ISSN: 2182-2069 / E-ISSN: 2182-2077

A Hybrid Learning and Quantum-Inspired Optimization Framework for Adaptive Task Offloading in Mobile Edge Computing

Shreenivas Badada[1*], and Dr. Akkalakshmi Muddana[2]

[1*]Research Scholar, Department of Computer Science and Engineering, GITAM (Deemed to be University), Hyderabad, Telangana, India. shreenivasbadada@gmail.com, https://orcid.org/0009-0005-6728-9536

[2]Professor, Department of Computer Science and Engineering, GITAM (Deemed to be University), Hyderabad, Telangana, India. amuddana@gitam.edu, https://orcid.org/0000-0002-9721-6507

Received: January 14, 2026; Revised: March 02, 2026; Accepted: April 07, 2026; Published: May 29, 2026

Abstract

The rapid development of IoT services has led to an increasing need for efficient resource scheduling at MEC in terms of both latency-sensitive and energy-limited environments. However, conventional optimization strategies can't effectively adapt to dynamically changing networks, heterogeneous devices, and the multiplicity of optimization objectives. This paper have presented a unified hybrid MEC system model integrated by DRL, GAT, MARL, QIO and DT to overcome these limitations, enabling intelligent adaptive task offloading, predictive workload control, and economically aware resource allocation in large scale MEC-IoT networks. The simulations have shown significant improvement of the proposed model over baseline and state-of-the-art techniques. Through simulations, this study have achieved low latency of 80ms, 110ms and 140ms for 50, 100 and 150 devices which is 20-30% lower than that of the second-best technique respectively. The system consumes less energy (0.60 J, 0.85 J and 1.10 J respectively) compared to the other methods. Fairness of devices (0.97, 0.96 and 0.95 respectively) shows fair allocation. Ablation study indicates that all individual modules (DRL for adaptive decisions, GAT for dependencies among network topology, MARL for equilibrium-based pricing, QIO for broader and efficient global exploration, and DT for predictive optimization) play an indispensable and complementary role. The outcomes exhibit great scalability, efficiency and adaptability of the framework to the MEC-IoT environment. The future work would involve deploying the model on real testbeds, integrating privacy and security-preserving solutions and extending the model to heterogeneous edge computing and 6G networks. This proposed hybrid MEC system would prove to be intelligent, adaptive and efficient framework for next generation resource scheduling on MEC-enabled IoT in many scenarios, such as smart cities, industrial IoT and autonomous vehicles.

Keywords: Mobile Edge Computing (MEC), Internet of Things, Deep Reinforcement Learning, Graph Attention Networks, Digital Twin (DT), Quantum-Inspired Optimization (QIO), Resource Scheduling, Task Offloading.

Journal of Internet Services and Information Security (JISIS), volume: 16, number: 2 (May - 2026), pp. 325-344.
DOI: *10.58346/JISIS.2026.I2.021*

*Corresponding author: Research Scholar, Department of Computer Science and Engineering, GITAM (Deemed to be University), Hyderabad, Telangana, India.

1 Introduction

The proliferation of the Internet of Things (IoT) has reshaped modern computing systems by bridging diverse devices over a vast spectrum ranging from smart cities, healthcare, industrial automation and intelligent transport systems (Chen et al., 2015; Zarandi & Tabassum, 2021). However, the explosive increase of IoT devices and services poses several constraints in terms of latency, bandwidth and energy efficiency that centralized cloud architectures are not capable of processing efficiently (Sardellitti et al., 2015; Balakrishnan et al., 2019). Mobile Edge Computing (MEC) can address these limitations by positioning computational resources close to the users thereby lowering latency and boosting Quality of Service (QoS) (Min et al., 2019; Vrahatis et al., 2024). Nonetheless, MEC-empowered IoT systems suffer from non-deterministic and real-time nature of workload, restricted edge resources, diverse characteristics of devices, and changing wireless environments which require advanced multi-objective optimization techniques.

In this paper, a synergistic hybrid optimization framework which amalgamates Deep Reinforcement Learning (DRL), Graph Attention Networks (GAT), Multi-Agent Reinforcement Learning (MARL), Quantum-Inspired Optimization (QIO) and Digital Twin (DT) is proposed. The framework facilitates efficient task offloading decision-making and proactive resource allocation, making it economically driven and increasing MEC utility in terms of latency, energy consumption, fairness etc. DRL handles the dynamic offloading of tasks; GAT captures inter-dependent network characteristics; MARL establishes a market-driven price mechanism; QIO facilitates global optimization and DT monitors and foresees the future behavior of the system.

The main contribution of the paper follows,

- Develop a hybrid DRL-GAT-MARL-QIO-DT framework for massive MEC-IoT systems.
- Design the multi-objective optimization problem incorporating latency, energy, fairness and utility.
- Present the experimental setup and verify through simulations achieving superior performance gain over baseline methods.
- Carry out ablation studies to evaluate individual and joint contributions of each proposed module.

The importance of this work is in furnishing a scalable, dynamic and intelligent framework for resource allocation which would boost the performance of MEC-IoT systems under such volatile situations.

The paper organization includes, Section 2 surveys the related works, Section 3 proposes individual modules and unified framework, Section 4 presents the simulation and ablation studies, and Section 5 concludes and provides future research prospects.

2 Related Works

One paradigm that comes to mind for solving the latency, bandwidth and energy constraints of massive IoT is Multiaccess Edge Computing (MEC). Many recent works have emphasized the importance of task offloading and dynamic resource allocation to address low-latency requirements in many of the applications that exist for the IoT (Abbas et al., 2017; Jiang et al., 2022). Showed that using MAEC along with smart decision making has boosted up the throughput and reduced the latency in 5G-enabled IoT networks (Liu et al., 2020) and a similar paper has implemented deep learning for task offloading in distributed MEC-IoT networks where there was an increase in system utility and energy efficiency (Arun & Azhagiri, 2025; Abdullaev et al., 2023).

Graph-based approaches are often useful for the networks with dependencies in wireless and edge systems (Sorooshian et al., 2022; Song et al., 2014). This paper emphasizes that the Graph Neural networks (GNNs) are a beneficial approach to describe the interference patterns and spatial locality of the devices and hence have enabled efficient resource allocation in the system to achieve good reliability (Shen et al., 2022) and later showed that Graph Attention Networks (GATs) could dynamically weigh the nodes according to their importance and are therefore beneficial for multi-device coordinated heterogeneous IoT networks (Vrahatis et al., 2024; Meng & Zeng, 2013).

The multi-agent reinforcement learning is often combined with the resource allocation in MEC for capturing the economic and cooperative interactions between entities. It discussed the concept of MARL in detail and stated its advantage in adapting dynamically and intelligently to multiple agents interacting among each other (Canese et al., 2021) and another work combining federated learning with edge intelligence achieved scalability with increased privacy with multiple devices (Chen et al., 2015).

The digital twin (DT) technologies have also been adopted for predictive resource allocation in MEC (Wang et al., 2019) and claimed that with DT based simulations one can dynamically manage resources by predicting the conditions in the network that ultimately helps in reducing latency and energy (Lu et al., 2020). In summary all of the above studies have revealed that to overcome the limitations of utilizing these techniques in isolation, combination of DRL, GAT, MARL, QIO, and DT would enable MEC-IoT networks to efficiently and adaptively perform task offloading, achieve greater scalability and high efficiency (Zhang et al., 2016).

This study's contributions are built upon this existing work by illustrating how the integration of learning-based, graph-aware, multi-agent, quantum-inspired, and predictive techniques leads to improvement of the overall performance in MEC-IoT networks, especially in terms of dynamic and massive scale.

3 Materials and Methods

The figure 1 indicates that the reinforcement learning with federated game scheduling optimization DRL-FGSO model is an improvement of the TS-Hybrid-SB model, where the heuristic global exploration is replaced by a data-driven intelligent decision-making mechanism. This framework integrates Deep Reinforcement Learning (DRL), Federated Learning (FL), and game-theoretic optimization to achieve adaptive, scalable, and privacy-conscious resource scheduling in MEC-enabled IoT. Systems. Consider a set of IoT devices $N = \{1,2,\dots,N\}$ and MEC servers $M = \{1,2,\dots,M\}$. Each IoT device is an autonomous agent of the DRL which interacts with the MEC environment. The scheduling problem is defined as a MDP with the following definition: (S, A, P, R) shown in equation (1-9). The state of device *i* at time *t* is expressed as

$$s_i^t = \{E_i^t, D_i^t, C_i^t, h_i^t, p_j^t\} \tag{1}$$

Action Space A:

Each agent selects an action:

$$a_i^t \in \{0,1,\dots,M\} \tag{2}$$

Where $a_i^t = 0$ denotes local execution and $a_i^t = j$ represents offloading to server j.

Reward Function R:

The objective is to minimize latency, energy, and economic cost:

$$R_i^t = -\left(\alpha L_i^t + \beta E_i^t + \gamma C_i^t\right) \tag{3}$$

Where α, β, γ are weighting coefficients. Each IoT agent learns an optimal policy $\pi_i(a|s)$ that maximizes cumulative discounted reward:

$$\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t R_i^t\right] \tag{4}$$

A Deep Q-Network (DQN) or Actor–Critic model is used. The Q-function is defined as:

$$Q(s, a; \theta) = E\left[R + \gamma \max_{a'} Q(s', a'; \theta)\right] \tag{5}$$

The loss function for training is:

$$L(\theta) = E\left[\left(y - Q(s, a; \theta)\right)^2\right] \tag{6}$$

Where $y = R + \gamma \max_{a'} Q(s', a'; \theta^-)$. Here, θ^- denotes the target network parameters. To ensure data privacy and scalability, DRL-FGSO adopts a Federated Learning (FL) paradigm. Instead of sharing raw data, devices locally train models and share only parameters. Global model aggregation is performed as:

$$\theta^{(t+1)} = \sum_{i=1}^{N} \frac{n_i}{\sum_{k=1}^{N} n_k} \theta_i^{(t)} \tag{7}$$

Where $\theta_i^{(t)}$ is the local model of device i, and n_i represents its data size. The DRL-FGSO model incorporates a Stackelberg game between MEC servers (leaders) and IoT devices (followers). MEC servers dynamically adjust pricing strategies, while IoT agents learn optimal responses.

The completed forms are:

Leader utility:

$$U_j = \sum_{i=1}^{N} p_{ij} f_{ij} - 2E_j - \mu D_j \tag{8}$$

Follower optimization:

$$\min_{a_i}\left(L_i + E_i + p_j f_{ij}\right) \tag{9}$$

Unlike static models, DRL allows agents to learn best-response strategies dynamically under uncertain environments.

Algorithm Workflow 1: Workflow of DRL-Based Federated Task Offloading with Game-Theoretic Pricing

Initialize DRL agents and global model

Each device observes state s_i^t

Select action a_i^t using policy π_i

Execute task (local/offload)

Receive reward R_i^t

Update local DRL model

Periodically perform federated aggregation

MEC servers update pricing via game feedback

Algorithm 1 shows the flow of operation for MEC-enhanced IoT devices through a combination of DRL, federated learning and game-theory based pricing. The devices observe local states, decide an action of local execution or offloading a task according to their policy and get rewards accordingly. The local DRL models are trained continuously whereas federated aggregation happens in specific rounds to get the globally optimum policy. MEC servers dynamically set the prices based on the game's feedback to balance resource distribution and system utility. Through these operations, the study achieves adaptability, scalability and efficiency in task offloading and also fairness and energy efficiency for large scale MEC environment.

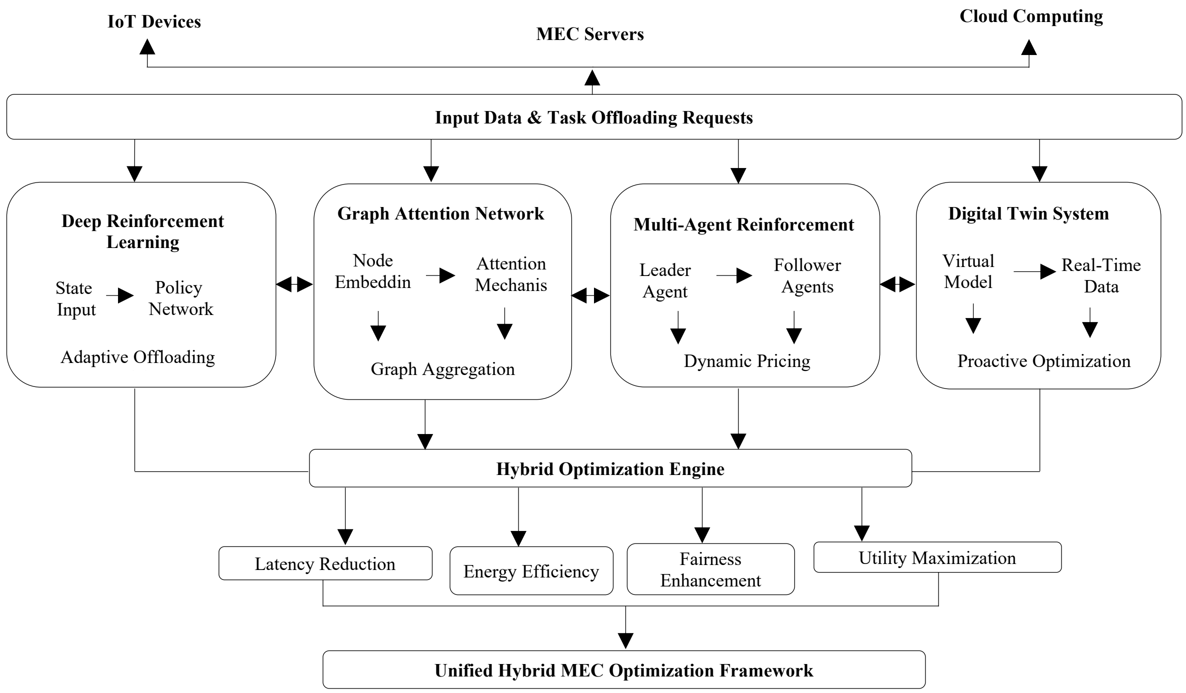

Figure 1: Proposed DRL-FGSO model in MEC-enabled IoT systems

3.1. GAT-MEC: Graph Attention-Based Resource Scheduling

The proposed GAT-MEC model improves the TS-Hybrid-SB and DRL-FGSO models by adding graph-based deep learning to represent the intricate spatial and relational relationships of MEC-capable IoT networks. In contrast to traditional optimization methods, which assume that devices operate in isolation, GAT-MEC represents the MEC system as a dynamic graph, which allows intelligent and context-aware scheduling decisions to be made using attention mechanisms. The MEC-IoT setup is modelled as a graph $G = (V, E)$ with node set V comprising IoT devices and MEC servers, and edge set modelling communication between them. Each node $i \in V$ is associated with the following feature vector are illustrated in equation (10-14):

$$x_i = [E_i, D_i, C_i, h_i, f_i] \tag{10}$$

AT computes attention coefficients for neighbors:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}(a^T[Wx_i \parallel Wx_j])\right)}{\sum_{k\in\mathcal{N}_i} \exp\left(\text{LeakyReLU}(a^T[Wx_i \parallel Wx_k])\right)} \tag{11}$$

The updated node embeddings are:

$$h_i' = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W x_j\right) \tag{12}$$

Each IoT device selects a server or local execution by minimizing the system cost:

$$f_i = \arg\ \min \sum_i (\text{latency}_i + \text{energy}_i + \text{price}_i) \tag{13}$$

Subject to the resource constraint:

$$\sum_i f_i \leq C_{\text{server}}, f_i \geq 0 \tag{14}$$

This structure not only allows to record all network wide dependencies but also enables the model to react to the dynamic environments and the scalability and the fairness to the user devices. This is the effective graph-based learning technique for the MEC resource scheduling.

3.2. TSG-MARL: Multi-Agent Reinforcement Learning-Based Stackelberg Game for MEC Resource Scheduling

TSG-MARL generalizes Stackelberg-Bandit pricing to support learning-based multi-agent resource allocation in MEC-enabled IoT environments where MEC servers are leaders while IoT devices are followers. Specifically, leaders formulate optimal dynamic pricing strategies and followers determine appropriate offloading schemes. Both leaders and followers are modeled as reinforcement-learning agents in a multi-agent Markov game defined by (S, A_L, A_F, R_L, R_F), where $S$is the global state comprising device energy, task size, channel conditions, and server resources.

The leader's objective is to maximize expected utility are shown in equation 15:

$$U_j = \sum_{i \in N_j} p_j f_{ij} - \lambda E_j - \mu D_j \tag{15}$$

Each follower minimizes its expected cost are illustrated in equation 16:

$$C_i = L_i + E_i + p_j f_{ij} \Rightarrow \min_{\pi_i^F} \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t C_i^t] \tag{16}$$

The framework uses central training with distributed execution (CTDE). Leaders and followers are trained using Q-functions are demonstrated by equation 17:

$$Q_L(s, p_j) = \mathbb{E}[U_j + \gamma \max_{p_j'} Q_L(s', p_j')] \tag{17}$$

Policies are learned via gradient-based algorithms such as MADDPG, and a regularization term $L_{reg} = \lambda \parallel \pi^t - \pi^{t-1} \parallel^2$ensures smooth updates. This design lets the agents adjust to changing environmental conditions, dynamic equilibrium, and fairness, at the same time optimizing delay, energy, and pricing.

The model reflects real economic interactions where servers provide finite computational power at fluctuating prices and devices have access to this power based on economic factors. Integrating multi-agent learning with Stackelberg pricing TSG-MARL can be extended to the developing MEC environments such as smart cities, industrial IoT, autonomous vehicles and 6G networks.

3.3. QIO-MEC: Quantum-Inspired Optimization for MEC resource scheduling

To further expand the exploration power and address the shortcomings of classical metaheuristic algorithms, this study presents a quantum-inspired optimization-based MEC (QIO-MEC) framework to efficiently offload tasks and schedule resources in MEC-enabled IoT systems. In contrast to traditional methods, such as PSO or GA, which utilize deterministic or stochastic updates in the solution space, QIO-MEC uses the concepts of quantum computing, namely quantum superposition and probabilistic representation, to provide better global search performance and prevent early convergence.

Quantum Representation of Solutions: In QIO-MEC, each candidate solution (i.e. offloading decision vector) is represented using a quantum bit (qubit) instead of a classical binary variable. A qubit is defined as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex probability amplitudes satisfying $|\alpha|^2 + |\beta|^2 = 1$. In this case, $|0\rangle$ represents local execution and $|1\rangle$ represents offloading to an MEC server. A quantum chromosome represents a population of solutions, and each gene is represented by a qubit that relates to the offloading decision of a device. The probability of selecting a particular state is determined as follows equation 18:

$$P(0) = |\alpha|^2, \quad P(1) = |\beta|^2 \tag{18}$$

This probabilistic encoding enables a single quantum chromosome to encode a superposition of multiple candidate solutions, which greatly enhances the exploration ability.

Observation and Solution Generation: The quantum state should be measured to reduce the state to a classical binary solution which will be used to estimate candidate solutions. For each qubit, a random number $r \in [0,1]$ is generated, and the state is determined as equation 19:

$$x_i = \{1, if\ r < |\beta_i|^2\ 0, otherwise \tag{19}$$

This results in a discrete offloading vector $X = [x_1, x_2, \ldots, x_N]$, which is evaluated using the system objective function $F(X) = \sum_{i=1}^{N}(\alpha L_i + \beta E_i + \gamma C_i)$, where, L_i, E_i, and C_i represent the latency, energy consumption, and cost, respectively.

Quantum Rotation Gate Update: The quantum rotation gate is the most important optimisation mechanism in QIO-MEC, which modifies the probability amplitudes of each qubit according to the quality of the solutions. The update rule is defined as follows equation 20:

$$[\alpha_i{}' \, \beta_i{}'] = [cos(\Delta\theta_i)\ - sin(\Delta\theta_i)\ sin(\Delta\theta_i)\ cos(\Delta\theta_i)\ ][\alpha_i\ \beta_i\ ] \tag{20}$$

Where $\Delta\theta_i$ is the rotation angle calculated by comparing the current solution and the global best solution. When the current solution is less good, the rotation angle is changed such that the likelihood of the better solution being chosen is higher. The rotation strategy guarantees a balance between exploration (diversity) and exploitation (convergence), which enables the algorithm to effectively explore the solution space. The quantum optimisation module is incorporated into the global exploration phase of MEC scheduling in the QIO-MEC framework. The algorithm constructs a set of candidates offloading strategies by exploiting quantum superposition, and the strategies are evaluated based on the system objectives; then, the probability amplitudes are updated using rotation gates. After identifying an optimal or near-optimal offloading setup, convex optimisation techniques can be used to further optimize continuous parameters, including transmission power and CPU allocation. Moreover, it is possible to use the QIO-MEC together with pricing methods, such as Stackelberg or MARL-based methods, to integrate economic factors into the optimisation.

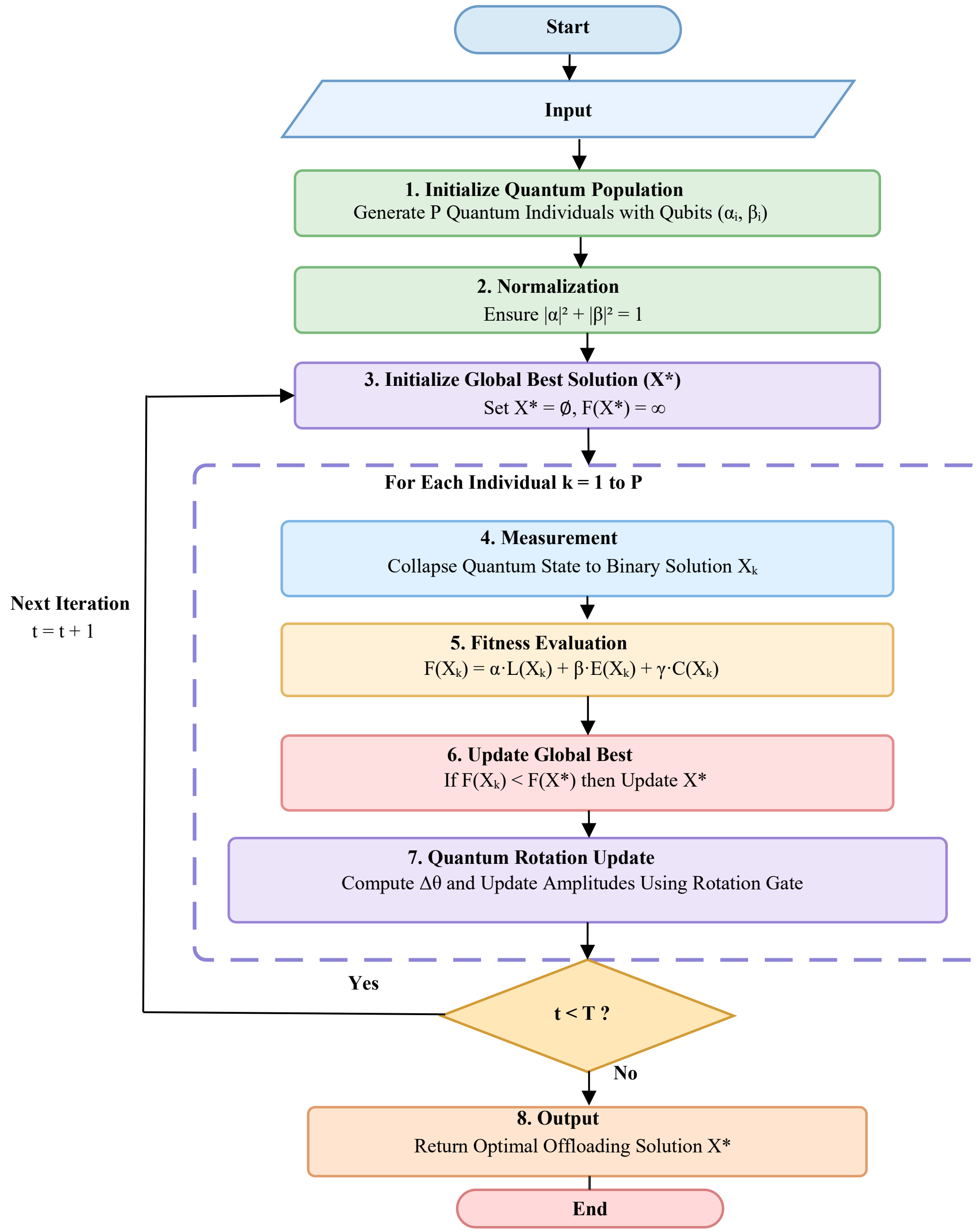

Figure 2: Offloading optimization of MEC-enabled IoT systems quantum-inspired principles

The QIO-MEC flowchart as shown in figure 2 shows a systematic approach to offloading optimisation of MEC-enabled IoT systems based on quantum-inspired principles. The step starts with input parameters, such as the population size, number of devices, and maximum iterations. It is then initialized with a quantum population, with every solution being represented on qubits with probabilistic amplitudes, and normalized to valid probability states. The algorithm is placed in an iterative loop, and each quantum person is measured, collapsing to a binary solution, which is an offloading decision. The evaluation of these solutions is based on a multi-objective fitness function that considers latency, energy, and cost. An improved fitness value was obtained, and the best solution was updated. A quantum rotation gate then alters the probability amplitudes and directs the search to the optimal regions. This is an iterative process until the stopping condition is satisfied. Finally, the algorithm provides the best offloading strategy, which guarantees the efficient use of resources and enhances the MEC system performance.

Algorithm Complexity

The computational complexity of the QIO-MEC primarily depends on the population size, number of qubits, and maximum iterations. The overall complexity can be expressed as $O(P \cdot N \cdot T)$. Similar to classical evolutionary algorithms, but with faster convergence owing to improved exploration. The QIO-MEC framework has several benefits compared to traditional optimisation techniques. First, quantum superposition is used to simultaneously search for several solutions at the same time, which greatly enhances the search efficiency. Second, the probabilistic representation avoids early convergence and improves diversity. Third, the rotation gate mechanism provides a smooth and guided convergence to optimal solutions. Finally, the algorithm is discrete optimisation, which is suitable for solving discrete optimisation problems, such as deciding on task offloading in MEC systems. Finally, QIO-MEC presents a new paradigm of quantum-inspired optimisation of MEC resource scheduling, providing better exploration, faster convergence, and quality of solutions. The offered solution, which combines the principles of quantum computing with classical optimisation methods, can serve as a potent substitute for conventional metaheuristics and a supplement to the learning-based models of DRL and MARL.

3.4. DT-MEC: Digital Twin-Based Predictive Optimization for MEC-IoT Systems

To overcome the issues of dynamic workloads, unpredictable network states, and reactive decisions in MEC-enabled IoT systems, this study proposes a digital twin-based MEC optimisation (DT-MEC) model. In contrast to traditional optimisation methods that can be based only on real-time measurements, the DT-MEC opens a virtual copy (digital twin) of the physical MEC-IoT system, where analysis of predictability, proactive scheduling, and efficient resource distribution can be performed. A digital twin is a virtual image of a real system that constantly reflects the state of IoT devices, MEC servers, and network conditions. Suppose the physical system state at time t is denoted as $s^t = \{E_i^t, D_i^t, C_i^t, h_i^t, f_j^t, p_j^t\}$. The digital twin maintains an estimated state $\hat{s}^t$ using real-time data synchronization $\hat{s}^t = F(s^{t-1}, u^{t-1}, \omega^t)$ with $F(\cdot)$ the system dynamics model, u^{t-1} control actions, and ω^t the stochastic disturbances, which include channel variations and workload variations. The digital twin is accurate by maintaining a low error in the state estimation $min \parallel s^t - \hat{s}^t \parallel^2$. One of the most important factors of DT-MEC is the future prediction of the system state. Using time-series forecasting models (e.g. LSTM or transformer-based predictors), the digital twin predicts the future workload and state of the network are shown in equation 21.

$$\widehat{D}_i^{t+1} = G(D_i^t, D_i^{t-1}, \dots) \ \hat{h}_i^{t+1} = H(h_i^t, h_i^{t-1}, \dots) \tag{21}$$

These forecasts enable the system to foresee congestion, peaks in latency, and energy usage patterns before it occurs. This leaves the opportunity to make resource allocation decisions proactively, as opposed to reactively. The DT-MEC framework creates a look-ahead optimisation problem based on the predicted states over a time horizon T as follows equation 22:

$$min \sum_{t=1}^{T} \sum_{i=1}^{N} (\alpha L_i^t + \beta E_i^t + \gamma C_i^t) \tag{22}$$

This formulation ensures that the performance of the system is optimized in the future and not in the short run. The DT-MEC model is designed as a closed system, as decisions made based on the digital twin are implemented on the physical system, and the feedback can be used to update the twin model. The control loop has the following form $u^t = \pi(\hat{s}^t)\ s^{t+1} = E(s^t, u^t)$ and $\pi(\cdot)$ is the decision policy, and $E(\cdot)$ is the actual dynamics of the environment. The new physical state is then fed into the digital twin to further refine future predictions. The DT-MEC module expands the unified hybrid framework by providing predictive inputs to other modules. The predicted states are used by the DRA agents to learn the policy, the predicted network topology changes are used by the GAT module, and the future interactions between pricing and demand are predicted by the MARL module. This integration greatly enhances the accuracy of decision-making and the stability of the system.

3.5. Unified DRL–GAT–MARL–QIO–DT Framework for MEC Resource Scheduling

Proposed architecture integrates DRL, GAT, MARL, QIO and Digital Twin (DT) to provide a collaborative decision-making pipeline for adaptable, scalable and predictive optimization. The unified DRL-GAT-MARL-QIO-DT architecture is represented in figure 3.

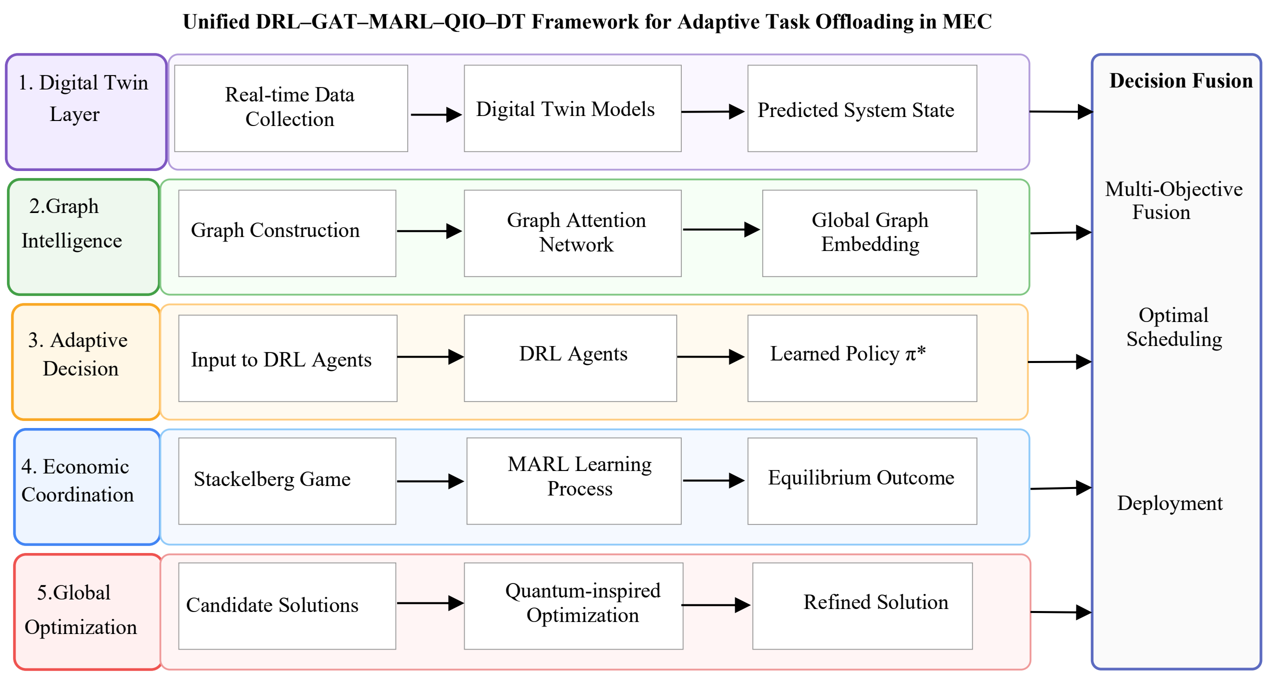

Figure 3: Unified DRL–GAT–MARL–QIO–DT architecture for adaptive task offloading and resource scheduling in MEC-enabled IoT systems

Decision Fusion Mechanism

The proposed framework combines outputs from all modules through a weighted multi-objective optimization process. The final decision vector is determined as equation 23:

$$D^* = \arg\min\,(w_1 L + w_2 E - w_3 F - w_4 U) \tag{23}$$

The Digital Twin provides predictive information, GAT supplies contextual embeddings, DRL performs adaptive policy learning, MARL ensures economic equilibrium, and QIO refines the global solution. The fused output represents the optimal resource scheduling strategy.

Training and Optimization Sequence

The integrated framework operates according to the following sequence:

Step 1: Collect real-time MEC-IoT system states.

Step 2: Synchronize the Digital Twin and predict future workloads and network conditions.

Step 3: Construct the MEC network graph and generate node embeddings using GAT.

Step 4: Provide graph embeddings and predicted states to DRL agents.

Step 5: Train IoT agents to learn adaptive task-offloading policies.

Step 6: Execute MARL-based Stackelberg interactions between MEC servers and IoT devices.

Step 7: Generate candidate scheduling solutions.

Step 8: Refine candidate solutions using QIO-based global optimization.

Step 9: Select the optimal scheduling strategy using multi-objective decision fusion.

Step 10: Deploy the selected strategy and update the Digital Twin using real-time feedback.

The algorithm 2 illustrates the proposed Unified DRL-GAT-MARL-QIO-DT resource scheduling framework for MEC enabled IoT systems. The algorithm combines the modeling of prediction, graph-based learning, RL, multi-agent-based economy and quantum inspired optimization for offloading the tasks and allocating the resources. Firstly, the algorithm collects all the information as real time state, update Digital twin for prediction, build up network graph for embedding generation. Then, the DRL agents were trained up. After that, the MARL based pricing mechanism is invoked and the solution is enhanced by QIO. Finally, based on multi-objective fusion mechanism, the optimal scheduling decision is selected, and the whole procedure repeats until the optimal condition achieved and the performance achieved with the desired latency, energy consumption, fairness and system utility

Algorithm 2: Unified DRL–GAT–MARL–QIO–DT Resource Scheduling Framework

Input: MEC system state S_t

Output: Optimal scheduling decision D^*

1. Initialize Digital Twin, DRL agents, GAT network, MARL agents, and QIO population.
2. Collect real-time system state S_t.
3. Update Digital Twin and predict future state $\hat{S}_{t+1}$.
4. Construct MEC graph $G = (V, E)$.
5. Generate graph embeddings using GAT.
6. Feed embeddings to DRL agents.
7. Learn task-offloading policies.
8. Execute MARL-based Stackelberg pricing and resource allocation.
9. Generate candidate scheduling solutions.
10. Apply QIO search to refine solutions.
11. Evaluate latency, energy, fairness, and utility objectives.
12. Select optimal solution using multi-objective fusion.
13. Deploy scheduling decision.
14. Update Digital Twin using system feedback.
15. Repeat until convergence.

4 Results and Discussion

The simulations showed an MEC-enabled IoT environment with multiple devices. The number of devices was N =50, 100, and 150, and the number of MEC servers was M=3. The computational tasks designed by each IoT device were random and had data sizes between 10^4 and 10^6 bits, requiring between 10^6 and 10^8 CPU cycles to execute. This study considered following parameters for system. Bandwidth B=10 MHz, Noise power noise power 10^{-13} W, Computational capacity of MEC server is 10 GHz and CPU frequency of IoT devices is in between 1 to 2 GHz. Noise power is within the standard values widely used in wireless communications simulation, and kept fixed throughout all experiment. The experiments were implemented in MATLAB R2023a and Python 3.11 by using PyTorch deep learning toolbox. MATLAB was adopted for system level simulation, performance evaluation and statistical analysis. Python and PyTorch were used for DRL, GAT, MARL and Digital Twin learning modules. All the simulations were carried out on a workstation (CPU: Intel Core i7 at 3.4 GHz, 16 GB RAM, GPU: NVIDIA acceleration, OS: Windows 11). The combination of MATLAB and Python provided an effective method for MEC supported IoT system modeling and training of proposed AI based optimization framework.

Table 1: System parameters

Parameter	Value
Number of IoT devices (N)	50, 100, 150
Number of MEC servers (M)	3
Task size (D_i)	$10^4 – 10^6$ bits
CPU cycles (C_i)	$10^6 – 10^8$ cycles
Bandwidth (B)	10 MHz
Noise power	10^{-13} W
MEC CPU capacity	10 GHz
Device CPU	1–2 GHz
Learning rate (η)	0.001
Discount factor (γ)	0.9
Exploration rate (ε)	0.1
Population size (QIO)	30
Iterations	100

Optimization modules were realized with different software tools according to their computational complexity: DRL, GAT and MARL models were realized in python with the PyTorch software tool, supporting both deep neural network and reinforcement learning, whereas QIO optimization algorithm and the MEC simulations were implemented in MATLAB 2023a. Data exchange between MATLAB and python was carried out using the input/output common interfaces.

The simulation environment is set in such a way that it realistically simulates a large scale MEC enabled IoT with heterogeneous devices and dynamic network conditions. To test scalability, the number of IoT devices is varied N= {50,100,150}, and the number of servers in the MEC is fixed at M=3. The input data sizes of each IoT device create computational tasks with 104 to 106 bits of input data and 10^6 to 10^8 cycles of the CPU, which are inferred based on different application requirements. The bandwidth of the communication was established as 10 MHz, and the noise power was assumed to be 10^{-13} W to represent the actual wireless channel environment. Each MEC server is predetermined with a computational capacity of 10 GHz, and the limit of the CPU frequency of the IoT devices is 1-2 GHz which is indicative of the resource limitation on the edge. In the case of learning-based models, the learning rate was 0.001, discount factor 0.9, and exploration rate 0.1 to strike a balance between exploration and exploitation. The maximum population number is 30 and the maximum iterations are 100 in the QIO-MEC. These parameters were chosen carefully to guarantee a fair comparison and steady convergence in all the methods considered, and also to keep the computationally viable, as listed in table 1.

To test the efficiency of the proposed hybrid framework, a full-fledged experimental design was developed considering both conventional and modern optimisation methods for MEC-enabled IoT systems. The results of the proposed methods were compared with those of a set of baseline and state-of-the-art algorithms, including TS-Hybrid-SB, DRL-FGSO, GAT-MEC, TSG-MARL, and QIO-MEC. Two-Stage Hybrid Optimisation with Stackelberg Bandit Pricing (TS-Hybrid-SB) is the benchmark, which incorporates confidence-based PSO, convex optimisation, and bandit-optimal pricing when scheduling resources adaptively. Although powerful, it is based on heuristic search and is not as adaptable to deep learning. The deep reinforcement learning with federated game scheduling optimisation (DRL-FGSO) technique uses reinforcement learning to facilitate adaptive offloading decisions under dynamic network conditions, which is inspired by the more recent DRL-based optimisation of MEC. The Graph Attention-Based MEC Scheduling (GAT-MEC) model uses graph neural networks to learn the spatial dependency and interference patterns within the network to make resource allocation more efficient. The TSG-MARL (Multi-Agent Reinforcement Learning Stackelberg Game) framework uses both MEC servers and IoT devices as learning agents, which allows for dynamic price adjustment and decision-making based on equilibrium. Finally, the quantum-inspired optimisation (QIO-MEC) algorithm uses probabilistic qubit representation and rotation gates to improve the ability to search for global optimisation and prevent premature convergence in discrete optimisation problems. The above approaches are an excellent representation of various optimisation paradigms, such as heuristic, learning-based, graph-based, game-theoretic, and quantum-inspired methods, which offer a solid benchmark to assess the proposed unified framework.

The table 2, Latency comparison of different resource scheduling algorithms in MEC-enabled IoT systems. TSG-MARL, DRL-FGSO and GAT-MEC achieves performance gains by using learning ability, graph awareness and multi-agent approach, respectively. QIO-MEC can bring medium gains by applying the quantum inspired global search, while DT-MEC obtains the best performance by adopting the predictive optimization. The benchmark TS-Hybrid-SB results in high latency because of the heuristic characteristic of the TSG. For different device scales (N=50, 100 and 150), The Unified DRL+GAT+MARL+QIO+DT algorithm significantly outperforms other compared algorithms and has better scalability than all. Latency values are reduced to 80 ms, 110 ms and 140 ms for N=50, 100 and 150, respectively.

Table 2: Average latency (ms)

Method	**N=50**	**N=100**	**N=150**
TS-Hybrid-SB	120	165	210
DRL-FGSO	105	140	180
GAT-MEC	98	130	170
TSG-MARL	92	125	160
QIO-MEC	110	150	195
DT-MEC	85	115	145
Unified DRL+GAT+MARL+QIO+DT	80	110	140

The energy consumption table 3 evaluates the six MEC-IoT resource scheduling approaches in three scale levels of N (N=50,100,150). From the energy consumption comparison, the study can find that TS-Hybrid-SB, a heuristic resource scheduling method, has the highest energy consumption. DRL-FGSO, GAT-MEC and TSG-MARL has successively reduced energy consumption through intelligent learning method, graph-based context awareness and multi agent interactions method. QIO-MEC has a moderate energy reduction compared with TS-Hybrid-SB with quantum inspired search mechanism, while DT-MEC has achieved the additional improvement using predicted optimization. As a comparison, the unified framework, DRL+GAT+MARL+QIO+DT, presents the minimum energy consumption on different scales of MEC environments, where it has achieved 0.60 Ja, 0.85 J and 1.10 J on N=50, N=100 and N=150 respectively. It can also show that combining of adaptive method, predictive method and global optimal method can effectively improve the system efficiency of large scale MEC environments.

Table 3: Energy consumption (J)

Method	**N=50**	**N=100**	**N=150**
TS-Hybrid-SB	0.85	1.20	1.60
DRL-FGSO	0.75	1.05	1.40
GAT-MEC	0.70	0.98	1.30
TSG-MARL	0.68	0.95	1.25
QIO-MEC	0.78	1.10	1.45
DT-MEC	0.62	0.88	1.15
Unified DRL+GAT+MARL+QIO+DT	0.60	0.85	1.10

The table 4 depicts the resource allocation fairness. The lowest value was attained by TS-Hybrid-SB because its heuristic scheduling policy was likely to miss an efficient resource allocation at the first place. TSG-MARL, GAT-MEC, and DRL-FGSO present increased fairness thanks to adaptivity, learning, graph-context awareness, and multi-agent collaboration while QIO-MEC demonstrates the moderate improvement due to quantum inspired exploration and DT-MEC presents more advance performance through predictive optimization. Fair resource allocation among MEC enabled IoT devices reached the maximum value in Unified DRL+GAT+MARL+QIO+DT where it obtains the fairness of 0.97, 0.96, and 0.95 when number of devices is 50, 100 and 150 respectively. This confirms the efficiency of the proposed method as it equally distributed resources and avoided monopolization over number of IoT devices.

Table 4: Fairness index (jain's index)

Method	**N=50**	**N=100**	**N=150**
TS-Hybrid-SB	0.88	0.86	0.84
DRL-FGSO	0.91	0.90	0.88
GAT-MEC	0.93	0.92	0.90
TSG-MARL	0.94	0.93	0.92
QIO-MEC	0.89	0.87	0.85
DT-MEC	0.96	0.95	0.94
Unified DRL+GAT+MARL+QIO+DT	0.97	0.96	0.95

The figure 4 compare the performance of six MEC-IoT scheduling algorithms and the unified DRL+GAT+MARL+QIO+DT framework when the number of nodes increases (50, 100, 150). Red line represents the unified framework and it always stays below the others, which means it always has the smallest latency/cost in every scale. For traditional heuristic algorithms, TS-Hybrid-SB always has the largest cost, DT-MEC and TSG-MARL has a smaller cost than TS-Hybrid-SB but still keep it above the unified framework. The study can find that the integration of learning, graph, multi-agent, quantum-inspired and predictive modules achieve an optimal resource allocation under time-varying MEC-IoT environment, in addition to the highest efficiency and scalability.

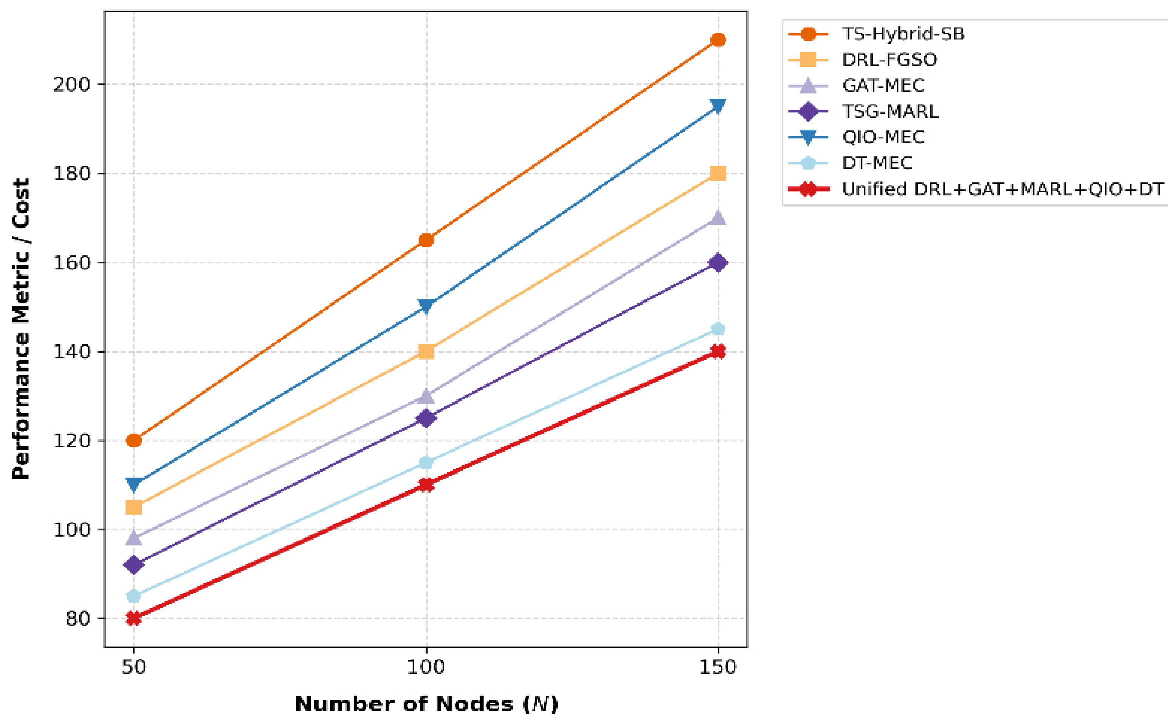

Figure 4: Latency comparison of baseline methods, and DT-MEC under varying numbers of IoT devices

The figure 5 shows the energy consumed by six MEC-IoT scheduling methods and the unified DRL+GAT+MARL+QIO+DT scheme, under the environment of varying number of nodes, 50, 100, 150. The red line, representing the unified scheme, achieves minimum energy consumption in every case. TS-Hybrid-SB consumes the most energy due to its heuristic and reactive natures. DRL-FGSO, GAT-MEC, TSG-MARL, QIO-MEC, DT-MEC show the performance order in intermediate cases, each scheme is better than the former, benefiting from learning ability, graph awareness, multi-agent characteristics, quantum-inspired intelligence, predictive execution. The proposed scheme achieves superior energy efficiency by the synergistic action of all five modules.

The figure 6 presents the fairness index (Jain's Index) for the 6 MEC-IoT resource scheduling methods as well as for Unified DRL+GAT+MARL+QIO+DT with an increasing number of nodes (N=50,100,150). Fairness of TS-Hybrid-SB is the least and fairness of DRl-FGSO, GAT-MEC, TSG-MARL, QIO-MEC is improved in increasing number of nodes. Fairness of DT-MEC is better as it is using prediction-based optimization and fairness of the Unified DRL+GAT+MARL+QIO+DT system is better than other methods for N=50, 100,150 respectively as the values are 0.97, 0.96, 0.95. It can be inferred that the use of learning, graph, MARL, quantum inspired and DT modules guarantees fair resource allocation for large scale MEC-IOT systems.

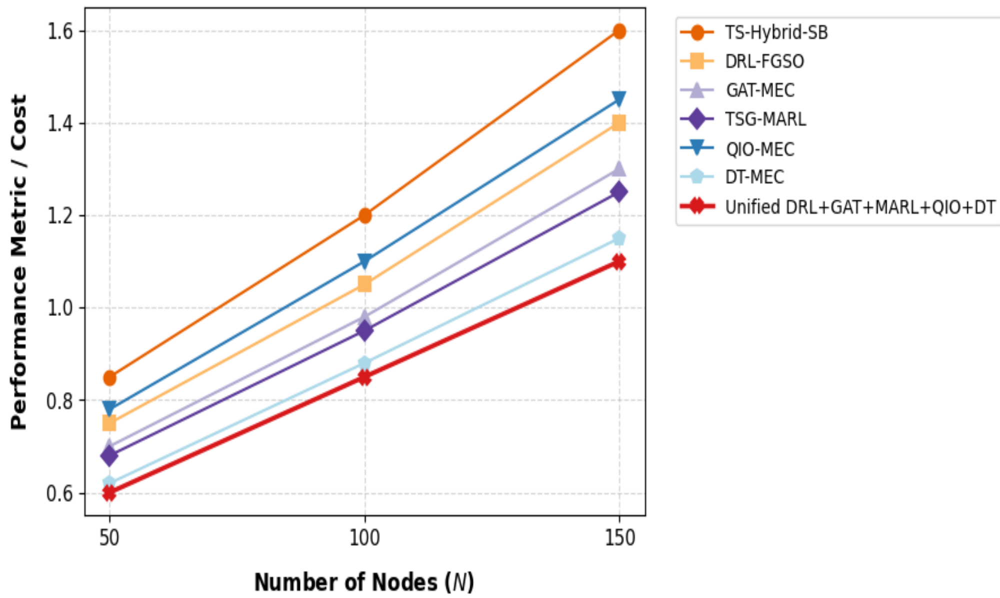

Figure 5: Energy consumption comparison of baseline methods, and DT-MEC under varying numbers of IoT devices

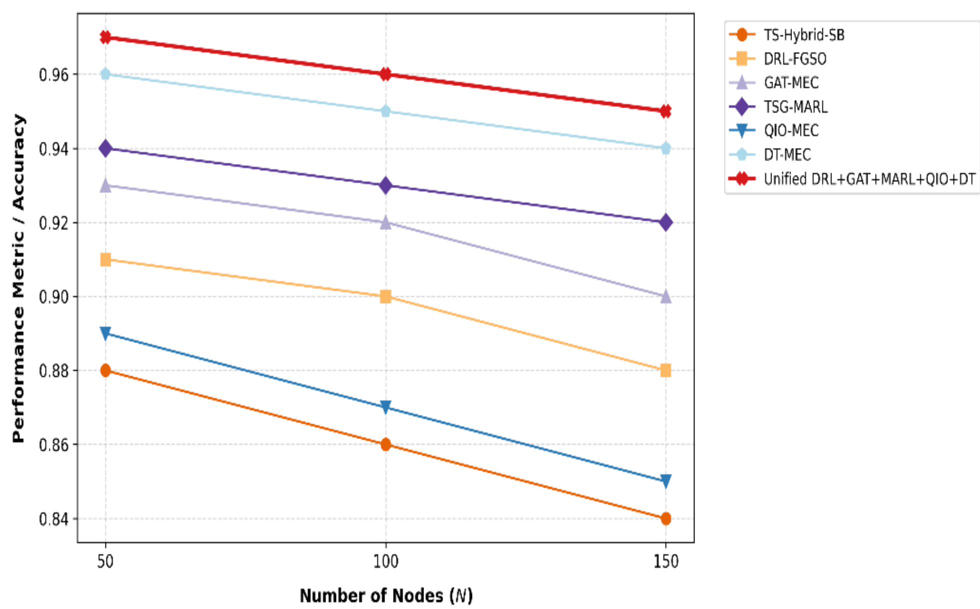

Figure 6: Fairness index comparison of MEC-IoT scheduling methods

Despite thorough simulations with MATLAB and Python based environments carried out to evaluate the presented architecture, simulation configuration was meticulously planned to replicate realistic MEC-enabled IoT systems with diverse hardware devices, variable workloads, time-variant wireless channels and bounded resources. Simulation based evaluation has become a well-received method in the context of MEC research as it allows rigorous and scalable examination of the sophisticated optimization frameworks prior to implementation. However, issues related to hardware diversity, communication overheads, synchronization delays and real-time computing limits may affect the performance of the system in real life deployment. Future work will address thorough verification of the proposed

DRL-GAT-MARL-QIO-DT framework using physical MEC testbeds and IoT hardware platforms. Experiments with real environments in smart cities, industrial IoT and 6G edge devices would also be considered to investigate deployment feasibility, scalability, resilience and adaptiveness, additionally provide new perspective on deployment costs and resource constraints.

Ablation Study of Proposed Hybrid Model

To evaluate the contribution of each component in the proposed DRL–GAT–MARL–DT unified framework, an ablation study is conducted by systematically removing or modifying key modules and observing performance variations. The study focuses on four configurations:

a) Full Model (DRL + GAT + MARL + DT)

b) Without DT (No Prediction Layer)

c) Without GAT (No Graph Learning)

d) Without MARL (No Game Learning)

e) Without DRL (Heuristic Replacement)

The table 5 clearly shows the contribution of each module in the proposed hybrid framework. The complete model exhibited the highest performance in all measures, proving the success of the combination of the elements of DRL, GAT, MARL, and DT. The elimination of the DT module causes a marked increase in latency and power consumption, making predictive optimisation important in the management of dynamic workloads. Cutting the GAT module makes the process less fair and throughput, indicating that graph-based learning is the key to learning network dependencies. The lack of MARL will lead to decreased utility and fairness as there will be no strategic pricing and learning of equilibrium. The removal of the DRL most significantly impairs performance, both in terms of latency and energy, proving that adaptive decision-making is the key to efficient scheduling. In general, the findings confirm the individual contribution of each component, and their integration will create a strong, scaled, and high-performance optimisation framework for MEC.

Table 5: Ablation study results

Model Variant	Latency (ms) ↓	Energy (J) ↓	Throughput (Mbps) ↑	Fairness ↑	Utility ↑
Full Model (All Modules)	**115**	**0.88**	**6.4**	**0.95**	**0.80**
Without DT	135	1.05	5.8	0.91	0.72
Without GAT	130	0.98	5.9	0.92	0.70
Without MARL	125	0.95	6.0	0.90	0.68
Without DRL	140	1.10	5.6	0.89	0.65

5 Conclusion

This work proposes the hybrid DRL-GAT-MARL-QIO-DT approach to adaptively manage offloading decisions and resources allocation for MEC-enhanced IoT networks. It combines deep reinforcement learning (DRL), graph attention network (GAT), multi-agent reinforcement learning (MARL), quantum-inspired optimization (QIO), and digital twin predictive modeling to consider the impact of latency, energy consumption, fairness, and utility within the dynamic IoT system. From the simulation results, it can be clearly seen that compared to the baseline and state-of-art methods, the proposed hybrid approach greatly enhances the overall performance. For instance, the latency has reduced to 80 ms, 110 ms, 140 ms for N=50, 100, and 150, and compared to DT-MEC, it is 20-30% reduction; the energy consumption is to 0.60 J, 0.85 J, 1.10 J and the fairness index is up to 0.97, 0.96, 0.95 for N=50, 100, 150 devices respectively; showing that resources are fairly allocated on the whole large system. It also shows through ablation studies that all modules of DRL-GAT-MARL-QIO-DT have a great contribution in system performance; the DRL is used for adaptive decision making, GAT for modeling the dependencies within the network, MARL for fair equilibrium-based pricing, QIO for global optimal searching and DT for predictive optimization. The synergy from the combination of learning, graph modeling, multi-agent economy, quantum-inspired exploration and predictive intelligent is exhibited on MEC environment. The future studies are looking forward to deploying this proposed framework on real MEC testbed for evaluation and feasibility analysis on the scalability. Furthermore, to explore a higher application level, such as integration with privacy-preserving approaches, blockchain-based resource sharing mechanisms, security-aware scheduling policies for industrial and smart city IoT systems. Integration with lightweight models, heterogeneous hardware and the 6G network could improve efficiency, reliability and robustness on practical scenarios. In general, the proposed hybrid approach can serve as a feasible and smart solution for large-scale MEC-enabled IoT systems in various future applications.

References

[1] Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2017). Mobile edge computing: A survey. *IEEE Internet of Things Journal*, *5*(1), 450-465. https://doi.org/10.1109/JIOT.2017.2750180

[2] Abdullaev, I., Prodanova, N., Bhaskar, K. A., Lydia, E. L., & Kadry, S. (2023). Task Offloading and Resource Allocation in IoT Based Mobile Edge Computing Using Deep Learning. *Computers, Materials & Continua*, *76*(2). https://doi.org/10.32604/cmc.2023.038417

[3] Arun, V., & Azhagiri, M. (2025). Hybrid deep reinforcement learning and genetic algorithm-based resource allocation framework for edge computing. *Peer-to-Peer Networking and Applications*, *18*(6), 320. https://doi.org/10.1007/s12083-025-02149-8

[4] Balakrishnan, R., Sankhe, K., Somayazulu, V. S., Vannithamby, R., & Sydir, J. (2019, December). Deep reinforcement learning based traffic-and channel-aware OFDMA resource allocation. In *2019 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-6). IEEE. https://doi.org/10.1109/GLOBECOM38437.2019.9014270

[5] Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., & Spanò, S. (2021). Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, *11*(11), 4948. https://doi.org/10.3390/app11114948

[6] Chen, M., Hao, Y., Li, Y., Lai, C. F., & Wu, D. (2015). On the computation offloading at ad hoc cloudlet: architecture and service modes. *IEEE Communications Magazine*, *53*(6), 18-24. https://doi.org/10.1109/MCOM.2015.7120041

[7] Chen, X., Jiao, L., Li, W., & Fu, X. (2015). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM transactions on networking*, *24*(5), 2795-2808. https://doi.org/10.1109/TNET.2015.2487344

[8] Jiang, H., Dai, X., Xiao, Z., & Iyengar, A. (2022). Joint task offloading and resource allocation for energy-constrained mobile edge computing. *IEEE Transactions on Mobile Computing*, *22*(7), 4000-4015. https://doi.org/10.1109/TMC.2022.3150432

[9] Liu, Y., Peng, M., Shou, G., Chen, Y., & Chen, S. (2020). Toward edge intelligence: Multiaccess edge computing for 5G and Internet of Things. *IEEE Internet of Things Journal*, *7*(8), 6722-6747. https://doi.org/10.1109/JIOT.2020.3004500

[10] Lu, Y., Liu, C., Kevin, I., Wang, K., Huang, H., & Xu, X. (2020). Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and computer-integrated manufacturing*, *61*, 101837. https://doi.org/10.1016/j.rcim.2019.101837

[11] Meng, F. L., & Zeng, X. J. (2013). A Stackelberg game-theoretic approach to optimal real-time pricing for the smart grid. *Soft Computing*, *17*(12), 2365-2380. https://doi.org/10.1007/s00500-013-1092-9

[12] Min, M., Xiao, L., Chen, Y., Cheng, P., Wu, D., & Zhuang, W. (2019). Learning-based computation offloading for IoT devices with energy harvesting. *IEEE Transactions on Vehicular Technology*, *68*(2), 1930-1941. https://doi.org/10.1109/TVT.2018.2890685

[13] Sardellitti, S., Scutari, G., & Barbarossa, S. (2015). Joint optimization of radio and computational resources for multicell mobile-edge computing. *IEEE Transactions on Signal and Information Processing over Networks*, *1*(2), 89-103. https://doi.org/10.1109/TSIPN.2015.2448520

[14] Shen, Y., Zhang, J., Song, S. H., & Letaief, K. B. (2022). Graph neural networks for wireless communications: From theory to practice. *IEEE Transactions on Wireless Communications*, *22*(5), 3554-3569. https://doi.org/10.1109/TWC.2022.3219840

[15] Song, L., Niyato, D., Han, Z., & Hossain, E. (2014). Game-theoretic resource allocation methods for device-to-device communication. *IEEE Wireless Communications*, *21*(3), 136-144. https://doi.org/10.1109/MWC.2014.6845058

[16] Sorooshian, S., Khademi Sharifabad, S., Parsaee, M., & Afshari, A. R. (2022). Toward a modern last-mile delivery: Consequences and obstacles of intelligent technology. *Applied System Innovation*, *5*(4), 82. https://doi.org/10.3390/asi5040082

[17] Vrahatis, A. G., Lazaros, K., & Kotsiantis, S. (2024). Graph attention networks: a comprehensive review of methods and applications. *Future Internet*, *16*(9), 318. https://doi.org/10.3390/fi16090318

[18] Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., & Chen, M. (2019). In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. *Ieee Network*, *33*(5), 156-165. https://doi.org/10.1109/MNET.2019.1800286

[19] Zarandi, S., & Tabassum, H. (2021). Delay minimization in sliced multi-cell mobile edge computing (MEC) systems. *IEEE Communications Letters*, *25*(6), 1964-1968. https://doi.org/10.1109/LCOMM.2021.3051558

[20] Zhang, K., Mao, Y., Leng, S., Vinel, A., & Zhang, Y. (2016, September). Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks. In *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)* (pp. 288-294). IEEE. https://doi.org/10.1109/RNDM.2016.7608300

Authors Biography

Shreenivas Badada is currently pursuing his Ph.D. as a Research Scholar in the Department of Computer Science and Engineering at GITAM (Deemed to be University), Hyderabad, Telangana, India. He received his Master of Technology degree in Computer Science and Engineering. His research interests include Internet of Things (IoT), Wireless Sensor Networks, Machine Learning, Deep Learning, Artificial Intelligence, and Mobile Edge Computing. He has published research papers in peer-reviewed journals and international conferences in the areas of intelligent systems and network optimization. His current research focuses on hybrid learning and quantum-inspired optimization frameworks for adaptive task offloading in edge computing environments.

Dr. Akkalakshmi Muddana is a Professor in the Department of Computer Science and Engineering at GITAM (Deemed to be University), Hyderabad, Telangana, India. She received her Ph.D. degree in Computer Science and Engineering and has over 15 years of teaching and research experience. Her research interests include Machine Learning, Data Security, Artificial Intelligence, and Cyber Security. She has guided several M.Tech and Ph.D. scholars and has published numerous research papers in reputed international journals and conferences indexed in Scopus and Web of Science. She serves as a reviewer for various peer-reviewed journals and is actively involved in funded research projects. Her current work focuses on secure intelligent systems and privacy-preserving data analytics.