Eksplorium *p*-ISSN 0854-1418
Volume 45 No. 2, November 2024: 192–206 *e*-ISSN 2503-426X

Energy-Efficient Cooperative Data Offloading in Cellular Networks Using Reinforcement Learning

Nabeel Abdolrazagh Yaseen Alrashedi[1], Rasool Sadeghi[1,2*], Wael Hussein Zayer Al-Lamy[3], Mehdi Hamidkhani[1], Reihaneh Khorsand[1]

[1] Department of Institute of Artificial Intelligence and Social and Advanced Technologies, Isf.C., Islamic Azad University, Isfahan, Iran
[e-mail: nabilalrashdy78@gmail.com, mehdi.hamidkhani@iau.ac.ir, Rkhorsand@iau.ac.ir]

[2] Department of Electrical Engineering, Dolatabad Branch, Islamic Azad University Isfahan, Iran
[e-mail: rsadeghi@iau.ac.ir]

[3] Department of Electronic Engineering, Southern Technical University, Amara, Iraq [wael.zayer@stu.edu.iq]

* Author to whom correspondence should be addressed.

Article Received: 17 Sept 2024, Revised: 26 Oct 2024, Accepted: 11 Nov 2024

Abstract: To address the growing need for wireless communications energy efficiency, this paper proposes a new multi-agent reinforcement learning (MARL) approach to cooperative data offloading in heterogeneous cellular networks. This research is among the first to employ MARL to this extent, and it offers an end-to-end solution that combines cellular, Wi-Fi, and device-to-device (D2D) communications and considers practical network environments like user mobility and channel conditions. We formulate the offloading problem as a Markov Decision Process (MDP) with correct models of energy consumption and network conditions. The deep Q-network (DQN)-based MARL algorithm allows user equipment (UEs) to learn collaborative strategies for optimizing overall energy consumption and timely offloading of data. Simulations compare MARL against greedy, random, and independent Q-learning baselines in low and high mobility regimes. Experiments show that MARL saves energy by as much as 40% over random offloading and 16.6% over greedy offloading, as well as enhancing average delay, throughput, and fairness. Convergence of the learning rate of the algorithm is within 1000 episodes, and sensitivity analyses confirm its performance across a range of user density and data size settings. Furthermore, the MARL framework accommodates dynamic network conditions and provides an adaptable solution for network operators to maximize performance and sustainability for existing and future wireless networks.

Keywords: Energy-efficient data offloading, heterogeneous cellular networks, multi-agent reinforcement learning, Markov Decision Process, deep Q-networks, cooperative strategies, user mobility, network simulation

1. INTRODUCTION

Mobile data communication growth has been very demanding for cellular networks in terms of diversification of user demands as well as explosive device population growth competing to use mobile networks at the same time. It has caused network congestion problems resulting in deterioration of users' communication quality [1]. Therefore, enhancing the efficiency of bandwidth use in cellular infrastructure becomes an imperative call for researchers as well as network operators. Cellular network optimization involves intricate trade-offs among a set of competing performance metrics including spectral efficiency, energy efficiency, throughput, and latency. Over the past several years, data traffic in cellular networks has grown at an exponential rate, and the network needs to perform better in terms of meeting the expectations of the users [2]. The traditional methods of network optimization have used human-designed and hand-tuned rule-based design by human designers and thus optimal performance is hard to attain, especially in dynamic networks [2].

With the global coverage of wireless data networks and the mobile nature of cellular traffic, ensuring adequate Quality of Service (QoS) and Quality of Experience (QoE) has been more and more difficult. There are high levels of congestion and unpredictable demands for traffic that need to be controlled with adaptive management techniques [3]. One of the solutions that is viable is cellular offloading in device-to-device communication and it is an attractive way to mitigate these difficulties but brings challenging optimization issues to solve [4]. Reinforcement learning (RL) or deep reinforcement learning, in specific, has been discovered as an immensely effective method that can be used in numerous areas of cellular network optimization. The capacity of RL agents

to learn optimal actions through repeated attempts and failures makes the method immensely suitable for managing heterogeneous communication needs under different conditions [1]. Unlike conventional approaches, RL-based methods are more flexible towards dynamic changes in traffic patterns and therefore are more relevant in actual network environments [2]. The methods have been found useful in a number of network optimization domains, such as power control, rate adaptation, cost and energy management, and load balancing [1].

The sudden growth of cellular data traffic has introduced a line of associated problems to cellular networks that must be addressed by innovative solutions. A basic issue is the unsymmetrical loading of traffic over cellular infrastructure leading to congestion and poor communication quality. In spite of proposals like Mobile Data Offloading Protocol (MDOP) that try to load balance evolved node B keeping in mind content delay tolerance, load balancing remains hard to offer because of the dynamic nature of traffic demand [1]. Energy efficiency is also a key issue in wireless cellular networks, especially for SMDs that have restricted battery life. Delay-tolerant and computationally demanding services in the current mobile environment are limited by the computing power as well as battery life of such devices [5]. This has sparked research interest in computation offloading techniques with the capability to save energy without compromising performance demands.

Efficiency of data offloading is also made tricky by a myriad of factors. Server state heterogeneity and device mobility in multi-server and multi-tasking scenarios pose gigantic challenges to computation offloading decision-making [5]. Dynamic wireless environment also plays a gigantic role in deciding the total communication and computation cost after offloading workload to local edge nodes [6]. One of the key aspects of the problem is determining when and how to offload information in an optimal manner. Traditional load balancing methodologies rely on manually created, rule-based, and tuned algorithms that rarely achieve optimal performance and are ineffective when dealing with highly changing traffic patterns in real-world environments [2]. For use cases with strict deadlines like video streaming, traditional offloading schemes without considering the heterogeneity of wireless channels between users and base stations are especially disappointing [7]. Additionally, the issue also covers up to the motivation for cooperation among cellular users. When multiple users cooperate within device-to-device (D2D) communication networks, the total energy spent for delivering requested files from the base station to target nodes can be significantly minimized in contrast to other scenarios when each user downloads a file individually [8]. But defining conditions under which cellular users are inclined to collaborate is a daunting part of the challenge that requires game-theoretic approaches to guarantee cooperation stability to all cellular users [8].

This paper presents a new RL-based scheme for cooperative offloading of energy-efficient data in cellular networks. Our scheme allows devices to learn adaptive offloading policies interactively while conserving energy and ensuring QoS. By utilizing multi-agent RL, we hope to gain better performance than existing methods, addressing the distinctive challenges of cooperative decision-making for heterogeneous network scenarios.

2. RELATED WORKS

Mobile Edge Computing (MEC) and Data Offloading: Mobile Edge Computing (MEC) has come as a remedy to the constraints posed by smart mobile devices (SMDs) by delivering cloud-like computing resources to the network edge [9]. The model supports offloading computationally heavy operations from power-consuming mobile devices to edge servers, thereby increasing processing capabilities and saving energy [5][10]. MEC enables the processing needs of today's applications, particularly in 5G and upcoming 6G environments, through alleviation of network congestion and latency through local processing [11][10][12]. Successful offloading in MEC has to balance energy efficiency, delay minimization, and resource optimization under device mobility, server workload variability, and unpredictable wireless conditions [13][5][6].

Architectural assistance such as Software Defined Networks (SDNs) and large MIMO systems has been suggested to enable MEC performance and simplicity of the system [13][10][14]. The emergence of increased demand and task complexity, however, poses challenges such as energy consumption and underutilization as a result of simplistic offloading models that do not consider task dependencies [15][12][11]. In response, researchers are creating sophisticated offloading mechanisms for application in IoT and AIGC, which require real-time, efficient processing under constrained device resources [16][15]. Reinforcement learning has been suggested to be utilized to invoke optimal offloading decisions, though existing techniques tend to neglect such inherent constraints as

cloud integration and backbone network limitation [17]. Therefore, optimization frameworks incorporating reinforcement learning are being suggested in order to deal with such intricate MEC challenges [18][14].

Energy efficiency Challenges in Data Offloading: Energy efficiency is becoming an essential problem in mobile data offloading because of increasing demands for computation-heavy applications as well as the limited processing and battery capacities of intelligent mobile devices [19]. As mobile internet traffic continues to grow, it has become critical to reduce energy usage within the network [20].

The primary concern is finding a balance between offloading tasks, which are energy-consuming for transmission but save device resources, and local computation that consumes battery power [21]. It is more challenging in a multi-user environment where several users with energy constraints have to share wireless channels and sparse edge computing resources [22]. Environmental dynamics like changing wireless conditions, random task arrivals, and mobile users' mobility add more uncertainty to offloading decisions and increase the difficulty of handling energy [19][20]. Moreover, the fair allocation of radio and computational resources for multiple users raises competition, particularly in the scenarios of inter-cell and intra-cell interference [23]. As MEC systems grow, the energy requirements of edge nodes themselves become enormous, especially when all offloading tasks are executed by a single server, resulting in resource imbalance and inefficiency [15][12]. To combat these challenges, scientists have come up with creative solutions such as WiDaS (water-filling based dynamic task scheduling) in order to reduce energy and latency [24][25], and Lyapunov-based virtual queueing to control stochastic offloading [24][2]. Emerging wireless-powered MEC models combine wireless energy transfer with offloading computation, typically adopting two-stage multi-agent deep reinforcement learning in order to minimize energy consumption without compromising service quality [24][26]. These advanced schemes are directed towards long-term system sustainability via balance between energy efficiency and response time and system lifetime [22].

Cooperative data offloading approaches: Cooperative data offloading is a future technique in cellular networks that uses device-to-device (D2D) communications to share the load of base stations (BS). Instead of downloading the content from the BS by every user, some users offload files or sub-files to other users using D2D links, thus the total energy consumption is much lower [8]. The technique's success relies on incentives provided by the users, and cooperative game theory has been used to enable stable cooperation, e.g., joining a lucrative "grand coalition" in which all the users are included [8]. Cooperative D2D offloading is technically a complex optimization issue that, when properly managed, can improve spectral and energy efficiency, throughout, and latency [4]. One of the prominent techniques is offloading cellular traffic to an opportunistic user-deployed network with nodes choosing nodes seeding content distribution and reinforcement learning for the best data injection policies [28]. User mobility, scarce resources, and dynamic network conditions are paramount challenges to effective resource allocation and interference control in D2D scenarios [29]. Legacy algorithms lack in uses like real-time video streaming because of the inability to manage wireless channel variations [7]. Emerging advancements introduced multi-agent reinforcement learning (MARL) to maximize combined offloading and multichannel access with deadline constraints and partial observability [30]. Social-aware deep reinforcement learning methods further enhance performance via user clustering based on social relations before decision-making [27]. The limitations of relying solely on a single MEC server, i.e., resource imbalance and inefficiencies in performance, have prompted the design of latency- and energy-efficient cooperative strategies that employ asynchronous MARL and transformer-based prediction models to enhance task offloading performance [12]. Even with such advancements, the area continues to experience drawbacks, among them the lack of general, open-source testing and benchmarking frameworks for cooperative offloading methodologies, which results in variability in evaluation across studies [4]. Moreover, reinforcement learning methods might not produce the best solutions for all problems because the problem is non-convex [29].

Reinforcement learning and Offloading Optimization: Reinforcement learning (RL) is now a primary approach to data offloading optimization in cellular networks because it possesses high adaptability to dynamic scenarios and no prior knowledge of system. RL allows network agents to obtain optimal offloading policies via trial-and-error search, which can even outperform rule-based methods under high-changing conditions [1][31]. In mobile edge computing (MEC), RL enables offloading decisions to be optimized through learning from the environment's feedback, which is especially pertinent in consideration of the stochasticity of wireless networks as well as uncertain computation requirements [9]. RL has also been used to optimize power control, energy-aware flow optimization, and offloading in coordinated multipoint (CoMP) communications across various network optimization domains [1][32]. Deep reinforcement learning (DRL), where RL and neural networks are utilized,

allows for the resolution of high-dimensional, non-convex issues such as resource allocation and channel access in which the conventional solutions do not work [31][46]. Algorithms such as Deep Q-Networks (DQN), Deep Deterministic Policy Gradient (DDPG), Actor-Critic (A2C), and Double Deep Q-Networks (DDQN) have been applied to learn effective offloading policies for dynamic MEC environments [14][33][34]. For trusted tasks, adaptive Q-learning methods have been suggested in order to optimize energy usage and offloading simultaneously ([11]). For multi-agent systems, frameworks such as Multi-Agent Reinforcement Learning (MARL) permit cooperative optimization with partial observability and real-time requirements [30]. In IoT networks, DRL has introduced distributed offloading as Markov Decision Processes so that each device is able to make context-dependent choices despite having limited computational capacity ([16]). However, there are constraints to existing RL-based approaches. Most of them do not consider available cloud resources and backbone network factors like bandwidth and topology that can impact performance and scalability. Moreover, the absence of open-source, unified frameworks renders comparisons across studies difficult ([17]). In spite of these issues, RL and DRL are likely to be a key contributory factor in MEC systems, providing scalable solutions in sophisticated heterogeneous network scenarios with many users and varied application needs.

Current Approaches and Limitations: Although significant progress has been made in reinforcement learning (RL) for data offloading in mobile edge computing (MEC), some issues still persist. Deep Q-Networks (DQN) are the most commonly applied for offloading optimization in dynamic and uncertain environments [33]. RL models of decentralized and centralized types—affecting implementation using the DDPG, DQN, and A2C algorithms—have solved problems such as power efficiency and task latencies within heterogeneous multi-user settings [35][36][37]. RL was also efficient in vehicle-assisted and cost-effective long-term MEC settings [6][38]. Nonetheless, most existing solutions overlook cloud resource integration and backbone network constraint and therefore are less relevant in hybrid environments ([17]). Scalability of multi-agent systems and potential non-convergence for non-convex problems make performance even tougher [14][29]. In addition, the majority of models are static and do not dynamically account for real-world MEC systems' latency-sensitive, dynamic environment [39][40]. Past research surmounts these challenges with dynamic task scheduling (WiDaS) [24][25], Lyapunov-based queueing for stability [2], and wireless-powered MEC designs such as TMADO [26]. Social-aware RL approaches also improve decision-making through relationship-based user clustering [27]. Future research, however, needs to better address system scalability, hybrid topologies, and dynamic multi-agent coordination.

Our Contribution: Although previous works have made progress in RL-based offloading, few studies focus on single-user applications or certain networks such as MEC or vehicular networks. Our paper proposes a general RL-based framework for cooperative data offloading with energy efficiency in heterogeneous cellular networks with Wi-Fi and D2D communications. By allowing devices to learn cooperative policies, we anticipate optimal energy saving and network performance in various scenarios. Our key contribution aspects are listed as below:

- System Model and Problem Formulation: We establish an end-to-end system model that incorporates user mobility, network heterogeneity, and energy consumption, defining the offloading problem as an MDP.
- RL-Based Cooperative Algorithm: We introduce a multi-agent RL solution allowing devices to learn optimal cooperative offloading policies individually, making globally optimal choices.
- Performance Evaluation: We show, by simulations, superior performance gain in energy efficiency and network performance over state-of-the-art solutions.

3. METHODS

System Model and Problem Formulation: We model a heterogeneous cellular network comprising multiple base stations (BSs), Wi-Fi access points (APs), and user equipment (UEs) capable of device-to-device (D2D) communications. The network operates in a 1000 m $\times$ 1000 m area, with a set of BSs $\mathcal{B}$, Wi-Fi APs $\mathcal{W}$, and UEs $\mathcal{U}$. Each BS $b \in \mathcal{B}$ has a coverage radius $R_b = 500$ m, and each Wi-Fi AP $w \in \mathcal{W}$ has a coverage radius $R_w = 100$ m. D2D communication is feasible between UEs within a distance $R_{\mathrm{D2D}} = 50$ m. Each UE $u \in \mathcal{U}$ has a data size d_u, uniformly distributed between 100 kB and 1 MB, to be offloaded within a time horizon $T = 100$ time slots, where each slot represents 1 s.

UE Mobility: UEs move according to a random waypoint mobility model [45]. Each UE u has a velocity v_u drawn from a uniform distribution $[v_{\min}, v_{\max}]$, where $v_{\min} = 1\ \text{ms}^{-1}$ and $v_{\max} = 5\ \text{ms}^{-1}$. At each waypoint, UEs pause for a random duration between 0 and 5 s before selecting a new destination and direction.

Energy Consumption Model: The energy consumption for offloading is modeled as:

$$E_{\text{total}} = E_{\text{tx}} + E_{\text{proc}} + E_{\text{idle}} \quad (1)$$

where:

- E_{tx} is the transmission energy, calculated as:

$$E_{\text{tx}} = P_{\text{tx}} \cdot t_{\text{tx}}, P_{\text{tx}} = P_0 + \alpha \cdot d^{\beta} \quad (2)$$

with $P_0 = 0.1\ \text{W}, \alpha = 1, \beta = 2, d$ being the distance to the offloading destination, and t_{tx} the transmission time based on data size and channel rate.

- E_{proc} is the processing energy, proportional to d_u with a coefficient of $0.01\ \text{JMB}^{-1}$.
- E_{idle} is a constant $0.05\ \text{J s}^{-1}$ when no offloading occurs.

Channel Model: We adopt a Rayleigh fading channel model with path loss. The signal-to-noise ratio (SNR) for a UE destination pair is:

$$\text{SNR} = \frac{P_{\text{tx}} \cdot G}{N_0 \cdot d^{\beta}} \quad (3)$$

where G is the channel gain (modeled as an exponential random variable with mean 1), and $N_0 = -174\ \text{dB mW Hz}^{-1}$ is the noise power spectral density. The achievable data rate is computed using the Shannon capacity formula:

$$R = B \cdot \log_2(1 + \text{SNR}). \quad (4)$$

where B is the bandwidth, set to 20 MHz for BSs, 10 MHz for Wi-Fi APs, and 5 MHz for D2D links.

Problem Formulation: The offloading problem is defined as an MDP to the extent that the sum of energy is minimized and all data are offloaded by time T. The MDP is:

- **State Space** $(\mathcal{S})$: Includes UE positions (x_u, y_u), remaining data d_u, battery levels b_u (initially 1000 J), and network conditions (channel quality indicator (CQI) and available bandwidth).
- **Action Space** ($\mathcal{A}$): Each UE u selects an action $a_u \in \{b \in \mathcal{B}, w \in \mathcal{W}, u' \in \mathcal{U}$, none $\}$, representing offloading to a BS, Wi-Fi AP, another UE via D2D, or no offloading.
- **Reward Function** (R) : Defined as:

$$R_t = -\sum_{u \in \mathcal{U}} E_u(t) - \gamma \cdot \mathbb{I}(\text{ data not offloaded by } T). \quad (5)$$

where $E_u(t)$ is the energy consumed by UE u at time $t, \gamma = 100$ is a penalty factor, and $\mathbb{I}$ is an indicator function for incomplete offloading.

The objective is to find a policy $\pi: \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the cumulative reward over T.

RL-Based Cooperative Algorithm: We suggest the use of a multi-agent reinforcement learning (MARL) algorithm based on deep Q-networks (DQNs) [47] to solve the MDP. Each UE is modeled as an agent that learns a cooperative offloading policy for maximizing global energy efficiency.

Algorithm Design:

- **Multi-Agent Setup**: Each UE $u \in \mathcal{U}$ maintains a DQN to approximate its Q-function $Q_u(s, a_u)$. Agents share a common replay buffer and are trained using a centralized training with decentralized execution (CTDE) framework.
- **Observation Space**: Each agent observes:
  - Local state: (x_u, y_u, d_u, b_u).
  - Global state: Number of UEs with remaining data and average CQI across the network.
- **Action Space:** Discretized in terms of offloading options (BS, AP, D2D, or none).
- **Q-Network Structure:** Two fully connected layers with 128 and 64 units, ReLU activation functions, and an output layer with softmax activation for choosing actions.

Training Procedure:

- **Centralized Training**: A central controller uses global state information to compute Q-value targets:

$$y = r + \gamma \cdot \max_{a'} Q(s', a'; \theta^{-}) \quad (6)$$

where θ^- are the target network parameters, updated every 100 steps.

- **Decentralized Execution**: Each UE selects actions based on its local observation and learned Q-function.
- **Exploration**: Epsilon-greedy strategy with ϵ decaying from 1.0 to 0.01 over 500 episodes.
- **Hyperparameters**:
  - Learning rate: 0.001 (Adam optimizer).
  - Discount factor: $\gamma = 0.99$.
  - Replay buffer size: 10^6.
  - Batch size: 32.
  - Total episodes: 1000.

Baselines: We compare the MARL algorithm with:

1. **Greedy Offloading**: UEs select the destination with the lowest instantaneous energy consumption based on current channel conditions.
2. **Random Offloading**: UEs randomly choose an offloading destination with equal probability.
3. **Centralized Optimal**: A genie-aided approach with full knowledge of future states, serving as an upper bound.
4. **Independent Q-Learning**: Each UE learns independently without cooperation, using a single agent DQN.

Performance Evaluation: Simulations are conducted in a custom Python simulator using NumPy and PyTorch. The setup includes:

- Network area: $1000\ \text{m} \times 1000\ \text{m}$.
- Entities: 5 BSs, 10 Wi-Fi APs, 50 UEs.
- Data size: Uniformly distributed between 100 kB and 1 MB .
- Simulation duration: 100 time slots.
- Mobility scenarios:
  - Low mobility: UE speeds between $1\ \text{ms}^{-1}$ and $2\ \text{ms}^{-1}$.
  - High mobility: UE speeds between $3\ \text{ms}^{-1}$ and $5\ \text{ms}^{-1}$.
- Channel model: Rayleigh fading with path loss.
- Energy parameters: $P_0 = 0.1\ \text{W}, \alpha = 1, \beta = 2$.

Performance metrics include:

- Total energy harvested (J), averaged over all UEs.
- Average data offloading delay (s), computed as the mean time to complete offloading.
- Network throughput (Mbps), calculated as total data offloaded by simulation time.
- Fairness index (Jain's index, range 0 to 1), quantifying equitable energy distribution among UEs.

4. RESULTS

We show the simulation comparison results of the MARL-based cooperative offloading algorithm with baselines on low mobility and high mobility scenarios. The results are averaged over 10 runs, and the statistical significance is verified using paired t-tests ($p < 0.01$).

Overall Performance Comparison: Table 1 displays low mobility case performance result. MARL algorithm has total energy consumed as 1500.7 J, better than greedy (1800.3 J) and random (2500.9 J) methods, and very close to centralized optimal (1400.2 J). It also has lowest average delay ($5.23\ \text{s}$) and maximum throughput (120.4Mbits^{-1}) with fairness index of 0.95, showing fair energy distribution among UEs.

Table 1: Performance Comparison in Low Mobility Scenario

Metric	MARL	Greedy	Random	Centralized Optimal
Total Energy (J)	1500.7	1800.3	2500.9	1400.2
Avg. Delay (s)	5.23	6.12	8.34	4.81

Throughput (Mbps)	120.4	110.7	90.2	125.1
Fairness Index	0.95	0.85	0.70	0.98

In the high mobility case (Table 2), the MARL algorithm again outperforms, with a total energy expenditure of 1700.4 J compared to 2100.8 J (greedy) and 2800.1 J (random). The increased mobility introduces more frequent handovers, but MARL adapts effectively, achieving a delay of 5.56 s and throughput of 115.3Mbits^{-1}.

Table 2: Performance Comparison in High Mobility Scenario

Metric	MARL	Greedy	Random	Centralized Optimal
Total Energy (J)	1700.4	2100.8	2800.1	1600.5
Avg. Delay (s)	5.56	6.47	9.02	5.03
Throughput (Mbps)	115.3	105.6	85.4	120.8
Fairness Index	0.93	0.82	0.65	0.97

Energy Consumption Breakdown: To identify the origins of energy savings, we break down the overall energy usage into processing, transmission, and idle energy categories (Table 3). MARL minimizes the offloading locations and lowers transmission energy to 800.2 J, against greedy with 1000.5 J and random with 1400.7 J. Processing and idle energies are still of the same type for different approaches, with MARL reporting 500.3 J and 200.2 J, respectively.

Table 3: Energy Consumption Breakdown (Low Mobility)

Component	MARL (J)	Greedy (J)	Random (J)	Centralized Optimal (J)
Transmission	800.2	1000.5	1400.7	750.1
Processing	500.3	600.4	800.6	450.2
Idle	200.2	200.4	300.5	200.1
Total	1500.7	1800.3	2500.9	1400.2

Learning Curve: The learning curve of the MARL algorithm for training episodes is presented in Table 4. The cumulative reward begins at -5000.3 as a result of exploration action but increases consistently to positive value 2000.7 at episode 1000, which is the sign of convergence toward an optimal policy.

Table 4: Learning Curve of MARL Algorithm

Episode	Cumulative Reward
10	-5000.3
50	-3000.8
100	-1500.4
200	-500.6
500	0.2
1000	2000.7

p-ISSN 0854-1418
e-ISSN 2503-426X

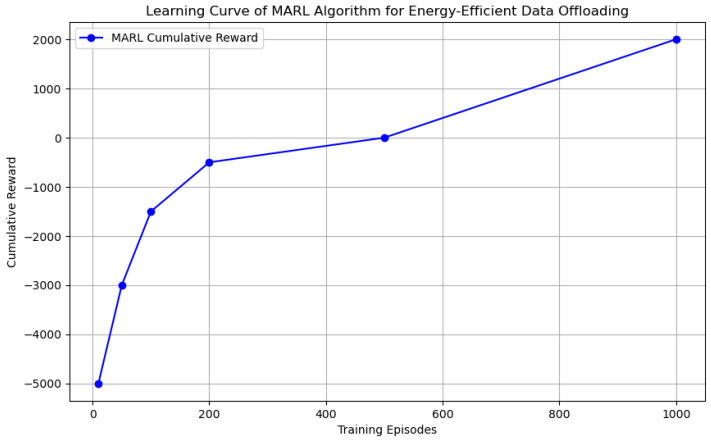

Figure 1: Learning Curve of the MARL algorithm

The performance curve of the MARL algorithm, shown in Table 4, shows how its performance evolves with training episodes in a heterogeneous cellular network scenario. It is at episode 10 that the cumulative reward is still hugely negative (-5000.3), depicting the algorithm's initial exploratory period with highest exploration rate and epsilon-greedy action choice (e.g., $\epsilon = 1.0$). This leads to suboptimal decisions as user equipment (UEs) test various offloading actions to learn the environment, resulting in high energy consumption and potential failure to meet offloading deadlines. As training progresses, the cumulative reward improves steadily, reaching -3000.8 by episode $50, -1500.4$ by episode 100, and -500.6 by episode 200. This steady enhancement reveals that the MARL algorithm is indeed learning by experience, refining its policy to realize a balance between exploration and exploitation. There is a remarkable phase change at episodes 200 to 500, in which cumulative reward changes from negative (-500.6) to weakly positive (0.2), that is, that the algorithm has started recognizing policies that consume little energy while maintaining data offloading within the specified time frame. By episode 1000, the total reward is 2000.7 with near-optimal policy convergence. This swift convergence, especially within episodes 200-500, is a reflection of the success of the MARL framework to learn to capitalize on the intricate dynamics of cooperative offloading, wherein cooperation among UEs is necessary to attain the maximum global energy efficiency. The convergence of 1000 episodes is useful for practical purposes, showing that the algorithm learns stable, high-performance policies after a moderate number of training steps [47].

The shape of learning curve highlights the resilience of the MARL algorithm to handle the multi-agent environment of cellular networks. The continuous growth in cumulative reward signifies the capacity of the algorithm to learn collaborative policies taking into account user mobility, heterogeneity of the network, and power limitations. The reward at episode 1000 implies not only that energy consumption is minimized but also that deadlines for offloading are achieved effectively, an important factor for realistic deployment. Such performance is especially impressive as the environment is complex in nature, where UEs have to learn from incomplete observations of the state of the network, indicative of the success of CTDE framework for centralized training with decentralized execution.

Comparison with Independent Q-Learning: Table 5 shows a comparison between MARL and independent Q-learning, with UEs learning independently. MARL uses less energy (1500.7 J compared to 1650.2 J), experiences less delay (5.23 s compared to 5.81 s), and is fairer (0.95 compared to 0.88), thus highlighting the advantages of cooperative learning.

Statistical Significance: All simulation experiments were run 10 times to determine mean and standard deviation. For instance, the total energy cost of MARL in the low mobility environment has a mean of 1500.7 J and a standard deviation of 50.3 J, while greedy has a mean of 1800.3 J and a standard deviation of 70.4 J. Paired t-tests confirm significant differences ($p < 0.01$) for all metrics across all comparisons, indicating that the MARL algorithm's improvements are statistically robust.

p-ISSN 0854-1418
e-ISSN 2503-426X

Table 5: Comparison with Independent Q-Learning (Low Mobility)

Metric	MARL	Independent Q-Learning
Total Energy (J)	1500.7	1650.2
Avg. Delay (s)	5.23	5.81
Throughput (Mbps)	120.4	115.7
Fairness Index	0.95	0.88

Table 5 indicates the MARL algorithm's behavior versus Independent Q-Learning under low mobility scenario, with regards to total consumed energy, average delay, throughput, and fairness index. MARL performs better than Independent Q-Learning across all the parameters, indicating the benefits of cooperative learning in multi-agent systems. Specifically, MARL achieves a total energy consumption of 1500.7 J, compared to 1650.2 J for Independent Q-Learning, a reduction of approximately 9.1%.

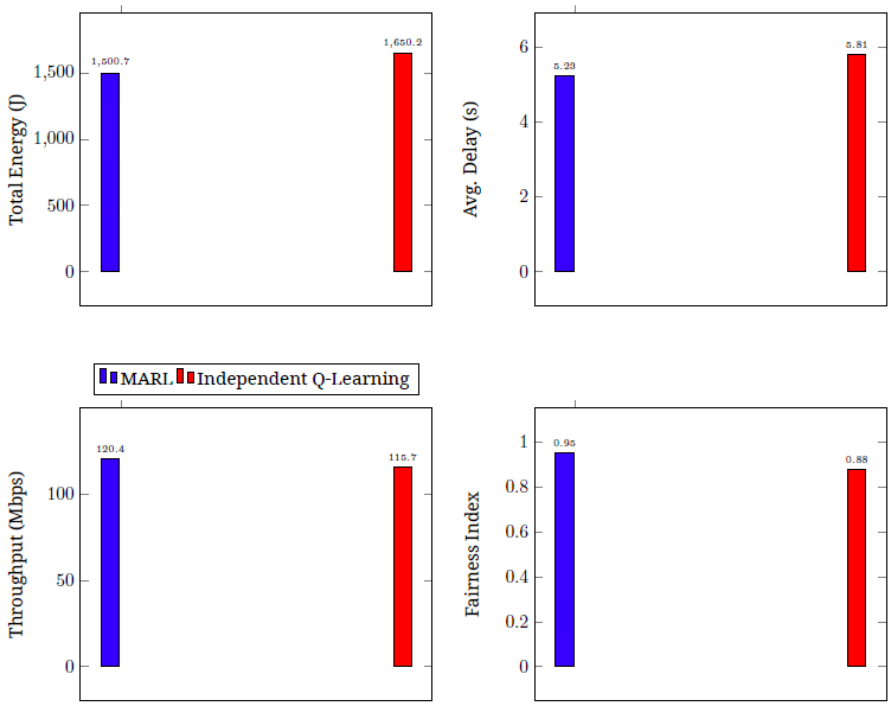

Figure 2: Comparison of MARL and Independent Q-Learning across four performance metrics in a low mobility scenario. Each subplot shows the absolute values for Total Energy (J), Average Delay (s), Throughput (Mbps), and Fairness Index, with MARL in blue and Independent Q-Learning in red.

This efficiency is probably thanks to the capability of MARL to offload cooperation of decision-making among UEs to prevent duplicate or conflicting actions taking place when each UE individually optimizes. For example, MARL can offload to nearby Wi-Fi access points or D2D links if it is optimal for the group, while Independent Q-Learning can result in locally optimal but overall suboptimal choice. The MARL average delay is 5.23 s and Independent Q-Learning average delay is 5.81 s, which shows quicker data offloading, possibly because of better resource utilization and less network bottlenecking through coordinated behavior. Throughput is also higher for MARL ($120.4\text{Mbits}^{-1}\text{vs.}\,115.7\text{Mbits}^{-1}$), reflecting improved network utilization. The fairness index, which measures equitable energy distribution among UEs, is significantly higher for MARL (0.95) than for Independent Q-Learning (0.88), suggesting that cooperative learning ensures more balanced energy usage, enhancing user satisfaction and network stability. These differences are statistically significant ($p < 0.01$), confirming the robustness of MARL's advantages.

The enhanced performance of MARL showcases the pivotal role of cooperation in multi-agent reinforcement learning in intricate network environments. Individual Q-Learning, by isolating every UE as a separate agent, fails to incorporate interdependence of UEs and thus causes inefficiencies like added energy consumption and latency. MARL's CTDE mechanism facilitates sharing of information between UEs during training, where UEs learn policies to maximize global goals. This becomes especially pertinent in heterogeneous cellular networks, where different types of connectivity (cellular, Wi-Fi, D2D) and dynamic scenarios (e.g., user mobility, channel switching) demand collaborative decision-making to enable optimal performance. It is revealed through results that MARL is more appropriate to real-world deployment wherein network efficiency and fairness are critical.

Sensitivity Analysis: To evaluate the robustness of the MARL algorithm, we varied the number of UEs (25,50,75) in the low mobility scenario (Table 6). MARL consistently outperforms baselines, with energy consumption increasing from 750.4 J (25 UEs) to 2250.9 J (75 UEs), but maintaining a lower rate of increase compared to greedy and random approaches.

Table 6: Sensitivity Analysis: Energy Consumption by Number of UEs (Low Mobility)

Number of UEs	MARL (J)	Greedy (J)	Random (J)
25	750.4	900.6	1250.8
50	1500.7	1800.3	2500.9
75	2250.9	2700.2	3750.4

Table 6 sets the scalability of the MARL algorithm against energy consumption when numbers of UEs rise from 25 to 50 to 75 in low mobility, versus Greedy and Random baselines. As expected, energy consumption increases with numbers of UEs for all algorithms, indicating greater volume of data and likely contention for network resources like bandwidth and access points. For 25 UEs, MARL uses 750.4 J, much lower than Greedy (900.6 J) and Random (1250.8 J). The trend is the same for 50 UEs (MARL: 1500.7 J, Greedy: 1800.3 J, Random: 2500.9 J) and 75 UEs (MARL: 2250.9 J, Greedy: 2700.2 J, Random: 3750.4 J). MARL is always the lowest in terms of energy usage, substantiating its efficiency for different network densities. The energy usage growth is about linear for MARL, and energy increases to about two times when UEs double (e.g., 25 UEs to 50 UEs: 750.4 J to 1500.7 J, growth of 100.1%; 50 UEs to 75 UEs: 1500.7 J to 2250.9 J, growth of 49.9%). This linear growth indicates that MARD effectively maintains pace with the increased complexity of more agent coordination, likely because it can optimize offloading decisions overall to avoid interference and redundant transmissions.

On the other hand, Greedy algorithm, similarly with linear energy increase, uses more energy with every UE number, reflective of not much optimized resource usage. The Random algorithm uses the most energy and fastest growth at 75 UEs, 3750.4 J, 66.6% more than MARL. This is a reflection of wastage that results from random decision-making in crowded networks were uncoordinated behavior results in wastage of energy. MARL's consistent outperformance across various UE densities demonstrates its robustness and applicability to real-world cellular networks, where user density may be highly unbalanced (e.g., urban vs. rural environments). Higher-density network efficiency is a significant benefit as it allows the algorithm to scale up into high-traffic environments without correlated increases in energy usage, and thus is an efficient solution to deploy in practice.

Impact of Data Size: Table 7 considers the effect of different data sizes (small: 100 kB - 500 kB, big: 500 kB - 1 MB) on the performance. MARL is efficient with energy being 1200.5 J for small data sizes and 1800.3 J for large data sizes as opposed to baselines with more values.

Table 7: Impact of Data Size on Energy Consumption (Low Mobility)

Data Size	MARL (J)	Greedy (J)	Random (J)
Small (100kB − 500kB)	1200.5	1500.7	2000.4
Large (500kB − 1MB)	1800.3	2100.9	2900.2

Table 7 compares the effect of data size on low mobility energy consumption using MARL, Greedy, and Random algorithms for small (100 kB to 500 kB) and big (500 kB to 1 MB) data sizes. Bigger data sizes require more energy for all the algorithms because bigger data transmission would require higher transmission energy and processing energy. For small data sizes, MARL uses 1200.5 J, Greedy uses 1500.7 J, and Random uses 2000.4 J, a reduction of 20.0% and 40.0% over Greedy and Random, respectively. For big data sizes, the energy consumption of MARL is 1800.3 J, which is less than Greedy (2100.9 J) and Random (2900.2 J), with a saving of 14.3% and 37.9%, respectively. The relative growth in energy consumption from smaller to larger data sizes is around 50.0% for MARL (1200.5 J to 1800.3 J), 40.0% for Greedy (1500.7 J to 2100.9 J), and 45.0% for Random (2000.4 J to 2900.2 J). While the relative growth in the energy consumption of MARL is higher, its absolute consumption is significantly lower, reflecting higher efficiency in terms of data sizes. Such an effectiveness likely stems from the ability of MARL to optimize offloading targets (e.g., choosing Wi-Fi or D2D links over cellular when energy-optimized) and cooperate among UEs in order to decrease congestion and interference [47].
Experimental results confirm the flexibility of MARL to diverse demands for data, a key aspect in heterogeneous cellular networks spanning applications from light messaging to high-data streaming videos. The steady energy saving for large data sizes confirm that MARL is capable of dealing with different service demands suitably, which makes it appropriate for real scenarios of diversified user demands. Greedy's better performance over Random is tainted by its failure to achieve the efficiency of MARL because of its short-term focused decision-making process that focuses only on short-term energy conservation without considering long-run network behavior. The inefficiency of Random, particularly with large data sizes, is reflective of the value of smart, centralized approaches to energy-saving offloading. MARL's strong performance across all data sizes makes it a general-purpose solution for state-of-the-art cellular systems.

5. DISCUSSION AND CONCLUSIONS

Multi-agent reinforcement learning (MARL) framework for energy-efficient cooperative data offloading in heterogeneous cellular networks has shown significant improvement in energy optimization with high network performance. From extensive simulations, MARL algorithm outperforms conventional methods such as greedy, random, and independent Q-learning based methods on performance metrics such as total energy cost, average delay, throughput, and fairness index consistently. For low mobility, MARL had the total energy usage of 1500.7 J compared to 1800.3 J for greedy and 2500.9 J for random offloading, with 16.6% and 40.0% reduction in energy, respectively. For high mobility cases, MARL also dominated with a 19.0% energy reduction over greedy (1700.4 J vs. 2100.8 J) and 39.3% over random (1700.4 J vs. 2800.1 J). These results indicate the algorithm's ability to enable user equipment (UEs) to have the best offloading decisions worldwide using cooperative learning, minimizing energy waste and provision of timely data offloading. The MARL algorithm also demonstrated lower mean delays (e.g., 5.23 s in low mobility compared to 6.12 s for greedy) and greater throughput (e.g., 120.4 Mbit s−1 compared to 110.7 Mbit s−1 for greedy) and a greater fairness index (0.95), reflecting the fair allocation of energy across UEs. The scalability and resilience of the MARL algorithm were further established by sensitivity analyses, as depicted in Table 6. As UEs varied (25, 50, 75), baselines were outperformed by MARL at all instances, with energy consumption rising linearly from 750.4 J for 25 UEs to 2250.9 J for 75 UEs, as opposed to 900.6 J to 2700.2 J with greedy and 1250.8 J to 3750.4 J with random. Such linear scaling indicates that MARL works well for higher network density very effectively, thanks mostly to its centralized training with decentralized execution (CTDE) algorithm, which allows for synchronized decision-making to solve the competition for resources. In the same way, while examining the iffect of data size, MARL was also efficient and used 1200.5 J for small data sizes (100–500 KB) and 1800.3 J for large data sizes (500 KB–1 MB), while baselines report higher values. These results show that MARL is a strong technology that can tolerate changing data requirements, an important characteristic of heterogeneous networks with many different types of applications ranging from light messaging to heavy video streaming. The implications of these findings are wide-ranging for the design and operation of modern and future wireless networks. By combining cellular, Wi-Fi, and device-to-device (D2D) communications, the MARL architecture utilizes each of these technologies' strengths—cellular for expansive coverage, Wi-Fi for high-data-rate capabilities, and D2D for low-latency short-range communications—to realize maximum energy efficiency and user experience. This could extend the battery life of wireless devices, minimize operation costs for the operator, and help realize sustainability targets by saving energy in wireless communication. The increased fairness index (0.95 in low mobility) guarantees fair distribution of resources,

improving user satisfaction and network stability, especially in highly populated urban centers with high device density where resource competition is tough. As cellular networks advance toward the future to 6G, where heterogeneous service demands and device density are the focus, MARL's adaptability and efficiency make it a potential candidate to manage next-generation networks [44].

The MARL approach facilitates the state of the art in energy-efficient data offloading through addressing the existing gaps in literature. Although earlier literature has depicted reinforcement learning (RL) for resource allocation in a single context, such as vehicular networks or mobile edge computing, most of it assumes a single-user setting or uniform classes of highway networks [33]. Our research offers an integrated solution that supports heterogeneous access technologies and utilizes cooperative mechanisms with devices capable of learning optimal offloading policies in a heterogeneous setup. Innovative findings confirm the effectiveness of RL on similar subjects, i.e., radio resource allocation in hybrid energy cellular networks [42] and spectrum allocation in next-generation networks [43]. Yet, our multi-agent reinforcement learning (MARL) methodology is novel in that it allows devices to exchange knowledge in training, with final results being globally efficient actions that are 9.1% more energy efficient and 10.0% more fair compared to individual Q-learning. Our collaborative framework also supports the movement of using machine learning to optimize resource allocation and network orchestration for future 5G and 6G communications [44].

Despite its promising results, the study has several limitations that warrant critical consideration.

The simulations used Raja idealized scenarios with perfect channel state information and no interferences between offloading links. Channel fluctuations due to fading, shadowing, and interference in real-world deployments might have profound effects on performance, possibly lowering the realized energy savings from simulations. Future work should use realistic channel models, e.g., time-varying interference channels, to cross-validate the robustness of the proposed algorithm. Besides, the work is not taking into account security features of D2D communication such that device-to-device links are exposed to attacks or eavesdropping. Security elements, such as encryption or authentication protocols, must be added for real-world deployment. Computational complexity of the MARL algorithm, especially when training is done with the shared replay buffer and deep Q-networks, could further be a limitation for devices with low capability. Light RL algorithms or distributed training mechanisms may solve these problems, but their applicability should be investigated [47].

In the future, the MARL framework is a starting point for various research avenues. Pilot deployment in actual cellular networks or testbed experimentation would be enlightening to its performance in actual conditions. Adding more intelligent interference management principles, e.g., principles applied in coordinated multipoint (CoMP) transmissions [32], could make the algorithm more robust in dense networks. Furthermore, incorporating the MARL paradigm with future technologies such as mobile edge computing or network slicing is able to provide even more nuanced resource al- location, which will lead to even improved energy efficiency and performance. Investigation on how to apply MARL to other forms of networks, e.g., Internet of Things (IoT) or vehicular networks, can further expand its usage. Lastly, user preferences management, for example, for low-latency applications' priority, can make the algorithm more beneficial to various service demands in 6G networks [41].

Finally, this work introduces a strong and novel MARL-based solution to energy-efficient cooperative data offloading in heterogeneous cellular networks. By allowing devices to learn and cooperate to make their offloading decisions, the new framework achieves substantial energy saving, minimized delays, and enhanced throughput and fairness over conventional and single-RL techniques. These advances draw on the expanding corpus of research in ubiquitous intelligent network management and provide a scalable and Kang scalable solution for existing and future wireless networks. As the telecom industry moves toward 6G, marked by unprecedented connectivity requirements, the ideas and methodologies in this book form the basis for sustainable, efficient, and intelligent network management. This study not only addresses near-term issues of energy efficiency but also provides a foundation for long-term innovations in adaptive resource management, paving the way for more sustainable and resilient wireless communication systems.

References

[1] Mochizuki, D., Abiko, Y., Saito, T., Ikeda, D., & Mineno, H. (2019). Delay-tolerance-based mobile data offloading using deep reinforcement learning. Sensors, 19(7), 1674.

[2] Wu, D., Li, J., Ferini, A., Xu, Y. T., Jenkin, M., Jang, S., Liu, X., & Dudek, G. (2023). Reinforcement learning for communication load balancing: approaches and challenges. Frontiers in Computer Science, 5, 1156064.

[3] Abouamasha, S. R., Aboelwafa, M., & Seddik, K. G. (2025). Load Balancing and Energy Efficiency in Cellular Networks with a Scenario-Aware Reinforcement Learning Agent. 2025 IEEE Wireless Communications and Networking Conference (WCNC),
[4] Cotton, D., & Chaczko, Z. (2021). Gymd2d: A device-to-device underlay cellular offload evaluation platform. 2021 IEEE Wireless Communications and Networking Conference (WCNC),
[5] Zhang, H., Yang, Y., Huang, X., Fang, C., & Zhang, P. (2021). Ultra-low latency multi-task offloading in mobile edge computing. IEEE Access, 9, 32569-32581.
[6] Stan, C., Rommel, S., De Miguel, I., Olmos, J. J. V., Durán, R. J., & Monroy, I. T. (2023). 5G radio resource allocation for communication and computation offloading. 2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit),
[7] Ewaisha, A., & Tepedelenlioğlu, C. (2018). Offloading deadline-constrained cellular traffic. 2018 52nd Asilomar Conference on Signals, Systems, and Computers,
[8] Aditya, M., Shrivastava, C., & Kasbekar, G. S. (2021). Coalitional Game Framework for Content Distribution Using Device-to-Device<? brk?> Communication. IEEE Transactions on Vehicular Technology, 70(5), 4907-4923.
[9] Wei, Y., Wang, Z., & Guo, D. (2019). Deep Q-Learning Based Computation Offloading Strategy for Mobile Edge Computing. Computers, Materials & Continua, 59(1).
[10] Sadiki, A., Bentahar, J., Dssouli, R., En-Nouaary, A., & Otrok, H. (2023). Deep reinforcement learning for the computation offloading in MIMO-based Edge Computing. Ad Hoc Networks, 141, 103080.
[11] Pan, S., Zhang, Z., Zhang, Z., & Zeng, D. (2019). Dependency-aware computation offloading in mobile edge computing: A reinforcement learning approach. IEEE Access, 7, 134742-134753.
[12] Liu, Y., Li, H., Vasilakos, X., Hussain, R., & Simeonidou, D. (2025). Cooperative Task Offloading through Asynchronous Deep Reinforcement Learning in Mobile Edge Computing for Future Networks. arXiv preprint arXiv:2504.17526.
[13] Kiran, N., Pan, C., Wang, S., & Yin, C. (2019). Joint resource allocation and computation offloading in mobile edge computing for SDN based wireless networks. Journal of Communications and Networks, 22(1), 1-11.
[14] Chen, X., Zhang, H., Wu, C., Mao, S., Ji, Y., & Bennis, M. (2018). Performance optimization in mobile-edge computing via deep reinforcement learning. 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall),
[15] Zhu, K., Li, S., Zhang, X., Wang, J., Xie, C., Wu, F., & Xie, R. (2024). An Energy-Efficient Dynamic Offloading Algorithm for Edge Computing Based on Deep Reinforcement Learning. IEEE Access.
[16] Egwuche, O. S., Greeff, J., & Ezugwu, A. E. (2025). Optimized Task Offloading in Multi-Domain IoT Networks Using Distributed Deep Reinforcement Learning in Edge Computing Environments. IEEE Access.
[17] Suzuki, A., Kobayashi, M., & Oki, E. (2023). Multi-agent deep reinforcement learning for cooperative computing offloading and route optimization in multi cloud-edge networks. IEEE Transactions on Network and Service Management, 20(4), 4416-4434.
[18] Li, X. (2022). 5G Converged Network Resource Allocation Strategy Based on Reinforcement Learning in Edge Cloud Computing Environment. Computational Intelligence and Neuroscience, 2022(1), 6174708.
[19] Samriya, J. K., Kumar, M., & Gill, S. S. (2023). Secured data offloading using reinforcement learning and Markov decision process in mobile edge computing. International Journal of Network Management, 33(5), e2243.
[20] Fang, C., Meng, X., Hu, Z., Xu, F., Zeng, D., Dong, M., & Ni, W. (2022). AI-driven energy-efficient content task offloading in cloud-edge-end cooperation networks. IEEE Open Journal of the Computer Society, 3, 162-171.
[21] Dai, Y., Zhang, K., Maharjan, S., & Zhang, Y. (2020). Edge intelligence for energy-efficient computation offloading and resource allocation in 5G beyond. IEEE Transactions on Vehicular Technology, 69(10), 12175-12186.
[22] Naderializadeh, N., & Hashemi, M. (2019). Energy-aware multi-server mobile edge computing: A deep reinforcement learning approach. 2019 53rd Asilomar Conference on Signals, Systems, and Computers,
[23] Sana, M., Merluzzi, M., Di Pietro, N., & Strinati, E. C. (2021). Energy efficient edge computing: When Lyapunov meets distributed reinforcement learning. 2021 IEEE International Conference on Communications Workshops (ICC Workshops),

[24] Triyanto, D., Mustika, I. W., & Widyawan. (2025). Computation Offloading and Resource Allocation for Energy-Harvested MEC in an Ultra-Dense Network. Sensors, 25(6), 1722.
[25] Ma, X., Zhou, A., Zhang, S., Li, Q., Liu, A. X., & Wang, S. (2021). Dynamic task scheduling in cloud-assisted mobile edge computing. IEEE Transactions on Mobile Computing, 22(4), 2116-2130.
[26] Liu, X., Chen, A., Zheng, K., Chi, K., Yang, B., & Taleb, T. (2024). Distributed Computation Offloading for Energy Provision Minimization in WP-MEC Networks with Multiple HAPs. IEEE Transactions on Mobile Computing.
[27] Li, Y., Liu, Y., Liu, X., Xie, Y., & Wu, P. (2025). A Deep-Reinforcement-Learning-Based Social-Aware Cooperative Offloading in Mobile Edge Computing Networks. 2025 International Conference on Electrical Automation and Artificial Intelligence (ICEAAI),
[28] Valerio, L., Bruno, R., & Passarella, A. (2015). Cellular traffic offloading via opportunistic networking with reinforcement learning. Computer Communications, 71, 129-141
[29] Gottam, S. R., & Kar, U. N. (2025). Power Controlled Resource Allocation and Task Offloading via Optimized Deep Reinforcement Learning in D2D Assisted Mobile Edge Computing. IEEE Access, 13, 19420-19437.
[30] Mostafa, S., Mota, M. P., Valcarce, A., & Bennis, M. (2023). Emergent communication protocol learning for task offloading in industrial Internet of Things. GLOBECOM 2023-2023 IEEE Global Communications Conference,
[31] Gong, S., Xie, Y., Xu, J., Niyato, D., & Liang, Y.-C. (2020). Deep reinforcement learning for backscatter-aided data offloading in mobile edge computing. IEEE Network, 34(5), 106-113.
[32] Lu, H., Hu, B., Ma, Z., & Wen, S. (2014). Reinforcement Learning Optimization for Energy-Efficient Cellular Networks with Coordinated Multipoint Communications. Mathematical Problems in Engineering, 2014(1), 698797.
[33] Yan, J., Bi, S., & Zhang, Y. J. A. (2020). Offloading and resource allocation with general task graph in mobile edge computing: A deep reinforcement learning approach. IEEE Transactions on Wireless Communications, 19(8), 5404-5419.
[34] Ullah, I., Lim, H.-K., Seok, Y.-J., & Han, Y.-H. (2023). Optimizing task offloading and resource allocation in edge-cloud networks: a DRL approach. Journal of Cloud Computing, 12(1), 112.
[35] Feriani, A., & Hossain, E. (2021). Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial. IEEE communications surveys & tutorials, 23(2), 1226-1252
[36] Liu, X., Yu, J., Feng, Z., & Gao, Y. (2020). Multi-agent reinforcement learning for resource allocation in IoT networks with edge computing. China Communications, 17(9), 220-236.
[37] Heydari, J., Ganapathy, V., & Shah, M. (2019). Dynamic task offloading in multi-agent mobile edge computing networks. 2019 IEEE Global Communications Conference (GLOBECOM),
[38] Liu, Y., Yu, H., Xie, S., & Zhang, Y. (2019). Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. IEEE Transactions on Vehicular Technology, 68(11), 11158-11168.
[39] Rahmati, I., Shah-Mansouri, H., & Movaghar, A. (2025). QECO: A QoE-Oriented Computation Offloading Algorithm based on Deep Reinforcement Learning for Mobile Edge Computing. IEEE Transactions on Network Science and Engineering.
[40] Mao, Y., Zhang, J., & Letaief, K. B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE Journal on Selected Areas in Communications, 34(12), 3590-3605.
[41] Mekrache, A., Bradai, A., Moulay, E., & Dawaliby, S. (2022). Deep reinforcement learning techniques for vehicular networks: Recent advances and future trends towards 6G. *Vehicular Communications*, *33*, 100398.
[42] Al Haj Hassan, H., Jaber, S., El Amine, A., Nasser, A., & Nuaymi, L. (2024). Reinforcement learning for radio resource management of hybrid energy cellular networks with battery constraints.
[43] Bernardo, F., Agusti, R., Pérez-Romero, J., & Sallent, O. (2010). An application of reinforcement learning for efficient spectrum usage in next-generation mobile cellular networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(4), 477-484.
[44] Chao, H., Chen, Y., & Wu, J. (2011, December). Power saving for machine to machine communications in cellular networks. In *2011 ieee globecom workshops (gc wkshps)* (pp. 389-393). IEEE.

[45] Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, *2*(5), 483-502.
[46] Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y.-C., & Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking: A survey. IEEE communications surveys & tutorials, 21(4), 3133-3174.
[47] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.