Journal of Advanced College of Engineering and Management

A Hybrid PPO and DDPG Algorithm for Resource Aware Task Offloading in Edge-Cloud Computing Paradigm

[1]Rojina Baral, [2]Anisha Pokhrel, [3]Subarna Shakya

[1]*MSc. in Computer System and Knowledge Engineering Pulchowk Campus, Institute of Engineering, Tribhuvan University Lalitpur, Nepal*

[2]*Department of Electronics and Computer Engineering, Advanced College of Engineering and Management, Kalanki, Kathmandu, Nepal*

[3]*Professor Pulchowk Campus, Institute of Engineering, Tribhuvan University Lalitpur, Nepal*

Email: rojinabaral28@gmail.com, anishapokhrel01@gmail.com, drss@ioe.edu.np

DOI: 10.3126/jacem.v11i1.84519

Abstract

Rapid expansion of the Internet of Things (IoT) and mobile applications has increased the demand for low latency and efficient computational resource management. Although offering high processing power, cloud computing suffers from network congestion and latency issues, making edge computing a viable alternative. Task offloading in cloud-edge environments is crucial for optimizing resource allocation, reducing delays, and enhancing system performance. For intelligent job offloading in an Edge-Cloud architecture, this study suggests a hybrid reinforcement learning model based on Deep Deterministic Policy Gradient (DDPG) and Proximal Policy Optimization (PPO). The model is trained on the datasets simulated in EdgeCloudSim simulator and consisting of tasks, network conditions, computational capabilities of servers, and energy efficiency constraints. The results of the experiment shows that the suggested model considerably reduces task execution time, lowers energy usage, and boosts system efficiency. This work provides a robust deep reinforcement learning-based solution for optimizing task offloading in future edge-cloud computing infrastructures.

Keywords*—CloudSim, DDPG, Edge-Cloud, IOT, PPO*

1. INTRODUCTION

Intelligent industrial systems, also known as Industrial IoT (IIoT), have emerged due to the rapid growth of the Internet of Things (IoT). These systems incorporate smart devices, sensors, cameras, and 5G technology to facilitate automatic data collection and analysis, enhance production efficiency, and address scalability issues. However, IoT devices have limited amounts of memory, processing power, and battery life [1]. Traditional cloud computing platforms have proven inadequate for the emerging Internet of Things (IoT) era due to the exponential growth of data generated by multiple users and the increasing demand for applications that are sensitive to computation and latency [2]. Decentralized computing architectures that can process workloads closer to their source are therefore becoming increasingly in demand. To bridge this gap, edge, fog, and cloud computing paradigms have been developed, offering improved responsiveness, reduced latency, and enhanced bandwidth utilization for real-time industrial applications. The edge and cloud computing have become one of the main factors in modern distributed computing infrastructures due to the capability provided by these infrastructures for lowlatency and highly scalable resource-intensive applications. Mobile edge computing, one of the emerging computing paradigm, it augments the computational capabilities of mobile devices by offloading the tasks that are computationally intensive to the edge clouds nearby with potential computation capability from resource-constrained smart mobile devices[3].

Edge computing has drawn a lot of interest as a supplement to cloud computing in order to overcome these constraints. Edge computing improves responsiveness for real-time applications, lowers latency, and relieves bandwidth demand on the cloud by moving computation closer to the data source. In industrial settings where prompt decision-making is essential, such as in predictive maintenance, quality control, and safety monitoring, this paradigm is very advantageous. Furthermore, distributed intelligence is supported by edge computing, which enables IIoT devices to process and act on data locally without entirely depending on centralized cloud infrastructure. So, task offloading and cloud resource allocation are the two key processes that enable the achievement of the operational objectives of such systems relating to expenditure reduction while keeping up with performance benchmarks. Also, efficient decision-making for task offloading as well as effective resource allocation is crucial for enhancing system performance [4].

2. LITERATURE REVIEW

In paper [5], the multi-objective task offloading problem within MEC for a scenario with numerous users and multiple base stations was examined, considering both the resource allocation problem and the impact of transmission power during offloading. In the work [6], researchers proposed a two-stage vehicular edge orchestrator based on machine learning that considers not only task completion success but also service duration. Using network base stations and queue based task scheduling, the suggested system [7] offered a task offloading decision. The base station, also known as the edge orchestrator, had chosen whether to compute locally or offload to edge servers or the cloud. In the study [8], the authors have utilized the Rainbow Deep Q-Network (DQN), a so phisticated Deep Reinforcement Learning (DRL) algorithm, to propose a novel method for task offloading in Device-to-Device (D2D)-Edge-Cloud computing. In the paper [9], the researcher presented a deep reinforcement learning (DRL) framework with an entropy and attention mechanism (DMOEA) for MEC compute offloading facilitated by D2D. In order to optimize the binary task offloading decision, the autonomous choice of where to carry out each computer operation, which depends on numerous factors, researchers presented a distributed Deep Reinforcement Learning (DRL) tool in the work [10]. In the paper [11], a meta-learning-based generalized FRL strategy is proposed that combines RL models elucidated by several smart devices into a generic model. To determine the CO and RA strategy in a Digital Twin (DT)-enabled UAV-assisted MEC system, the study divided a complex network into small-scale units using a normalized characteristic matrix and offered a normalized network model for complex network situations based on the FRL metacritic method. With the use of non-orthogonal multiple access (NOMA), and reconfigurable intelligent surfaces (RIS), the study [12] presented a unique VEC architecture in which vehicles executed tasks by outsourcing partial or binary computations to edge nodes (eNs). To minimize latency, the work [13] formulated and investigated the topic of combined task offloading, resource allocation, and access point selection in heterogeneous edge environments. According to simulation results, the suggested method in the research cited in [14] can converge to local optimal solutions and save 20% to 40% of energy compared to the reference schemes.

3. METHODOLOGY

This section outlines the step-by-step process for developing A Hybrid PPO and DDPG Algorithm for Resource Aware Task Offloading in Edge-Cloud Computing Paradigm. The goal is to develop the hybrid model using DDPG and PPO algorithm for resource-aware task offloading and minimizing the energy consumption, latency and hence cost of edgecloud computing.

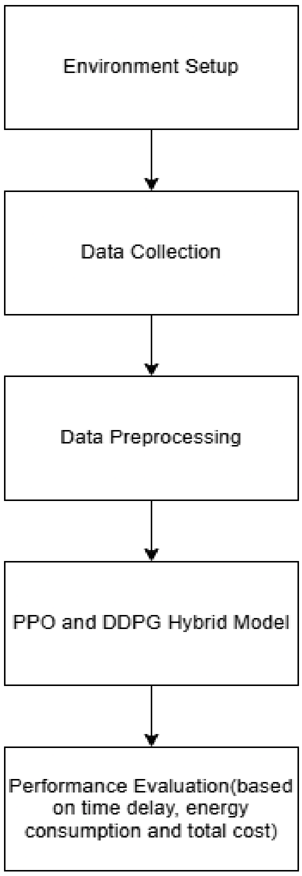

Figure 3.1: Methodology

3.1 Environment Setup

The EdgeCloudSim simulator has been used, which was developed based on the CloudSim simulator, adding the module for edge servers. The core simulation module, networking module, load generator module, and mobility module are used from the EdgeCloudSim Environment. I have created the environment, consisting of end devices, edge servers, and cloud servers.

3.2 Data Collection

Before any preprocessing, a total of 67,810 tasks were generated after the simulation environment was set up. Along with related characteristics, including task type, maximum allowable delay, priority level, data size, and computation size (in millions of instructions, or MI), each task was given a unique task ID. These activities were divided into three primary application domains: Infotainment, Danger Assessment, and Traffic Management. The activities were given priority levels of 1, 2, and 3, respectively. The system (cloud, edge, device) states information, storage in megabytes, RAM in megabytes, average utilization and Millions of Instructions Per Second(MIPS) were also simulated. For network characteristics, WAN bandwidth, WLAN bandwidth, WAN propagation delay, and LAN internal delay were collected.

3.3 Data Preprocessing

To ensure data consistency and model relevance, the preprocessing stage was crucial. The simulated data was originally in JSON format and included more information than just task characteristics, network details, cloud servers, edge servers, and mobile servers, such as mobility, virtual machines (VMs), and many other details. During data preprocessing, only the required parameters were extracted, and extraneous information was removed. Relevant elements, including task ID, task type, priority, computation size, data size, and maximum delay, as well as the associated system and network characteristics, were extracted from the JSON structure through parsing.

Initially, the collection contained around 67,000 tasks, many of which were duplicates. To ensure that the final dataset represented a varied and non-redundant task distribution, deduplication was performed based on identical values across key variables. After removing repeated tasks, 27,843 unique tasks remained. Among 27,843 datasets, 80% of the data were used for learning purposes and 20% of the data were used for evaluation.

3.4 PPO and DDPG Hybrid Model

Due to the inherent trade-offs among performance indicators such as latency, energy consumption, and resource utilization, task offloading in this context is a multi-objective problem. The hybrid approach seeks to optimize the three main objectives.

$$E = \sum_{i=1}^{N} e(x_i) \tag{1a}$$

$$T = \sum_{i=1}^{N} t(x_i) \tag{1b}$$

$$C = \sum_{i=1}^{N} c(x_i) \tag{1c}$$

subject to

$$x_i \in \{0, 1, 2\}, \forall i = 1, \ldots, N. \tag{1d}$$

$$0 \le p_{i,j} \le p_i^{\max}, \forall i = 1, \ldots, N, \forall j = 1, \ldots, M. \tag{1e}$$

$$p_{i,j} = 0, \forall x_i = 0. \tag{1f}$$

$$T_i \le t_i, \forall i = 1, \ldots, N. \tag{1g}$$

$$\sum_{i=1}^{N} \lambda_i^j \le 1, \forall j = 1, \ldots, M. \tag{1h}$$

The first objective function 1a refers to the minimization of energy consumption during the computation and transmission of tasks. Transmission time is considered in the case of task offloading to the edge or cloud. The second objective function 1b refers to minimizing computation and transmission time for tasks. Transmission time is considered in the case of task offloading to the edge or cloud. The third objective function 1b refers to the minimization of total costs required to complete tasks. Currently, it is the sum of energy consumed and time taken.

In the given objective functions, there are lists of constraints. The constraint equation 1d states that the offloading decision must be either 0, 1, or 2, corresponding to executing locally, offloading to the edge, or offloading to the cloud, respectively. The constraint equation 1e and equation 1f indicate the upper and lower limits of transmission power during task offloading. This transmission power is not considered when executing the task locally. The constraint equation 1g represents the time delay that must not exceed the deadline. And the constraint equation 1h indicates that the allocation of computation resources to the tasks must be less than or equal to the available resources, where available resources is considered as one.

The following essential elements make up the deep reinforcement learning algorithm, in addition to its mathematical formulation:

State Space:

It represents the current state of the system that affects the task-offloading decision. The features that are taken as the state space in this research are:

- Task features like computation size, maximum delay, priority, and data size.
- Server features like storage, RAM, MIPS, and utilization available on mobile, edge, and cloud platforms.
- Network conditions including WLAN transmission latency, LAN transmission latency, WLAN bandwidth, and WAN bandwidth

Action Space:

It includes the task offloading decision, i.e., to execute the task on local devices or to offload it to the edge or cloud.

$x_i \in \{0, 1, 2\}, \forall i = 1, \ldots, N.$

where:

$x_i = 0$: Task executed locally.

$x_i = 1$: Task offloaded to edge server.

$x_i = 2$: Task offloaded to cloud server.

The Hybrid DDPG-PPO model combines the Deep Deterministic Policy Gradient (DDPG) and Proximal Policy Optimization (PPO) algorithms in reinforcement learning. The model actively experiments with different actions throughout the exploratory phase of training to comprehend the task-environment dynamics. In this case, DDPG's weight is set higher (80%) since its deterministic action selection makes navigating through the continuous action space easier. Although PPO's policy-driven, clipped-surrogate strategy helps to preserve learning stability, its weight is purposefully reduced (20%) to avoid being overly conservative, which could impede the development of stronger policies.

Learning becomes less important than performance once training is over and the agent enters evaluation. Now, the objective is to carry out activities as efficiently as possible by applying the learned policies. PPO, which is renowned for using value estimates and clipped updates to create stable and generalized policies, is now granted more decision-making authority (40%) than before. To ensure that the acquired continuous control abilities are still used, DDPG continues to contribute, albeit with a little lower weight (60%) than before. This change preserves DDPG's dynamic reactivity while ensuring that the assessment phase benefits from PPO's reliable policy updates.

3.5 Testing and Performance Evaluation

Based on the selected node(local device, edge, cloud) the task will be executed. Then the latency L_i and energy consumption E_j will be measured along with the deadline factor, resource utilization, and priority of the task. Based on the reward functions of PPO and

DDPG designed to minimize energy consumption and latency, while also ensuring a minimum deadline miss for model reliability. Additionally, efficient resource utilization is also a priority for us, so this thesis focuses on maximizing resource utilization as well. The performance of this study is evaluated based on time delay, energy consumption, and total cost.

4. RESULTS AND DISCUSSION

With the inclusion of PPO and DDPG, the efficiency of offloading has increased by making near-optimal decisions on the processing of a task locally or offloading it to the cloud or edge server, depending on the conditions of the underlying networks, bandwidth, and other resources.

```
Task TRAFFIC_MANAGEMENT:
  Offloaded to: Local Server
  Total Time: 2.81 s (Deadline: 3.0 s) --> Deadline Met
  Energy Required: 3.09

Task DANGER_ASSESSMENT:
  Offloaded to: Edge Server
  Total Time: 1.00 s (Deadline: 6.0 s) --> Deadline Met
  Energy Required: 0.79

Task TRAFFIC_MANAGEMENT:
  Offloaded to: Edge Server
  Total Time: 0.32 s (Deadline: 3.0 s) --> Deadline Met
  Energy Required: 0.24

Task DANGER_ASSESSMENT:
  Offloaded to: Local Server
  Total Time: 10.52 s (Deadline: 6.0 s) --> Deadline Missed
  Energy Required: 11.57

Task INFOTAINMENT:
  Offloaded to: Edge Server
  Total Time: 2.05 s (Deadline: 9.0 s) --> Deadline Met
  Energy Required: 1.62
```

Figure 4.1: Output for Resource Aware Task Offloading

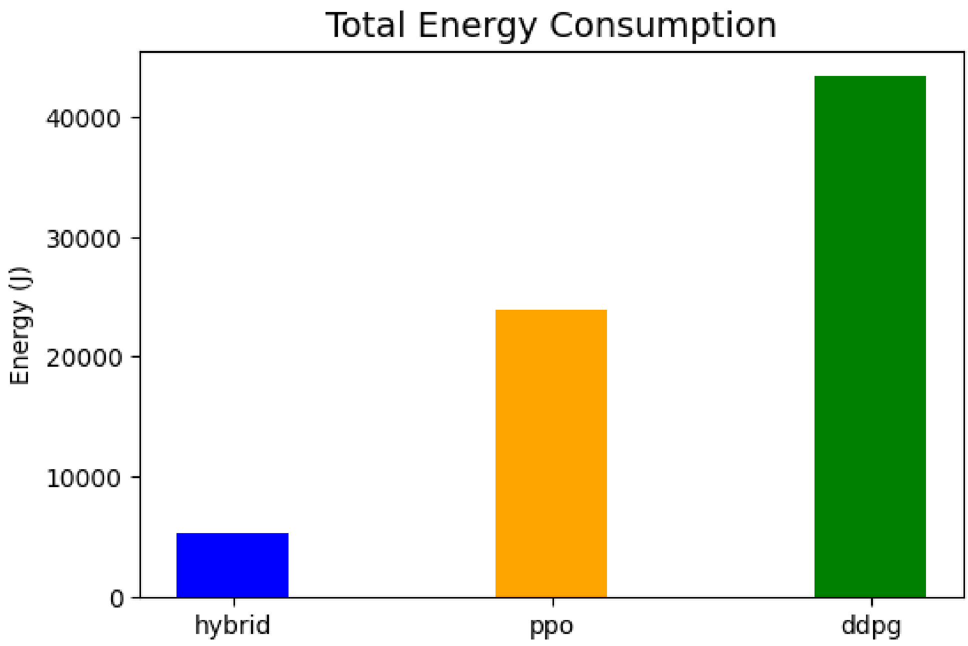

Figure 4.2: Energy Consumption Graph for PPO, DDPG and Hybrid Model

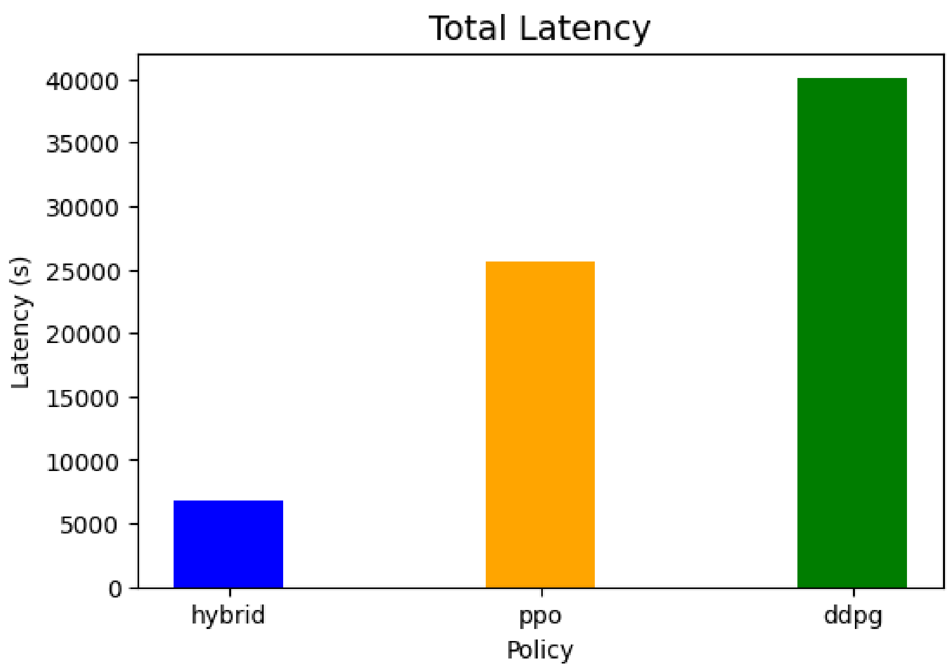

Figure 4.3: Latency Graph for PPO, DDPG and Hybrid Model

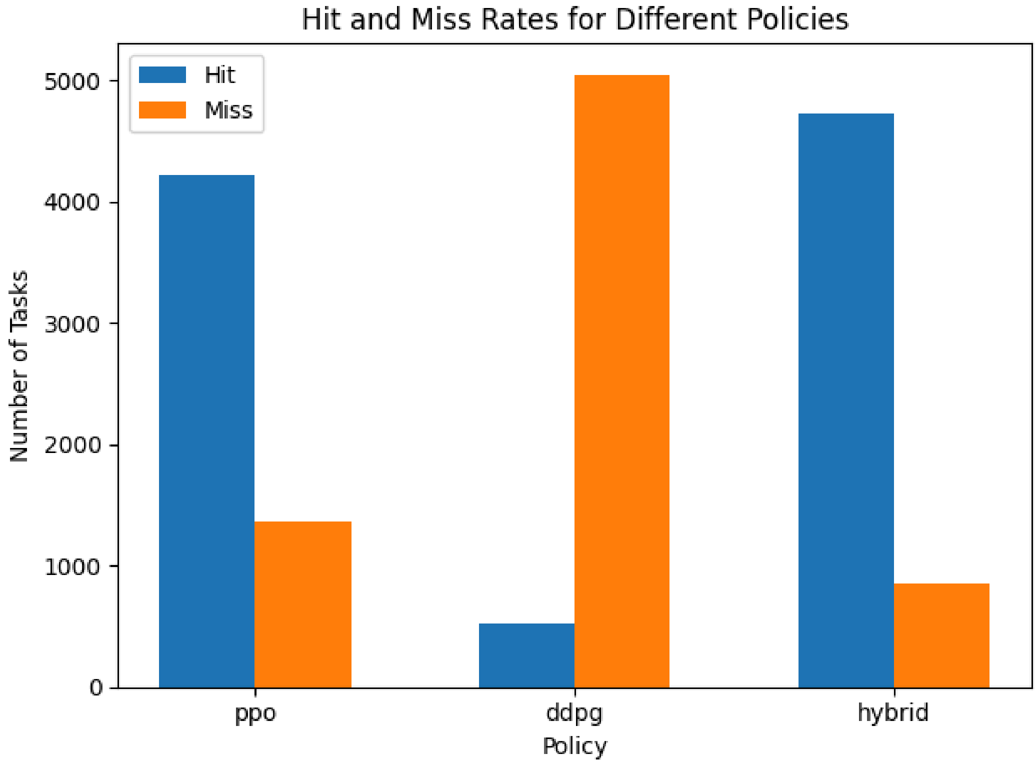

Figure 4.4 Deadline Hit/Miss graph for DDPG, PPO, and Hybrid Model

5. CONCLUSIONS

The experimental results showed that, while DDPG and PPO learned optimal task distribution rules over time, they incurred large energy and latency costs. PPO performed moderately in terms of training stability and resource efficiency, achieving a better balance than DDPG but falling short of optimal performance in limited situations. Upon closer examination of energy usage and latency data, it was found that the hybrid model performed more efficiently than both DDPG and PPO. In particular, DDPG used the most resources out of the three models, while the hybrid model had the lowest average energy consumption and task delay, followed by PPO. This outcome further supports the hybrid model’s capacity to offer the best possible balance between resource management and system performance. Additionally, the task deadline hit and miss rates were used to assess the dependability of job completion. With an 84.75% hit rate, the hybrid model showed the best reliability, indicating that most jobs were finished on the due date. With a moderate success percentage of 75.61%, PPO outperformed DDPG, which missed more deadlines than it hit. These findings demonstrate that the hybrid approach greatly improves task execution consistency and reliability in dynamic contexts in addition to optimizing energy and latency.

6. SUGGESTIONS AND RECOMMENDATIONS

Although the suggested framework has yielded encouraging outcomes, several areas remain for improvement and expansion. These essentials might be addressed in future studies, which would also investigate other settings and parameters like:

- Including Parameters Like Mobility and Waiting Delay
- Integrating Fog Computing Environments
- Using Real World Datasets
- Incorporating Security and Privacy

7. ACKNOWLEDGEMENTS

I would like to thank Prof. Dr. Subarna Shakya for supervision and my family for their support. The Department of Electronics and Computer Engineering , Pulchowk Campus, Institute of Engineering provided the necessary infrastructure and academic environment.

REFERENCES

[1] Huong Mai Do, Tuan Phong Tran, and Myungsik Yoo. Deep reinforcement learningbased task offloading and resource allocation for industrial iot in mec federation system. IEEe Access, 2023.

[2] Dongdong Yang, Bin Li, and Dusit Niyato. Energy-aware task offloading for rotatable star-ris-enhanced mobile edge computing systems. IEEE Internet of Things Journal, 2025.

[3] Hongyan Yu, Quyuan Wang, and Songtao Guo. Energy-efficient task offloading and resource scheduling for mobile edge computing. In 2018 IEEE international conference on networking, architecture and storage (NAS), pages 1–4. IEEE, 2018.

[4] Yintao Li, Chuanguo Tang, Bohao Hou, Feibo Jiang, and Yuhang Fu. Transformer and meta-reinforcement learning-based task offloading for mec systems in dynamic environments. IEEE Access, (99):1–1, 2025.

[5] YANG LIU JIAN CHENG CHUNYANG YU, YIBO YONG and QIANG TONG. A multi-objective evolutionary approach: Task offloading and resource allocation using enhanced decomposition-based algorithm in mobile edge computing. IEEE Access, 2024.

[6] Cagatay Sonmez, Can Tunca, Atay Ozgovde, and Cem Ersoy. Machine learning-based workload orchestrator for vehicular edge computing. IEEE Transactions on Intelligent Transportation Systems, 22(4):2239–2251, 2020.

[7] Habtamu Mohammed Birhanie and Mohammed Oumer Adem. Optimized task offloading strategy in iot edge computing network. Journal of King Saud University-Computer and Information Sciences, 36(2):101942, 2024.

[8] Komeil Moghaddasi, Shakiba Rajabi, Farhad Soleimanian Gharehchopogh, and Ali Ghaffari. An advanced deep reinforcement learning algorithm for three-layer d2d-edgecloud computing architecture for efficient task offloading in the internet of things. Sustainable Computing: Informatics and Systems, 43:100992, 2024.

[9] Cong Wang, Xiaojuan Chai, Sancheng Peng, Ying Yuan, and Guorui Li. Deep reinforcement learning with entropy and attention mechanism for d2d-assisted task offloading in edge computing. IEEE Transactions on Services Computing, 2024.

[10] Gorka Nieto, Idoia de la Iglesia, Unai Lopez-Novoa, and Cristina Perfecto. Deep reinforcement learning techniques for dynamic task offloading in the 5g edge-cloud continuum. Journal of Cloud Computing, 13(1):94, 2024.

[11] Prakhar Consul, Ishan Budhiraja, Deepak Garg, Neeraj Kumar, Ramendra Singh, and Ahmad S Almogren. A hybrid task offloading and resource allocation approach for digital twin-empowered uav-assisted mec network using federated reinforcement learning for future wireless network. IEEE Transactions on Consumer Electronics, 2024.

[12] Abdul-Baaki Yakubu, Ahmed H. Abd El-Malek, Mohammed Abo-Zahhad, Osamu Muta, and Maha M. Elsabrouty. Task offloading and resource allocation in an risassisted noma-based vehicular edge computing. IEEE Access, 12:124330–124348, 2024.

[13] Yu Liu, Yingling Mao, Zhenhua Liu, Fan Ye, and Yuanyuan Yang. Joint task offloading and resource allocation in heterogeneous edge environments. IEEE Transactions on Mobile Computing, 2023.

[14] Mingxiong Zhao, Jun-Jie Yu, Wen-Tao Li, Di Liu, Shaowen Yao, Wei Feng, Changyang She, and Tony QS Quek. Energy-aware task offloading and resource allocation for timesensitive services in mobile edge computing systems. IEEE Transactions on Vehicular Technology, 70(10):10925–10940, 2021.