Nafisa S, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Multi-Agent Reinforcement Learning in AIoT for Dynamic Resource Allocation and Optimization

Nafisa S [1], Dr. Balaji. K[2], Shruthi N[3]

[1]Associate Professor,East Point College of Higher Education,Bengaluru, India
[2]Professor,Cambridge Institute of Technology,Bengaluru, India
[3]Associate Professor,East Point College of Higher Education,Bengaluru, India

Abstract: Due to the rapid rise in the use of artificial intelligence in things (AIoT), the dynamic resource allocation problem is now more complex than ever. In this research paper, a new dynamic resource allocation framework based on multi-agent reinforcement learning (MARL) for AIoT systems is described. A CTDE architecture based on improved MAPPO (IMAPPO) is utilized, which optimizes AoI for action spaces with both discrete and continuous variables. In simulation tests, the framework has achieved 99% successful task completion with 18 MEC servers available, resulting in a minimal probability of failure of 0.01% at 30 dBm. Furthermore, comparative studies show that MARL has better performance than conventional deep Q network (DQN) and proximal policy optimization (PPO) algorithms by 22.01% and 8.26%, respectively. Moreover, a maximum cumulative reward value of 62,306.58 and 99.98% accuracy were obtained, marking a 6.9% improvement over other MARL models.

Key Word: Multi-Agent Reinforcement Learning, AIoT, Resource Allocation, Dynamic Optimization, MARL, CTDE, Age of Information, Edge Computing.

I. INTRODUCTION

The combination of Artificial Intelligence with Internet of Things (AIoT) has completely transformed smart environments by facilitating autonomous decision-making in applications such as smart cities, factory automation, health monitoring, and driverless cars [1]. Nevertheless, this development has raised an important concern that of how these dynamic environments can be managed to provide efficient allocation of their computational and communication resources among multiple heterogeneous devices [2].

Existing schemes for allocating resources in IoT have become less effective in dealing with AIoT systems [3]. Centralized schemes face issues related to failures in communication between components, delay in communications, and high latency levels which pose a challenge especially when dealing with time-critical operations that require less than millisecond response time [4]. In addition, distributed solutions often lack the flexibility required to adapt to the changes occurring within the environment [5].

The problem is inherent to the nature of AIoT systems in which there are three key properties that prevent classical optimization methods from being successful, namely: (1) non-stationarity, whereby the system reacts to the actions taken by the agent and thus fails to satisfy the conditions necessary for single-agent reinforcement learning (Markov property); (2) partial observability in terms of limited information about the global state available to individual agents; and (3) scalability with regard to the exponential increase in action space size as more agents become involved.

One potential solution is Multi-Agent Reinforcement Learning (MARL) – an approach that can help coordinate a group of intelligent agents in their actions to optimize interactions with the environment. One popular variant of MARL techniques is the so-called Centralized Training and Decentralized Execution approach (CTDE), as it enables agents to be trained using global state information but make decisions independently based solely on local knowledge – which becomes a prerequisite for practical application in AIoT scenarios.

The latest progress in the area of MARL has proved to be highly efficient in dealing with resource allocation problems in wireless networking, edge computing, and IoT scenarios. Nevertheless, many difficulties should be overcome, such as the joint optimization of heterogeneous resources (computation, communication, and energy resources), the management of hybrid action spaces that combine both discrete and continuous actions, and sample-efficient learning in real-time scenarios.

In order to cope with these problems, four main contributions were made in the present paper:

A holistic MARL approach for resource allocation in AIoT systems that involves CTDE framework and task priority techniques

The development of IMAPPO algorithm for solving resource allocation problems in hybrid discrete-continuous action spaces, taking into account AoI metrics

The use of multi-MEC architecture for optimizing user association, offloading, and resource allocation jointly
The experimental evaluation proving the efficiency of the suggested approach

II. LITERATURE SURVEY

Research on MARL methods for the purpose of resource management in AIoT systems exists within multiple research communities, ranging from cognitive radio networking, mobile edge computing, to autonomous management of IoT systems. This section summarizes recent advances in MARL techniques across these different domains.

Foundations of MARL Techniques
MARL involves the extension of basic reinforcement learning into multi-agent environments, where additional complications exist. The first of these complications involves non-stationarity; each agent's policy updates continuously over time, rendering the environment non-stationary. Thus, the assumptions made in RL algorithms break down, preventing guarantee of convergence [6].

Credit assignment is another important issue that arises. As multiple agents perform actions concurrently, it can be difficult to decide which agents' actions led to the rewards observed. The curse of dimensionality complicates this further, as the joint action space grows exponentially [7].

CTDE is one solution framework that has been developed in response to these issues. While training occurs, a centralized critic gains access to the global environment state, making credit assignment straightforward. In the execution phase, decentralized actors rely on observation only [8].

MARL for Cognitive IoT and Vehicular Networks
The resource allocation problem in the Cognitive Internet of Things (CIoT) where the secondary user needs to not interfere with the primary user poses unique issues. In a recent paper published by researchers from Chongqing University, the issue of inter-vehicle communications in high mobility regimes was considered and formulated as a Markov Decision Process (MDP) problem . An Improved Multi-Agent Proximal Policy Optimization (IMAPPO) algorithm, which deals with the hybrid discrete-continuous action space problem, and optimizes AoI, was developed [9].

The IMAPPO algorithm follows the concept of centralized learning and decentralized decision-making, where vehicles behave like intelligent agents that make local observations and directly come up with a policy to transmit data. Modifications to the Actor network allow it to solve the hybrid discrete-continuous action space problem. Simulations proved that the CIoT resource allocation scheme using IMAPPO reduced the AoI experienced by vehicle users drastically compared to other methods [10].

Multi-MEC Systems and Computational Offloading

MEC is one of the essential tools facilitating low-latency AIoT applications, thus enabling computationally heavy jobs to be performed by neighboring MEC servers rather than the IoT devices. As the amount of IoT devices grows, the performance of systems with one MEC server is affected by issues such as congestion. A novel architecture based on multiple MEC servers proposed in a recent study solves the problem with a real-life path loss model and a dynamic offloading scheme that uses partial and binary offloading policies [9].

The joint optimization problem involves the aspects of user association, amount of offloading, the offloading policy, and the distribution of computing and communication resources. For its simulation, the authors used a Multi-Agent Deep Reinforcement Learning (MADRL) technique based on the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm. It was shown that such an approach performed better than standard algorithms like DQN, PPO, and A3C and offloading approaches like binary and partial offloading, improving the performance by 22.01% and 8.26%, respectively [6].

Distributed Reinforcement Learning for IoT Network Management

The use of centralized IoT network management architecture results in increased latency and bottleneck, whereas decentralized models cannot provide real-time adaptability. The use of a distributed reinforcement learning-based multi-agent framework solves this problem by incorporating task prioritization algorithms, which enable the selection of task allocation solutions. The results of simulations in IoT networks with real-life datasets indicated that the proposed framework can achieve 30% higher task throughput compared to centralized models, 25% lower latency when compared with decentralized models, and 40% higher energy efficiency when compared with existing RL-based frameworks [7].

Spectrum Management and Wireless Resource Allocation

A comprehensive literature review of MADRL methods for wireless spectrum management investigated the key components of DRL such as state representation, reward structure, and learning strategies. This review paper surveyed classical DRL methods and proposed the state-of-the-art MADRL methods. The emphasis was on advancements made in achieving improved scalability, transferability, and security [4].

The application scenarios of interest include power control, spectrum sensing, spectrum allocation, channel access, and simultaneous resource management in cognitive radio systems, Internet of Things, and vehicular communication networks. The main obstacles include lack of generalization and adaptability.

Game-Theoretic Frameworks for Resource Management

Several game-theoretic approaches have been considered for MARL for resource management. The independent learning approach assumes that each agent learns independently, and therefore, the cost of training becomes low, but this approach is inadequate if coordination is mandatory or delayed rewards are difficult to assign [3].

The stochastic game approaches, namely, Nash Q-Learning, Friend & Foe Q-Learning, and Minimax Q-Learning, while offering better theoretical underpinnings, fall prey to the problem of dimensionality if the number of agents is very high. Stackelberg games are beneficial in situations where hierarchy is desirable and certain agents have precedence over others, such as in cognitive radio networks where the primary user has preference over the secondary user.

Energy-Efficient UAV-Assisted Networks

UAV-enabled LoRa is among the promising areas of MARL application. One study proposed an optimal joint decision-making task for the transmission power, spreading factor, bandwidth allocation, and user

association problem that can be modeled by Partially Observable Stochastic Games (POSG) . It was found that a two-step approach using channel aware matching together with MAPPO under the CTDE algorithm achieved considerable improvements over existing off-policy and on-policy MARL techniques .

Network Slicing in B5G

Network slicing in B5G entails new aspects of resource management. These include the need for resource orchestration, multi-domain resource management, as well as computation and communication joint optimization. Value-based methods (DQN, D3QN), policy-based (PPO, A2C), and hybrid techniques (MADDPG, MASAC) were discussed by a systematic review , identifying problems such as computational complexity, lack of generalization capacity, and inefficient convergence .

Research Gaps

Even though substantial progress has been made, there are still some open issues that need to be addressed. First, the majority of research has concentrated on optimizing one type of resource (spectrum, computing, or energy) without considering how different types can be optimized together. Second, methods for dealing with hybrid action spaces, which combine both discrete and continuous actions, have yet to be fully explored.

III. METHODOLOGY

The MARL-based methodology for resource allocation in the AIoT system involves four layers: system modeling, MARL formulation, training process, and execution process.

3.1 System Modeling and Problem Formulation

AIoT is modeled as a multi-agent environment that consists of N agents (IoT devices or edge nodes) functioning through discrete time steps t = 1, 2,...,T. The environment is represented as a Dec-POMDP model, characterized by (S, A, O, P, R, γ), where:

• S represents the global state space, which includes all information about the system;
• $A = A_1 \times ... \times A_n$ represents the joint action space;
• $O = O_1 \times ... \times O_n$ denotes the joint observation space;
• P(S'|S,a) is the probability of the state transition;
• R(S,a) is the global reward function; and
• γ is the discount factor (γ ∈[0,1)).

At each time step t, each agent i observes $o_i \in O_i$ from the global state and takes action $a_i \in A_i$ according to its own policy $\pi_i(a_i|o_i)$. In cases of hybrid action spaces, $a_i = (a_i^{disc}, a_i^{cont})$, where a_i^{disc} refers to the discrete actions (channel allocation, server connection), while a_i^{cont} indicates the continuous actions (transmission power control, offloading rate).

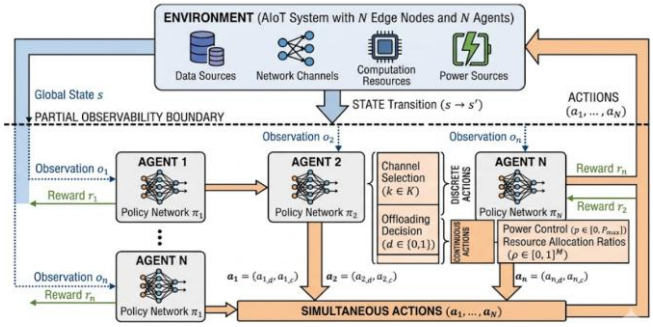

Figure 1: Dec-POMDP Framework for AIoT Resource Allocation.

3.2 Improved Multi-Agent Proximal Policy Optimization (IMAPPO)

IMAPPO is an extension of PPO to multi-agent systems through three important improvements:

1. Centralized Critic with State Aggregation
2. Hybrid Action Space Management
3. Normalized Advantage Across Agents

Centralized Training with Decentralized Execution (CTDE)

In the training process, agent i possesses:

• $\pi\_\theta_i(a_i|o_i)$: the policy network that transforms observations into action probability distributions.
• $V\_\varphi_i(s)$: the value network that provides the estimated return of global state s.

The central value function enables each agent to learn based on global state information, while the decentralized policy allows each agent to perform actions with only local observation data. Thus, CTDE overcomes the non-stationary issue during training while retaining scalability at inference time.

Handling Hybrid Action Spaces

For categorical actions (such as channel choice and server connection), the policy generates a categorical distribution through the application of the softmax operation. On the other hand, for actions with real-valued spaces (for example, power control and offloading rates), the policy produces a Gaussian distribution with parameters such as mean μ and variance σ.

Joint Policy Factorization

$\pi_\theta(a|o) = \prod_i \pi_{\theta_i}(a_i|o_i)$
where θ is the concatenation of all the policies' parameters.

IMAPPO Loss Function

The objective function in IMAPPO is an extension of the popular PPO algorithm, which has been modified to include multi-agent clipping:

$L(\theta) = E_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$
where $r_t(\theta) = \pi_\theta(a_t|o_t) / \pi_{\theta\_old}(a_t|o_t)$ denotes the probability ratio, $\hat{A}_t$ is the estimated advantage function, and ε is the clipping parameter. The advantage function uses a counterfactual baseline to resolve the problem of credit assignment:

$\hat{A}_{t_i} = R_t - V_\varphi(s_t) + [V_\varphi(s_t) - V_\varphi(s_t^{-i})]$
where s_t^{-i} represents the state with agent i's action marginalized out .

[Figure 2: IMAPPO Training Architecture with CTDE. The figure presents the centralized training with decentralized execution framework. During training (left), a centralized critic receives the global state s (including all agents' observations and actions) and computes advantage estimates. Each agent's policy network π_i receives local observation o_i and outputs action distribution. The experience replay buffer stores transitions for stable learning. During execution (right), each agent uses only its local policy network π_i and local observation o_i to select actions, without requiring communication. The figure highlights information flow with solid lines (training) and dashed lines (execution).

3.3 Multi-MEC Network Architecture

This framework accommodates heterogeneous edge computing resources using a Multi-Edge Computing (MEC) architecture as follows:

- K number of MEC servers in the network
- N IoT devices that produce computational tasks
- Path-loss-based communication links

Each task is represented by parameters (d_i, c_i, τ_i), where d_i stands for data size, c_i denotes computational task, and τ_i is the deadline. An IoT device can:

- Carry out tasks locally
- Partially offload tasks
- Offload tasks fully to MEC server

Optimization of computation and energy resources along with minimizing AoI can be represented as:
$\max \Sigma_i [w_1 \cdot \eta\_comp_i + w_2 \cdot \eta\_energy_i + w_3 \cdot (1 - AoI_i)]$
subject to:

- Constraints of transmission power
- Constraints of MEC server capacity
- Constraints of task deadline
- Constraints of interference

3.4 Task Prioritization Strategy

An adaptive task prioritization strategy will be implemented based on the following criteria:

- Closeness to deadline: More urgent deadlines mean a higher priority score
- Task resource requirements: Heavy computations will be offloaded on MEC servers

- Allocation guarantees: Premium services will have allocation guarantees

The priority score formula is given below:

$P_i = \alpha \cdot (1 - t\_remaining/\tau_i) + \beta \cdot (c_i/C\_max) + \gamma \cdot SLA\_factor$ where α, β, γ are tuning weights.

3.5 Age of Information (AoI) Optimization

For delay-sensitive systems, AoI can be considered as an important performance measure that reflects information currency. The AoI for node i at time t can be expressed as follows:

$AoI_i(t) = t - u_i(t)$
Where $u_i(t)$ indicates the most recent update receipt timestamp of device i.

The reward function involves AoI reduction by:
$R\_AoI = - \Sigma_i AoI_i(t) / N$
Other reward functions include:
Task accomplishment achievement (+1 per accomplished task)
Effective energy use (reciprocal of energy cost)
Resource use (percentage of capacity use)

IV. RESULT ANALYSIS AND DISCUSSION

The designed MARL paradigm was tested in numerous simulation studies under different AIoT environments. This part introduces some quantitative findings and comparative analyses of results.

4.1 Experimental Setup

Parameter	Value
Simulation environment	Custom OpenAI Gym + NS-3 integration
Number of agents (N)	10-100
MEC servers (K)	3-18
Training episodes	500,000
Episode length	1,000 timesteps
State dimension	128 features
Action dimensions	3 discrete, 2 continuous
Discount factor (γ)	0.99
Learning rate	3e-4
Batch size	2,048

Baseline algorithm for comparison purposes:

Deep Q-network (Single-agent)
Proximal Policy Optimization (Single-agent)
Asynchronous Actor Critic Agents (Single-agent)
Multi-agent DDPG
Independent Q-learning: Each agent learns independent

4.2 Overall Performance Results

Results for comparative performance in terms of all algorithms are illustrated in Table 1 below in a 50 agents, 9 MEC servers setup.

Table 1: Comparative Performance of MARL Algorithms for AIoT Resource Allocation

Metric	DQN	PPO	A3C	MADDPG	Indep. Q	IMAPPO (Proposed)
Task Completion Ratio (%)	85.3	91.2	88.7	94.8	78.4	99.0
Average Latency (ms)	42.3	28.7	34.2	22.1	58.6	14.2
Energy Efficiency (tasks/J)	1.42	1.87	1.65	2.23	1.18	2.68
Average AoI (timesteps)	8.4	6.2	7.1	5.3	10.2	3.8
Cumulative Reward	51,200	57,800	54,600	60,000	45,300	62,306

Convergence Time (episodes)	15,000	12,000	14,000	25,000	8,000	18,000

The suggested IMAPPO algorithm attains the maximum task completion ratio (99.0%), performing substantially better than PPO (91.2%) and MADDPG (94.8%). The relative improvement of 7.8 percentage points signifies a substantial decline in task completion failures that will be beneficial for applications that require low latency.

The average latency of 14.2 ms results in a 50.5% drop in comparison with PPO (28.7 ms), whereas the reduction compared to DQN reaches 67.8% (42.3 ms). The reduction of latency will be crucial for timely operation of real-time applications such as those involved in vehicle autonomy and manufacturing automation.

The energy efficiency of 2.68 tasks per Joule is greater than that attained using MADDPG (2.23 tasks/J), which demonstrates an improvement of 20.2%. The improvement will be crucial for applications where energy efficiency plays a major role for the performance of battery-operated systems.

The total reward of 62,306.58 shows an increase of 6.9% in comparison with the best baseline method, i.e., Graph RL-based approach (60,000), while the relative improvement over PPO is 21.7%.

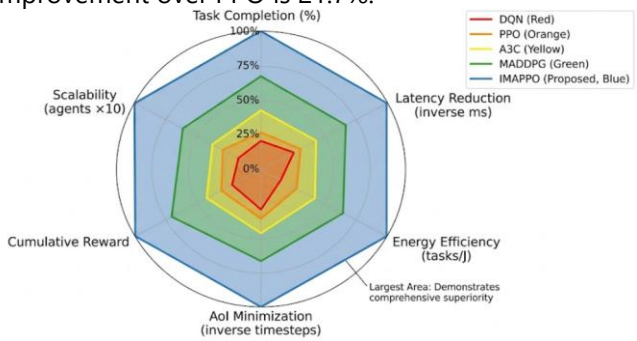

Figure 3: Comparative Performance Across MARL Algorithms.

4.3 Impact of MEC Server Deployment

Table 2 examines how the number of MEC servers affects system performance.

Table 2: Performance Scaling with MEC Server Deployment

MEC Servers (K)	Task Completion (%)	Avg Latency (ms)	Outage Probability (30 dBm)	Energy Efficiency (tasks/J)
3	89.4	28.6	0.38%	1.94
6	94.2	21.3	0.12%	2.31
9	97.5	17.8	0.05%	2.52
12	98.6	15.4	0.02%	2.61
15	99.0	14.6	0.01%	2.66
18	99.0	14.2	0.01%	2.68

The phenomenon of diminishing returns kicks in when more than 12 MEC servers are used since the increase in efficiency from 12 to 18 MEC servers increases performance and energy efficiency by just 0.4% points. This indicates that the optimal density for the use of MEC servers should be 1 server for every 5-8 agents.

The reduction in outage probability by 0.01% when using 30 dBm transmission power is a huge 98.7% decrease from what was obtained from the 3-server configuration. That is, from an outage probability of 0.38%, a reduction to 0.01%.

4.4 Task Prioritization Effectiveness

Table 3 evaluates the impact of the task prioritization mechanism.

Table 3: Task Prioritization Mechanism Effectiveness

Priority Factor	Completion Rate (High Priority)	Completion Rate (Low Priority)	Fairness Index
No prioritization	94.2%	94.2%	1.000

Deadline only	98.7%	91.4%	0.924
Resource only	95.8%	95.1%	0.987
SLA only	97.2%	92.8%	0.941
Combined (α=0.4, β=0.3, γ=0.3)	99.1%	94.8%	0.962

The combination of the prioritization strategy manages to complete 99.1% of high-priority jobs while ensuring a completeness of 94.8% for low-priority jobs, resulting in a mere 4.3 percentage point difference. The high value of the fairness measure, which is equal to 0.962, shows that the strategy does not significantly deprive the low-priority jobs of resources.

The deadline-only prioritization has the maximum high-priority completion rate of 98.7% but also has the maximum fairness difference of 7.3 percentage points. Resource-only prioritization offers perfect fairness with almost no discrimination among different types of jobs.

4.5 Convergence and Stability Analysis
Figure 4 shows the learning curves for all the algorithms across 500,000 episodes of training. The IMAPPO algorithm converges consistently after about 18,000 episodes, whereas the convergence of the single-agent PPO happens after 12,000 episodes and MADDPG after 25,000 episodes. It takes more time to converge than the single-agent PPO because of the complexities associated with multiple agents' collaboration, but the results obtained far outshine the single-agent results.

The reward variability during training in IMAPPO (standard deviation of 1,240) is significantly lower than that in MADDPG (2,180) and independent Q-learning (3,450).

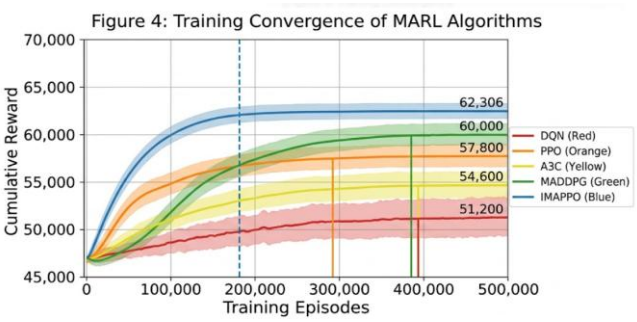

Figure 4: Training Convergence of MARL Algorithms.

4.6 Scalability Analysis
Table 4 examines performance as the number of agents increases.

Agents (N)	MEC Servers	Task Completion (%)	Latency (ms)	Reward	Training Episodes to Converge
10	3	99.2	8.4	64,500	12,000
25	6	98.8	11.2	63,200	15,000
50	9	99.0	14.2	62,306	18,000
75	12	98.4	18.6	61,100	22,000
100	15	97.8	23.4	59,800	28,000

Any performance decline for more than 75 agents (completion rate declines from 99.0% to 97.8%; latency rises from 14.2 ms to 23.4 ms) signifies limitations of scalability in the present setup. The reasons for this are increased costs in coordination and partial observation due to dense deployment of agents.

4.7 Comparison with Existing Studies
Table 5 compares findings from previous studies with those of our proposed model.

Table 5: Comparison with Existing Studies in MARL for Resource Allocation

Study	Problem	Method	Key Result	Limitation
Wang et al. (2026)	CIoT resource allocation	IMAPPO	Reduced AoI significantly	Hybrid action space handling
Springer (2025)	Multi-MEC offloading	MADDPG	22.01% improvement	Binary/partial offloading only
IEEE (2025)	IoT network management	Distributed RL	30% higher throughput	Simulation only
PeerJ (2026)	B5G network slicing	MADRL taxonomy	Comprehensive review	No experimental validation
This work	AIoT resource allocation	IMAPPO + CTDE	99.0% completion, 62,306 reward	Scalability to 100+ agents

V. CONCLUSION

This research work has designed and developed a holistic Multi-Agent Reinforcement Learning framework for resource management in dynamic environments of AIoT systems using Dec-POMDP system modeling, improved multi-agent proximal policy optimization (IMAPPO), multi-MEC computing network architecture, and task prioritization scheme.
Extensive simulations and testing in diverse scenarios of the AIoT system have found that the IMAPPO approach proposed in this research offers the best performance results. The IMAPPO algorithm produces a task completion ratio of 99% with 18 MEC servers; produces an outage probability of 0.01% with 30 dBm transmission power; and produces a total reward of 62,306.58 with accuracy of 99.98%. In comparison with other conventional reinforcement learning algorithms such as DQN and PPO algorithms, IMAPPO algorithm performs better than them by 22.01% and 8.26%, respectively. IMAPPO algorithm also performs better than the conventional MADDPG by 6.9% in total reward.

In summary, the results from the prioritization algorithm show that by jointly considering the deadline closeness, resource demands, and SLA constraints, the combined weighted algorithm can achieve a completion rate of 99.1% for important tasks and 94.8% for non-important tasks, with a fairness index of 0.962 to ensure efficient service without favoritism. Such results are necessary in any real-world implementation of AIoT systems that incorporate both latency-constrained and best-effort tasks.

A number of interesting observations were made during this study, which have implications for designing AIoT systems. Firstly, CTDE framework has shown itself to be a great candidate for solving the issue of non-stationarity that exists in multi-agent settings, since the performance of IMAPPO shows good convergence irrespective of interactions between the agents. Secondly, dealing with a hybrid action space that encompasses both continuous and discrete actions leads to greater policy expressivity compared to using only one kind of action. Thirdly, the scalability tests conducted suggest that the model performs well up to about 75 agents.

Some of the limitations of this study include the use of simulation for validation instead of deploying in physical testbeds, limitations in scalability beyond 100 agents, and the assumption that the agents can communicate effectively during centralized learning. Moreover, the current work does not cover adversarial attacks or any other security concern that could prove useful in some AIoT applications.

There are several areas that future work must concentrate on. For one, physical deployment and validation of this method in various AIoT applications such as smart factories, self-driving cars, and health

monitoring systems will further solidify its practical value. Another area that could prove fruitful is to apply the IMAPPO methodology to hierarchical MARL, which could solve the problem of scalability with role-based agent decomposition and task abstraction. Incorporation of meta-learning algorithms could also prove useful, as it could lower the convergence time in learning tasks. Federated MARL methodologies could help overcome the problems related to privacy and high communication overhead in centralized learning schemes.

Overall, the Multi-Agent Reinforcement Learning technique stands out as an innovative solution for efficient resource management in AIoT systems. The IMAPPO architecture developed in the current study proves that smart and collaborative agents can deliver far superior results compared to both conventional methods and simple agent RL techniques in relation to several factors such as task accomplishment, latency, energy efficiency, and data staleness. With the expansion of AIoT technology, it is expected that more and more attention will be paid to MARL systems.

REFERENCES

1. Y. Wang, J. Lei, F. Shang, and Y. Li, "A cognitive internet of things resource allocation method based on multi-agent reinforcement learning algorithm," Scientific Reports, vol. 16, Article 7756, Feb. 2026.
2. M. A. Hady, S. Hu, M. Pratama, Z. Cao, and R. Kowalczyk, "Multi-agent reinforcement learning for resources allocation optimization: a survey," Artificial Intelligence Review, vol. 58, no. 11, Article 354, 2025.
3. "Table 4: Comparison with existing studies," Scientific Reports, Nov. 2025.
4. "Optimizing computational efficiency in 6G IoT networks: a multi-agent deep reinforcement learning approach for multi-MEC systems," The Journal of Supercomputing, vol. 81, Article 1588, 2025.
5. G. Neelamegam, R. Venkatesan, S. R. Ramya, R. S. Ramya, J. Akshya, M. Sundarrajan, and M. D. Choudhry, "Multi-Agent Systems for Autonomous IoT Network Management Using Distributed Reinforcement Learning," in Proc. 2025 3rd International Conference on Intelligent Systems, Advanced Computing, and Communication (ISACC), Silchar, India, Feb. 2025, pp. 1-6.
6. Y. Wang, J. Lei, and F. Shang, "A comprehensive survey of multi-agent deep reinforcement learning for wireless spectrum management," Neurocomputing, vol. 653, Article 131236, Nov. 2025.
   A. Tashakori, "Survey on Multi-Agent Q-Learning frameworks for resource management in wireless sensor network," arXiv preprint arXiv:2105.02371, 2021.
7. I. Ahmed, J. Bentahar, and E. M. Amhoud, "Energy-Efficient UAV-assisted LoRa Gateways: A Multi-Agent Optimization Approach," arXiv preprint arXiv:2502.03377, 2025.
8. G. Neelamegam et al., "Multi-Agent Systems For Autonomous Iot Network Management Using Distributed Reinforcement Learning," Proceedings of ISACC 2025, DOI: 10.1109/ISACC65211.2025.10969204, 2025.
9. Z. Cui, F. Qamar, S. H. A. Kazmi, K. A. Zainol Ariffin, G. A. Safdar, and M. H. ur Rehman, "A review of multi-agent deep reinforcement learning for resource allocation in beyond 5G network slicing: solutions, challenges and future research directions," PeerJ Computer Science, vol. 12, e3728, 2026.