_{off} : offloading target of each mobile user

Action Space(A): $\{ <Ap, Ab, Ot> \}$

A_p : Allocating computing resources for each mobile user's task

A_b : Allocating migration bandwidth for each mobile user's task

O_t : Offloading target of each mobile user

Reward Function:

$$d_{\text{trans}} = d_{\text{req}} - d_{\neg\text{trans}}$$

$$T_{\text{avg}} = \frac{\sum_{i=1}^{n} d_{\text{trans}}}{n}$$

$$R = \mu \cdot T_{\text{avg}} + (1 - \mu) \cdot E_c$$

Experimental Setup - MDP Model

1. User and Edge Server Configuration:

Cases	#Users	#Edge Servers	Evaluation Criteria
Case 1	10	10	Balanced resource distribution
Case 2	15	10	Higher computing resource demands
Case 3	10	15	Demand exceeds resources availability

2. Tuning parameters :

Symbol	Definition	Default Value
L_{RA}	Actor learning rate	0.0001
LR_C	Critic learning rate	0.0002
γ	Discount factor	0.9
τ	Soft replacement	0.01
N	Batch size	32
E	Maximum no. of steps	1000
M	Maximum no. of episode	500

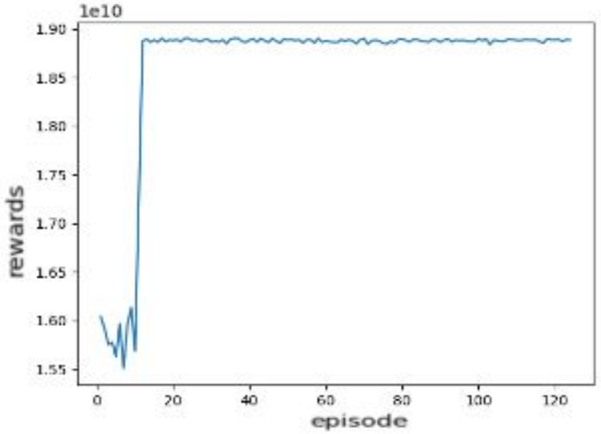
3. Performance evaluation factors : Analysis of reward, throughput, edge capability are done by generating plots against episodes.

Analysis 1: Evaluation of user and edge server configurations on model learning

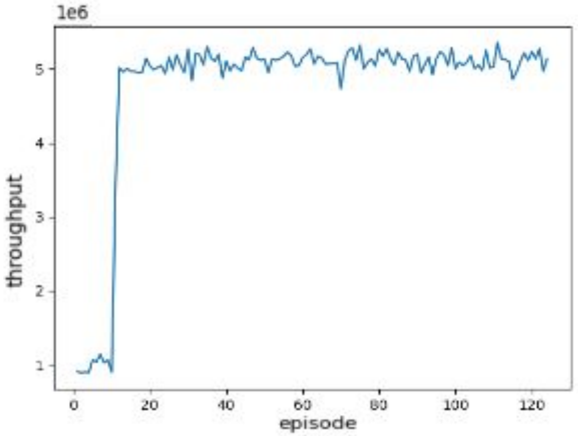
DDPG Algorithm Results

- Case 1: Balanced Resource Distribution with Equal Number of Users and Edge Servers

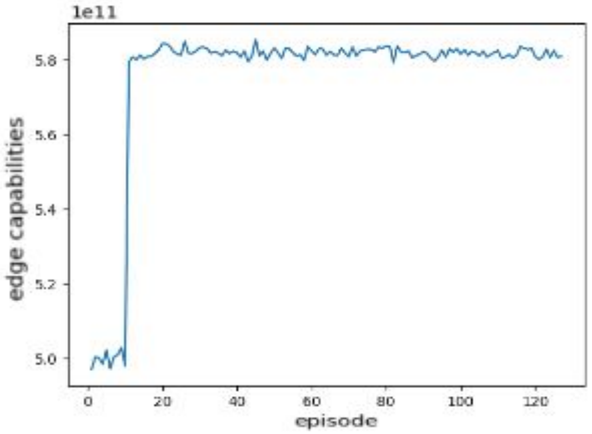
user_num: 10
edge_num: 10



(a) Cumulative Reward vs Episode



(b) Throughput vs Episode

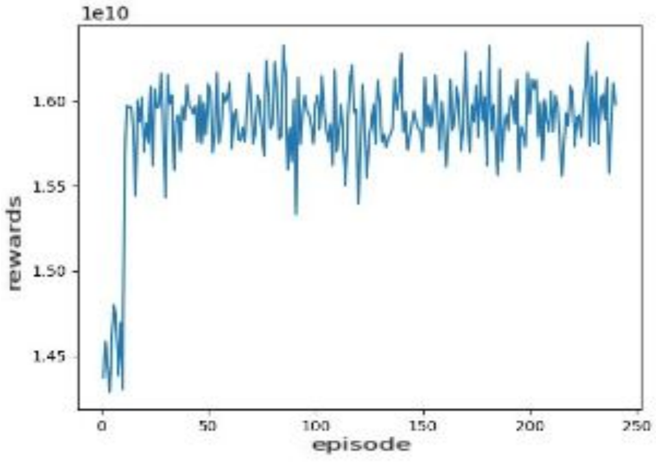


(c) Edge capabilities vs Episode

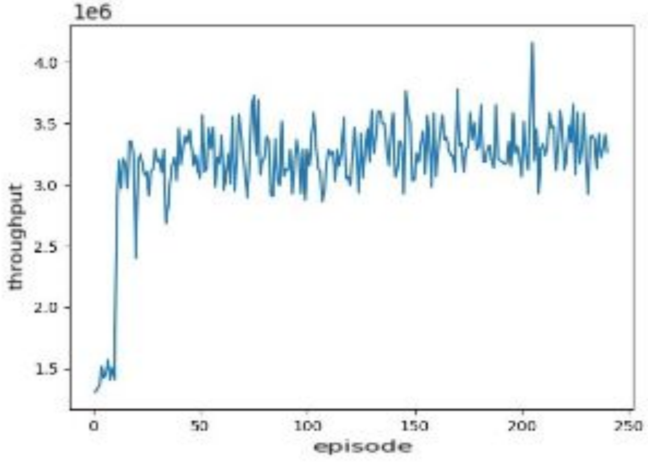
DDPG Algorithm Results

- Case 2: Higher User Demand with More Users than Servers

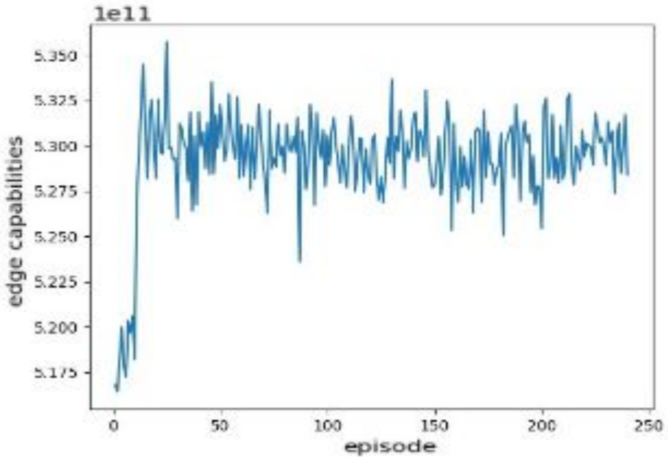
user_num: 15
edge_num: 10



(a) Cumulative reward vs Episode



(b) Throughput vs Episode

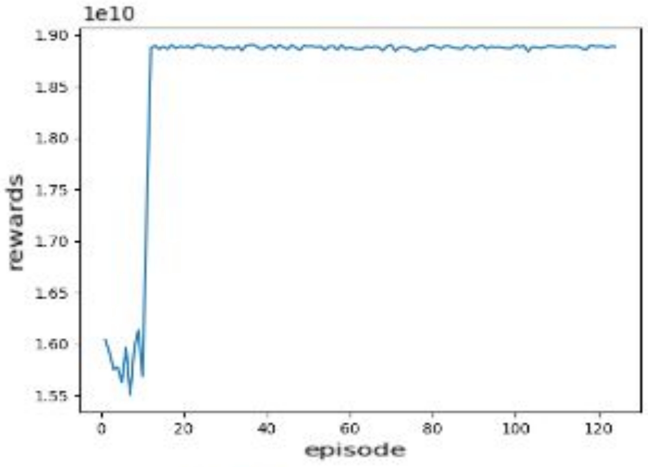


(c) Edge capabilities vs Episode

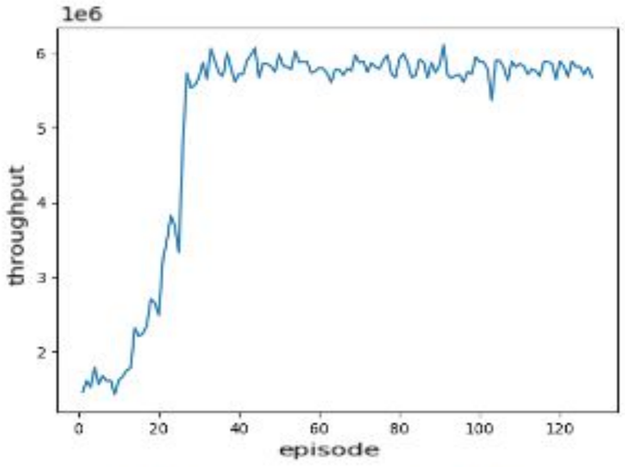
DDPG Algorithm Results

- Case 3: Excess Resource Distribution with More Servers than Users

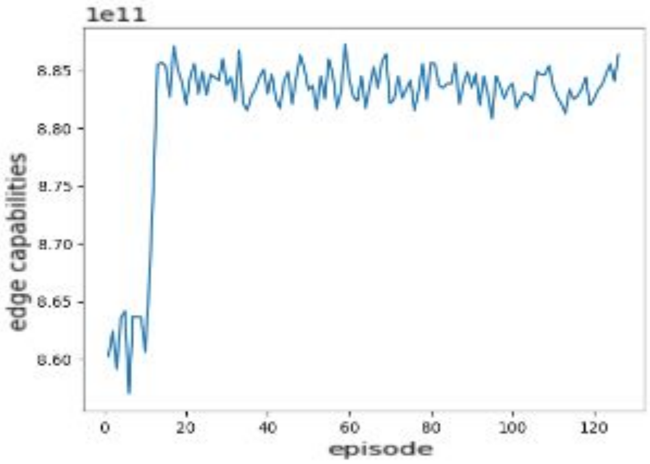
user_num: 10
edge_num: 15



(a) Cumulative Rewards vs Episode



(b) Throughput vs Episode

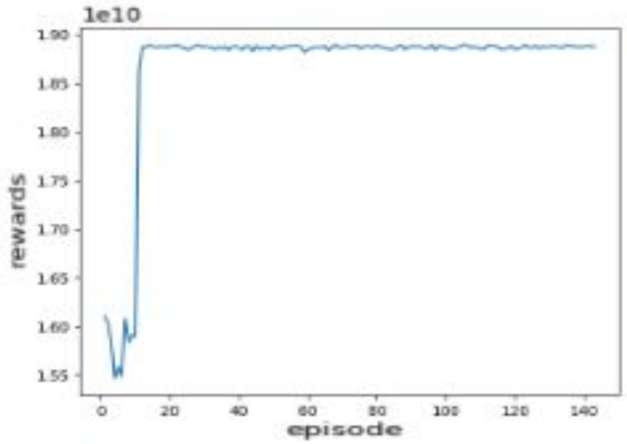


(c) Edge capabilities vs Episode

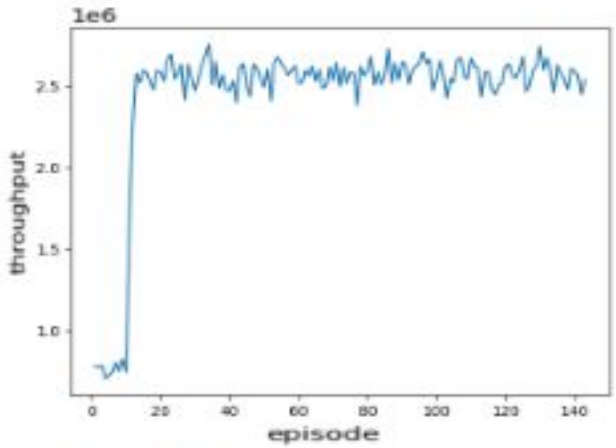
DQN Algorithm Results

- Case 1: Balanced Resource Distribution with Equal Number of Users and Edge Servers

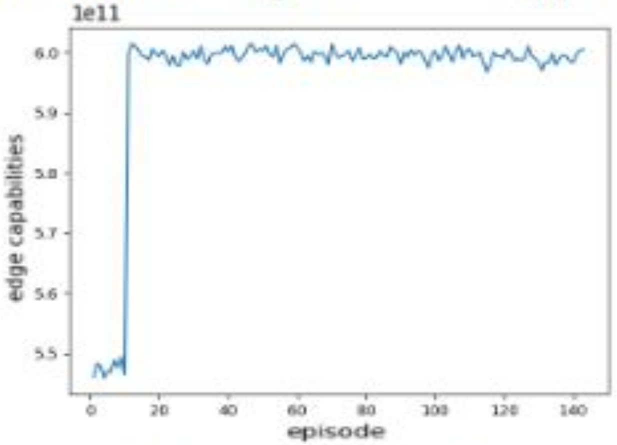
user_num: 10
edge_num: 10



(a) Cumulative Reward vs Episode



(b) Throughput vs Episode

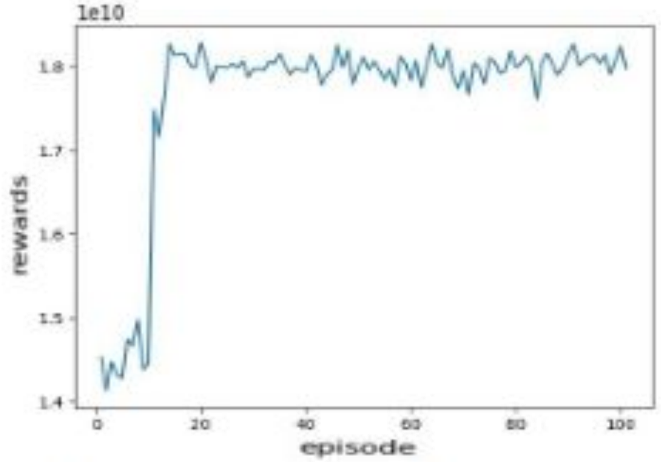


(c) Edge capabilities vs Episode

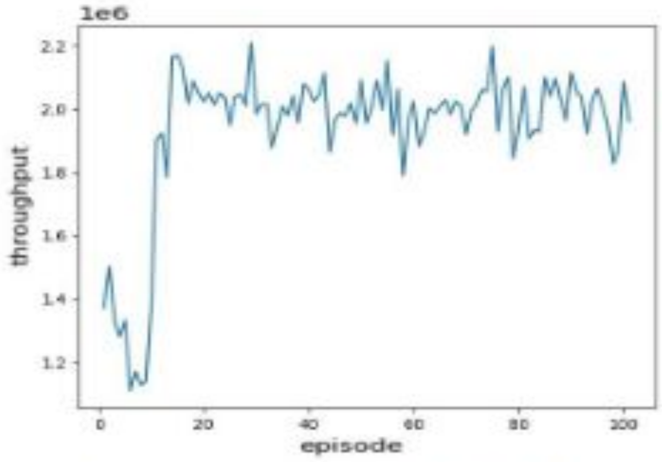
DQN Algorithm Results

- Case 2: Higher User Demand with More Users than Servers

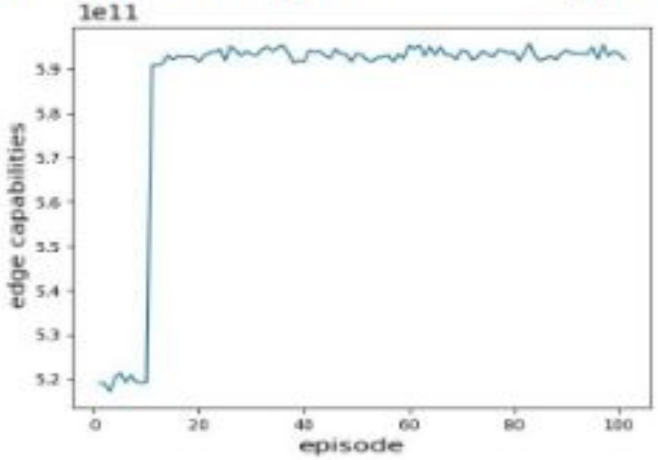
user_num: 15
edge_num: 10



(a) Cumulative reward vs Episode



(b) Throughput vs Episode

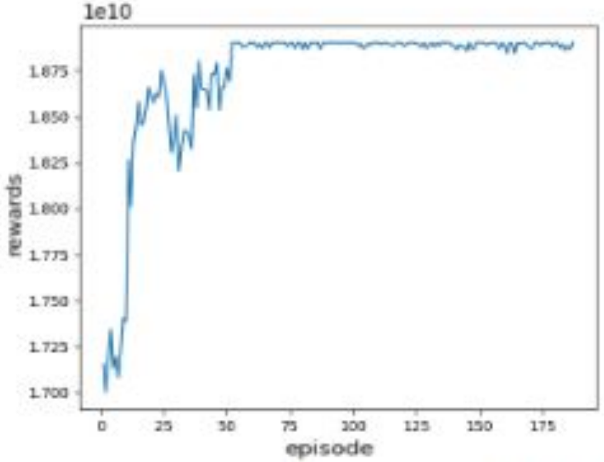


(c) Edge capabilities vs Episode

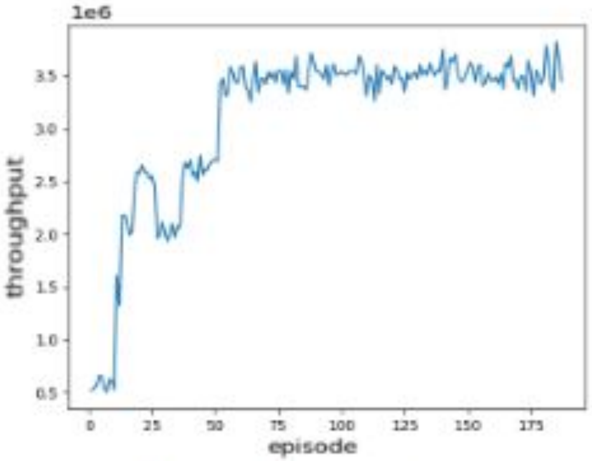
DQN Algorithm Results

- Case 3: Excess Resource Distribution with More Servers than Users

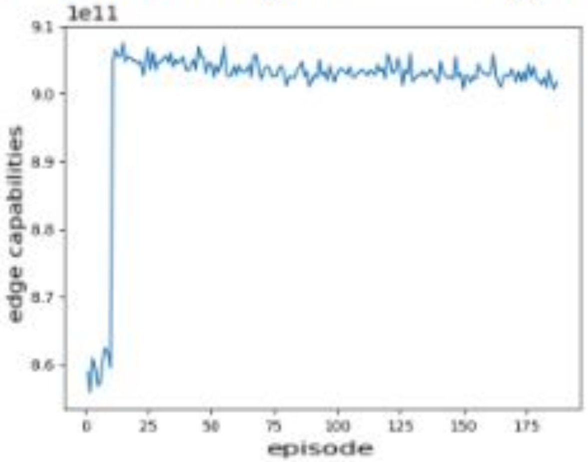
user_num: 10
edge_num: 15



(a) Cumulative Rewards vs Episode



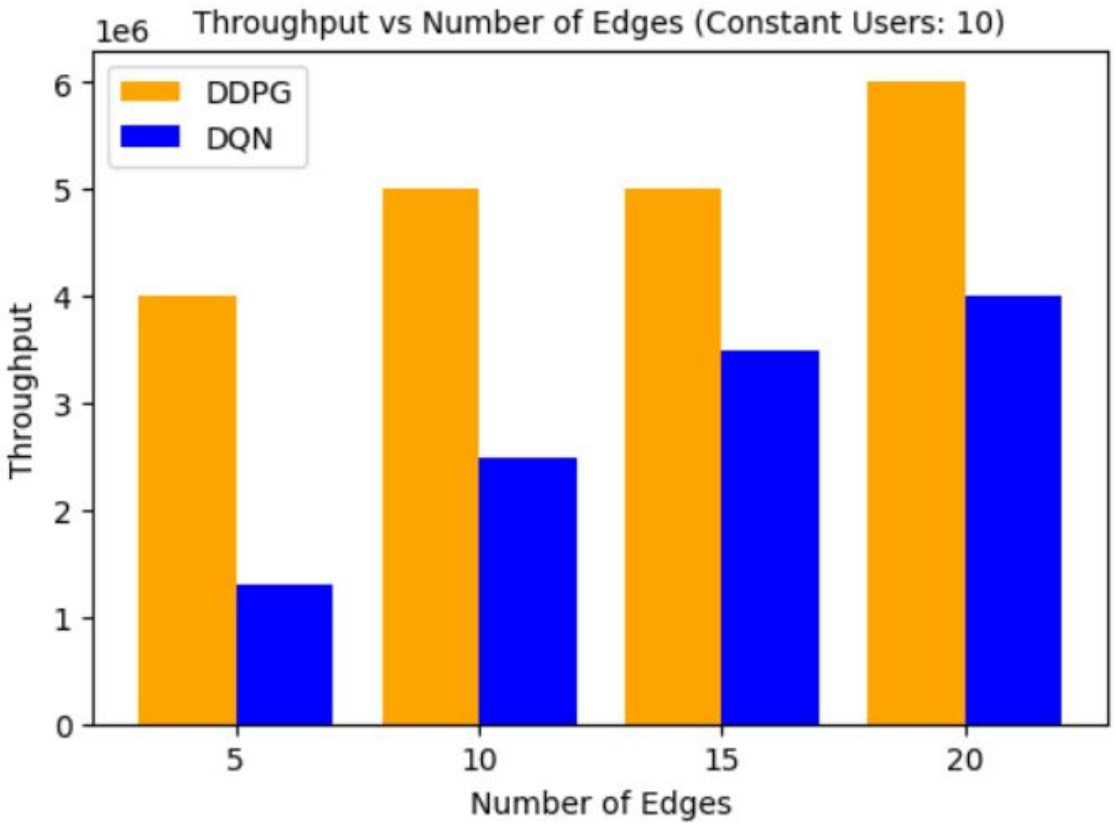
(b) Throughput vs Episode

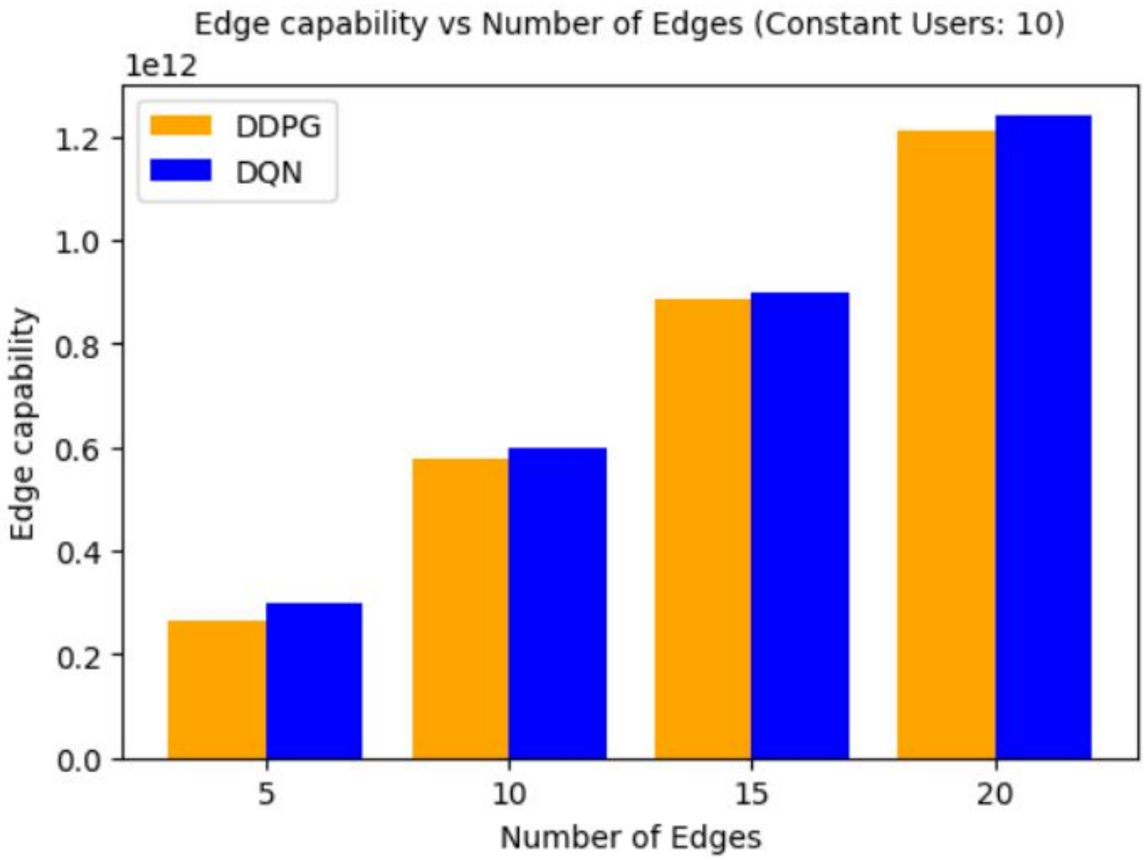


(c) Edge capabilities vs Episode

Analysis 2: Evaluation of throughput and edge capability in dynamic environments considering varying user demands and resource capacity

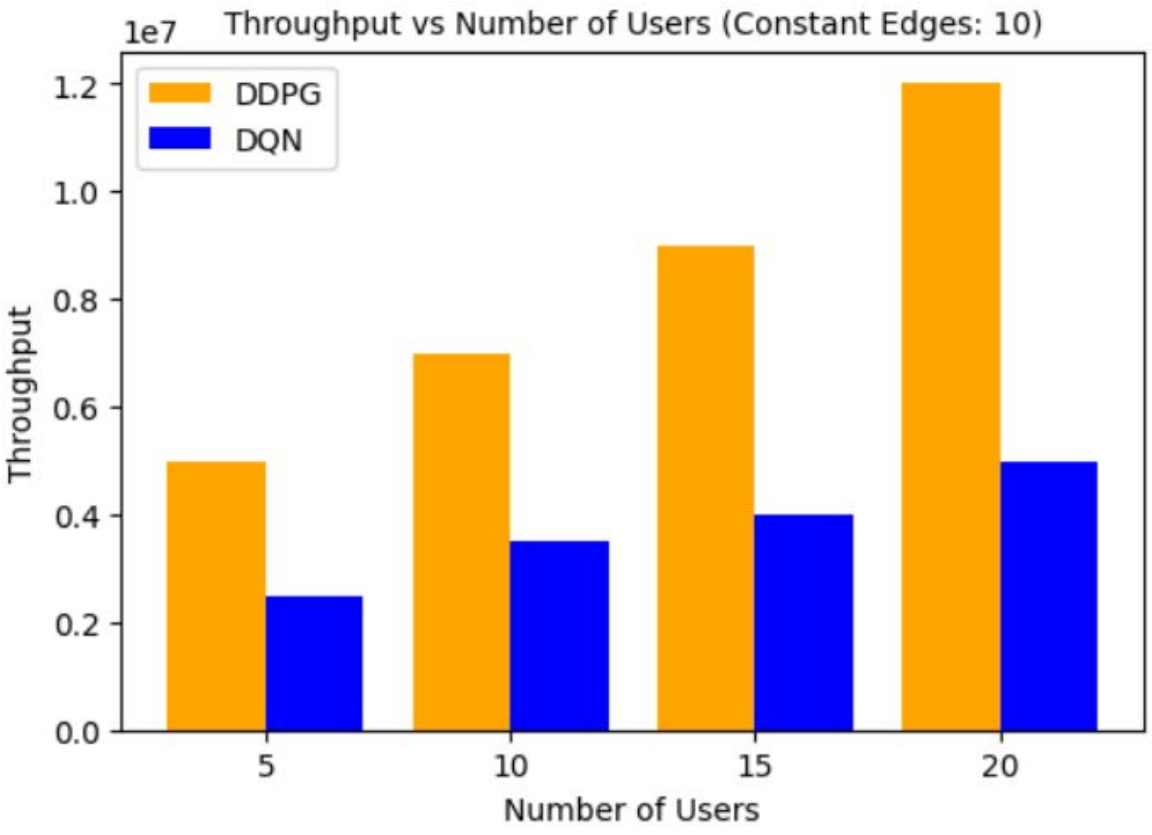
Case 1:Varying Number of Edges with Constant Users:

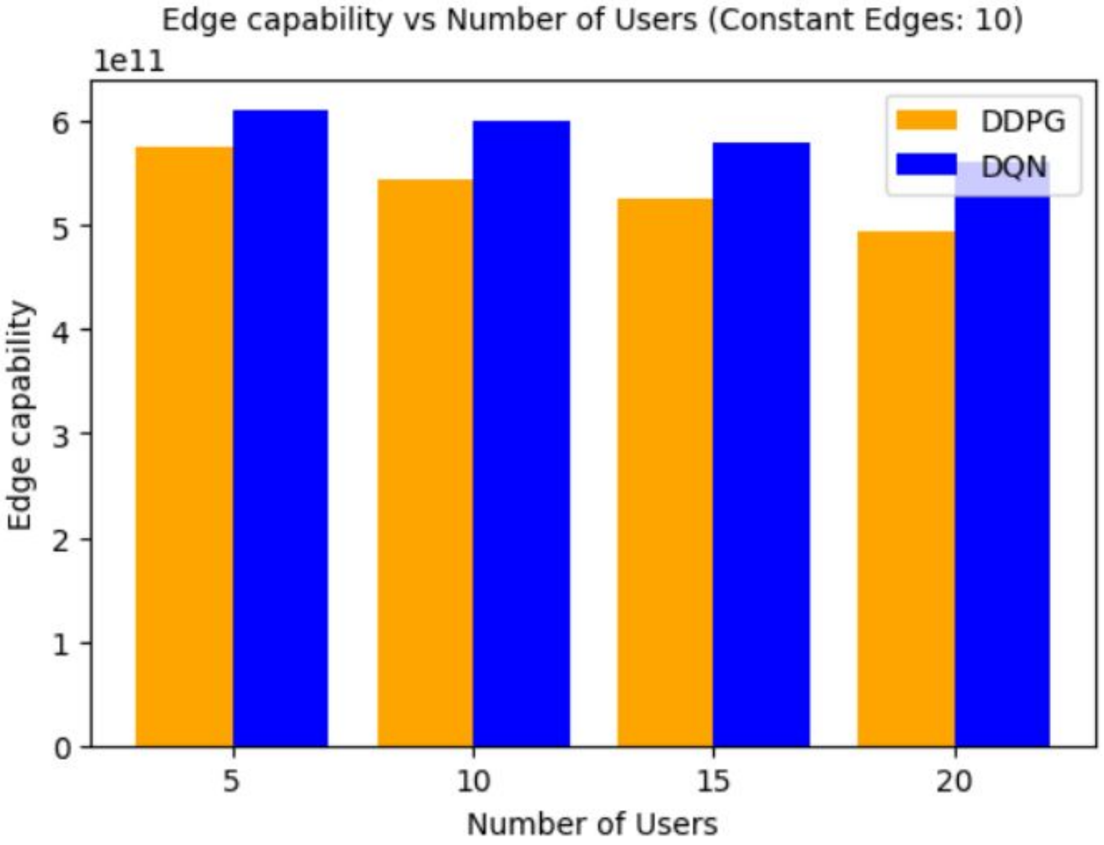




Analysis 2: Evaluation of throughput and edge capability in dynamic environments considering varying user demands and resource capacity

Case 2:Varying Number of Users with Constant Edges:





Findings

Reward Convergence : The generated plots for reward against episode converges ensuring that the models are actually learning.

Constant Users Scenario**:**

DDPG: Edge Capability: Increases from $2.65 * 10^{11}$ to $1.21 * 10^{12}$ as the number of edges increases from 5 to 20.
Throughput: Increases from $4 * 10^6$ to $6 * 10^6$ with the increase in the number of edges.

DQN: Edge Capability: Shows a similar increasing trend with the number of edges.
Throughput: Also increases with the number of edges.

Constant Edges Scenario**:**

DDPG: Throughput and edge capability remain constant across different numbers of users.

DQN: Throughput varies with the number of users but remains relatively stable.
Edge capability remains almost constant.

Effect of Number of Edges**:**

Increasing the number of edges leads to higher throughput and edge capability, as seen in both algorithms.

Effect of Number of Users**:**

- DQN shows variations in throughput with different numbers of users in the constant edges scenario.
- DDPG maintains a constant throughput and edge capability regardless of the number of users.

Conclusion

- DDPG and DQN adapt their resource allocation strategies based on real-time data and performance metrics, potentially outperforming static techniques like FCFS and SJF.

- Both DDPG and DQN show promising performance in optimizing throughput and edge capability in edge computing scenarios

  .

- DDPG generally achieves higher maximum throughput compared to DQN in most scenarios, making it more suitable for certain edge computing use-cases.

Conferences Targeted

International Journal of Information Technology - https://link.springer.com/journal/41870

References

[1] Lu, Shuaibing, et al. "A Dynamic Service Placement Based on Deep Reinforcement Learning in Mobile Edge Computing." Network 2.1 (2022): 106-122.

[2] Chen, Zhao, and Xiaodong Wang. "Decentralized computation offloading for multi-user mobile edge computing: A deep reinforcement learning approach." EURASIP Journal on Wireless Communications and Networking 2020.1 (2020): 1-21.

[3] Hortelano, Diego, et al. "A comprehensive survey on reinforcement-learning-based computation offloading techniques in Edge Computing Systems." Journal of Network and Computer Applications 216 (2023): 103669.

[4Chen, Yan, et al. "Dynamic task allocation and service migration in edge-cloud iot system based on deep reinforcement learning." IEEE Internet of Things Journal 9.18 (2022): 16742-16757.]

[5] Zhang, Cheng, and Zixuan Zheng. "Task migration for mobile edge computing using deep reinforcement learning." Future Generation Computer Systems 96 (2019): 111-118.

[6] Salameh, Ahmed I., and Mohamed El Tarhuni. "From 5G to 6G—challenges, technologies, and applications." Future Internet 14.4 (2022): 117.

[7] Rui, LanLan, et al. "Service migration in multi-access edge computing: A joint state adaptation and reinforcement learning mechanism." Journal of Network and Computer Applications 183 (2021): 103058.

Link to the Project Demo: demo_finalyr.mp4

GitHub Link: Final year project code