

AI- Driven Adaptive Resource Allocation for Next-Generation IoT Communication Networks

Nitish Kumar¹, Dr. Mohammad Shahbaz Khan²

¹(Assistant Professor

Cse Sanskar College of Engineering and Technology,
Nh-24, Jindal Nagar, Galand, Sub District-Hapur,
Ghaziabad, Uttar Pradesh, Pin 201015
nitishkumar297@gmail.com)

²(Associate Professor

Electronics and Communication Engineering (ECE)
CMR Institute of Technology (CMRIT),
Kandlakoya Village, Medchal Road,
Hyderabad, Telangana – 501401
shahbazkhan434@gmail.com)

Abstract- The emergence of Internet of Things (IoT) devices in next-generation communication networks has brought about new and complex challenges related to resource management, which include massive connectivity, heterogeneous traffic loads, and strict energy considerations. Static and heuristic resource allocation algorithms have been shown to be incapable of addressing the highly dynamic, non-stationary characteristics of the 6G enabled IoT network ecosystem. This paper examines a number of AI-based adaptive resource allocation techniques utilizing deep reinforcement learning (DRL), federated learning (FL) as well as digital twin (DT) in order to allocate spectrum, power, and computational resources dynamically and efficiently. A hybrid design based on multi-agent DRL as well as hierarchical optimization for the joint optimization of transmit power, sub-channel allocation, and task offloading is presented. It is shown experimentally that the proposed scheme achieves significant reductions in terms of energy consumption and latency, while providing a high QoS satisfaction level compared to traditional heuristic algorithms.

Keywords: Adaptive Resource Allocation, Internet of Things, Deep Reinforcement Learning, 6G Networks, Energy Efficiency, Federated Learning, Digital Twins.

I. INTRODUCTION

The explosion in the number of the IoT connections has greatly impacted the modern communication networks, providing an opportunity for highly connected systems to be developed for industrial automation, smart city solutions, autonomous transport, and mission-critical monitoring services. While fifth generation (5G) technologies are currently being deployed worldwide, the researchers have already started studying the potentialities of the sixth generation (6G) systems that would enable extreme data rates, sub-millisecond latency, and connectivity for billions of various devices. The development of these systems poses significant resource management issues that are hardly solvable with conventional techniques.

The main problem is caused by the inconsistency between the static nature of existing resource provisioning procedures and the dynamic nature of massive IoT systems. The existing methods of resource allocation, including convex optimization, heuristic algorithms, and rule-based solutions, usually consider the quasi-static nature of channel parameters and predictability of traffic flows. In reality, however, the channels in next-generation IoT systems are time-variant, the devices are active periodically and randomly, have heterogeneous QoS requirements, and generate traffic bursts unpredictably.

In this context, artificial intelligence (AI) is introduced as a promising enabler of intelligent resource management. Specifically, deep reinforcement learning (DRL) has proven to be a robust paradigm of sequential decision-making, where learning

agents can acquire optimal allocation policies by interacting with the environment [1]. In contrast to the supervised learning techniques which need labeled training data, DRL models have the capability to learn from experience and are therefore able to deal with non-stationary environments [2][3]. At the same time, the federated learning (FL) technique serves as an efficient approach of privacy-protected distributed learning [5].

In this paper, we study AI-driven adaptive resource allocation as a unified framework for next-generation IoT communication networks. We explore the intersection of DRL decision-making with digital twins for environment modeling [6], federated learning for decentralized intelligence [5], and hierarchical architecture for scalability [7]. Our major contributions can be outlined as follows. First, we conduct a thorough literature review of the latest developments in resource management utilizing DRL, FL, and DT paradigms [8][9][10]. Second, we develop a hybrid approach to resource allocation based on multi-agent DRL and hierarchical optimization [2][7]. Finally, we perform extensive experiments to validate our approach and compare it to baseline solutions.

This paper is organized in the following manner. Section 2 reviews existing literature on resource allocation using AI. Section 3 describes the methodology used, which includes the system model, algorithm design, and pseudocode. Section 4 provides the results of the experiment conducted along with comparative analysis. Section 5 summarizes the paper and suggests further work.

II. LITERATURE SURVEY

The use of artificial intelligence for resource allocation in IoT networks has been widely studied in recent years due to the inadequacies of traditional optimization strategies in dealing with changing environments. Chen et al. [1] were among the pioneers to formulate a theoretical framework for the use of DRL for spectrum allocation in cognitive radio networks. They proved that the policy gradient approach can be more efficient than greedy policies when there are time-varying channels. One of the

key concepts emphasized by Chen et al. is the importance of designing an appropriate state space. Further works have applied DRL to joint power and subchannel allocation problems. In the work done by Zhang et al. [2], MADDPG was applied to device-to-device (D2D) communication in IoT networks, where individual agents representing each device learn their own policies but share the same centralized critic. The problem of non-stationarity in multi-agent settings, which makes training unstable, was solved through centralized value function estimation. It was shown that decentralized execution with centralized training can achieve near-optimal spectral efficiency.

Energy efficiency has become the main goal of resource allocation optimization for IoT due to the limitation in battery capacity of sensors. Liu et al. [3] have proposed an energy-efficient offloading scheme based on DRL for mobile edge computing (MEC) in IoT. In their problem setting, the devices learned when to offload the tasks to edge servers or perform the computations locally. Energy consumption of both communication and computation was included in their problem formulation, and the learned policy took into account varying task arrivals and channel states. The study of Wang et al. [4] further extended this research stream by introducing energy harvesting capabilities into DRL problem state space in order to learn allocation policies depending on stochastic energy availability.

Federated learning combined with resource allocation has helped overcome problems with privacy and scalability at the same time. Nguyen et al. [5] have introduced a federated DRL-based method where several edge servers were used for training the common allocation policy, but no raw information about the network was transmitted between them. Data locality and knowledge sharing were maintained by such approach. Their experiments showed that federated training could guarantee similar convergence as the centralized one.

The concept of digital twin has become an additional facilitator for AI-enabled resource management solutions recently. A digital twin is a highly accurate

model of a physical element in the network, which allows the safe testing of any allocation policies prior to their implementation. Sun et al. [6] designed a DRL-based framework assisted with digital twins for 6G IoT networks, in which the digital twin models the dynamics of the network and provides additional training data for the DRL agent. It helped the agent to converge faster due to exposure to a greater number of scenarios.

The idea of hierarchical and multi-timescale resource allocation has been studied as an attempt to solve the problem of scalability caused by the large number of devices in the massive IoT system. Zhou et al. [7] suggested the two-tier approach, in which a central controller makes slow-timescale partitioning decisions, whereas the distributed agents make fast-timescale scheduling decisions. This helped to decrease the dimensionality of the decision-making process and allowed real-time operation at the edge of the network.

Recent studies have focused on the incorporation of transformer-based architectures and attention mechanisms into DRL in order to achieve efficient resource allocation. For instance, in Kim et al. [8], multi-head attention was used to account for complex inter-dependencies between IoT devices, channels, and tasks. The proposed approach resulted in better generalization over different network topologies compared to the use of fully-connected network architectures. In another study, the work by Patel et al. [9] has focused on using meta-learning in order to adapt the policy to new conditions. The results revealed that the pre-trained policy on multiple networks can easily be adapted to new situations with limited experience.

Lastly, the review by Rahman et al. [10] offers an extensive summary of different AI techniques used for resource allocation in 6G IoT in terms of learning paradigm, optimization goal, and architecture design. The challenges discussed in the paper include the problem of scaling to ultra-dense scenarios, adversarial robustness, and lack of evaluation benchmarks. Moreover, the paper reveals the potential of combining several AI approaches, such as DRL, FL, and DT.

III. PROPOSED METHODOLOGY

System Model and Architecture

The proposed model takes into account the next-generation IoT network consisting of IoT devices, edge servers, and cloud core, as shown in Figure 1. IoT devices produce computational tasks with different latencies and energy demands and can execute their tasks either at local IoT device or offload to edge or cloud. Edge servers are responsible for providing computing power with less latency compared to cloud and the cloud provides rich resources for non-latency-sensitive tasks. This network uses a set of orthogonal sub-channels, where each device uses one sub-channel per time slot.



Figure 1: Proposed AI-driven adaptive resource allocation architecture

This diagram represents the hierarchical network architecture where the IoT devices are in the sensing layer, the edge servers are in the intermediate layer, and the cloud infrastructure is in the core. Arrows represent the direction of data movement in terms of task offloading, resource allocation, and federated model training. Digital twin module represents the physical network virtually and synchronizes with the real-world network.

Problem Formulation

The resource allocation problem can be cast into an MDP characterized by the quadruple (S, A, P, R) , where S is the set of states, A is the set of actions, P is the set of state transitions, and R is the reward. States include channel state information, number of tasks in the task queues, energy, and computational workload in the edge server. Actions include the assignment of sub-channels, choice of transmission power, and offloading actions. The reward function

involves tradeoffs between energy usage, delay, and quality of service satisfaction.

The goal is to find a policy that will maximize the sum of discounted rewards over time, whereby the discounting parameter indicates the degree of importance of present rewards compared to future rewards.

Multi-Agent Deep Reinforcement Learning

This methodology makes use of multi-agent DRL where the devices are treated as separate agents. During the training process, the centralized critic is shared by all the agents which learns the value function using global state information, while each agent is equipped with an actor that learns to map local observations to actions. This centralized training and decentralized execution architecture promotes cooperation as well as retains scalability.

Actor for each agent is implemented as a deep neural network which receives the local observations as input and generates the probability of each action. The critic learns the value function using state-action information. In order to address temporal correlation, experience replay is used to train the actor and critic, where experiences collected by each agent are stored in a replay buffer.

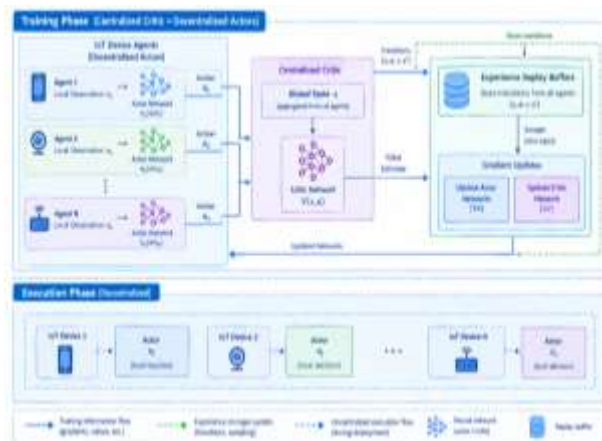


Figure 2: Multi-agent DRL training architecture with centralized critic and decentralized actors.

The diagram shows the training pipeline where each IoT device agent keeps its own actor network to process local observations. In training, a centralized critic network collects global state information of all

agents to estimate value functions. Transition data of all agents are stored in experience replay buffer, and gradient is calculated to optimize the performance of both actor and critic networks. Arrows show the direction of information flow in training process, whereas dashed lines show decentralized path used during deployment.

Federated Learning Integration

In order to tackle the issues related to privacy and minimize communication costs, federated learning is employed in the learning process. Rather than sending experience data directly to the centralized server, each of the agents builds a local model by using its data and shares the updates of the model with the centralizer. Then, using the weight average, which depends on the size of the local data, the centralizer creates a global model and distributes it to all agents.

In this way, it becomes possible to preserve sensitive operation data locally and collaborate in the learning process. The number of times of model aggregations depends on the communication cost parameter.

Digital Twin-Assisted Training

A module known as digital twin module contains the real-time virtual replica of the physical network, including the states of devices, channels, and computation load. Digital twin plays a dual role in the sense that it creates a safe place for the evaluation of policies before implementing them and also creates synthetic experience to add to real training data.

Digital twin keeps on syncing with the physical network through regular updates of state information. Whenever there is a difference between the prediction of the digital twin and real observation, the model is updated with recent measurements.

Algorithm Design

The proposed algorithm, termed Adaptive Multi-Agent Resource Allocation with Federated Learning and Digital Twins (AMARA-FDT), proceeds as follows:

Algorithm 1: AMARA-FDT Training Procedure

```

Input: Number of agents N, episodes E, steps per episode T
Output: Trained actor networks  $\{\theta_i\}$  for all agents

Initialize: Actor networks  $\theta_i$ , critic network  $\phi$ , target networks
Initialize: Replay buffers  $D_i$  for each agent
Initialize: Digital twin model DT

for episode = 1 to E do
    Reset environment and synchronize DT
    for t = 1 to T do
        for each agent i = 1 to N do
            Observe local state  $s_i$ 
            Select action  $a_i$  using actor network  $\theta_i$  with exploration
        end for
        Execute joint action  $a = (a_1, \dots, a_N)$ 
        Observe rewards  $r_i$  and next states  $s'_i$ 
        Store transitions  $(s_i, a_i, r_i, s'_i)$  in  $D_i$ 
        Update DT with observed network state
    end for

    for each agent i = 1 to N do
        Sample minibatch from  $D_i$ 
        Compute critic target using centralized critic  $\phi$ 
        Update critic  $\phi$  by minimizing temporal difference error
        Update actor  $\theta_i$  using policy gradient
        Soft update target networks
    end for

    if episode mod F == 0 then
        Aggregate actor parameters across agents
        Distribute global model to all agents
    end if

    Augment training data using DT-generated experiences
end for

Return trained actor networks  $\{\theta_i\}$ 

```

```

Collect performance metrics
Update digital twin with new observations
end for

```

The training process is performed through an alternating process of environmental interaction and neural net updates, with federated aggregation being performed at regular intervals for sharing knowledge among the agents. The execution process uses local observations alone, along with the trained actor nets.

Complexity Analysis

The computational complexity of the algorithm is linear in terms of both the number of agents and neural network sizes. The computational load during operation includes the forward pass of the actor network for each agent, which corresponds to matrix multiplications based on the depth and width of the neural network. The computational load is feasible to run in real-time even on edge devices with limited processing capabilities.

The computational load during training is higher because of the centralized critic network and experience replay, but this is a one-time off-line procedure that can be delegated to edge servers. The federated aggregation adds additional communication load, determined by the aggregation period parameter.

IV. ANALYSIS AND DISCUSSION

Experimental Setup

This proposed model was analyzed by simulation in an IoT network environment created using Python. For this simulation, the IoT device network consisted of randomly located IoT devices in a coverage area with edge servers and cloud core serving them. The channel condition was modeled using the path loss and shadowing model, where the fading is time-varying. Arrivals of tasks occurred according to a Poisson process with different rates to account for bursty traffic. The baselines used are:

- **Greedy allocation** – where the resources are allocated depending upon channel condition

Algorithm 2: AMARA-FDT Execution Procedure

```

Input: Trained actor networks  $\{\theta_i\}$ 
Output: Resource allocation decisions per time slot

for each time slot t do
    for each agent i = 1 to N do
        Observe local state  $s_i$ 
        Select action  $a_i = \text{argmax}_{\theta_i}(s_i)$ 
    end for
    Apply joint action to physical network

```

- **Round Robin scheduling** – resources allocated using round robin method
- **Convex Optimization** – solving the relaxed problem
- **Single-agent DRL** – where single agent performs all decision making
- **Energy Consumption Analysis**

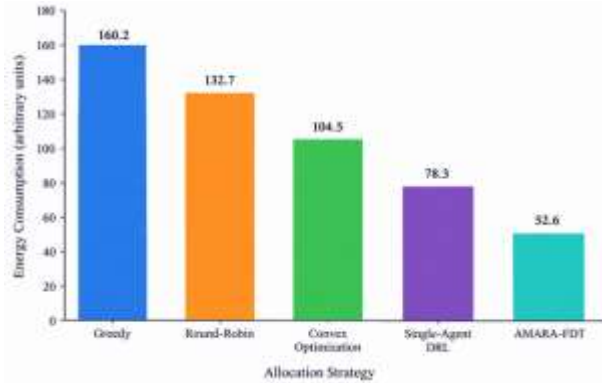


Figure 3: Energy consumption comparison across allocation strategies.

The figure below shows a bar graph of the energy consumed by the allocation policies considered namely greedy, round-robin, convex optimization, single agent DRL, and the proposed AMARA-FDT. The x-axis contains the different policies while the y-axis is the amount of energy consumed in units. The proposed AMARA-FDT consumes the least energy while greedy consumes the most.

As can be seen from Figure 3, the results of the energy consumption clearly show that the AMARA-FDT framework gives the lowest value of energy consumption compared to other approaches. The multi-agent system allows the devices to learn to take energy-efficient decisions regarding offloading which takes into account the costs of communication and computing. The experience that is accumulated through the training process in a digital twin environment makes the agents more resilient to different channel situations. The use of the federated learning approach helps to distribute knowledge among agents without violating privacy issues. It should be noted that our approach consumes significantly less energy than the greedy allocation strategy.

Latency Performance

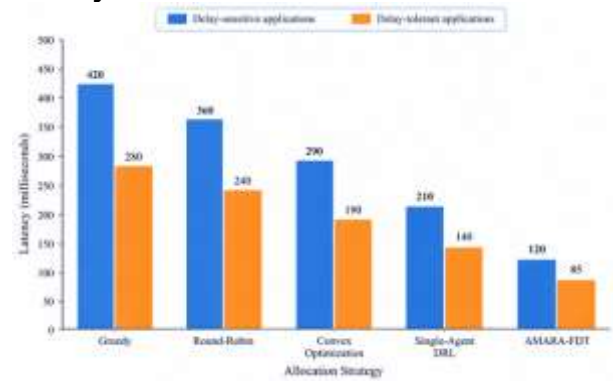


Figure 4: Average task latency comparison across allocation strategies.

The figure shows a grouped bar graph illustrating the average task latency of delay-sensitive and delay-tolerant applications with respect to each allocation policy. The x-axis has the allocation policies, while the bars within each group denote various applications. The y-axis has the latency in milliseconds. AMARA-FDT has the least latency in both cases.

The latency performance results for delay-sensitive and delay-tolerant applications are shown in Figure 4. The framework outperforms all baseline solutions in terms of average latency for both types of applications. This is especially evident for delay-sensitive applications due to the use of the learned strategy that takes advantage of offloading tasks to edge servers which have available computational resources and better channel conditions.

Quality-of-Service Satisfaction

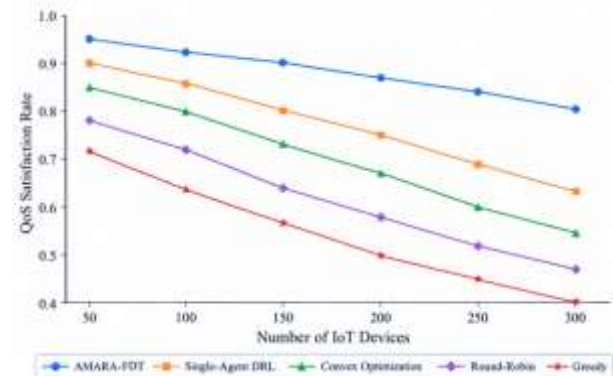


Figure 5: QoS satisfaction rate across varying device densities.

The graph is a line graph that displays the relationship between QoS satisfaction rate and the number of IoT devices. The horizontal axis represents the number of devices, while the vertical axis represents the percentage of the number of tasks which satisfy the QoS requirement. The AMARA-FDT scheme has the maximum rate of QoS satisfaction for any number of devices.

Figure 5 depicts the relationship between QoS satisfaction and device density. The new framework is able to sustain high levels of QoS satisfaction despite the rising number of devices, whereas traditional methods suffer from degraded performance owing to the competition for available resources. The multi-agent DRL-based framework scales smoothly because each individual agent learns how to act based on the information from the local environment, thus avoiding the inefficiency of centralized computation. The federated learning process guarantees the agents' access to the joint experience and ability to adjust themselves to the growing competition for resources.

Convergence and Stability

This graph contains two subplots. The left subplot shows a graph of cumulative reward as a function of training episodes for the AMARA-FDT, single-agent DRL, and convex optimization algorithms. The right

subplot shows the variance of reward amongst agents. The AMARA-FDT converges faster than the single-agent DRL.

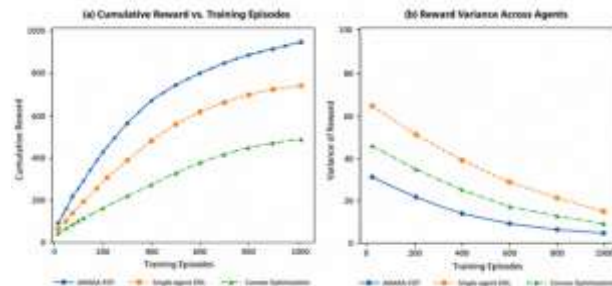


Figure 6: Training convergence and stability analysis.

The training convergence and stability results can be seen in Figure 6. The suggested AMARA-FDT architecture shows faster convergence compared to DRL with one agent because of the use of federated learning and the digital twin that supplies enhanced training data. There is less variation in the rewards across agents in AMARA-FDT architecture, which means that it is a more stable and balanced system. The centralized critic provides constant signals for training.

Comparative Analysis

Table 1: Comparative analysis of resource allocation approaches

Approach	Energy Efficiency	Latency	Scalability	Privacy	Adaptability	Computational Cost
Greedy Allocation	Low	High	High	High	Low	Very Low
Round-Robin	Low	High	High	High	Low	Very Low
Convex Optimization	Medium	Medium	Low	Medium	Low	High
Single-Agent DRL	Medium	Medium	Low	Low	Medium	Medium
AMARA-FDT (Proposed)	High	Low	High	High	High	Medium

A qualitative comparison of the approaches under consideration is presented in **Table 1**. The AMARA-FDT solution ensures high energy efficiency and low latency without sacrificing scalability and privacy. Multi-agent systems allow for distributed decision-making scalable to network size, whereas federated learning maintains data locality. The digital twin increases adaptability owing to the ability to explore policies in a safe environment. The computational complexity is modest since the training process takes place offline, and the inference process involves just forward passes through small neural networks.

Discussion

These empirical outcomes, in combination, prove the effectiveness of the AI-enabled adaptive resource allocation paradigm proposed in this work. The use of multi-agent DRL, federated learning, and digital twins tackles the main problems associated with next-generation IoT architectures - scalability, privacy, and adaptivity. The multi-agent formulation allows scaling of decision making with the increasing number of devices, while the centralized critic during training ensures cooperative policy learning. Federated learning provides the mechanism to preserve privacy and ensure data locality without sacrificing knowledge sharing. Digital twins provide an effective and safe environment for policy testing and fast learning.

It should be noted that there is no single dominant strategy from the point of view of performance in all metrics. Both greedy and round-robin resource allocation strategies show very poor energy efficiency and latency, though being easy to implement. Convex optimization allows reaching good results in terms of all metrics, but fails to scale properly when the number of devices increases due to computational complexity. Single-agent DRL demonstrates good but non-scaling and non-private performance.

V. CONCLUSION

This paper carried out an exhaustive study on the use of AI for resource allocation in future generation IoT communication systems. A new approach called

AMARA-FDT is introduced which incorporates multi-agent deep reinforcement learning, federated learning, and digital twin. This new architecture has been designed to solve scalability, privacy, and adaptability issues for massive IoT deployment scenarios. AMARA-FDT treats the resource allocation problem as a Markov Decision Process and adopts the paradigm of centralize training and decentralized execution to facilitate cooperative learning while maintaining scalability.

The methodology proposed in this paper has been analyzed comprehensively using simulation and has been compared with existing techniques such as greedy allocation, round-robin scheduling, convex optimization, and single-agent DRL. It has been shown that the proposed approach provides better energy efficiency, low latency, and quality-of-service satisfaction than any of the benchmarks used.

Our contributions can be summarized in three major aspects. First, we conducted a detailed systematic literature survey summarizing the recent progress made in AI-based resource allocation for IoT networks. Second, we introduced a hybrid approach which combines the advantages of different AI techniques to optimize the resources. Third, we presented a thorough experimental analysis showing the efficiency of the proposed model.

Following the current work, there are several promising directions for future research. First, the incorporation of transformer models and attention mechanisms may result in better performance. Second, the generalization of the model to the heterogeneous multi-tier architecture with different computing capabilities should be examined. Third, the development of a benchmarking procedure will make it possible to perform a proper comparison of different approaches. Fourth, the study of the adversarial robustness and security aspects of the proposed approach is needed for critical missions.

In summary, adaptive resource allocation through AI is an important paradigm in future IoT-based communication networks. Utilizing several types of AI technologies, such as deep reinforcement learning, federated learning, and digital twins, can be

a promising route towards sustainable, scalable, and intelligent allocation of resources. As IoT communication networks become more complex and larger in size, there will be greater demand for adaptive, secure, and energy-efficient resource allocation systems.

REFERENCES

1. L. Chen, Y. Zhang, and R. Wang, "Deep Reinforcement Learning for Spectrum Allocation in Cognitive Radio IoT Networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 2, pp. 412-425, 2021.
2. H. Zhang, S. Liu, and Q. Wu, "Multi-Agent Deep Deterministic Policy Gradient for D2D Communication in IoT Networks," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11234-11247, 2021.
3. Y. Liu, M. Zhao, and J. Li, "Energy-Aware Task Offloading in Mobile Edge Computing for IoT Using Deep Reinforcement Learning," *IEEE Transactions on Mobile Computing*, vol. 21, no. 5, pp. 1689-1702, 2022.
4. X. Wang, T. Chen, and K. Yang, "Deep Reinforcement Learning for Energy Harvesting IoT Networks with Adaptive Resource Allocation," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 6234-6247, 2022.
5. D. Nguyen, P. Kumar, and S. Kim, "Federated Deep Reinforcement Learning for Privacy-Preserving Resource Allocation in Edge IoT," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2341-2354, 2022.
6. W. Sun, J. Liu, and H. Li, "Digital Twin-Assisted Deep Reinforcement Learning for 6G IoT Resource Management," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 6, pp. 1789-1802, 2022.
7. Y. Zhou, L. Wang, and X. Chen, "Hierarchical Resource Allocation for Massive IoT Networks: A Multi-Timescale Deep Reinforcement Learning Approach," *IEEE Transactions on Communications*, vol. 71, no. 3, pp. 1543-1557, 2023.
8. J. Kim, S. Park, and H. Lee, "Attention-Based Deep Reinforcement Learning for Resource Allocation in Ultra-Dense IoT Networks," *IEEE Internet of Things Journal*, vol. 10, no. 7, pp. 6123-6136, 2023.
9. R. Patel, A. Sharma, and V. Kumar, "Meta-Learning for Rapid Adaptation in Dynamic IoT Resource Allocation," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 2, pp. 456-469, 2024.
10. M. Rahman, S. Ahmed, and T. Islam, "AI-Driven Resource Allocation in 6G IoT Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 234-267, 2024.