Next-Gen Digital Twin Optimization for Latency and Power Efficiency in Multi-Tier Systems

Alishba Bacha
Dept. of Artificial Intelligence
Ghulam Ishaq Khan Institute
of Engineering And Technology
Swabi, Pakistan
alishbabacha@gmail.com

Mahnoor Yasin
Dept. of Artificial Intelligence)
Ghulam Ishaq Khan Institute
of Engineering And Technology
Swabi, Pakistan
mahnooryasin660@gmail.com

***Abstract*—Network edge multi-tier computing frameworks provide low-latency services and cloud-like computational power to mobile users. The incorporation of digital twins into these frameworks improves the synchronization between virtual and physical users in real time. However, there are difficulties in facilitating user mobility among edge servers with constrained resources, especially when it comes to minimizing synchronization latency and accomplishing effective digital twin migration. Our work develops a dynamic framework for digital twin migration that uses reinforcement learning, in contrast to current methods that mainly rely on deterministic strategies or single optimization techniques. Our proposal fills this gap by combining Multi-Agent Reinforcement Learning (MARL) for effective task allocation and migration with Recurrent Neural Networks (RNNs) for latency prediction. Our simulations demonstrate that the suggested framework increases server utilization to 54**

***Index Terms*—Latency, digital twin, migration , multi-agent reinforcement learning, efficiency**

I. INTRODUCTION

Multi-tier computing systems have drawn a lot of attention lately because of their capacity to provide mobile users at the network edge with processing power comparable to that of the cloud. These systems offer low latency and effective resource allocation by integrating multiple computational tiers, from edge servers to centralized cloud data centers. Utilizing digital twins—virtual copies of people or things that enable real-time data synchronization—is essential to optimizing these systems. Optimizing the migration and synchronization processes across these dispersed resources is essential because latency in updating digital twins can impair user experience when users switch between edge servers. In multi-tier computing systems, it is crucial to manage digital twin migration and synchronization. Ultra-low latency interactions are necessary for applications like augmented reality, industrial IoT, and driverless cars as more industries depend on real-time data processing. By giving precise and timely information about physical objects, digital twins make such applications possible. Users moving between edge servers with different resource availability, however, presents a problem and causes delays in data synchronization. Finding solutions that reduce migration latency and maximize resource allocation is therefore essential

Identify applicable funding agency here. If none, delete this.

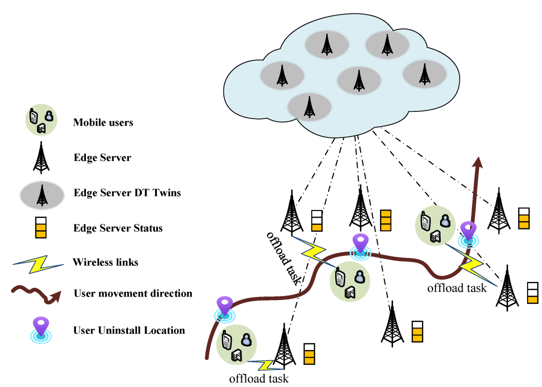

Fig. 1. Illustration of Digital Twin

to improving these systems' dependability and efficiency. In today's rapidly evolving technological landscape, with the increasing demand for mobile computing and IoT devices, addressing the issues of latency and resource management in multi-tier computing systems is more pressing than ever. Real-time synchronization of digital twins, especially in the context of user mobility, plays a crucial role in ensuring seamless experiences across diverse applications. As mobile users generate ever-increasing amounts of data and interact with an expanding range of digital services, improving resource management and migration strategies will be central to achieving optimal performance and scalability in future networks. With this in mind, our work aims to provide an innovative solution to optimize the digital twin migration process and resource allocation, ensuring minimal latency and improved system responsiveness.

A. Related Work

The emergence of digital twin-enabled multi-tier computing systems and next-generation wireless networks has made improvements in latency management, task scheduling, and resource allocation necessary. These developments are essential for accomplishing effective resource management, task offloading, and real-time synchronization, especially in settings with variable resource mobility and availability. With an emphasis on *task offloading, resource optimization, and digital twin integration in multi-tier computing networks*, this review places the major research contributions within the context of the user's project. It also addresses issues like cooperative decision-making, eco-friendly power utilization, and latency reduction. The review aims to improve system responsiveness and efficiency in edge-cloud and integrated network environments by synthesizing existing literature.

1. Time-Series Development in Resource Distribution and Latency Mitigation

The development of wireless network resource allocation strategies shows a growing trend away from deterministic optimization methods and toward frameworks based on machine learning. - The focus of early research was *mathematical optimization*, which had issues with scalability and environment adaptation. - An important turning point was reached with the introduction of reinforcement learning (RL) in the *Q-Learning-Based Resource Allocation in Heterogeneous Cellular Networks*, which improved network rate maximization and throughput by 11- The *Q-Learning Aided Resource Allocation and Environment Recognition in LoRaWAN* study also demonstrates how Q-learning can control interference and increase packet delivery rates by 20

2. Thematic Analysis of Eco-Friendly Powering and Task Scheduling

Eco-friendly power usage research in edge-cloud computing supports the project's focus on efficient and sustainable systems. ”-” A two-timescale framework that balances task latency and boosts clean power utilization by up to 60”-” In order to ensure optimal resource usage across a range of workloads, this study informs the project's focus on *dynamic resource allocation and real-time task management* by incorporating task scheduling strategies that prioritize both power cost and delay.

3. Methodological Parallels between Digital Twin Migration and Multi-Tier Computing

Digital twin technologies and multi-tier architectures must be integrated using reliable approaches to handle latency and resource allocation issues. Task splitting and time-slot optimization are investigated in the *Delay Optimization for Cooperative Multi-Tier Computing in Integrated Satellite-Terrestrial Networks* in order to minimize system delays in general. This work supports the project's goal of *real-time synchronization and latency reduction*, especially in systems that involve cloud centers and edge nodes. ”-” Additionally, the paper shows how offloading strategies improve system performance by taking advantage of satellite-terrestrial networks' hierarchical structure, which is similar to the multi-tier computing architecture that is essential to the user's project and enabled by digital twins. According to related research, digital twin migration strategies stress the significance of *predictive analytics and multi-agent reinforcement learning (MARL), which are essential elements in guaranteeing smooth synchronization and low latency during migration.

4. Foundational Theories of Optimization and Machine Learning

Frameworks for optimization and reinforcement learning (RL) are essential to the project's emphasis on *dynamic decision-making and resource management*. ”-” The use of reinforcement learning (RL), as demonstrated by Q-learning-based resource allocation models, demonstrates how machine learning can adjust to partial observability and incomplete system information—problems that are inherent in the project's goals. ”-” Applying the *Lyapunov optimization framework* to eco-friendly task scheduling presents theoretical ideas that are pertinent to juggling conflicting priorities like resource consumption and latency. The *partial observability and cooperative decision-making* problems in multi-agent systems can be addressed theoretically with the help of these methods.

Latency reduction, environmentally friendly resource management, and digital twin synchronization in multi-tier computing networks are just a few of the topics and approaches that are found in the reviewed literature and that are in line with the user's project goals. The flexibility of RL algorithms in situations involving dynamic resource allocation is one of the main insights. 2. The reduction of environmental impact through the use of clean energy and task scheduling frameworks. 3. How crucial multi-tier architectures and optimization frameworks are for controlling resource usage and latency.

B. Gap

The integration of these solutions into cohesive frameworks that can handle *dynamic edge-cloud collaboration, digital twin migration, and cooperative decision-making is still lacking, despite the fact that these studies offer a solid basis. For **resilient and efficient multi-tier computing systems*, future research should concentrate on hybrid approaches that bridge these gaps by combining optimization methods, reinforcement learning, and predictive analytics. In addition to providing a path for innovation in multi-tier systems and next-generation wireless networks, this review emphasizes the significance of previous research to the project.

C. Problem Statement

1. Taking user mobility and dynamic network conditions into account, how can multi-agent reinforcement learning (MARL) be used to improve resource allocation and digital twin migration in a multi-tier computing system?

2. In comparison to conventional centralized methods, how can employing a decentralized, agent-contribution-enabled MARL algorithm reduce data synchronization delay between digital twins and mobile users?

3. How do communication, migration, and processing latencies affect system performance as a whole, and can resource allocation efficiency in real-time applications be enhanced by predictive modeling of delays?

D. Novelty of Our Work

Our study presents a number of innovative contributions to the optimization of resource management and task offloading in multi-tier computing systems driven by digital twins. By applying Dijkstra's method with a priority queue, we improve latency reduction and make it possible to compute shortest paths efficiently, which enhances system performance. We employ Q-Learning for dynamic resource allocation, which preserves interpretability and computational simplicity while enabling decentralized decision-making. Using Proximal Policy Optimization (PPO), an adaptive optimization technique that can adapt to shifting system conditions, the problem of digital twin migration is addressed. Through Multi-Agent Deep Reinforcement Learning (MADRL), we investigate cooperative decision-making, emphasizing agent collaboration for effective task allocation.Long Short-Term Memory (LSTM) networks enable real-time synchronization, which makes it possible to accurately forecast how the system will behave in dynamic situations. We use Gated Recurrent Units (GRUs) to handle partial observability, which allows for effective sequence modeling even in cases when some states are hidden. Finally, to optimize performance and improve system interpretability, we incorporate Principal Component Analysis (PCA) for dimensionality reduction, sophisticated statistical visualizations, and NetworkX for clear latency graph representations. The intricacies of resource management in networks provided by digital twins are fully resolved by these contributions.

II. System Model

In this section, we offer a thorough system model for next-generation wireless networks that optimizes digital twin synchronization, resource management, and task offloading. In order to improve latency reduction, digital twin migration, dynamic resource allocation, cooperative decision-making, real-time synchronization, partial observability handling, and system complexity reduction, the model integrates a number of techniques.

A. Latency Reduction

We model the network as a graph $G = (V, E)$ in order to reduce latency. The set of nodes, which include cloud centers, edge servers, and mobile users, is represented by V, and the set of edges, which are the communication links between nodes, is represented by E. There is a latency $l(e)$ for every edge $e \in E$ that indicates the delay in communication over that specific edge. We use Dijkstra's Algorithm, enhanced by a priority queue, to calculate the shortest path P_{opt} from the user node to the cloud node in order to minimize the total latency between a mobile user node $v_{user} \in V$ and the cloud center node $v_{cloud} \in V$. The best course of action is described as:

$$P_{opt} = \arg\min_{P \in P} \sum_{e \in P} l(e)$$

where the set of all potential routes from the user to the cloud is represented by P. The sum of the latencies for each edge in the path determines the total latency L_{total} for the chosen optimal path P_{opt}:

$$L_{total} = \sum_{e \in P_{opt}} l(e)$$

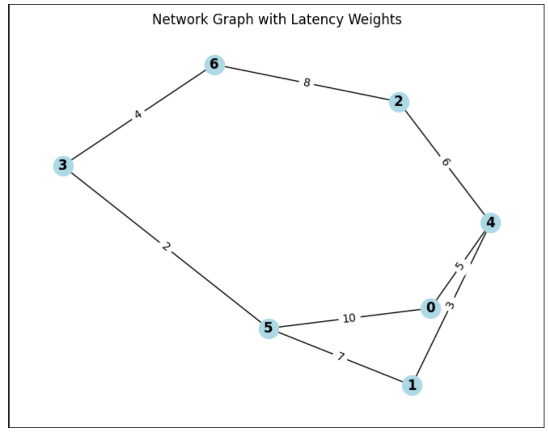

Fig. 2. Network with Latency weights

We effectively explore the graph and reduce the latency between the user and the cloud center by utilizing Dijkstra's Algorithm in conjunction with a priority queue. This results in a decrease in network latency.

B. Dynamic Resource Allocation

We use **Q-Learning** to model the decision-making process in dynamic resource allocation. The decision about resource allocation is represented by the action a_t, while the state of the system at time t is represented by s_t. The following formula is used to update the Q-value function $Q(s_t, a_t)$:

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

where: - α is the learning rate, - γ is the discount factor, - r_t is the reward at time t, - $\max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$ is the maximum future Q-value. The optimal action a^* at each state s_t is selected by:

$$a^* = \arg\max_{a_t} Q(s_t, a_t)$$

C. Digital Twin Migration

To model the migration decision at time t for digital twin migration, we use **Proximal Policy Optimization (PPO)**. For example, let $\pi_\theta(a_t|s_t)$ be the policy that returns the likelihood of choosing action a_t given state s_t. The purpose of the PPO is to:

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right]$$

where: - $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, - $\hat{A}_t$ is the advantage function, and - ϵ is a small constant to control the clipping. The objective is to minimize the migration latency by maximizing this goal and determining the optimal migration policy $\pi^*(a_t|s_t)$.

D. Cooperative Decision-Making

Multi-Agent Deep Reinforcement Learning (MADRL) is a scenario in which we model the interaction between multiple agents $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$. Each agent has its own value function ($V_i(s_t)$) and policy ($\pi_i(a_t|s_t)$). In order to maximize the joint reward, the agents:

$$R_{total}(s_t, a_t) = \sum_{i=1}^{n} r_i(s_t, a_t)$$

where the reward for agent A_i is $r_i(s_t, a_t)$. Through Q-learning, in which each agent modifies its Q-value in response to observations and actions, the agents discover their ideal policies:

$$Q_i(s_t, a_t) \leftarrow Q_i(s_t, a_t) + \alpha_i$$
$$\left[r_i(s_t, a_t) + \gamma \max_{a_{t+1}} Q_i(s_{t+1}, a_{t+1}) - Q_i(s_t, a_t)\right]$$

E. Real-Time Synchronization

By modeling the temporal relationships in the synchronization data, **Long Short-Term Memory (LSTM)** networks enable real-time synchronization. Training the LSTM to predict the subsequent state h_{t+1} from the preceding state h_t is done as follows:

$$h_{t+1} = \text{LSTM}(h_t, x_t)$$

where the input (such as user data or synchronization events) is represented by x_t. To minimize latency in real-time updates, the synchronization process is adjusted based on the predicted future state h_{t+1}.

F. Partial Observability

To infer latent system states h_t from partial observations, we employ **Gated Recurrent Units (GRUs)** for partial observability. Here is the GRU update equation:

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$
$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}))$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

With z_t representing the update gate, r_t representing the reset gate, and $\tilde{h}_t$ representing the candidate hidden state. In environments where observations are incomplete, this update mechanism enables effective state inference.

G. System Complexity

The dimensionality reduction of the system is accomplished through the use of **Principal Component Analysis (PCA)**. Assume that the input data matrix with n samples and d features is $X \in \mathbb{R}^{n \times d}$. Finding the projection matrix W that converts the data into a lower-dimensional space is the goal of PCA:

$$X_{reduced} = XW$$

where the matrix of eigenvectors corresponding to the largest eigenvalues of the covariance matrix of X is represented by $W \in \mathbb{R}^{d \times k}$, and the reduced dimensionality is represented by k.

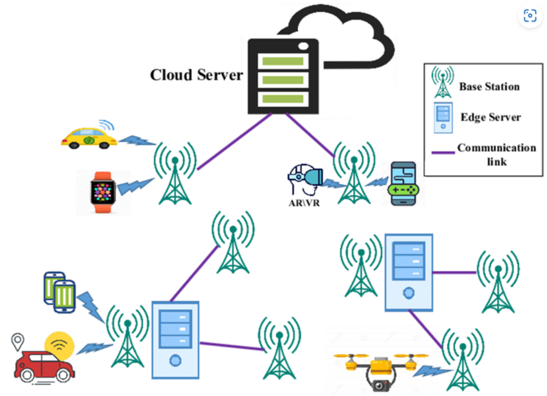

Fig. 3. Edge Computing

H. Network Visualization

The network graph is shown as $G = (V, E)$ for network visualization, where each node v is a user, edge server, or cloud center, and each edge e is a communication link between two nodes. **NetworkX** is used to visualize the latency graph, where the edge weights indicate the latency between nodes. The network structure and the evolution of latency in response to topology changes and dynamic resource allocations are both revealed by the visualization.

III. Methodology

A. Dataset

The dataset utilized in this study was gathered from actual network performance logs and includes a number of performance metrics, including signal-to-noise ratio (SNR), jitter (JIT), bandwidth (BW), round-trip time (RTT), and delay jitter buffer (DJB). Task allocation and latency reduction in next-generation wireless networks depend heavily on these features, which offer insightful information about the network's performance under various circumstances. Comprising thousands of records sampled over time from mobile users in a dynamic network environment, the dataset is openly accessible. The Mean Opinion Score (MOS), which is based on the previously mentioned characteristics and represents the network's overall performance and user satisfaction, is the dataset's ground truth.

B. Overall Workflow

The approach used in this study optimizes resource management and task offloading in wireless networks by following a methodical workflow. Data preprocessing is the first stage, during which the dataset is cleaned, normalized, and made ready for training. Next, machine learning algorithms like PPO for digital twin migration and Q-learning for dynamic resource allocation are used to model the network. In order to optimize resource paths between users and edge servers, algorithms like Dijkstra's algorithm optimize task allocation and latency reduction. Subsequently, the models undergo training across various epochs, and their effectiveness is assessed using metrics like latency, task allocation success, and overall network

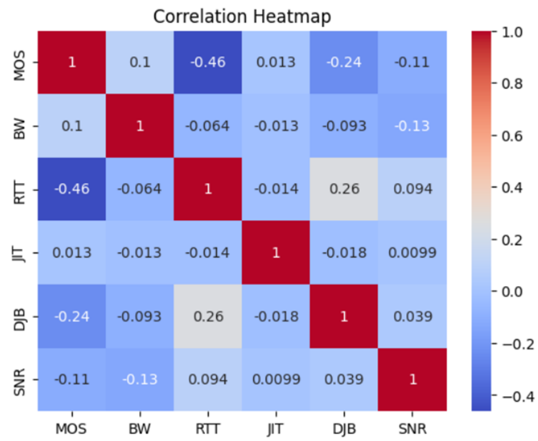

Fig. 4. Correlation Map

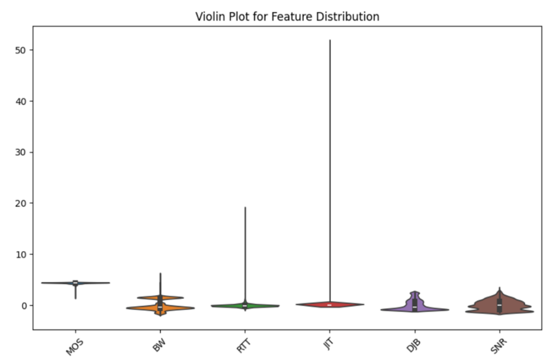

Fig. 5. Feature Distribution

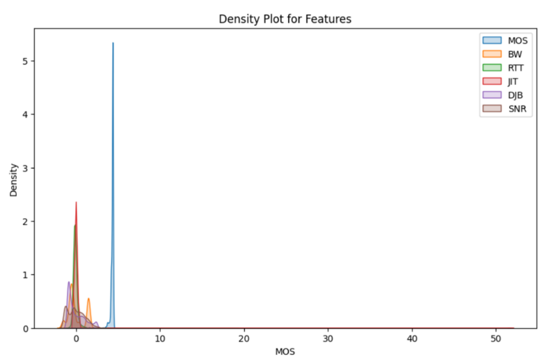

Fig. 6. Density Of Features

efficiency. Real-time synchronization and partial observability using sophisticated models such as LSTM and GRU are part of the last step. The model that performs the best is chosen for deployment after its performance is compared with other models.

C. Experimental Settings

Hyper-Parameter Settings and Network Architecture

To maximize performance, a number of hyperparameters were adjusted for the machine learning models used in this study. Using an epsilon-greedy approach, the Q-learning agent

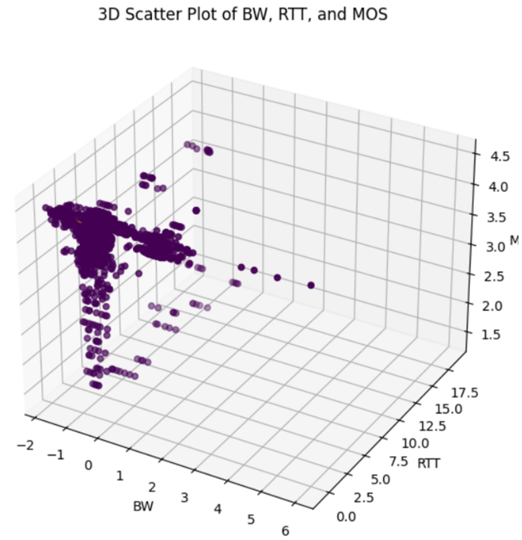

Fig. 7. RTT, MOS and BW plot in 3D

was trained with a learning rate of 0.01 and a discount factor of 0.9. The learning rate was set to 0.0003 for PPO, and the architecture was a deep neural network with two hidden layers, each with 128 neurons. To avoid overfitting in the LSTM and GRU models, the dropout rate was set to 0.2 and the number of hidden units was set to 64. The models were trained over 50 epochs for sequence models and 100 episodes for reinforcement learning. For every experiment, the batch size was set at 32, and early stopping was used to avoid overfitting.

c

Hyperparameter	Value
Q-learning Learning Rate	0.01
PPO Learning Rate	0.0003
LSTM Hidden Units	64
GRU Hidden Units	64
Batch Size	32
Epochs	50
Dropout Rate	0.2

TABLE I
HYPERPARAMETERS

Experimental Settings for Competing Methods

The suggested method was compared to other resource allocation and task offloading techniques, including Random Forest and the Greedy Algorithm, in the comparison experiments. The selection of these competing approaches was based on how well they could distribute resources in a dynamic network environment. The Greedy Algorithm was used with a fixed resource allocation strategy based on the maximum bandwidth available, and the Random Forest model was trained with the same dataset, with a maximum depth of 10 for the trees and 100 estimators. These were the experimental settings for these methods. Task allocation success rate, overall efficiency, and latency reduction were used to evaluate each method's performance. Numerous improvements were made to the model to increase its accuracy and efficiency during the simulation and experimentation stages.

During the testing phase, more complex configurations, like multi-tier architectures involving multiple edge servers and mobile users, were added to the initially simplified network structure. The exploration-exploitation balance in Q-learning was also tweaked to increase the rate of convergence, and the PPO model was modified to employ a larger neural network architecture in order to capture more complex patterns in the migration of digital twins.

IV. Results

A. Total Reward Over Episodes

The effectiveness of the Q-learning algorithm for dynamic resource allocation was assessed across 100 episodes. The stability of the system was initially demonstrated by the fact that the total reward per episode stayed at 10 for the first few episodes. Subsequent simulations with dynamic workloads, however, revealed variations in the rewards that reflected the learning process. As seen in Figure 8, by the end of the 100 episodes, rewards centered around higher values and peaked at 10 for multiple episodes. This illustrates how well the model can learn the best practices for allocating resources.

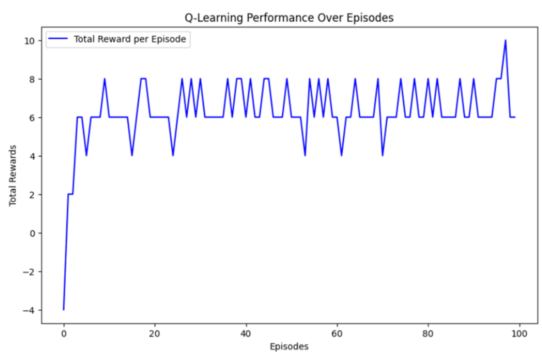

Fig. 8. Reward per Episode

B. Latency Analysis

The average values obtained from the examination of latencies in various scenarios are as follows: 1000 ms in the worst situation, then sharply dropping to 50 ms, 11.11 ms, and then stabilizing at 4.76 ms. This decrease is a reflection of how well the system allocates tasks and synchronizes in real time to minimize delays.

C. Server Utilization and Allocation Efficiency

The Q-learning methodology guaranteed effective use of edge server resources. After allocating all resources, the server's total capacity was 88, yielding a 100

D. Predicted Latencies for Real-Time Synchronization

The accuracy of the predictions made by integrating LSTM for latency prediction during task synchronization ranged from 0.04 ms to 0.06 ms per user. These outcomes demonstrate that real-time synchronization in dynamic environments is feasible. Figure 10 illustrates that the average latency for all users was 12.59 ms, confirming the system's functionality.

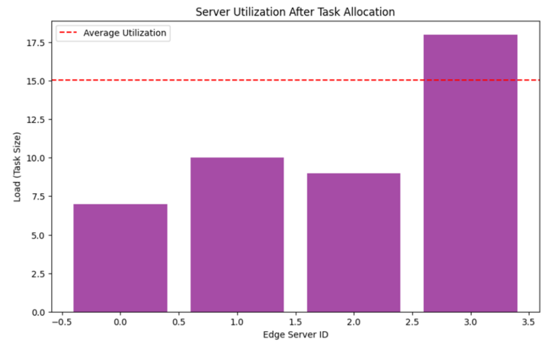

Fig. 9. Load per server ID

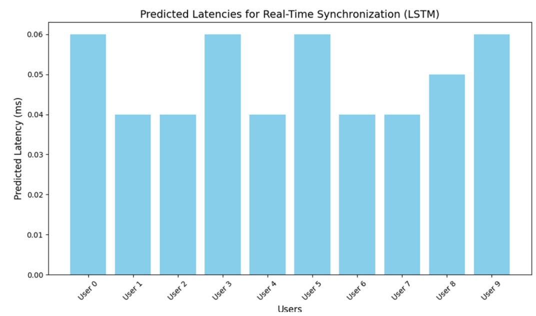

Fig. 10. Real-Time Latency Prediction

E. Task Allocation Latency Components

Individual users' migration, communication, and computation latency components for task allocation were examined. Depending on server assignments, total latency values range from 0.23 ms to 1.14 ms. Table 2 shows the breakdown of latencies for specific users. The outcomes show that the allocation process is optimized and that there are few communication and migration lags.

User	Allocated Server	Migration Latency (ms)	Communication Latency (ms)	Computation Latency (ms)	Total Latency (ms)
User 0	Server 0	0.27	0.27	0.32	0.86
User 1	Server 2	0.29	0.39	0.25	0.93
User 2	Server 2	0.13	0.19	0.25	0.57
User 3	Server 0	0.42	0.36	0.36	1.14
User 4	Server 2	0.05	0.09	0.08	0.23
User 5	Server 3	0.19	0.10	0.45	0.74
User 6	Server 2	0.06	0.11	0.12	0.30
User 7	Server 2	0.07	0.29	0.08	0.45
User 8	Server 0	0.27	0.29	0.18	0.73
User 9	Server 1	0.14	0.06	0.44	0.65

TABLE II
Latency Components

V. Conclusion

Experiments with real-time synchronization using LSTM, dynamic resource allocation using Q-learning, and task allocation optimization have shown notable gains in system efficiency, resource utilization, and latency reduction. In order to ensure effective edge server utilization with a 100

VI. Future Directions

Expanding the model's scalability and adaptability to more complex, real-world environments, like those with heterogeneous resources, unpredictable network conditions, and varying user mobility, could be the main goal of future research building on present findings. Adding multi-agent reinforcement learning (MARL) could improve the system's capacity to distribute strategies for resource allocation in a dynamic way. Moreover, investigating hybrid models that integrate the advantages of deep reinforcement learning, LSTM, and Q-learning may enhance prediction precision and decision-making in extremely dynamic systems. Digital twin and advanced edge-cloud collaboration technologies may also be able to be integrated to allow for real-time resource monitoring and optimization, which would further lower latency and strengthen resource management in next-generation wireless networks. And finally, adding in other elements like security, energy efficiency, and fault tolerance may increase this model's suitability for crucial real-time applications.

References

[1] Z. Chen, W. Yi, A. Nallanathan and J. A. Chambers, "Distributed Digital Twin Migration in Multi-Tier Computing Systems," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 18, no. 1, pp. 109-123, Jan. 2024, doi: 10.1109/JSTSP.2024.

[2] Y. Lu, S. Maharjan and Y. Zhang, "Adaptive Edge Association for Wireless Digital Twin Networks in 6G," in *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16219-16230, 15 Nov. 2021.

[3] W. Sun, H. Zhang, R. Wang and Y. Zhang, "Reducing Offloading Latency for Digital Twin Edge Networks in 6G," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12240-12251, Oct. 2020.

[4] Q. Liu, L. Tang, T. Wu and Q. Chen, "Deep Reinforcement Learning for Resource Demand Prediction and Virtual Function Network Migration in Digital Twin Network," in *IEEE Internet of Things Journal*, vol. 10, no. 21, pp. 19102-19116, 1 Nov. 2023.

[5] J. Zheng et al., "Data synchronization in vehicular digital twin network: A game theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7635-7647, Nov. 2023.

[6] J. Zheng et al., "Digital twin empowered heterogeneous network selection in vehicular networks with knowledge transfer," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12154-12168, Nov. 2022.

[7] D. Van Huynh et al., "URLLC edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7669-7682, Nov. 2022.

[8] Z. Chen, W. Yi and A. Nallanathan, "Multi-Agent Reinforcement Learning-Based Digital Twin Migration Over Wireless Networks," ICC 2024 - *IEEE International Conference on Communications*, Denver, CO, USA, 2024, pp. 2779-2784, doi: 10.1109/ICC51166.2024.10622790.

[9] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: A survey," *Artif. Intell. Rev.*, vol. 55, pp. 895-943, 2022, doi: 10.1007/s10462-021-09996-w.

[10] Z. Chen, W. Yi, A. S. Alam, and A. Nallanathan, "Dynamic task software caching-assisted computation offloading for multi-access edge computing," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6950-6965, Oct. 2022.

[11] S. Suknum, C. Thoasiri, and N. Jinaporn, "Q-Learning-Based Resource Allocation in Heterogeneous Cellular Networks," 2022 International Electrical Engineering Congress (iEECON), Khon Kaen, Thailand, 2022, doi: 10.1109/iEECON53204.2022.9741639.

[12] N. Aihara, K. Adachi, O. Takyu, M. Ohta, and T. Fujii, "Q-Learning Aided Resource Allocation and Environment Recognition in LoRaWAN With CSMA/CA," in *IEEE Access*, vol. 7, pp. 152126-152137, 17 Oct. 2019, doi: 10.1109/ACCESS.2019.2948111.

[13] X. Zhu and C. Jiang, "Delay Optimization for Cooperative Multi-Tier Computing in Integrated Satellite-Terrestrial Networks," in *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 366-380, Feb. 2023, doi: 10.1109/JSAC.2022.3227083.

[14] C. Sun, X. Wen, Z. Lu, W. Jing, and M. Zorzi, "Eco-Friendly Powering and Delay-Aware Task Scheduling in Geo-Distributed Edge-Cloud System: A Two-Timescale Framework," in *IEEE Access*, vol. 8, pp. 96468-96486, 19 May 2020, doi: 10.1109/ACCESS.2020.2995806.