Received: 11 June 2025 | Revised: 19 July 2025 | Accepted: 24 July 2025

DOI: https://doi.org/10.65114/aide.4j9e91dk

Journal of Human–AI Decision-Making Systems

Graph-Enhanced Hierarchical Multi-Agent Reinforcement Learning for Adaptive Healthcare Coordination in Smart Fog Systems

Noman Gul | Bashir Hayat

School of Computer Science & IT, IMSciences, Hayatabad, Peshawar, Pakistan

Correspondence

Noman Gul, IMSciences, Hayatabad, Peshawar, Pakistan
Email: noman.gul@imsciences.edu.pk

Bashir Hayat, IMSciences, Hayatabad, Peshawar, Pakistan
Email: bashir.hayat@imsciences.edu.pk

Abstract

Fog computing has emerged as a promising solution for resource-constrained, real-time applications, particularly in the healthcare domain. However, efficient task scheduling remains a significant challenge in dynamic environments. This paper introduces the GE-HMARL (Graph-Enhanced Hierarchical Multi-Agent Reinforcement Learning) framework to address task scheduling in healthcare fog computing systems. The proposed framework combines hierarchical reinforcement learning with graph-based context modeling to enhance task allocation, resource management, and real-time decision-making. We evaluate GE-HMARL against traditional scheduling methods, including Random Task Scheduling, Priority-Based Scheduling, Flat RL, and Non-Adaptive Scheduling, using key performance metrics such as task completion time, load balancing efficiency, emergency response time, and energy consumption. The experimental results show that GE-HMARL consistently outperforms the baseline methods, achieving up to 44.7% reduction in task completion time, 28.6% lower energy consumption, and up to 17.6% improvement in load balancing efficiency. Additionally, GE-HMARL achieves the fastest response times, improving by 60% over the next best method. These findings demonstrate the effectiveness of GE-HMARL in optimizing task scheduling for healthcare applications in fog environments, offering a more efficient and scalable solution for real-time, resource-constrained systems.

Keywords

Fog computing, GE-HMARL, multi-agent systems, healthcare.

1. Introduction

The healthcare sector is undergoing a significant technological shift with the rise of smart devices, the Internet of Medical Things (IoMT), and patient monitoring systems. These systems produce vast amounts of real-time data that must be processed promptly and securely. Nevertheless, traditional cloud computing architectures often fail to meet the demanding latency and privacy requirements of healthcare workloads. This is especially problematic in time-sensitive applications like emergency treatment, remote surgeries, and intensive care unit (ICU) operations, where a delay in the decision-making process can impact the patient [1]. Fog computing is the possible solution that provides the connection between the cloud infrastructure and the end devices. It also allows processing of data at the edge of the network thereby reducing latency and making the best use of bandwidth. Such an arrangement enables localized services to be independent from centralized systems [2]. Applied to the healthcare sector, edge-based decision-making can accelerate diagnostic workflows and enhance resource provision. However, the heterogeneity of workloads, devices, and evolving patient needs, makes it hard to manage fog computing systems in healthcare. Conventional resource management techniques are not always able to adjust to these dynamic settings [3].

To solve these difficulties, reinforcement learning (RL) is being explored to optimize scheduling and coordination in fog computing systems. In comparison with fixed-rule systems, RL allows the system to improve by the actions it takes as it interacts with the environment on a regular basis. Hierarchical reinforcement learning (HRL) extends this by further decomposing complex decisions into simpler manageable sub-tasks. This can makes it suitable to be applied to fog environments where multi-level control needs to be applied, including central planners, regional coordinators and local executors [4]. Most RL-based fog systems solutions do not consider the interactions between the agents. Which can lead to sub-optimality of decisions made in the network. The work of a single node in a healthcare fog system can largely affect others. As an example, resource management in the hospitals would be greatly efficient with real-time information on the availability of resources and connectivity. Graph Neural Networks (GNNs) are a solution to this problem as they are able to learn the relationships in the data, taking into account node features, edge attributes, and network topology. GNNs improve localized action decision-making by giving it a global perspective. [5], [6]. When applied to multi-agent systems, GNNs let agents to better coordinate their actions and share information about the network.

In this paper, we propose a new framework called Graph-Enhanced Hierarchical Multi-Agent Reinforcement Learning (GE-HMARL). It incorporates GNN modules into a hierarchical MARL framework. This method enhances the coordination in the distributed healthcare fog nodes. Individual agents are trained on their local environment but also benefit with knowledge of graph-based features of the network as a whole. The model enhances individual and system-level decision-making by adjusting to real-time variations in workload and using numerous layers of control. The major contribution of the research is as follows:

- A novel GE-HMARL framework that integrates Graph Neural Networks (GNNs) into a Hierarchical Multi-Agent Reinforcement Learning architecture to enable adaptive healthcare coordination in fog computing systems.
- A graph-based context modeling to improve task allocation by considering the relationships between tasks, resources, and agents.
- Comprehensive evaluation using realistic simulations shows improved task response, load balance, and resource use over standard RL and non-graph MARL models.

2. Literature Review:

Fog computing in healthcare has received considerable interest, because of its ability to limit latency and enable real-time decisions at the network edge. Most of research has been done on how fog computing can be utilized in critical healthcare processes, including early diagnosis, remote patient monitoring, and emergency situations. As an example, Choppara et al. [7] suggested a thorough classification of fog computing and its benefits in time-sensitive healthcare services. Similarly, in another research [8], it was mentioned how fog-driven medical and cyber-physical systems might lead to a more cost-effective use of resources. As much as architectural design progress has been achieved, most of these systems continue to use the static resource management methods, which find it hard to keep up with the real time changes. To address these limitations, reinforcement learning (RL) has emerged as a more dynamic, data-driven approach to resource scheduling and task allocation in fog and edge environments. Recent research has demonstrated that RL can optimize adaptive offloading and bandwidth allocation in IoT systems, often outperforming traditional rule-based methods. For example, Ji et al. [9] used deep RL to manage the resources to process videos on edge servers. However, the conventional RL models have a serious problem in applying to the large-scale or multi-layered systems. Such challenges can be linked to the complexity of the state-action space and the lack of a structured coordination mechanism in the system.

The challenges can be potentially solved using hierarchical reinforcement learning (HRL), which adds a layered control hierarchy. By allowing high-level agents to break down complex tasks into simplified subtasks, this approach can enhance the learning efficiency as well as the scalability. HRL, on the one hand, has already shown its effectiveness in such areas as robotics / task scheduling, however, there are few studies that apply HRL to the context of fog-based healthcare systems. Besides, HRL does not solve the essential interdependencies among distributed agents on its own, which is an important issue in healthcare settings where agent collaboration is essential. This is the role of Graph Neural Networks (GNNs). GNNs provide a strong basis to model agent-agent relationships and have already demonstrated potential in traffic control, energy networks, and multi-agent pathfinding applications where agent-agent relationships are central. Xue et al. [10] thoroughly surveyed the GNN methods, and Nham et al. [11] managed to employ GNNs to traffic system interaction modeling, with successful results. However, using GNNs in healthcare fog systems, especially in a learning-based multi-agent system, has not been thoroughly investigated.

In existing studies, there is a lack of research that tried to combine HRL, GNNs, and multi-agent systems into a unified model of healthcare fog environments. Most other methods address coordination, graph modeling and learning, as independent aspects. In this work, we aim to fill this gap by directly applying GNNs to the policy networks of hierarchical multi-agent reinforcement learning (MARL) systems. Such integration enables agents to make informed decisions, combining local observations with shared, graph-based representations of the whole system.

3. System Architecture & Methodology

The GE-HMARL framework improve the coordination of distributed healthcare systems to integrate the graph-modelling, fog-computing, and multi-agent learning. It resembles the actual healthcare systems in which services are distributed in edge devices, local clinics, and hospitals. The framework learns inter-device dependencies using graph structures. In layer-based hierarchical multi-agent reinforcement learning (MARL), the complexity, workload variation, and policy learning are addressed in a layered manner. It can enhance efficiency and coordination in dynamic healthcare environments.

3.1 System Overview:

The architecture is based on the three-tier control hierarchy where each tier performs the different levels of decision-making. The framework reflected the hierarchical structure of smart healthcare systems. On the top, there is Regional Coordinator Agents (RCAs) that control macro-level policies and decision. They are found in regional fog controllers and manage several healthcare zones. They are supposed to distribute emergency resources among the cities/districts, settle any disputes between hospitals and modify policies in response to demand patterns and public health alerts. Zone Manager Agents (ZMAs) act on the scale of a single healthcare zones, e.g., city/hospital networks. They manage hospitals, clinics and fog nodes in their zones, including intra-zone task allocation, load balancing, and optimization of data flows, and when necessary, escalate problems to RCAs. Edge Node Agents (ENAs) are implemented on the lowest level in hospitals, ambulances, and wearable medical devices. These agents implement fine-grained control functions such as: prioritizing patient monitoring, triaging sensor signals and task scheduling on local fog-enabled devices.

3.2 Graph-Based Context Modeling

The agents in a distributed fog-based healthcare system run in a highly dynamic and connected environment. The behavior and state of other components in the system will tend to affect the performance of one component. To effectively learn such interdependencies, the GE-HMARL framework formulates the environment as a dynamic heterogeneous graph $G = (V, E)$ In this graph, each node $v \in V$ represents an entity such as a hospital server, fog node, ambulance, or wearable medical device. The nodes are instrumented with real-time feature vectors x_v , containing valuable measurements such as processing load, memory utilization, patient queue information, and device role. The relationships and the communication links between the nodes are modeled by the edges $e_{uv} \in E$. These relations may be physical like direct wireless/fiber-optic connections, or logical relations, such as confidence in coordination, handoff dependencies between patients/emergency broadcast connections. Each edge further has a feature vector that monitors important dynamic statistics, such as latency, bandwidth, reliability, and security confidence.

The graph is actually dynamic and is constantly updated to reflect changing conditions like system failures, congestion events or mobility patterns. A Graph Neural Network (GNN) is used to utilize this abundant structural information. The GNN updates node embeddings h_v which contain the local state of a node as well as the effect of its neighbors node. These embeddings are based on graph convolution operations, which collect information about neighborhood $\mathcal{N}(v)$ of each node. The operation is defined as:

$$h_v = \mathrm{GNN}(x_v, \{x_u : u \in \mathcal{N}(v)\}) \qquad (1)$$

This is a transformation of the raw feature vectors and graph topology into meaningful context representations. These representations are in turn fed to the policy networks of the respective agents. The embeddings allow the agents to decide not only about their states but also to co-ordinate with others. Consequently, the decisions become localized and globally informed. This context modelling is essential towards adaptive and efficient decision support in dynamic healthcare settings. To take a specific example, an edge node might not process a task locally when the graph shows that the load is lighter and the availability is higher on the neighbouring nodes. On the same note, an agent on the zone level could also reallocate tasks by anticipating future congestion through the embeddings. The agents become structurally aware by adding Graph Neural Networks (GNNs) to the observation space. This allows agents to cooperate on a large scale without explicit communication.

3.3 Hierarchical Multi-Agent Learning

The suggested framework relies on an HMARL (Hierarchical Multi-Agent Reinforcement Learning) model to manage adaptive healthcare coordination in a multi-fog layer. Such a design reflects the composition of actual smart healthcare systems where the decision is taken at different levels of abstraction and power. The agents act in an common environment where each agent acts independently to optimize its behavior to be beneficial locally as well as globally to system performance. This system is formalized as a multi-agent Partially Observable Markov Decision Process (POMDP), which is defined as: $\mathcal{M} = \langle N, S, A, T, R, O, \Omega, \gamma \rangle$. Here total number of agents is denoted by N. S is the global system of states. $A = A_1 \times A_2 \times \ldots \times A_n$ Is the joint action space of all agents, and $T(s' \mid s, a)$ defines the probability of transitioning from state s to s' after action a. R the reward function that returns a numerical value for each agent's performance, O is the local observation space for individual agents, Ω is the joint observation space, and $\gamma \in [0,1)$ is the discount factor controlling the importance of future rewards. Each agent i receives a local observation $o_i \in O_i$, along with a graph-encoded embedding h_i, representing its local and contextual environment. The agent selects actions based on a stochastic policy π_i, optimized to maximize the long-term expected return:

$$J_i(\pi_i) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^{\,t} R_i(s_t, a_t) \right] \tag{2}$$

In this equation, J_i is the objective function for the agent i, $R_i(s_t, a_t)$ is the reward received at the time step t for taking action a_t in state s_t, and γ^t reduces the weight of future rewards over time. This framework allows agents to think about the short-term effect and the long-term effect of their actions. This feature is necessary in time-based fields such as healthcare. In order to simplify the learning procedure, and to capture control hierarchies, the policy of each agent is factored into two layers. The high-level policy π^H selects sub-goals and directives, while the low-level policy π^L maps those sub-goals to concrete actions:

$$J_i(\pi_i^H, \pi_i^L) = \mathbb{E} \left[\sum_{t=0}^{T} \gamma^t \; R_i(s_t, a_t) \right] \tag{3}$$

This formulation takes a similar structure to the single-layer objective, with one important difference: the actions a_t are the result of a goal-action decomposition. This decomposition enables reuse of learned sub-skills, and improves sample efficiency. For example, a high-level goal would be offloading a patient to an adjacent zone, whereas the low-level policy would decide on which node to target and how to direct the data. The training is performed in accordance with the Centralized Training with Decentralized Execution (CTDE) methodology. In this approach, agents can access information globally during training, courtesy of a common value function $V(s)$ that evaluate the quality of each system state s:

$$V(s_t) = \mathbb{E}_\pi \left[\sum_{t'=t}^{T} \gamma^{t'-t} R(s_{t'},a_{t'})\right] \tag{4}$$

Whereas, $V(s_t)$ in the above equation, it denotes the state value function, which approximates the sum of the expected reward starting at time t given that the current policy, π s followed. This value is useful in determining how good a certain state is upon learning, irrespective of the actual action performed. Each agent's policy π_i is then improved through gradient-based optimization by using the following update rule equation:

$$\nabla_{\theta_i} J(\theta_i) \approx \mathbb{E}\left[\nabla_{\theta_i} \log \pi_i(a_i| o_i,h_i)\cdot\hat{A}_i\right] \tag{5}$$

Here, θ_i are the parameters of the agent i's policy network, and $\hat{A}_i$ is the advantage estimate, a scalar that quantifies how much better or worse an action was compared to the average behavior. This expression is derived from the actor-critic framework. It focuses the learning updates on behaviors that are beneficial. To support role-specific optimization, each agent is trained with its own customized reward function. In the case of the Regional Coordinator Agents (RCAs), the objective is to minimize the total delay and to balance the workload among the different zones:

$$R_{\mathrm{RCA}}^{t} = -\mathrm{GlobalDelay}_t + \lambda \cdot \mathrm{ZoneBalance}_t \tag{6}$$

In the above equation, $\mathrm{GlobalDelay}_t$ is the cumulative response latency at time t, and $\mathrm{ZoneBalance}_t$ measures the balance of the distribution of the workloads among the various regions. The parameter λ balances the objectives of minimizing delay and maximizing fairness. The Zone Manager Agents (ZMAs) work to balance local performance and to make sure that no region is overloaded:

$$R_{\mathrm{ZMA}}^{t} = \mathrm{LocalUtilization}_t - \delta \cdot \mathrm{Overload}_t \tag{7}$$

Whereas the $\mathrm{LocalUtilization}_t$ rewards the utilization of the local capacity within a zone, whereas $\mathrm{Overload}_t$ penalty the behavior that causes local overload. The coefficient δ controls the weight of the penalty. The Edge Node Agents (ENAs) aim at achieving high execution efficiency with minimal power:

$$R_{\mathrm{ENA}}^{t} = \mathrm{TasksCompleted}_t - \beta \cdot \mathrm{EnergyUsed}_t \tag{8}$$

Here, the reward is proportional to the number of tasks that were able to be processed at the edge, and the term $\mathrm{EnergyUsed}_t$ penalizes high power usage. The parameter β allows the system designer the flexibility to adjust between responsiveness and sustainability.

4. GE-HMARL Framework Design

In the last section, the architectural hierarchy and the agent roles were discussed, as well as the graph-based modeling. This section now discusses the design and integration of Graph-Enhanced Hierarchical Multi-Agent Reinforcement Learning (GE-HMARL) framework. It also defines the interconnections between the learning modules. This collaboration enables flexible and responsive decision-making at fog computing systems in real-time healthcare systems. The learning framework focusses on merging structural graph encoding and hierarchical reinforcement learning. The policy network of each agent receives two inputs, namely its local observation o_i and a graph embedding h_i that provides context-aware features obtained through the Graph Neural Network (GNN). With this design, agents are able to take decisions not only with respect to their current state but also with respect to the system dynamics captured in the graph. A multi-layer GNN encodes the dynamic system graph $G = (V, E)$ and aggregates and transforms node and edge features across the network. The embedding of any node v, is calculated according to equation 1. Where x_v is the feature vector of node v, and $\mathcal{N}(v)$ represents its direct neighbors. These embeddings are input to the RL policy:

$$\pi_i(a_i \mid o_i, h_i) \tag{9}$$

This approach enables all agents to make informed choices which are coordinated with others. It balances local control and system awareness. The decision-making process in GE-HMARL is two tiered. Each agent at the higher level employs a high-level policy π^H o specify high-level goals, like load balancing, task offloading and reallocation of resources. Having set these goals, a low-level policy π^L then takes over. It enforces certain operations such as node selection, bandwidth allocation and the sequence in which the tasks in the queue are to be executed. This two level hierarchy assists agents in planning across time horizons and levels of abstraction. It simplifies complex decision making. Although the two levels are trained jointly, they are each tuned to different rewards and feedback. This makes sure that the short term performance is compatible with the long term coordination objectives.

4.1 Agent Collaboration Through Shared Structure

Each agent acts autonomously according to its policy but collaboration between different agents is made possible through the same graph embeddings. Each of the agents shares a GNN-encoded view of the system. Consequently, they naturally tend to have their policies influenced by the state of other agents. This minimizes explicit communication requirements, and hence the system is scalable and tolerant to bandwidth-limited healthcare settings. This structure is modified to suit each kind of agent and its role. RCAs coordinate priorities among zones with global graph signals. ZMAs are concerned with the distribution of resources in their zones and prevent local congestion. ENAs utilize GNN-informed observations to react to patient needs in real-time and address energy limitations.

4.2 Policy Optimization and Training Flow

GE-HMARL optimizes policy according to the Centralized Training with Decentralized Execution (CTDE) strategy. The global information is given to the agents during the training stage. They are updated with policy gradient algorithms like Multi-Agent Proximal Policy Optimization (MAPPO). Equation 5 given above is the core update rule, and the learning process is stabilized by a shared critic network. Decentralized execution enables the individual agents to operate independently in deployment, following training. The outstanding aspect of GE-HMARL is that it can generalize to other network topologies. The agents learned in one area/zone of the hospital/city can transfer their knowledge to another area with a different layout/patient flow since GNNs are not constrained by fixed input dimensions but rather learn based on graph structures. This generalizing capability is especially useful in emergency scale-outs, mobile healthcare roll-outs, and adapted to infrastructure upgrades. GNNs are also able to capture high-order proximities in the graph in addition to direct neighbours. This allows the system to determine the impact of local changes, such as it can determine the impact of rerouting within one fog cluster on other clusters. Such predictive power is essential to preventive healthcare planning, and optimization of responses in smart environments.

5. Experimental Setup and Results

The experiments aims to evaluate the performance of the GE-HMARL framework against four popular methods of task scheduling in a healthcare fog computing. The performance measures are Task Completion Time, Load Balancing Efficiency, Emergency Response Time, and Energy Consumption. These metrics are fundamental towards evaluating the overall performance of healthcare fog systems where real-time task scheduling, effective load management, low response time, and energy-efficiency are paramount. The following frameworks were compared in-terms of their performance: Random Task Scheduling, Priority-Based Scheduling, Flat Reinforcement Learning, and Non-Adaptive Scheduling. Hierarchical Multi-Agent Reinforcement Learning GE-HMARL is specifically designed to function effectively in complex and dynamic environments, which are characteristics of the field.

We combined different simulation tools in order to simulate the fog-based healthcare system and analyze the work of GE-HMARL. The fog network was modelled with YAFS (Yet Another Fog Simulator), which simulates the interactions between fog nodes, task offloading, and resource management. YAFS is a flexible, Python-enabled simulator, which allows high scalability and complex modeling of fog environments. It has resource simulation, latency, and node communication features, which makes it a perfect choose to evaluate fog-based healthcare systems. Besides YAFS, we modelled the urban traffic using SUMO (Simulation of Urban Mobility) to trace the mobile healthcare units, i.e., ambulances. We used the MIMIC-III/eICU clinical datasets to simulate healthcare workloads, including patient data, physiological signals, and task demands. These datasets provide a practical base to test healthcare-related activities and decision making.

5.1 Performance Metrics

To evaluate the performance of GE-HMARL, we refer to certain performance metrics such as Task Completion Time which measures the sum of time taken to accomplish a healthcare task, e.g. processing patient data or sending an ambulance. Load Variance monitors the workload balance among fog nodes and the lower the variance the more balanced the workload. Response Time is the amount of time the system requires to react to an emergency task, e.g. prioritizing a critical patient case or mobilizing the required resources. The Energy Consumption measures the amount of energy that the system consumes when processing tasks, which is essential in resource-limited fog environments.

As baselines, we compare GE-HMARL to Random Task Scheduling, Priority-Based Scheduling, Flat Reinforcement Learning and Non-Adaptive Scheduling. In random task scheduling, the tasks are assigned randomly without considering system load or resource availability. In priority-based scheduling, the tasks are assigned based on predefined priority rules, such as patient condition or task type. A flat RL model is used for task scheduling, without hierarchical decision-making or graph-based context modeling. In non-adaptive scheduling, a static method is used where tasks are assigned based on fixed rules, without considering dynamic changes in system conditions.

5.2 Results and Evaluation

The experimental results demonstrate that GE-HMARL consistently outperforms the baseline methods in key areas such as task completion time, load balancing, and resource utilization.

Task Completion Time Comparison: Figure. 1 compares the task completion times of GE-HMARL with the baseline methods in various healthcare scenarios. GE-HMARL achieves the fastest task completion times, with a significant improvement over methods such as Random Task Scheduling and Flat RL. In particular, GE-HMARL reduces task completion time by up to 44.7% compared to non-adaptive scheduling methods. Figure 1 clearly demonstrates that GE-

HMARL can significantly reduce task completion time, which is crucial for real-time applications in healthcare. By optimizing task allocation and minimizing delays, GE-HMARL enables faster decision-making and more responsive healthcare services.

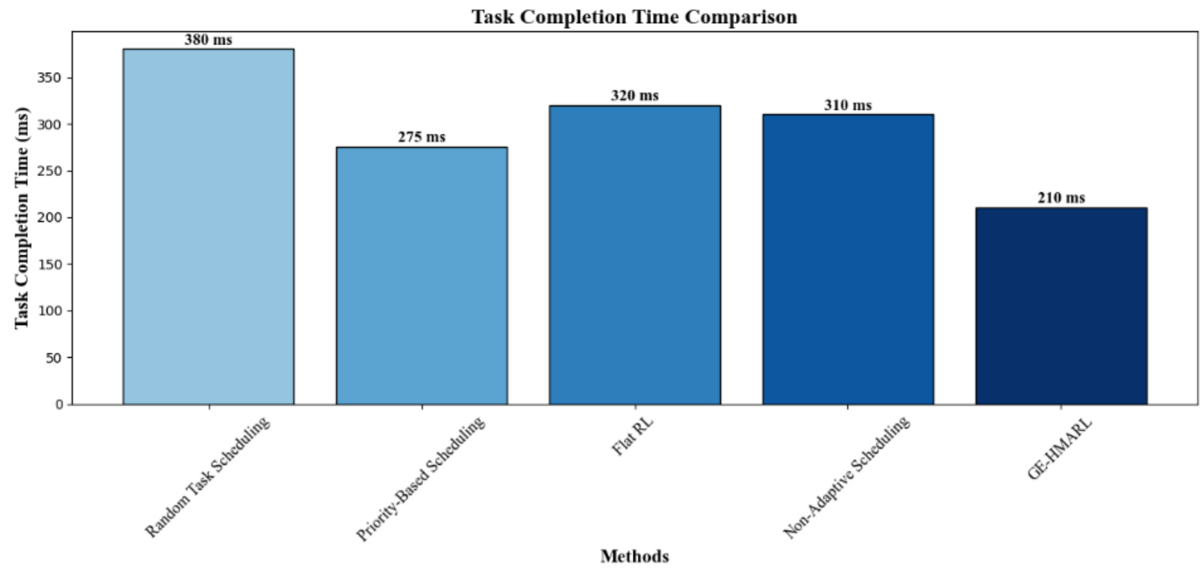

Figure 1. Task Completion Time comparison across different methods in healthcare task scheduling

Load Balancing Efficiency: As shown in the figure. 2, the load variance for GE-HMARL and the baseline methods. Smaller load variance indicates improved load balancing. GE-HMARL is efficient in minimizing the variance of loads, and it provides more balanced workloads among fog nodes. This aids in avoiding congestion and increasing the effectiveness of the system. On this graph, GE-HMARL demonstrates the least load variance, which reflects its capability to deal with dynamic and varying workloads and keep the system stable. This aspect is especially essential in fog computing scenarios where resources are decentralized and tasks may be very dynamic.

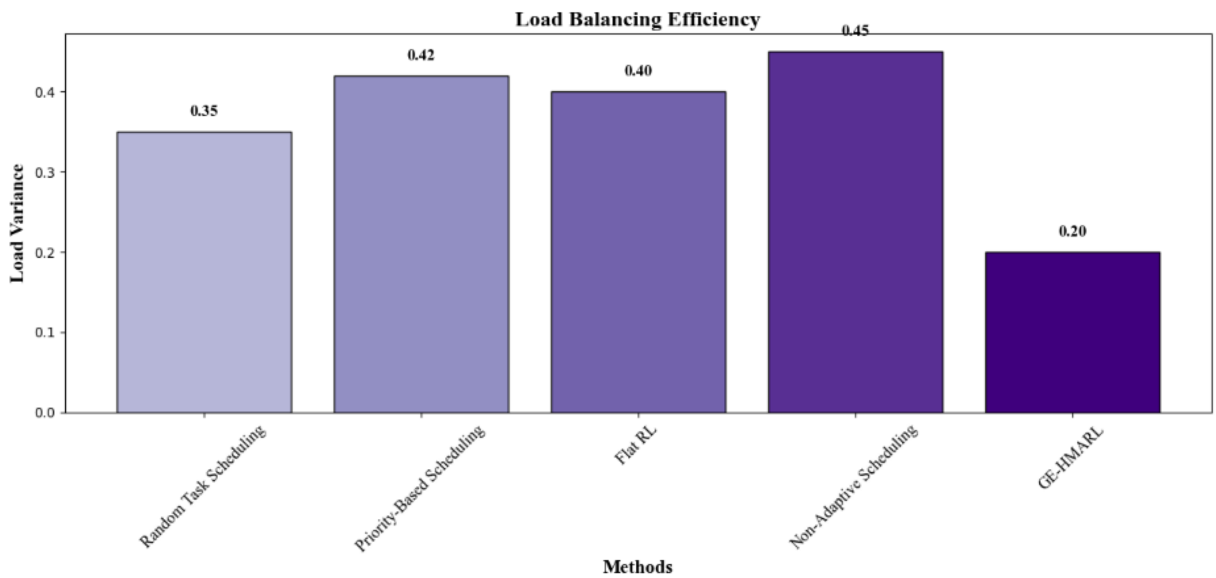

Figure 2. Baseline methods comparison, Load Balancing Efficiency of GE-HMARL

Emergency Response Time: Figure 3 compares the emergency response time in GE-HMARL and the baseline approaches. Among these methods, GE-HMARL has the quickest response time. This demonstrates its high capacity to prioritize time-sensitive healthcare activities and efficiently allocate resources in situations where time is critical. In practical emergency situations such as when dealing with critical patient cases / managing disaster response, fast decision-making may be a life-saver. GE-HMARL is highly adaptable to such environments because in real time, its responses are accurate and timely. Its performance underlines its dependability in fog-based healthcare applications that require fast response.

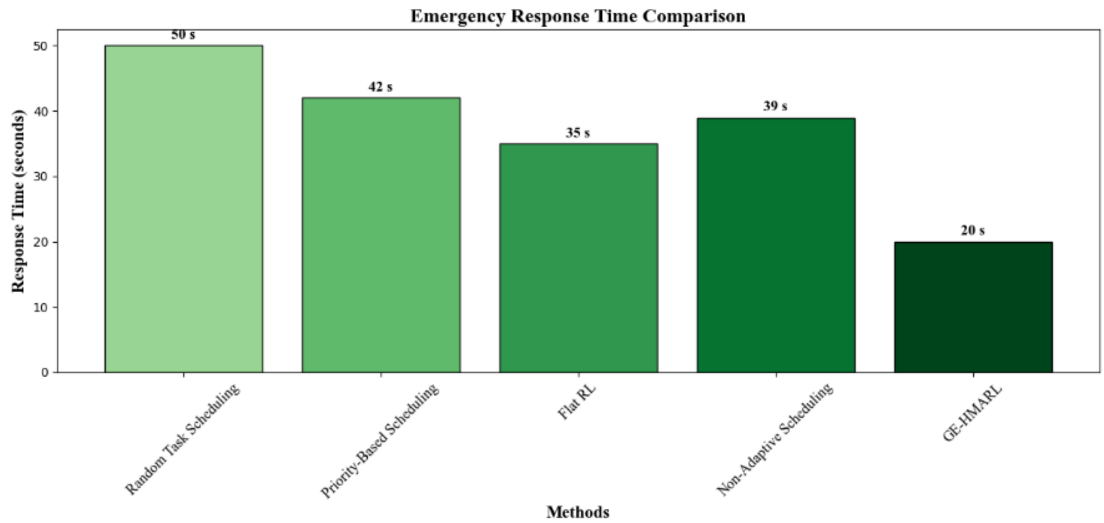

Figure 3. Comparison of GE-HMARL with the baseline approaches in terms of emergency response time

Energy Consumption: Figure 4 shows the energy usage of GE-HMARL to the baseline approaches. GE-HMARL saves 28.6 percent of energy usage compared to non-adaptive scheduling techniques, thus it is an energy-efficient approach that does not compromise in performance. The GE-HMARL framework proves its ability to maintain high performance while minimizing energy usage. Particularly, it is relevant in fog settings with limited resources, where energy efficiency is vital to lower operation costs and achieve sustainability.

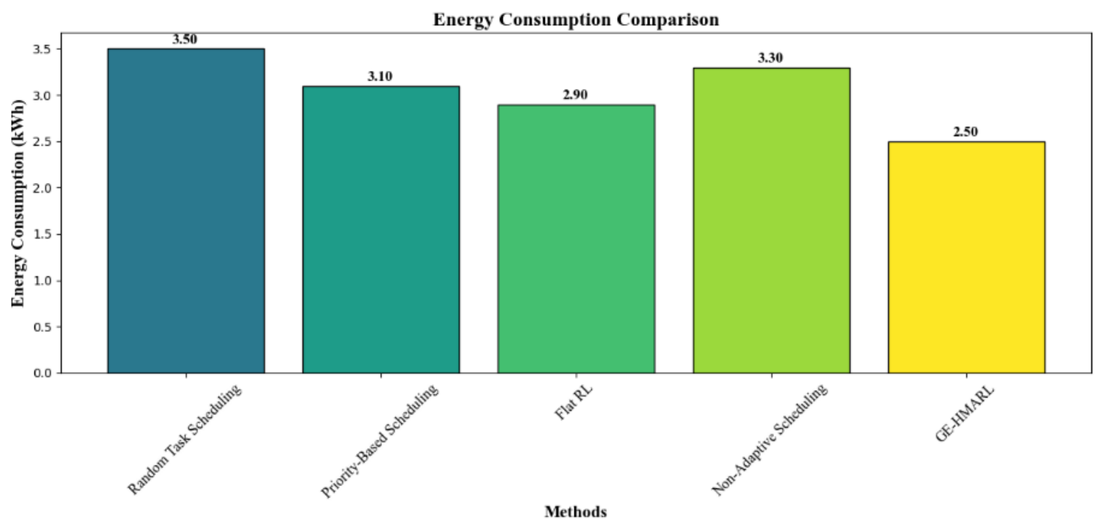

Figure 4. Comparison of energy consumption of the proposed model and the baseline methods

Conclusion

In this research, we presented the GE-HMARL framework for enhancing healthcare task scheduling in fog computing environments. We demonstrated through extensive experimentation that GE-HMARL outperforms traditional scheduling methods, such as Random Task Scheduling, Priority-Based Scheduling, Flat RL, and Non-Adaptive Scheduling, in a variety of important matrices including task completion time, load balancing, emergency response time, and energy consumption. Our findings shows that GE-HMARL consistently delivers faster task completion, better resource utilization, and more efficient load balancing while reducing energy use. These advantages are especially important for time-sensitive healthcare applications where both efficiency and sustainability are critical. By using Hierarchical Multi-Agent Reinforcement Learning, GE-HMARL adapts to the dynamic and resource-limited nature of healthcare environments, supporting real-time decision-making. Additionally, its use of graph-based context modeling enhances collaboration between agents, ensuring the system can manage both local and global decision-making processes effectively.

ORCID

Noman Gul https://orcid.org/0009-0009-7011-8489

Bashir Hayat https://orcid.org/0000-0003-3448-9804

Decision Impact Summary

This study informs operational decisions about scheduling time-critical healthcare workloads across edge and fog resources. In simulation, the proposed approach reduces completion and response times and lowers energy use relative to common baselines, which suggests potential clinical benefits if similar gains hold in practice. To translate this into real settings, organizations should introduce human-in-the-loop safeguards: clinicians or operators approve policy changes, clear thresholds trigger fallbacks to simple rules, and dashboards expose queues and confidence to guide overrides. Before deployment, teams should test under stress and failure scenarios, monitor for distribution shift, and document privacy-preserving telemetry. The immediate next step is a controlled pilot that maps infrastructure improvements to clinical service metrics–such as time-to-treatment or throughput–while logging incidents and operator interventions. Code, configuration files, and a concise model card would help others reproduce results and evaluate readiness for their environments.

References:

1. E. Huaranga-Junco, S. González-Gerpe, M. Castillo-Cara, A. Cimmino, and R. García-Castro, "From cloud and fog computing to federated-fog computing: A comparative analysis of computational resources in real-time IoT applications based on semantic interoperability," Future Generation Computer Systems, vol. 159, pp. 134–150, Oct. 2024, doi: 10.1016/J.FUTURE.2024.05.001.

2. D. Alsadie, "A Comprehensive Review of AI Techniques for Resource Management in Fog Computing: Trends, Challenges, and Future Directions," IEEE Access, vol. 12, pp. 118007–118059, 2024, doi: 10.1109/ACCESS.2024.3447097.

3. H. M. Khater et al., "Empowering Healthcare with Cyber-Physical System - A Systematic Literature Review," IEEE Access, vol. 12, pp. 83952–83993, 2024, doi: 10.1109/ACCESS.2024.3407376.

4. Z. Shamsa, A. Rezaee, S. Adabi, A. M. Rahimabadi, and A. M. Rahmani, "A distributed load balancing method for IoT/Fog/Cloud environments with volatile resource support," Cluster Comput, vol. 27, no. 4, pp. 4281–4320, Jul. 2024, doi: 10.1007/S10586-024-04403-9/METRICS.

5. S. G. Paul, A. Saha, M. Z. Hasan, S. R. H. Noori, and A. Moustafa, "A Systematic Review of Graph Neural Network in Healthcare-Based Applications: Recent Advances, Trends, and Future Directions," IEEE Access, vol. 12, pp. 15145–15170, 2024, doi: 10.1109/ACCESS.2024.3354809.

6. M. Ali, F. Duchesne, G. Dahman, F. Gagnon, and D. Naboulsi, "New Approaches for Network Topology Optimization using Deep Reinforcement Learning and Graph Neural Network," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3569236.

7. P. Choppara and B. Lokesh, "Efficient Task Scheduling and Load Balancing in Fog Computing for crucial Healthcare through Deep Reinforcement Learning," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3539336.

8. S. S. Tripathy, S. Bebortta, and T. R. Gadekallu, "Sustainable Fog-Assisted Intelligent Monitoring Framework for Consumer Electronics in Industry 5.0 Applications," IEEE Transactions on Consumer Electronics, vol. 70, no. 1, pp. 1501–1510, Feb. 2024, doi: 10.1109/TCE.2023.3332454.

9. X. Ji, F. Gong, N. Wang, J. Xu, and X. Yan, "Cloud-Edge Collaborative Service Architecture With Large-Tiny Models Based on Deep Reinforcement Learning," IEEE Transactions on Cloud Computing, 2025, doi: 10.1109/TCC.2024.3525076.

10. J. Xue, R. Tan, J. Ma, and S. V. Ukkusuri, "Data Science in Transportation Networks with Graph Neural Networks: A Review and Outlook," Data Science for Transportation 2025 7:2, vol. 7, no. 2, pp. 1–27, May 2025, doi: 10.1007/S42421-025-00124-6.

11. S. Nham, J. Lee, S. Yang, J. Kim, and S. Kamijo, "Scenario-Based Segmentation: Traffic Image Segmentation by GNN Based Driver's Scenario," IEEE Access, vol. 12, pp. 13088–13099, 2024, doi: 10.1109/ACCESS.2024.3354379.