10.48047/jocaaa.2025.34.07.24

COMPUTATIONAL ANALYSIS OF DEEP REINFORCEMENT LEARNING-BASED TASK SCHEDULING FOR LARGE-SCALE INTERNET OF THINGS NETWORKS

[1]Kumbala Pradeep Reddy
Associate Professor in CSE
CMR Institute of Technology, Hyderabad
pradeep529@gmail.com

[2]B. Narendra Kumar
Professor of CSE
Sridevi Women's Engineering College, Hyderabad
bnkphd@gmail.com

[3]S Jagadeesh
Professor ECE
Sridevi Women's Engineering College, Hyderabad
jaaga.ssjec@gmail.com

[4]Palabatla Anish
Technical Consultant, Oracle
anishpalabatla@gmail.com

Abstract - The rapid expansion of the Internet of Things (IoT) has resulted in the deployment of billions of interconnected devices that continuously generate large volumes of data and computational tasks. Efficient task scheduling has become a critical requirement in large-scale IoT networks because scheduling decisions directly influence system latency, resource utilization, energy consumption, and quality of service. Traditional scheduling approaches such as First-Come-First-Serve (FCFS), Round Robin, and heuristic-based methods often struggle to adapt to dynamic network conditions and heterogeneous computing environments. The increasing complexity of IoT infrastructures necessitates intelligent scheduling mechanisms capable of learning optimal decision policies from continuously changing system states. Deep Reinforcement Learning (DRL) has emerged as a promising solution by combining reinforcement learning with deep neural networks to enable adaptive and data-driven resource management. This paper presents a Computational Analysis of Deep Reinforcement Learning-Based Task Scheduling for Large-Scale Internet of Things Networks. The proposed framework employs a DRL agent that observes network conditions, evaluates resource availability, and dynamically schedules tasks to optimize execution performance. The scheduling policy is learned through continuous interaction with the environment using state, action, and reward mechanisms. Experimental evaluation is conducted under varying network loads and task arrival rates to assess the effectiveness of the proposed approach. Performance is compared with FCFS, Round Robin, Genetic Algorithm-based scheduling, and Particle Swarm Optimization-based scheduling methods. Results demonstrate that the proposed DRL framework significantly reduces scheduling delay, improves task completion rates, and enhances overall resource utilization. The findings indicate that deep reinforcement learning provides an effective and scalable solution for intelligent task scheduling in next-generation large-scale IoT networks.

Keywords— *Internet of Things, Deep Reinforcement Learning, Task Scheduling, Edge Computing, Resource Management, Network Optimization, Intelligent Scheduling.*

I. INTRODUCTION

The Internet of Things (IoT) has transformed modern computing by enabling billions of interconnected devices to sense, collect, process, and exchange data across distributed environments. Recent advances in wireless communications, embedded systems, cloud computing, and edge intelligence have accelerated the deployment of IoT applications in domains such as smart cities, healthcare, industrial automation, transportation, and environmental monitoring. As the scale of IoT infrastructures continues to expand, the efficient management of computational tasks generated by these devices has become a critical challenge [3], [4]. Large-scale IoT networks are characterized by heterogeneous devices with diverse computational capabilities, communication requirements, and energy constraints. These devices continuously generate workloads that must be processed within strict latency requirements. For example, industrial control systems, autonomous vehicles, and healthcare monitoring applications

10.48047/jocaaa.2025.34.07.24

require timely task execution to ensure reliable operation and service quality. Consequently, task scheduling plays a fundamental role in determining system performance, resource utilization, throughput, and response time [3], [5].

Traditional scheduling approaches such as First-Come-First-Serve (FCFS), Round Robin, and priority-based scheduling have been extensively adopted in distributed computing systems. Although these methods are computationally simple and easy to implement, they typically rely on static decision rules and are unable to adapt effectively to rapidly changing network conditions. In large-scale IoT environments, dynamic workload arrivals, varying resource availability, and fluctuating communication delays reduce the effectiveness of such conventional scheduling strategies [4], [6]. The emergence of edge computing has introduced new opportunities for improving task execution efficiency in IoT systems. By placing computational resources closer to end devices, edge computing reduces communication latency and alleviates the burden on centralized cloud infrastructures [3], [5]. However, determining where and when tasks should be executed remains a complex optimization problem. Effective scheduling mechanisms must simultaneously consider resource availability, network load, queue conditions, task priorities, and quality-of-service requirements.

Artificial Intelligence (AI) techniques have increasingly been applied to address complex resource management problems in distributed environments. Among these approaches, Reinforcement Learning (RL) has demonstrated considerable promise because it enables agents to learn optimal decision-making policies through interactions with the environment [1]. Unlike traditional optimization techniques, RL does not require explicit mathematical models and can adapt to dynamic operating conditions. This capability makes RL particularly suitable for IoT scheduling applications where system states change continuously. Deep Reinforcement Learning (DRL) extends conventional reinforcement learning by integrating deep neural networks for high-dimensional state representation and policy learning [2]. The combination of deep learning and reinforcement learning enables intelligent agents to learn sophisticated scheduling strategies from complex environmental information. DRL has achieved significant success in areas such as robotics, autonomous systems, network optimization, and resource management [2].

Motivated by these developments, this paper presents a computational analysis of a Deep Reinforcement Learning-based task scheduling framework for large-scale IoT networks. The proposed framework employs a Deep Q-Network (DQN) agent that continuously observes network conditions and dynamically allocates computational tasks to available edge and cloud resources. By learning from reward-driven interactions with the environment, the scheduling agent aims to minimize scheduling delay, maximize task completion rates, and improve overall resource utilization. Experimental evaluations are conducted under varying workload conditions and compared with FCFS, Round Robin, Genetic Algorithm (GA), and Particle Swarm Optimization (PSO)-based scheduling approaches. The results demonstrate the effectiveness of DRL in enabling intelligent and adaptive task scheduling for next-generation IoT environments.

II. RELATED WORK

Task scheduling has been extensively studied in distributed computing, cloud computing, edge computing, and Internet of Things (IoT) environments because scheduling decisions directly influence system responsiveness, resource utilization, throughput, and energy efficiency. As IoT deployments continue to grow in scale and complexity, researchers have explored numerous scheduling techniques ranging from traditional heuristic methods to advanced machine learning-based approaches. Conventional scheduling algorithms such as First-Come-First-Serve (FCFS), Round Robin, and priority-based scheduling have long been employed in distributed systems due to their simplicity and low implementation overhead. These methods perform adequately under stable operating conditions but often exhibit poor adaptability when workload characteristics change dynamically. In large-scale IoT environments, variations in task arrival rates and resource availability can result in increased delays and inefficient utilization of computational resources [4]. Consequently, more adaptive scheduling techniques have been investigated. The rapid development of edge computing has significantly influenced research in IoT task scheduling. Shi et al. [3] introduced the concept of edge computing and highlighted its ability to reduce latency by moving computational resources closer to end devices. Satyanarayanan [5] further emphasized the importance of edge computing in supporting latency-sensitive applications. Mao et al. [4] surveyed mobile edge computing architectures and identified task scheduling and computation offloading as critical research challenges affecting system performance and service quality.

10.48047/jocaaa.2025.34.07.24

Metaheuristic optimization techniques have been widely applied to scheduling problems because of their capability to explore large solution spaces. Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) have demonstrated effectiveness in identifying near-optimal scheduling decisions under complex conditions. These methods improve task allocation efficiency by iteratively searching for better solutions. However, their computational complexity often increases with network scale, making real-time deployment challenging in highly dynamic IoT environments. Reinforcement Learning (RL) has emerged as a promising alternative for addressing dynamic scheduling problems. Sutton and Barto [1] established the theoretical foundation of reinforcement learning, demonstrating how agents can learn optimal policies through environmental interactions and reward feedback. Unlike traditional optimization techniques, RL continuously adapts its behavior according to observed system conditions, making it highly suitable for dynamic IoT environments. The introduction of Deep Reinforcement Learning (DRL) further expanded the applicability of reinforcement learning to complex decision-making problems. Mnih et al. [2] demonstrated the effectiveness of Deep Q-Networks (DQN) in learning optimal control policies from high-dimensional inputs. This breakthrough inspired extensive research into DRL-based resource management and scheduling solutions.

Several studies have applied DRL to IoT and edge computing environments. Li et al. [8] proposed a DRL-based computation offloading framework for mobile edge computing and reported significant improvements in latency reduction and resource utilization. Zhou et al. [9] developed a DRL-based scheduling strategy for delay-sensitive IoT applications and demonstrated enhanced scheduling efficiency under varying network conditions. Similarly, Sheng et al. [10] employed DRL for task scheduling in edge-enabled IoT systems and showed that learning-based scheduling outperformed conventional heuristic approaches. Recent studies have also investigated energy-aware and multi-objective scheduling strategies. Sellami et al. [11] proposed a DRL-based scheduling framework for software-defined IoT networks that simultaneously considered energy efficiency and quality-of-service requirements. Shresthamali et al. [12] explored reinforcement learning for multi-objective resource scheduling and demonstrated its effectiveness in balancing latency, resource utilization, and energy consumption. Despite these advances, existing studies often focus on specific scheduling objectives or limited deployment scenarios. Many approaches prioritize delay minimization while overlooking resource utilization and throughput considerations. Furthermore, the computational performance of DRL scheduling under large-scale IoT workloads requires further investigation. Therefore, this work develops a comprehensive DRL-based scheduling framework and evaluates its effectiveness across multiple performance metrics, including scheduling delay, task success rate, resource utilization, and throughput in large-scale IoT environments.

III. PROPOSED METHODOLOGY AND EXPERIMENTAL SETUP

This study proposes a Deep Reinforcement Learning (DRL)-based task scheduling framework for large-scale Internet of Things (IoT) networks. The framework is designed to address the limitations of conventional scheduling techniques by enabling adaptive and intelligent task allocation decisions based on real-time network conditions. In large-scale IoT environments, computational tasks are generated continuously by heterogeneous devices with diverse processing requirements and latency constraints. The proposed framework employs a learning-driven scheduling mechanism that dynamically assigns tasks to available edge and cloud resources with the objective of reducing scheduling delay, maximizing task completion success, and improving resource utilization. The overall architecture of the proposed framework is illustrated in Fig. 1, which presents the interaction among the task generation layer, state observation module, DRL agent, scheduling engine, and execution resources.

10.48047/jocaaa.2025.34.07.24

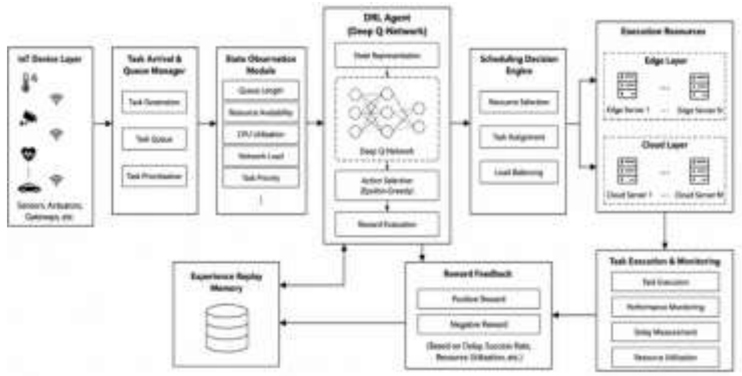

Figure 1. Deep Reinforcement Learning-Based Task Scheduling Framework for Large-Scale IoT Networks.

The framework consists of five major components: task generation, state observation, DRL agent, scheduling decision engine, and task execution layer. These components work together to enable adaptive scheduling decisions in dynamic IoT environments.

A. System Architecture and State Representation

The considered IoT environment consists of heterogeneous devices, edge servers, and cloud computing resources. IoT devices continuously generate computational tasks with different priorities, processing requirements, and execution deadlines. These tasks are collected within a task queue and forwarded to the scheduling module for execution. To support intelligent scheduling decisions, the framework employs a state observation mechanism that continuously monitors network and resource conditions. The observed parameters include queue length, resource utilization, available bandwidth, network load, and task priority. These parameters collectively describe the current state of the system and provide the DRL agent with real-time information about resource availability and workload distribution. By maintaining an updated representation of the operating environment, the framework can adapt scheduling decisions according to changing network conditions.

B. Deep Reinforcement Learning-Based Scheduling Strategy

The central component of the framework is a Deep Reinforcement Learning agent based on a Deep Q-Network (DQN). The agent receives the current system state and determines the most suitable scheduling action. Available actions include assigning tasks to edge servers, offloading tasks to cloud resources, delaying execution, or migrating tasks between resources when necessary. The learning process is driven by a reward mechanism that evaluates the effectiveness of each scheduling decision. Positive rewards are assigned when scheduling delay is reduced, resource utilization is improved, and tasks are completed successfully. Negative rewards are generated when resource congestion, excessive delay, or task failures occur. Through repeated interactions with the environment, the DRL agent gradually learns scheduling policies that maximize long-term performance. To improve learning stability, experience replay memory is employed to store previous interactions. The stored experiences are periodically used to update network parameters and refine scheduling policies. The complete DRL-based scheduling procedure is summarized in Algorithm 1.

Algorithm 1: DRL-Based Task Scheduling

Input:
    Task Queue T
    Resource Pool R
    Current Network State S

Output:

10.48047/jocaaa.2025.34.07.24

Optimized Task Scheduling Policy

```
1: Initialize Deep Q-Network (DQN) parameters
2: Initialize replay memory M
3: Observe initial network state S
4: for each training episode do
5:     while tasks remain in queue do
6:         Observe current state St
7:         Select action At using ε-greedy policy
8:         Execute scheduling action:
             • Assign task to edge server
             • Assign task to cloud server
             • Delay execution
             • Migrate task
9:         Receive reward Rt
10:        Observe next state St+1
11:        Store (St, At, Rt, St+1) in memory M
12:        Sample mini-batch from M
13:        Update DQN weights using Bellman equation
14:        Set St ← St+1
15:    end while
16: end for
17: Return optimized scheduling policy
```

C. Experimental Setup

The proposed framework is evaluated using a simulated large-scale IoT environment comprising 1,000 IoT devices, 20 edge servers, and 5 cloud servers. Dynamic workloads with varying arrival rates and computational demands are generated to emulate realistic operating conditions. The DRL model is trained over multiple episodes using the hyperparameter settings summarized in Table 1.

TABLE 1. DRL HYPERPARAMETERS USED IN THE PROPOSED FRAMEWORK.

Parameter	Value
Learning Algorithm	Deep Q-Network (DQN)
Learning Rate (α)	0.001
Discount Factor (γ)	0.95
Replay Memory Size	10,000
Batch Size	64
Initial Exploration Rate (ε)	1.0
Minimum Exploration Rate	0.01
Exploration Decay Rate	0.995
Hidden Layers	2
Neurons per Hidden Layer	128
Activation Function	ReLU
Optimizer	Adam
Training Episodes	500
Maximum Queue Length	100 Tasks

10.48047/jocaaa.2025.34.07.24

Performance is compared with First-Come-First-Serve (FCFS), Round Robin, Genetic Algorithm (GA), and Particle Swarm Optimization (PSO)-based scheduling methods. Evaluation focuses on average scheduling delay, task success rate, resource utilization, and throughput. These metrics provide a comprehensive assessment of the effectiveness of the proposed DRL-based scheduling framework under dynamic IoT workloads.

IV. RESULTS AND DISCUSSION

This section presents the performance evaluation of the proposed Deep Reinforcement Learning (DRL)-Based Task Scheduling Framework for large-scale Internet of Things (IoT) networks. The objective of the experimental analysis is to investigate the effectiveness of the DRL scheduling agent in reducing scheduling delay, improving task success rates, and enhancing resource utilization under dynamic network conditions. The proposed framework is compared with First-Come-First-Serve (FCFS), Round Robin, Genetic Algorithm (GA)-based scheduling, and Particle Swarm Optimization (PSO)-based scheduling approaches. The experiments were conducted using varying network loads and task arrival rates to simulate realistic IoT operating conditions. The evaluation focused on scheduling delay, task execution success, and overall system efficiency.

A. Scheduling Delay Analysis

Scheduling delay is a critical performance metric in IoT systems because many applications require real-time or near-real-time task execution. Delays in scheduling decisions can negatively affect service quality and application responsiveness. The average scheduling delay achieved by different scheduling algorithms is presented in Fig. 2**.**

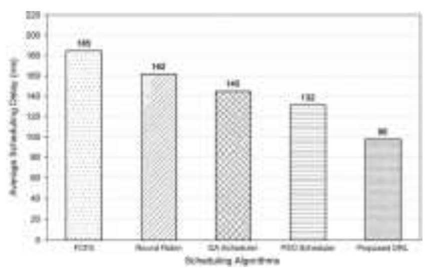

Figure 2. Average Scheduling Delay Comparison of Task Scheduling Algorithms.

The results indicate that traditional FCFS scheduling exhibits the highest delay, reaching approximately 185 ms. Since FCFS processes tasks strictly according to arrival order, it cannot adapt to variations in task priority or resource availability. Round Robin scheduling improves performance slightly by distributing tasks more uniformly, resulting in an average delay of 162ms. Metaheuristic scheduling approaches demonstrate better performance. The GA-based scheduler achieves a delay of approximately 145 ms, while the PSO-based scheduler further reduces delay to 132ms. These improvements result from their ability to optimize task allocation according to system conditions. The proposed DRL framework achieves the lowest scheduling delay of approximately 98 ms. This reduction is attributed to the adaptive decision-making capability of the DRL agent, which continuously learns efficient scheduling policies from environmental feedback. By considering queue status, resource utilization, and network load simultaneously, the framework effectively minimizes waiting time and processing delays. The overall performance comparison of the evaluated scheduling algorithms is summarized in Table 2.

TABLE 2. PERFORMANCE COMPARISON OF SCHEDULING ALGORITHMS

Method	Scheduling Delay (ms)	Task Success Rate (%)	Resource Utilization (%)	Throughput (Tasks/s)
FCFS	185	69	71	220
Round Robin	162	74	76	245
GA Scheduler	145	78	82	271
PSO Scheduler	132	82	86	289
Proposed DRL	98	87	93	324

10.48047/jocaaa.2025.34.07.24

B. Task Success Rate Analysis

Task success rate represents the percentage of tasks successfully completed within their allocated deadlines. High task success rates indicate efficient resource management and improved scheduling decisions. The variation of task success rate under increasing network load conditions is illustrated in Fig. 3**.** As network load increases, all scheduling approaches experience a gradual decline in task success rate due to increasing resource contention and queue congestion. FCFS exhibits the most significant performance degradation, decreasing from approximately 82% task success at low load levels to 69% under maximum load conditions. Round Robin scheduling demonstrates improved stability but still suffers from performance reduction at higher loads. GA and PSO schedulers provide better task completion performance because their optimization mechanisms improve resource allocation efficiency. The proposed DRL framework consistently maintains the highest task success rate across all workload conditions. Even at maximum network load, the framework achieves a success rate of approximately 87%, outperforming all competing approaches. This behavior demonstrates the ability of the DRL agent to adapt scheduling decisions according to changing network conditions and resource availability.

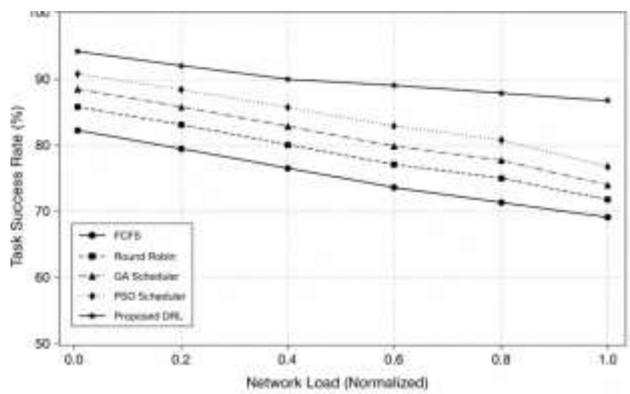

Fig 3. Task Success Rate Under Increasing Network Load Conditions.

C. Resource Utilization Analysis

Efficient utilization of computational resources is essential for maintaining high system productivity and minimizing infrastructure costs. Resource underutilization leads to wasted computational capacity, whereas excessive utilization may cause bottlenecks and performance degradation. The results presented in Table 2 indicate that the proposed DRL scheduler achieves approximately 93% resource utilization, significantly higher than FCFS (71%), Round Robin (76%), GA (82%), and PSO (86%).The improved utilization can be attributed to the intelligent decision-making capability of the DRL agent. By continuously observing resource availability and workload characteristics, the framework allocates tasks more effectively across available computing resources. Consequently, idle resources are minimized while system throughput is increased. Higher resource utilization also contributes to improved task completion performance and reduced scheduling delay.

D. Throughput Analysis

Throughput represents the number of tasks successfully processed per second and serves as an indicator of overall system efficiency. According to Table 2, the proposed DRL scheduler achieves the highest throughput of approximately 324 tasks per second. In comparison, FCFS processes only 220 tasks per second, while Round Robin achieves 245 tasks per second.GA and PSO schedulers improve throughput to 271 and 289 tasks per second, respectively. However, the proposed DRL framework demonstrates superior performance by continuously adapting scheduling decisions according to environmental conditions. The higher throughput achieved by the proposed

10.48047/jocaaa.2025.34.07.24

framework indicates that the scheduling policy effectively balances workloads across available resources while avoiding excessive queue buildup.

E. Discussion

The experimental results demonstrate that Deep Reinforcement Learning provides significant advantages for task scheduling in large-scale IoT networks. Unlike conventional scheduling approaches that rely on fixed rules, the DRL framework continuously learns and refines scheduling policies through interaction with the environment. A key observation is the substantial reduction in scheduling delay achieved by the proposed framework. Lower delay directly improves application responsiveness and service quality, which are essential requirements for many IoT applications. Furthermore, the framework maintains high task success rates even under heavy workload conditions, indicating strong adaptability and robustness. The results also highlight the relationship between resource utilization and overall scheduling performance. Efficient resource allocation contributes to higher throughput, reduced queue congestion, and improved task completion rates. By learning optimal scheduling policies, the DRL agent effectively balances competing objectives and maximizes overall system efficiency. Overall, the proposed DRL-based scheduling framework outperforms FCFS, Round Robin, GA, and PSO scheduling approaches across all evaluated performance metrics. These findings demonstrate the potential of deep reinforcement learning as an effective solution for intelligent task scheduling in next-generation large-scale IoT environments.

V. CONCLUSION AND FUTURE WORK

This paper presented a Computational Analysis of Deep Reinforcement Learning-Based Task Scheduling for Large-Scale Internet of Things Networks. The rapid growth of IoT infrastructures has increased the need for intelligent scheduling mechanisms capable of handling dynamic workloads, heterogeneous resources, and varying network conditions. Traditional scheduling approaches often struggle to maintain efficient performance under such complex environments because they rely on static decision-making strategies. To address these limitations, a Deep Reinforcement Learning (DRL)-based scheduling framework was proposed to learn adaptive scheduling policies through continuous interaction with the network environment. The proposed framework employed a Deep Q-Network-based learning model that utilized system state information, including queue length, resource availability, network load, and task priority, to make scheduling decisions. A reward-driven learning mechanism enabled the scheduling agent to optimize task allocation dynamically while minimizing scheduling delay and improving task completion performance. The framework was evaluated under varying workload conditions and compared with conventional scheduling approaches, including FCFS, Round Robin, Genetic Algorithm-based scheduling, and Particle Swarm Optimization-based scheduling. Experimental results demonstrated that the proposed DRL framework consistently outperformed the compared methods across multiple performance metrics. The framework achieved the lowest average scheduling delay of 98 ms, the highest task success rate of 87%, resource utilization of 93%, and throughput of 324 tasks per second. These improvements were achieved through adaptive learning and intelligent resource management, allowing the scheduling agent to respond effectively to changing network conditions. The results confirmed that Deep Reinforcement Learning can significantly enhance scheduling efficiency in large-scale IoT environments. Future research may focus on extending the framework to multi-agent reinforcement learning environments where multiple scheduling agents cooperate across distributed IoT infrastructures. The integration of energy-aware scheduling objectives, security constraints, and federated learning mechanisms may further improve scheduling performance and scalability. Additionally, validation using real-world IoT deployments and large-scale industrial datasets would provide further insights into the practical applicability of the proposed approach in next-generation intelligent IoT networks.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.
[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Human-Level Control Through Deep Reinforcement Learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A Survey on Mobile Edge Computing: The Communication Perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
[5] Mahadev Satyanarayanan, “The Emergence of Edge Computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

10.48047/jocaaa.2025.34.07.24

[6] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and Rajkumar Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in IoT, Edge and Fog Computing Environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.
[7] S. Wang, J. Xu, N. Zhang, and Y. Liu, "A Survey on Service Migration in Mobile Edge Computing," *IEEE Access*, vol. 6, pp. 23511–23528, 2018.
[8] J. Li, H. Gao, T. Lv, and Y. Lu, "Deep Reinforcement Learning Based Computation Offloading and Resource Allocation for Mobile Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 1–14, 2019.
[9] C. Zhou, W. Wu, H. He, P. Yang, F. Lyu, N. Cheng, and X. Shen, "Deep Reinforcement Learning for Delay-Oriented IoT Task Scheduling in Space-Air-Ground Integrated Networks," 2020.
[10] S. Sheng, P. Chen, Z. Chen, L. Wu, and Y. Yao, "Deep Reinforcement Learning-Based Task Scheduling in IoT Edge Computing," *Sensors*, vol. 21, no. 5, pp. 1–20, 2021.
[11] B. Sellami, A. Hakiri, S. Ben Yahia, and P. Berthou, "Energy-Aware Task Scheduling and Offloading Using Deep Reinforcement Learning in SDN-Enabled IoT Networks," *Computer Networks*, vol. 210, 2022.
[12] S. Shresthamali, M. Kondo, and H. Nakamura, "Multi-Objective Resource Scheduling for IoT Systems Using Reinforcement Learning," *Journal of Low Power Electronics and Applications*, vol. 12, no. 4, pp. 1–18, 2022.