Online ISSN: 2345-377X **Print ISSN:** 2345-3796

DOI: 10.57647/j.mjee.2025.17785

Original Research

Optimizing Network Performance in Mobile Data Offloading Using Cooperative IEEE 802.11 MAC Protocol and Deep Reinforcement Learning

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

1- Institute of Artificial Intelligence and Social and Advanced Technologies, Isf.C., Islamic Azad University, Isfahan, Iran

2- Department of Electrical Engineering, DOL.C., Islamic Azad University, Isfahan, Iran

3- Department of Electronic Engineering, Southern Technical University, Amara, Iraq

nabil@uomisan.edu.iq - rsadeghi@iau.ac.ir - wael.zayer@stu.edu.iq - mehdi.hamidkhani@iau.ac.ir - rkhorsand@iau.ac.ir

Corresponding author: rsadeghi@iau.ac.ir - https://orcid.org/0000-0003-1353-2789

Abstract

With the exponential growth of mobile data traffic and the increasing demand for latency-sensitive applications, wireless networks face critical resource management challenges. In particular, heterogeneous architectures such as LTE and Wi-Fi necessitate intelligent and adaptive offloading mechanisms to ensure efficiency and Quality of Service (QoS). This paper exploits Deep Reinforcement Learning (DRL) and cooperative communication within the MAC layer of the IEEE 802.11n standard to design an adaptive and scalable framework for data offloading in wireless networks. The proposed model employs the Partially Observable Markov Decision Process (POMDP) structure to optimally select the communication path and the type of communication (direct or multi-hop) in real-time. The proposed learning algorithm, which leverages the Deep Q-Learning structure and the Policy Improvement mechanism, demonstrates significant performance gains compared to conventional methods. Simulation results indicate that the average cumulative reward achieved by the DRL algorithm is 3.4, a

substantial improvement in decision-making effectiveness compared to 2.2 for Q-Learning and 0.7 for Heuristic methods. Moreover, the energy efficiency of the proposed method improved by 87% compared to CoopMAC, and the throughput increased by 18%. These results establish the proposed framework as an effective and low-power solution for implementation in Fifth-Generation (5G) network architectures.

KEYWORDS: Mobile data offloading; Deep reinforcement learning; Cooperative relay selection; CoopMAC; Real-time optimization; Q-learning.

Introuction

The dramatic rise of smart mobile devices (SMDs) and the rapid expansion of Internet of Things (IoT) ecosystems have together led to an exponential growth in global data traffic. Forecasts suggest that by 2030, the number of connected devices within the IoT ecosystem will exceed 42 billion, generating approximately 80 zettabytes of data [1]. This unprecedented surge in data exchange, coupled with the growth of latency-sensitive applications such as cloud gaming and real-time video streaming, is placing significant pressure on existing communication infrastructure. This issue becomes even more challenging in industrial and mission-critical applications. To address these issues, it is essential to manage network resources intelligently, ensuring users benefit from optimal Quality of Service (QoS) while simultaneously utilizing network capacity to its full potential [2, 3].

One of the fundamental challenges in heterogeneous wireless communication environments is the shortage of spectral resources and the critical need for their efficient management.1 With the increasing number of users and connected devices, the frequency spectrum in cellular networks has become a limited resource, resulting in greater congestion and reduced data transfer rates [4]. In this regard, mobile data offloading has been proposed as a key solution that utilizes complementary wireless networks such as Wi-Fi to transfer large volumes of transmitted data from cellular networks to these lower-cost communication platforms [5]. This method not only helps mitigate congestion in cellular networks but also increases spectral efficiency and improves overall network performance. However, the efficacy of data offloading is highly dependent on the efficiency of the complementary network (Wi-Fi), the optimal utilization of access points (APs), and proper coordination between heterogeneous network resources [6, 7].

Despite the potential benefits of mobile data offloading, it faces several challenges that, if not managed properly, can reduce overall network performance. Crucially, one of the most significant of these challenges is the coordination of Quality of Service (QoS) in heterogeneous network environments [8]. Since Wi-Fi and cellular networks possess different data rates, bandwidths, and network coverage, an intelligent coordination mechanism is necessary to manage the data offloading process and the handover of connections between these distinct networks. In this

context, latency-sensitive applications such as cloud gaming and industrial communications require Ultra-Reliable Low-Latency Communication (URLLC) [9], which is extremely challenging to maintain during the switch between cellular and Wi-Fi networks. Another important challenge lies in optimizing the number of Wi-Fi Access Points (APs) to improve data offloading efficiency. In densely populated scenarios, such as high-traffic urban areas, improper utilization of APs often leads to unbalanced loading and a subsequent reduction in offloading efficiency [10].

Deep Reinforcement Learning (DRL) is an advanced artificial intelligence approach that combines classical Reinforcement Learning (RL) with Deep Neural Networks (DNNs) to solve complex dynamic decision-making problems [11]. This method is particularly utilized in wireless network resource management, including mobile data offloading, spectrum allocation, and connection handover optimization [12, 13]. In applications related to heterogeneous network management, the DRL learning agent can offload its decisions to distributed computing nodes to ensure real-time optimization of network resources and reduce computational overhead on end devices [14]. Specifically, within the mobile data offloading process, DRL can determine the most optimal data transmission path, the connection handover schedule, and bandwidth allocation by offloading the learning and prediction processes to cloud servers or edge computing points. This approach not only reduces energy consumption and processing latency but also exploits the full potential of the network for data distribution and load balancing between Wi-Fi and cellular networks [15].

In this paper, we propose a cooperative Wi-Fi data offloading framework that leverages the IEEE 802.11 cooperative MAC protocol and Deep Reinforcement Learning (DRL) to significantly improve mobile data offloading performance. The proposed method employs relay nodes and a DRL algorithm to make intelligent decisions regarding optimal data transmission path selection, network resource allocation, and the reduction of connection handover latency. Specifically, a Deep Neural Network (DNN) is utilized as a learning agent to analyze the current network status and traffic conditions, enabling it to make the best decisions for increasing data offloading efficiency. The most important innovations and contributions of this work are as follows:

1. MAC Layer Integration of DRL and Cooperation: Unlike most previous studies focusing solely on higher network layers, our model is implemented at the MAC layer and exploits the CoopMAC mechanism for efficient multi-hop communication.
2. POMDP-Based Decision Modelling: The complex decision-making process—including relay node selection, resource allocation, and determining the type of communication (direct or cooperative)—is rigorously modelled using a Partially Observable Markov Decision Process (POMDP) and learned with the aid of a deep neural network.
3. Dynamic AP Offloading Optimization: In contrast to similar works that rely on static or pre-defined strategies, the proposed model dynamically adjusts the placement and utilization of Wi-Fi Access Points (APs) based on real-time network conditions and traffic distribution.

4. Enhanced Energy Efficiency: By substantially increasing the offloading ratio from LTE to Wi-Fi, the proposed framework demonstrates a significant reduction in energy consumption in high-traffic scenarios.
5. Robust Statistical Validation: A detailed statistical analysis using a two-sample t-test rigorously proves a significant performance difference between the DRL algorithm and other methods across all key indicators (Throughput, Offloading Ratio, Energy Efficiency, and Delay).
6. Accelerated and Sustainable Learning: The integration of the Policy Improvement mechanism within the DRL structure leads to accelerated convergence and demonstrably more accurate, sustainable decision-making in the face of changing network conditions.

The remainder of this paper is organized as follows. Section 2 reviews related research in the field of Wi-Fi-based data offloading and DRL algorithms for network optimization. Section 3 details the proposed model, including the relay-based cooperative data offloading scheme and its integration into the MAC layer of IEEE 802.11n networks. Section 4 presents the performance evaluation results, focusing on the analysis of throughput and the data offload ratio of Access Points (APs). Finally, Section 5 provides the conclusions and suggests directions for future research.

Related works

The increasing demand for high-throughput and low-latency mobile data transmission has led to extensive research into mobile data offloading strategies [16]. Existing approaches can typically be categorized into: delay-aware offloading, decision-making-based offloading, and reinforcement learning-based cooperative offloading. These strategies primarily aim to optimize offloading performance by addressing network congestion, enhancing resource utilization, and ensuring seamless data transmission. However, each category presents certain limitations, which necessitate the development of more adaptive, scalable, and intelligent offloading mechanisms [17].

Delay-Aware Offloading Techniques

Delay-aware offloading techniques focus on managing data transmission in heterogeneous networks while explicitly considering delay constraints. These approaches are particularly relevant for both delay-tolerant and delay-sensitive applications, such as real-time video streaming and IoT-based communications.

One notable study proposed a traffic offloading model that accounts for user-specific constraints, including battery life, storage capacity, and mobility patterns [18]. This model employs buffer-aware algorithms to adjust offloading policies dynamically based on call rates and data lifetimes. However, its applicability is limited to delay-tolerant networks (DTNs), rendering it unsuitable for real-time applications requiring strict latency control.

Another significant contribution investigated offloading via Device-to-Device (D2D) communication, introducing a QoS-aware scheduling algorithm that prioritizes data transmission based on link length and bit rate [19]. Although this method enhances spectral efficiency, it suffers from latency degradation in dense D2D environments, where maintaining stable connections becomes challenging. Similarly, a study analyzed Wi-Fi offloading efficiency under delayed and non-delayed transmission modes, concluding that delayed offloading significantly improves energy efficiency but is highly dependent on the chosen latency threshold [20]. While beneficial for optimizing network load, this approach lacks the adaptability required for real-time offloading decisions in dynamic environments.

A more comprehensive approach was presented in a study focusing on DTN-based offloading, integrating multi-network routing strategies to improve packet delivery reliability [21]. Evaluations conducted using the ONE simulator demonstrated that most of the network traffic could be offloaded effectively, but the model was constrained by its reliance on a single DTN routing version, highlighting the need for multi-path feedback mechanisms to optimize offloading efficiency. Other related works have explored opportunistic networking approaches [22], Reinforcement Learning-based cost and energy efficiency optimizations [23], and security-aware offloading mechanisms [24], with additional research addressing vehicular network offloading in dynamic mobility scenarios [25].

Decision-Making-Based Offloading Strategies

Decision-making-based offloading strategies focus on determining the optimal Access Points (APs) and user participation in the offloading process. These methods frequently incorporate adaptive algorithms, heuristic-based optimization, and machine learning techniques to enhance offloading efficiency and network adaptability.

One work introduced a Lyapunov-based optimization model for dynamic energy-efficient offloading, proposing an adaptive framework that categorizes tasks into offloading and non-offloading components [26]. Despite its ability to minimize energy consumption, this approach is computationally intensive, limiting its feasibility in real-time, high-density network environments.

Another investigation developed an autonomous Reinforcement Learning (RL)-based offloading strategy, where network agents interact with the environment, observe system behavior, and adjust offloading decisions accordingly [27]. Although this method enables self-optimization, its reliance on extensive training data introduces a high initial computational overhead, making it less efficient for rapidly changing network conditions. An alternative contribution introduced an incentive-driven offloading strategy, encouraging users to participate in offloading by sharing their network resources [28]. While this framework effectively enhances AP-based offloading performance, it faces scalability issues, as user engagement in incentive-based offloading may decline over time.

A model on Wi-Fi AP capacity constraints proposed a hybrid greedy algorithm that optimizes offline and online offloading strategies based on data sensitivity levels [29]. While the model effectively

reduces network congestion, it employs a static optimization approach, failing to adapt dynamically to network fluctuations. Additionally, research on trust-aware offloading has investigated secure node selection mechanisms to ensure reliable data transmission in cooperative networks [30]. However, these approaches often introduce a security overhead, which can increase latency and processing costs.

Table 1. Mobile Data Offloading method compressions

Reference	Category	Key Contribution	Limitations
[18]	Delay-Aware Offloading	Proposed an offloading model considering battery, storage capacity, and user-specific demands	Limited to delay-tolerant networks, lacks real-time adaptability
[19]		Introduced a D2D offloading scheme with fair scheduling based on link length and bit rate	QoS degradation in high-latency D2D environments
[20]		Evaluated Wi-Fi capacity for offloading in delay-tolerant and non-delay scenarios	Does not optimize real-time offloading decisions
[21]		Investigated DTN-based offloading, considering guaranteed delivery and heterogeneous routing	Limited to single DTN routing protocol, lacks multi-path feedback
[26]	Decision-Based Offloading	Proposed a Lyapunov-based optimization for dynamic offloading with energy efficiency	High computational overhead, limited to computation offloading
[27]		Applied reinforcement learning to develop an autonomous offloading strategy	Requires extensive training data for RL convergence
[28]		Designed an incentive mechanism for user participation in AP-based offloading	Incentives are not scalable, affecting long-term adoption
[29]		Proposed a hybrid greedy algorithm for Wi-Fi AP offloading under capacity constraints	Static optimization, lacks dynamic adaptation

[30]]	Trust-Aware Offloading	Implemented trust-based node selection for secure and efficient data offloading	Security overhead increases latency and computational complexity
This Paper	Reinforcement Learning-Based Offloading	Cooperative relay-assisted offloading with deep RL-based relay selection	Higher complexity but enables real-time, adaptive offloading

Reinforcement Learning-Based Cooperative Offloading

Given the limitations of traditional offloading techniques, this paper proposes a Reinforcement Learning (RL)-based cooperative offloading mechanism, which integrates relay selection with MAC-layer cooperation to significantly enhance offloading efficiency. Unlike previous models, the proposed approach dynamically learns from real-time network conditions, optimizing relay selection to minimize delay and maximize throughput. Traditional RL-based offloading schemes often rely on predefined policies, which can restrict adaptability. In contrast, our approach leverages DRL to enable real-time optimization and intelligent decision-making [31]. The integration of cooperative relay selection at the MAC layer allows the proposed model to outperform conventional network-layer offloading techniques by reducing latency and improving spectral efficiency. Unlike existing decision-based models that optimize offloading at the user or network layer, our approach is seamlessly embedded into the IEEE 802.11n MAC protocol, ensuring compatibility with standard Wi-Fi communication protocols [32].

Table 1 summarizes the existing research works across four categories: delay-aware methods, decision-based strategies, and Reinforcement Learning (RL) models. While each of these approaches has contributed to reducing congestion, improving spectral efficiency, or supporting Quality of Service (QoS), they still present critical gaps. Delay-aware schemes are primarily suitable for non-real-time applications and lack adaptability under dynamic traffic. Decision-based methods often suffer from high computational complexity and scalability limitations, despite their adaptive nature. Furthermore, RL approaches provide promising optimization capabilities, but they are usually implemented at higher layers, thereby overlooking the potential of the MAC layer for cooperative communication.

To address these shortcomings, our work introduces a novel framework that integrates Deep Reinforcement Learning (DRL) with cooperative MAC-layer mechanisms in IEEE 802.11n networks. This design enables efficient relay selection, significantly reduced latency, higher offloading ratios, and optimized Access Point (AP) offloading in a way that remains compatible with current infrastructures.

Proposed method

The proposed system model represents a heterogeneous wireless network consisting of an LTE eNB (Evolved Node B), multiple Wi-Fi Access Points (APs), and User Equipment (UEs). Unlike traditional offloading models that rely on static or predefined allocation strategies, this model integrates Deep Reinforcement Learning (DRL) to dynamically optimize both data offloading and relay selection. The primary goal is to enhance network performance by intelligently distributing traffic between LTE and Wi-Fi networks, whilst simultaneously minimizing relay costs and improving spectral efficiency.

One of the key advancements in this model is the dynamic optimization of Wi-Fi AP placement for offloading efficiency. Instead of assuming a fixed AP placement, the model dynamically adjusts the location and density of Wi-Fi APs based on real-time network conditions. This is achieved through a reinforcement learning framework that continuously evaluates network congestion, user density, and signal quality to optimize AP positioning. By doing so, the model ensures that network resources are utilized efficiently, reducing LTE congestion and enhancing Wi-Fi coverage. The placement of APs is further optimized using a Partially Observable Markov Decision Process (POMDP), allowing the system to make intelligent offloading decisions even in environments with incomplete or uncertain network information.

To further refine offloading efficiency, the model employs a dual-objective Reinforcement Learning (RL) approach. The first objective is to select the optimal offloading path for each UE, determining whether data should be transmitted via direct LTE communication, direct Wi-Fi access, or cooperative Wi-Fi relay-based transmission. The second objective is to dynamically adapt the offloading ratio in Wi-Fi APs to ensure that network capacity is distributed effectively and user connectivity remains stable. By continuously learning from real-time network conditions, the system is capable of making adaptive and intelligent offloading decisions that maximize throughput, minimize delay, and optimize spectral utilization.

In summary, this enhanced system model offers a comprehensive solution for Reinforcement Learning (RL)-based mobile data offloading. By integrating cooperative MAC-layer communication and DRL, the proposed framework not only improves offloading efficiency but also reduces infrastructure costs and enhances overall network scalability. The ability to dynamically adjust both offloading strategies and network topology ensures that the system remains robust and efficient, even in dynamic and high-density network environments. Fig. 1 shows the system model of the proposed method.

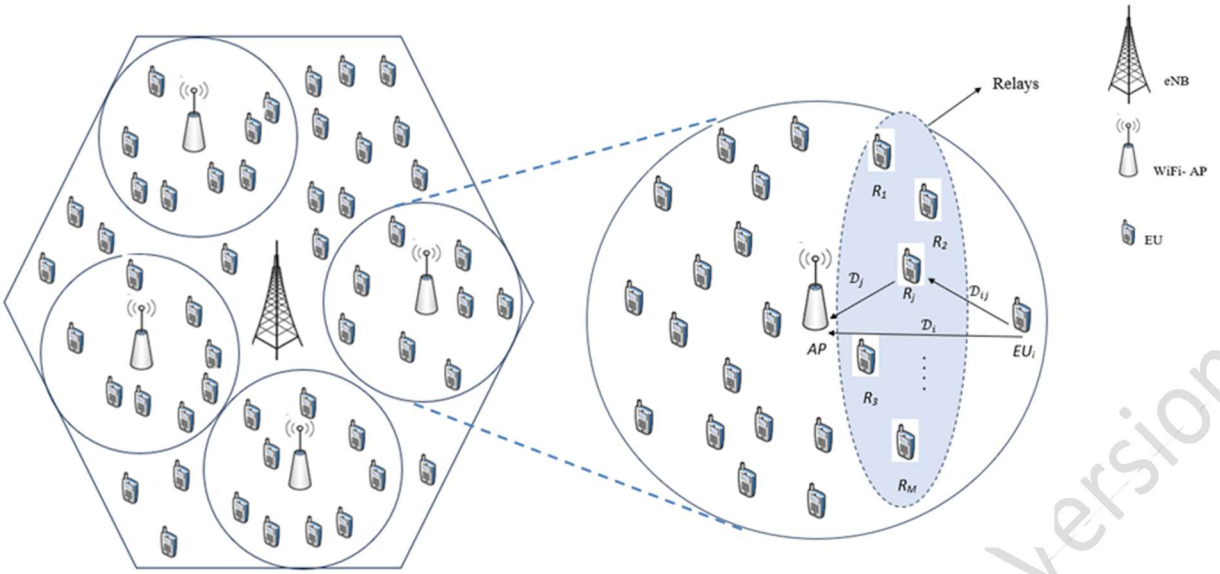

Fig. 1. Data offloading using collaborative and non-cooperative links (proposed system model)

In this section, a system model including the LTE and Wi-Fi networks, and a cooperation scenario will be explained. Then, this model integrates DRL to dynamically optimize both data offloading through cooperation.

System model: As indicated in Fig. 1.a, we consider a network including one LTE eNB (Evolved Node B), NAP number of Wi-Fi access points (APs), N_{ov} number of user equipment (UEs). Every AP_k can provide coverage of N_k^W UEs. These UEs ($\sum_{k=1}^{N_{AP}} N_k^W$)) have both access of LTE and Wi-Fi networks and the remained UEs ($N_{ov} - \sum_{k=1}^{N_{AP}} N_k^W$) have only access to LTE. In the cooperation mode of Wi-Fi network (Fig. 1), we suppose M relays (R_1, R_2,…, R_m ,.. R_M) for every UE_i. The overhearing can achieve three data rates of source and Rm ($\mathcal{D}_{SR_m}$), Rm and destination ($\mathcal{D}_{R_mD}$) and source-destination ($\mathcal{D}_{SD}$). To find the best relay node, we need to define a metric called CG as follows:

$$CG_m = \left(\frac{{\mathcal{D}_{SR_m}}^{-1} + {\mathcal{D}_{R_mD}}^{-1}}{{\mathcal{D}_{SD}}^{-1}} \right)^{-1} \quad (1)$$

The CG_m value determines the gain provided by relay R_m. The greater the value of CG_m, the faster relay R_m can establish the connection with lower delay.

After the selection of the best relay node, the exchange of the control packets and data packets is performed similarly to the CoopMAC protocol [10]. We need a more control packet called Helper to send (HTS) with a similar packet size of CTS for coordination to the relay node. Therefore, Eq. (4) in cooperation mode can be modified as follows:

$$T_s^{Coop} = \sum_{i=1}^{l} f_i \left[\left(P_i^{CG_m} \, t_i^{CG_m} \right) + \left(1 - P_i^{CG_m} \right) t_i^D \right] \quad (2)$$

Where $l, f_i, t_i^D, P_i^{CG_m}$ and $t_i^{CG_m}$ denote the number of supported data rate in Wi-Fi AP, the ratio of users with a data rate of di over the total number of users, the successful time of direct transmission with a direct data rate of di, the probability of finding a relay with CG_m and the successful time in cooperation mode. f_i and t_i Moreover, the value of $t_i^{CG_m}$can be modified as follows:

$$t_i^{CG_m} = T_{RTS} + 2T_{CTS} + \frac{L}{d_i\, CG_m} + T_{ACK} + 3\,T_{SIFS} + 4\sigma + T_{DIFS} \quad (3)$$

Where, T_{RTS}, T_{CTS}, T_{ACK} denote the time of control packet transmission and T_{SIFS} and T_{SIFS} denote the interframes time and σ is the propagation time. Then, the average throughput of an AP_k in cooperation mode can be expressed as follows:

$$S_k^{Coop} = \frac{P_s\ P_{tr} L}{N_k^w\left[(1-P_{tr})\sigma + P_{tr}\,P_s\ T_s^{Coop} + P_{tr}\,(1-P_s\ )T_c^{Coop}\right]} \quad (4)$$

Where, P_s, P_{tr} are the successful probability and transmission probability of a packet during a time slot and T_c^{Coop} denotes the collision time in cooperation mode. Moreover, to measure the energy efficiency provided by cooperation in the coverage of AP_k can be expressed as

$$EE_k^{Coop} = \frac{P_s\,P_{tr} E[P]}{N_k^w[(1-P_{tr})E_i + P_{tr}\,P_s\ E_s^{Coop} + (1-P_s)P_{tr}E_c]} \quad (5)$$

Where E_i, E_s^{Coop} and E_c denote the energy consumption in idle duration, successful transmission in cooperation mode and collision duration. Clearly, the energy consumption of relay node is taken account in Eq. (5)

In order to measure the impact of cooperation in data offloading, we define a metric called *throughput efficiency* (S_{eff}) which is the ratio of average throughput in cooperation mode of Wi-Fi and LTE in an offloading scenario to the average throughput of only LTE as follows:

$$S_{eff} = \left(\frac{\left(\sum_{k=1}^{N_{AP}} N_k^w\, S_k^{Coop}\right)/\sum_{k=1}^{N_{AP}} N_k^w + R_C^{UL}/(N_{ov} - \sum_{k=1}^{N_{AP}} N_k^w)}{R_{LTE}/N_{ov}}\right) \quad (6)$$

Where R_{LTE} denotes the bandwidth of LTE. Similarly, we can measure the Energy Efficiency (E_{eff}) ratio of average energy consumption in two modes of cooperation and non-cooperation and it can be expressed as

$$E_{eff} = \left(\frac{\left(\sum_{i=1}^{N_{AP}} N_k^w\, EE_k^{Coop}\right)/\sum_{i=1}^{N_{AP}} N_k^w + P_{LTE}/\left(R_{LTE}(N_{ov} - \sum_{i=1}^{N_{AP}} N_k^w)\right)}{P_{LTE}/\left(R_{LTE} N_{ov}\right)}\right) \quad (7)$$

Where, EE_k^{Coop} and P_{LTE}/R_{LTE} denote the energy efficiency in a cooperative Wi-Fi network [33] and LTE network, respectively.

Considering the proposed heterogeneous network structure and the key role of relay nodes in enhancing the data offloading process, the selection of the appropriate relay for establishing

cooperative communication between the User Equipment (UE) and the Access Point (AP) is regarded as a critical and decisive factor in the overall network performance. This decision not only directly affects critical indicators such as data transfer rate, latency, and energy consumption, but must also be made in real-time within a dynamic and changing environment.

Since the behavior of the decision-maker depends solely on the current state of the system and does not require the complete history, the relay selection process can be naturally modelled as a Markov Decision Process (MDP). In the following, the main components of this MDP—including the state space, the set of actions, the reward function, and the transition probabilities between different states—are introduced and analyzed.

To provide a clearer understanding of the equations, each key metric can be intuitively interpreted. The Cooperation Gain (CG) indicates the improvement in data rate achieved by using a relay compared to direct transmission, thereby highlighting the role of cooperative communication. The Throughput reflects how effectively the network utilizes its available capacity for data delivery. The Energy Efficiency captures the ratio of transmitted data to energy consumed, directly reflecting the energy-optimality of the offloading process. Finally, the Reward Function combines these metrics in a balanced manner, guiding the learning agent toward making optimal decisions under varying network conditions.

MDP analysis of relay selection: In this section, the cooperative communication process is modeled as a series of state transitions. Because the decision to select a relay or take other actions depends solely on the current state and behavior, a Markov Decision Process (MDP) is used for relay selection. The MDP is defined by the four-tuple (A, S, R, P), as following:

State S**:** The system's state space, denoted by S, is finite and encompasses all possible system states. The state space is defined by all possible cooperation gains (CG) between the source, relay and destination in the current state: $S = [0, 1, CG_1, CG_2, \dots, CG_K]$, where

$$S = \begin{cases} 0 & Direct\ LTE \\ 1 & Direct\ Wi\text{-}Fi \\ CG_i & Cooperation\ with\ CGi \end{cases} \quad (8)$$

Action (A): The set of possible actions, A, depends on the current state of the system and direct data rates of the user to LTE, Wi-Fi and cooperation Wi-Fi scenarios. A ={a_t} consists of actions a_t where t = -1,0,1,2, …. M. Every action a_t can be one of the values {-1, 0, 1, 2, ..., m, ..., M}. If at = -1, the source node S transmits directly to eNB in LTE communication. If at = 0, the source node S transmits directly to the Wi-Fi AP. If at = m, the source node S uses relay R_m for cooperative communication in Wi-Fi scenario.

Reward (R): The reward function, *R*, is crucial in reinforcement learning, providing feedback based on the agent's actions and we select a reward function dependent to the performance metric including throughput and energy efficiency as following:

$$R(s_t, a_t) = 0.5\ \frac{S_{eff}}{S_{eff}^{max}} + 0.5\ \frac{E_{eff}}{E_{eff}^{max}} \quad (9)$$

Where S_{eff}^{max} and E_{eff}^{max} denote the maximum of S_{eff} and E_{eff}, respectively.

State transition probability (P)**:** The CGs are modeled as *p+1* states of a Markov chain, where $cg_i \in \{CGa\}_{1\le a\le N}$ (i = 0,1, …, *p*). The state transition probability from CG_m to CG_n at time-slot k is reflected in (7):

$$P_{mn} = P_r\left\{cg_i^{(k)} = CG_n \,\middle|\, cg_i^{(k+1)} = CG_m\right\} \quad (10)$$

Considering the Markov decision-making structure introduced for relay node selection, in this section, a DRL algorithm in the form of the POMDP model is used so that, by utilizing the neural network, the learning agent is able to adaptively learn the optimal data unloading policy under conditions of uncertainty and incomplete information.

3.1. Deep Reinforcement Learning

The optimization problem in this research is formulated as a sequential decision-making process, where a Reinforcement Learning (RL) agent iteratively refines its policy to converge toward an optimal offloading strategy.1 The decision-making framework is rigorously modelled as a Partially Observable Markov Decision Process (POMDP), which effectively captures the inherent uncertainty in the network environment. In this setting, the agent interacts with an environment where the complete system state is not fully observable at each time step. Instead, the agent relies on partial observations to infer the underlying state dynamics and make optimal decisions.

Within the Q-learning framework, the RL agent explores the action space by executing different policies and evaluating the associated rewards. Through continuous interaction with the environment, the agent learns an optimal offloading policy by iteratively updating its state-action value function (Q-function). The fundamental elements of the POMDP-based optimization model include the state space (S), action space (A), reward function (R), and transition dynamics (T).

The state space (S) represents the network conditions, specifically including the Cooperation Gain (CG) provided by different communication modes: direct LTE (CG=0), Direct Wi-Fi (CG=1), and relays (CG=2,…,P). The action space (A) consists of possible offloading decisions such as direct LTE transmission, Wi-Fi-based offloading, or relay-assisted cooperation. Finally, the reward function (R) evaluates the efficacy of each action in terms of throughput and energy efficiency, guiding the agent toward selecting the most optimal data offloading strategy.

DRL in POMDP Framework for Cooperative Offloading: The proposed model leverages DRL within a POMDP framework to address uncertainties in network conditions and dynamic channel variations. Through trial-and-error learning, the agent optimizes offloading performance by

balancing throughput and energy efficiency, enabling real-time adaptation to network fluctuations for enhanced system performance and user experience.

A. State Representation**:** Let (x,y)∈D denote a data sample, where x∈X⊂R^n represents the observed network state and y is the target optimal value. The state space includes:

- A finite set of CG values: F={f1,…,fn}.
- A cost function c: F→R mapping each CGi to a research cost c(fi).

The agent observes a partial state s=(x_i, f_i), where $f_i \in \tilde{F} \subseteq F$ is a subset of available CG values for the current state xi.

B. Action Space: The action space A=$A_C \cup A_f$ consists of:

- Intermediate actions **(A_f):** Select a CG value $f_i \in \tilde{F} \subseteq F$ to optimize offloading.
- Terminal actions **(A_C):** Finalize the offloading decision (e.g., select direct LTE or Wi-Fi transmission).

C. Reward Function: The reward function r: $\tilde{S}$×A→R is defined as:

$$r((x,y,\tilde{F}),a) = \begin{cases} -\lambda \cdot c(fi) & \text{if } a \in A_f \text{ (CG selection),} \\ 0 & \text{if } a \in A_C \text{ and } a=y \text{ (correct decision),} \\ -1 & \text{if } a \in A_C \text{ and } a \neq y \text{ (wrong decision),} \end{cases} \quad (11)$$

Where λ∈ℝ+ balances cost and delay reduction. High λ prioritizes cost reduction and low λ provides delay reduction.

D. State Transitions: The transition function τ: $\tilde{S}$×A→$\tilde{S}$∪{τ} is deterministic:

$$\tau((x,y,\tilde{F}),a) = \begin{cases} \tau & \text{if } a \in A_C \text{ (terminal state),} \\ (x,y,\tilde{F}\backslash\{fi\}) & \text{if } a \in A_f \text{ (continue state).} \end{cases} \quad (12)$$

Reinforcement learning algorithm Improvement: To enhance the performance of the RL algorithm in this study, we employ a policy improvement method based on Q-function approximation using a neural network. The Q-function, denoted as Qπ(s,a), represents the expected cumulative discounted reward when taking action a in state s and subsequently following policy π:

$$Q^{\pi}(s,a) = r\,(s,a) + E\left[\gamma\, Q^{\pi}\big(S,\pi\,(S)\big)\right] \quad (13)$$
$$s \sim t\,(s,a)$$

Here, *r(s,a)* is the immediate reward, and *γ* (the discount factor) determines the importance of future rewards. A terminal state satisfies *Q(τ,0)=0*. Since the target CG values in this work are discrete, the environment guarantees algorithm termination, making *γ* a critical training parameter.

The optimal Q-function Q^* satisfies the Bellman optimality equations:

$$\pi^*\,(s) = \; argmax_a\, Q^*\,(s,a) \quad (14)$$
$$Q^*(s_1 a) = r(s_1 a) + E_{S \sim t\,(s,a)}\,(Y\, max\, Q^*\,(s',a') \quad (15)$$

In small, discrete state spaces, dynamic programming can efficiently compute *Q*. However, for large or continuous state spaces, exact computation becomes infeasible. To address this, we adopt Deep Q-Learning (DQL) [11], where a neural network with weights *θ* approximates *Q* as Q_θ.

Neural Network Training via Experience Replay: The network learns by minimizing the Mean Squared Error (MSE) between the predicted $Q_\theta(s,a)$ and the target *q*, derived from sampled transitions *(s,a,r,s')*. Using a greedy policy π_θ:

$$\pi^\theta(S) = argmax_a Q^\theta(s,a) \tag{16}$$

For a batch of transitions B, the loss function L_θ is:

$$L_\theta(B) = \frac{1}{|B|} \sum_{(s,a,r,s)\epsilon B} (q(r,s) - Q^\theta(s,a))^2 \tag{17}$$

Here, $q(r,s')$ is the target Q-value, held constant during optimization:

$$q(r,s) = r + y\, max\, Q^\theta(s,a) \tag{18}$$

As training progresses, $Q\theta$ converges toward $Q*$, reducing approximation error and improving policy performance. Algorithm 1 illustrates the RL and the policy improvement method based on Q-function approximation using a neural network. Moreover, the flowchart of the proposed method is presented in Fig. 2.

Algorithm 1: Deep Q-Learning-Based Cooperative Data Offloading

Inputs:

- Network environment (UEs, APs, eNB, relays $R_1 ... R_m$)
- Initial Q-network Qθ with random weights θ
- Replay buffer B = ∅
- Hyperparameters: α (learning rate), γ (discount), |B| (batch size)
- Exploration: Initial ϵ, decay rate ϵ_decay
- Training limits: E episodes, T steps per episode

Output: Optimal offloading policy π*: S → A

for episode =1 to E **do**
  Initialize environment; observe partial state $s_0 = (x_0, f_0)$.
  for t=1 to T **do**
    With probability ϵ, select random $a_t \in A$.
    Otherwise, select $a_t = \arg max_a\, Q_\theta\,(s_t, a)$.
    Execute a_t :
      if $a_t \in A_C$:
        if a_t = y: r_t = 0, terminate.
        else: r_t =−1.
        else if $a_t \in A_f$:
          $r_t = -\lambda . c(f_i)$.
    Observe next state s_{t+1}; store (s_t, a_t, r_t, s_{t+1}) in *B*.
    Sample mini-batch from *B*; update Q_θ via gradient descent.
    if terminal state reached: **break**.
  end for
  Decay exploration: ϵ←ϵ×decay_rate.

17: **end for**
18: **return** π*(s)= $\arg max_a\, Q_\theta\,(s,a)$.

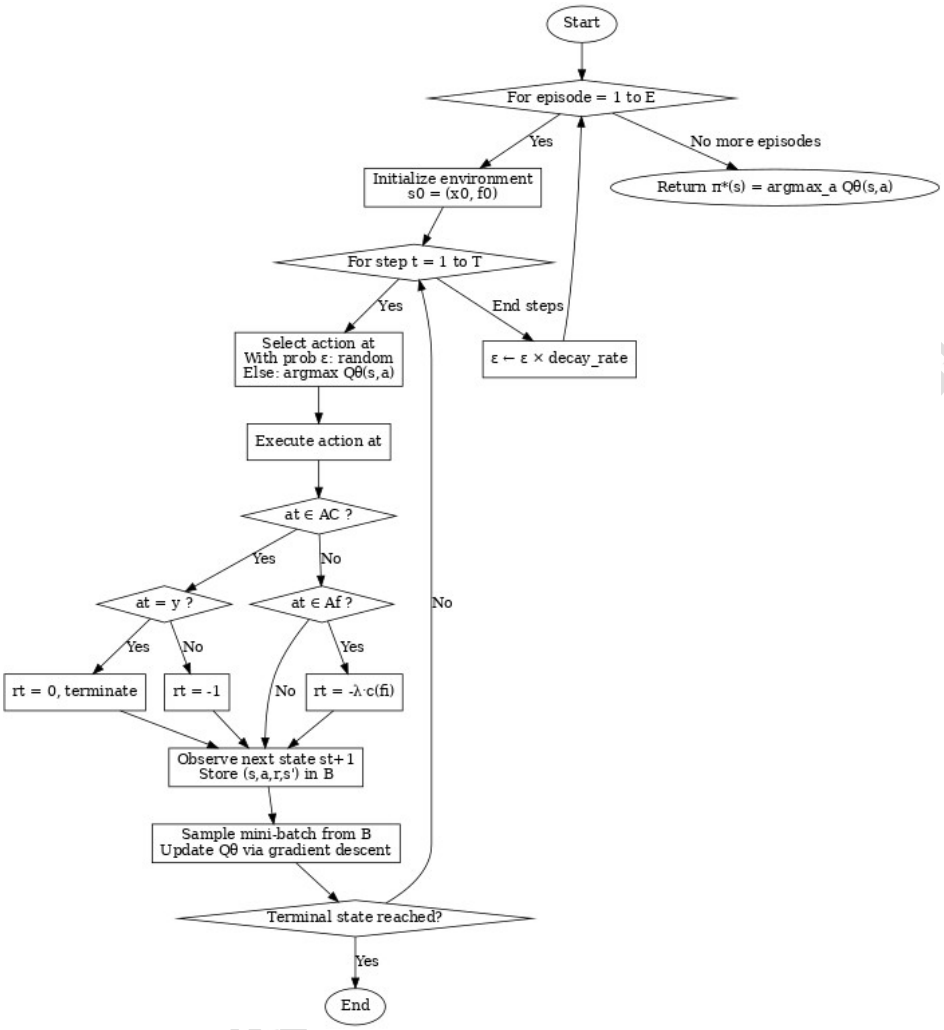

Fig. 2. The flowchart of Deep Q-Learning-Based Cooperative Data Offloading

Results

In order to accurately analyze the impact of cooperative communication on data offloading performance in Wi-Fi-based networks, this research utilizes the IEEE 802.11n standard as the Wi-Fi technology and LTE as the cellular communication platform. The performance evaluation is carried out through a series of extensive simulations in the MATLAB environment, where parameters based on real network conditions are employed to ensure the experimental validity and generalizability of the results.

Moreover, a set of key parameters must be carefully tuned at different levels of the network and the learning agent before running the simulation process. Table 2 details the network settings, including the topology, Wi-Fi, and LTE parameters. Table 3 indicates the parameters related to the DRL learning agent, such as the neural network structure, learning rate, discount rate, exploration policy, and number of training episodes. Finally, Table 4 is dedicated to the precise definition of the state space and action space, which specify the decision-making framework of the learning

agent for selecting the optimal data offloading path. These settings provide the basis for the accurate, repeatable, and comparable evaluation of the proposed method against other reference algorithms.

In this evaluation, four performance metrics are measured: reward convergence, throughput, energy efficiency, and offloading ratio.

Reward Convergence: Fig. 2 illustrates the reward changes of four distinct algorithms during the process of selecting the appropriate path for data offloading within the network. These algorithms include CoopMAC, Heuristic, Classical Q-learning, and the proposed DRL-based algorithm (Proposed-DRL). The Proposed-DRL algorithm converged the fastest and ultimately obtained the highest stable reward value compared to the other methods. This significant advantage is attributed to the utilization of the Deep Q-Network (DQN) structure alongside policy improvement and modelling as a Partially Observable Markov Decision Process (POMDP), which facilitates rapid adaptation to changing network conditions. Moreover, employing a neural network to approximate the Q-function has led to reduced fluctuations and increased accuracy in selecting optimal data transfer paths.

Table 2. Network Settings

Parameter	Value
Number of Users (UEs)	200 – 2000
Number of Wi-Fi Access Points (APs)	12
Number of LTE Base Stations (eNB)	1
Wi-Fi Coverage Radius	40 – 115 meters
Data Generation Rate per UE	0.5 – 2 Mbps
Wi-Fi Standard	IEEE 802.11n
LTE Bandwidth	56.394 Mbps
LTE Cell Coverage	500 meters
Wi-Fi packet payload	1500 bytes

Table 3. DRL Agent Configuration

Parameter	Suggested Value
RL Algorithm Type	Deep Q-Learning (DQN)
Neural Network Structure	2 hidden layers, 64 neurons each
Replay Buffer Size	10,000 samples
Learning Rate (α)	0.001
Discount Factor (γ)	0.95
Initial ε (Exploration Rate)	0.9 with decay
Number of Episodes	500 – 1000
Steps per Episode	100 – 200

Table 4. State and Action Space Design

Parameter	Description
State Space	SNR, RSSI, CG, user density
State Space Type	Continuous / normalized
Action Space	LTE, Wi-Fi direct, Wi-Fi with Relay R_m
Action Space Type	Discrete with M+2 options

In comparison, the Classical Q-learning algorithm, although it demonstrates acceptable performance, exhibits less stability in complex conditions. This is primarily due to its dependence on the Q-table and its inherent inability to generalize in a continuous state space, resulting in a lower reward than DRL. The Heuristic algorithm, which operates based on static and predetermined policies, stalled at approximately 1.5 rewards despite its initial improvement, owing to its fundamental inability to adapt to environmental changes. Finally, the CoopMAC algorithm, which lacks a learning mechanism, showed the lowest performance and consistently exhibited a negative reward with severe fluctuations over time. This clearly indicates the severe weakness of this method in dealing with dynamic conditions and underscores the critical need for intelligent decision-making in selecting data offloading paths.

Overall, the results of Fig. 2 demonstrate that utilizing a DRL algorithm in the target network provides significantly superior performance compared to other methods, establishing it as a suitable model for advanced applications such as 5G networks, IoT, and industrial systems.

Fig. 4 illustrates the average cumulative reward received by the four algorithms—CoopMAC, Heuristic, Q-learning, and the Proposed-DRL—across the entire training process as a bar chart. These averages reflect the final efficacy of each algorithm in selecting optimal data offloading paths under the dynamic conditions of the network.

The results presented in this Fig. clearly confirm the significant superiority of the proposed DRL algorithm, which achieved the most optimal performance with an average reward of 3.41. This success stems from the utilization of the deep learning structure and the ability to capture complex relationships between environmental parameters such as cooperation rate, channel quality, and user density. This capability enables the learning agent to make decisions that lead to increased efficiency and reduced energy costs and latency.

The Q-learning algorithm, with an average reward of 2.28, shows that classical reinforcement learning can improve performance to some extent. However, its inherent limitations in generalizability in environments with a large and continuous state space prevent it from reaching the performance level of DRL. The Heuristic method ranks third with an average reward of 0.7, a result of its decisions being based on fixed rules that are not adaptable to environmental changes. In sharp contrast, the basic CoopMAC method had the weakest performance with a negative average reward of −3.55. This clearly shows that the absence of a learning mechanism and the

failure to account for dynamic and time-varying network conditions lead to inefficient decisions and a consequent decrease in overall efficiency.

Overall, the analysis of Fig. 4 complements the results of Fig. 2 and demonstrates that the proposed DRL algorithm outperforms other methods not only in convergence speed but also in final reward rate and stability. This compelling evidence confirms its suitability for deployment in real networks with variable environments, including 5G, IoT, and industrial networks.

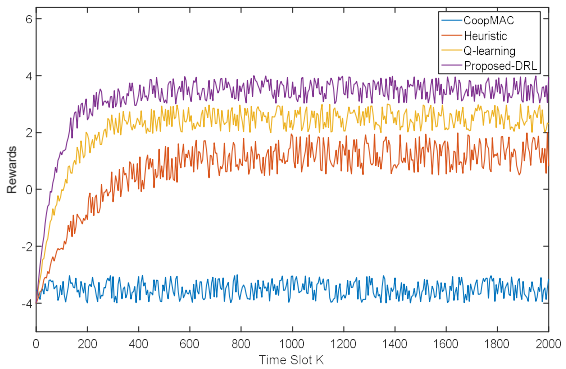

Fig.3. Comparison of cumulative reward of different algorithms over time for choosing the data discharge path in the network.

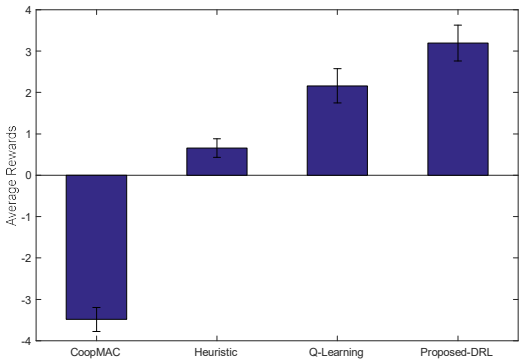

Fig.4. Comparison of average rewards of different algorithms in data dumping in the desired network

Throughput: Fig. 5 examines and compares the performance of different algorithms in optimizing energy efficiency over time. In this graph, the horizontal axis represents the number of time slots (K) that the learning agent operates in the network environment, and the vertical axis represents the Energy Efficiency value—the ratio of transmitted data to the energy consumed during the data offloading process.

The results show that the proposed DRL algorithm has been able to achieve the highest level of energy efficiency in a relatively short time and maintain this value stably at a level close to 2.5. This significant improvement is the result of the decision-making model design, which allows the learning agent to choose paths that, in the face of dynamic network changes, not only maintain the Quality of Service (QoS) but also minimize energy consumption. In this method, the utilization of the Cooperation Gain (CG) as a decision input, in combination with the POMDP structure, enables the agent to make an optimal choice between direct LTE, Wi-Fi direct, or relay cooperation routes. The Q-learning algorithm also shows that traditional reinforcement methods are somewhat effective in optimizing energy consumption, achieving a relatively favorable performance with an efficiency level of 2.05. However, the limitation in fully understanding the environmental conditions and the continuous state space prevents it from reaching the efficiency of the proposed algorithm. In contrast, the Heuristic method, with an efficiency of 1.56, shows improvement over CoopMAC but does not provide stable optimization due to the lack of adaptation to environmental changes. The weakest performance belongs to the CoopMAC algorithm, which, with an efficiency value of

1.21, demonstrates that the lack of a learning mechanism and intelligent route selection leads to a waste of energy resources, particularly in high-traffic or high-density scenarios.

Overall, Fig. 5 confirms the fact that the proposed DRL algorithm not only outperforms other algorithms in terms of decision accuracy and convergence speed but also in terms of energy efficiency. This feature makes it a suitable choice for application in 5G infrastructures, industrial smart networks, and energy-constrained IoT systems.

In Fig. 6, the average throughput of the four different algorithms—CoopMAC, Heuristic, Q-learning, and the Proposed-DRL—is compared in a bar chart. This metric indicates the ability of the algorithm to effectively utilize data transmission resources to achieve maximum network efficiency in the data offloading process. According to the graph, the Proposed-DRL algorithm has performed significantly better than other algorithms by achieving an average efficiency of 2.24. This success is the result of the intelligent selection of data offloading paths and the utilization of the POMDP structure to adapt to changing network conditions. Moreover, the utilization of the CG index in decision-making has enabled the learning agent to select paths with an appropriate balance between delay and transmission efficiency.

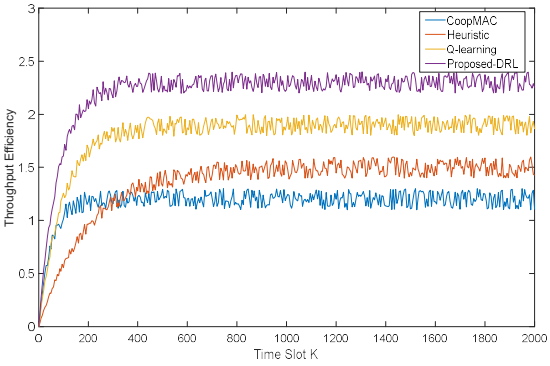

Fig. 5. Comparison of Throughput Efficiency of different algorithms in the data offloading process in the desired network.

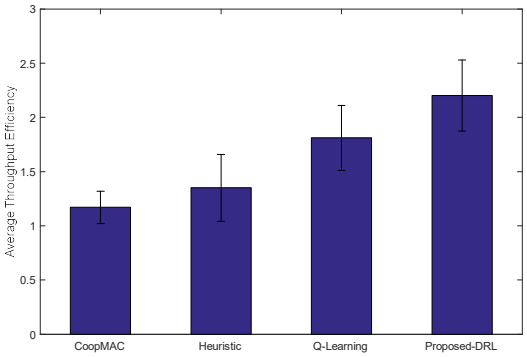

Fig. 6. Comparison of average Throughput Efficiency of different algorithms in the target network

The Q-learning algorithm ranks second with an average of 1.84, indicating that classical Reinforcement Learning (RL) is also capable of improving throughput performance, but its level is lower than DRL due to limitations in generalizability to new conditions. The Heuristic method ranks third with an average of 1.3, which is superior to CoopMAC but exhibits limitations in efficiency owing to its lack of learning and adaptation to dynamic network conditions. The CoopMAC algorithm, operating without a learning mechanism, shows the lowest throughput with an average of 1.2. This poor performance stems from its inability to adapt to user density, channel conditions, and traffic distribution, resulting in sub-optimal resource allocation and reduced network efficiency.

In conclusion, Fig. 6 clearly demonstrates that the proposed DRL algorithm, owing to its deep learning structure and intelligent decision-making capability at low levels of the network, offers the highest level of efficiency in terms of throughput for data offloading. This feature establishes it as

a prime choice for utilization in applications with high requirements for Quality of Service (QoS) and network efficiency, such as 5G networks and industrial communications.

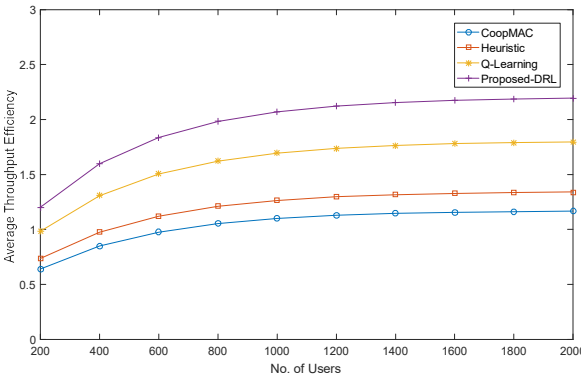

Fig. 7. Average Throughput Efficiency with different numbers of Users

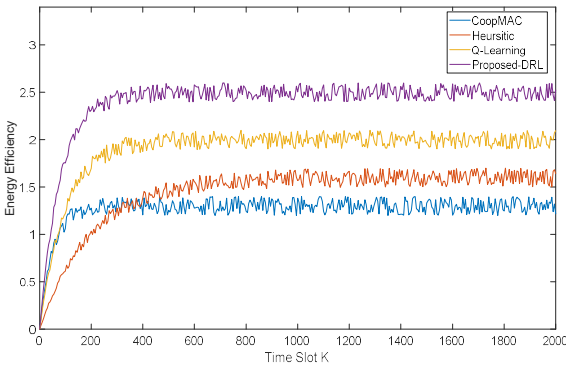

Fig.8. The trend of Energy Efficiency change of different algorithms over time in the network under consideration.

Fig. 7 illustrates the performance comparison of the four different algorithms under varying user numbers. The proposed DRL-based algorithm (Proposed-DRL) consistently outperforms the Q-Learning, Heuristic, and CoopMAC algorithms across the entire user range, from 200 to 2000.

In particular, the DRL approach, owing to its POMDP structure and adaptive learning via a deep neural network, is uniquely able to adapt in real-time to dynamic changes in user density and channel conditions. This feature results in a continuous increase in throughput, reaching 2.23 under conditions of maximum user density.

Unlike DRL, the Q-Learning algorithm, despite showing a relative improvement (up to 1.8), does not reach the performance level of DRL due to structural limitations in modelling continuous states and a lack of full adaptability. The Heuristic and CoopMAC methods remain constrained at efficiency levels of 1.4 and 1.2, respectively, due to the absence of a learning and adaptation mechanism.

The results of this Fig. ultimately confirm that the utilization of DRL combined with multi-hop cooperation (Cooperative Relaying) at the MAC layer leads to maximizing the use of network capacity and ensuring Quality of Service (QoS) in dense next-generation networks.

Energy Efficiency: Fig. 8 illustrates the energy efficiency trend for four different algorithms—CoopMAC, Heuristic, Q-Learning, and the Proposed-DRL—over a period of 2000 time slots. This metric indicates the efficacy of the algorithms in consuming energy efficiently relative to the transmitted data, a factor especially important in resource-constrained networks such as IoT or battery-powered devices. The performance of the Proposed-DRL algorithm is significantly superior to other methods, reaching a stable level of close to 2.5 from approximately slot 200 onwards. This high stability, combined with the optimal energy efficiency value, highlights the ability of DRL to intelligently select data transmission paths with the lowest energy consumption. The learning agent

in this model utilizes parameters such as Cooperation Gain (CG), channel quality, and traffic load to select routes that minimize energy consumption while ensuring Quality of Service (QoS).

In comparison, the Q-Learning algorithm maintains an acceptable energy efficiency but lags behind DRL due to its reliance on the Q-table and its inability to generalize to the continuous state space. The Heuristic algorithm ranks lower, with an efficiency of 1.61, as its lack of learning from the environment prevents the routes from being consistently optimal. The CoopMAC algorithm, operating without any learning or adaptation mechanism, recorded the lowest energy efficiency of 1.31. The high fluctuations in this method also indicate instability in route selection and data transmission scheduling decisions.

Overall, Fig. 8 confirms that the utilization of the DRL algorithm leads to a significant optimization of energy consumption in the target network, making this algorithm a suitable option for implementation in environments with limited energy resources, such as wireless sensor networks, IoT, and industrial applications.

Fig. 9 shows the average energy efficiency of the four algorithms in the form of a bar chart. This chart summarizes the final performance of each algorithm in terms of energy consumption relative to the transmitted data over the entire training process, a critical criterion in high-user or energy-constrained networks.

According to the results, the Proposed-DRL algorithm, with an average energy efficiency of 2.49, outperforms other methods and is able to provide the most optimal utilization of energy resources while maintaining QoS. This performance is attributed to the ability of deep learning to identify the network state, intelligently select transmission paths, and utilize the decision-making structure based on POMDP and the CG index.

In second place, the Q-Learning algorithm (average efficiency of 1.91) shows that traditional RL can positively affect energy consumption, but its limitations in modelling complex conditions and lack of generalizability place it at a lower level than DRL. The Heuristic method, with a value of 1.45, performs better than CoopMAC but still has limited efficiency due to its lack of learning ability. The weakest performance belongs to CoopMAC, which only reached an average efficiency of 1.33. This low performance clearly indicates the inefficiency of non-learning methods in optimizing energy consumption in complex and highly volatile network environments.

In conclusion, Fig. 9 complements the results of Fig. 7 (implied reference, likely meant to be Fig. 8) and once again emphasizes the superiority of the proposed DRL algorithm in the energy efficiency criterion. This advantage makes the Proposed-DRL a suitable choice for applications such as wireless sensor networks, IoT, and energy-constrained industrial systems.

Fig. 10 compares the average throughput of the four different algorithms in a bar chart. This metric indicates the algorithm's ability to effectively utilize data transmission resources to achieve maximum network efficiency in data offloading.

The Proposed-DRL algorithm, with an average throughput of 2.23, performed significantly better than other algorithms. This success is the result of intelligent path selection and the utilization of

the POMDP structure to adapt to changing network conditions. The Q-learning algorithm ranks second (average of 1.91), while the Heuristic (average of 1.3) and CoopMAC (average of 1.2) methods trail due to their lack of learning and adaptability.

In conclusion, Fig. 10 clearly shows that the proposed DRL algorithm, due to its deep learning structure and intelligent decision-making capability at low network levels, offers the highest level of efficiency in terms of throughput. This feature establishes it as a prime choice for utilization in applications with high requirements for QoS and network efficiency, such as 5G networks and industrial communications.

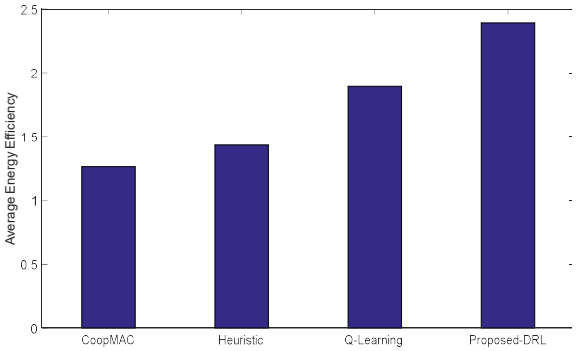

Fig. 9. Comparison of the average Energy Efficiency of different algorithms

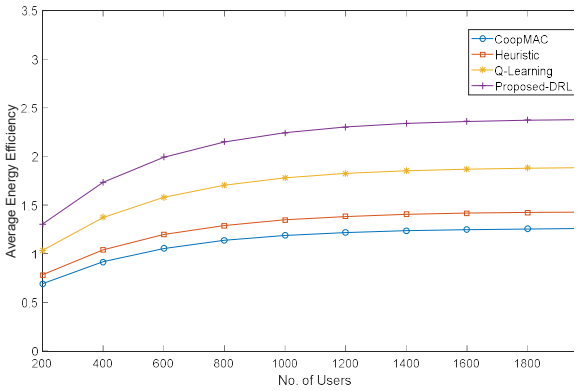

Fig. 10. Comparison of the average Energy Efficiency of different number of users

Offloading Ratio: Fig. 11 examines the changes in the Offloading Ratio against the number of network users, providing a comparison between the four algorithms: CoopMAC, Heuristic, Q-Learning, and Proposed-DRL, across the range of 200 to 2000 users. The results show that the proposed DRL algorithm, by utilizing real-time decision-making based on POMDP and incorporating the Cooperation Gain (CG) index within the MAC cooperation context, has been able to achieve the highest possible offloading ratio with a continuous and ascending trend. This superior performance is observed while the Q-Learning and Heuristic algorithms remain at lower levels, and CoopMAC, which operates without a learning mechanism, shows the lowest offloading ratio (below 1.1). The stable performance and gradual improvement of DRL across all heavy and dense load scenarios clearly demonstrate the importance of applying DRL in simultaneously optimizing energy and data offload capacity in next-generation 5G and IoT architectures.

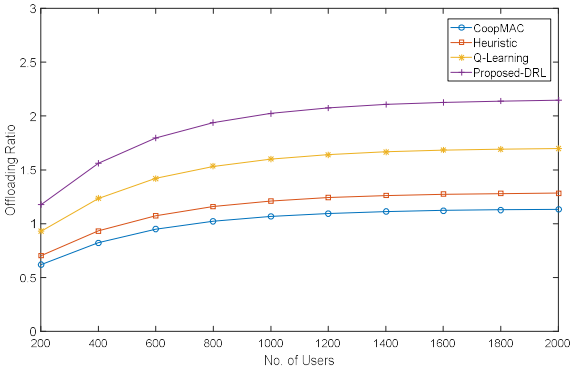

Fig. 11. Offloading ratio with different number of Users

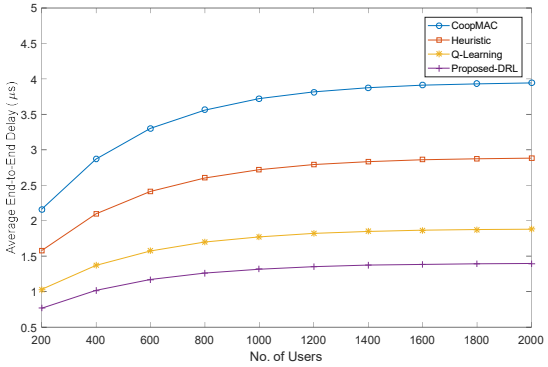

Fig. 12. Average End-to-End delay with different number of Users

Average End-to-End Delay: Fig. 12 illustrates the variation of Average End-to-End Delay (μs) as the number of network users increases from 200 to 2000, comparing the performance of CoopMAC, Heuristic, Q-Learning, and the Proposed-DRL algorithm. The results indicate that the Proposed-DRL algorithm consistently achieves the lowest delay across all user scenarios. By exploiting Deep Reinforcement Learning (DRL) and adaptive decision-making, it maintains stable and efficient performance even under heavy network loads. In comparison, Q-Learning and Heuristic achieve moderate delay reductions, with Q-Learning slightly outperforming Heuristic. CoopMAC, lacking a learning mechanism, records the highest delay values, exceeding 4μs as the number of users grows. These findings confirm the ability of the Proposed-DRL to minimize latency in dense networks, highlighting its potential for delay-sensitive applications.

Discussion and Future Work: In this paper, a novel framework based on Deep Reinforcement Learning (DRL) and cooperative communication was successfully proposed at the MAC layer of the IEEE 802.11n standard. By leveraging intelligent relay selection and optimized resource allocation, the framework achieved significant improvements in key performance indicators, namely throughput, energy efficiency, offloading ratio, and end-to-end delay. Despite these contributions, several limitations remain, which open important avenues for future research.

First, although the evaluation incorporated four major performance metrics, other important indicators, such as the Bit Error Rate (BER), were not explicitly addressed. Since BER is closely tied to physical-layer aspects such as modulation and coding, its assessment requires detailed PHY-layer modelling and simulation, which was beyond the scope of this work. Including BER in future investigations would, however, enable a more holistic evaluation of the framework.

Second, the individual energy consumption of relays was not separately modelled, as the focus was placed on overall network-level energy efficiency. Extending the analysis to incorporate relay-specific energy models could provide a more detailed and realistic understanding of the operational cost of cooperative communication.

Third, with respect to offloading feasibility, this study was limited to MATLAB-based simulations. However, the MAC-layer design ensures compatibility with existing Wi-Fi devices. In practice, the proposed framework could be implemented through software or firmware updates without requiring

fundamental hardware modifications. Furthermore, the computationally intensive DRL training phase can be executed offline on cloud or edge servers, while only the lightweight inference is deployed at network nodes, enabling real-time operation with negligible overhead. This architectural design highlights the adaptability of the framework to current infrastructures and its strong potential for practical deployment.

Fourth, although preliminary results confirmed that the framework is not highly sensitive to variations in the discount factor (γ), conducting a more detailed sensitivity analysis would provide deeper insights into convergence stability and long-term adaptability. Moreover, extending the framework utilizing advanced approaches such as Multi-Agent Reinforcement Learning (MARL) or Federated Learning (FL) could further enhance scalability and resilience in distributed and dynamic environments.

Finally, future work will focus on extending the evaluation to include BER, integrating relay-specific energy modelling, comparing against more advanced DRL-based approaches, and validating the system on real-world testbeds. These directions will help bridge the gap between simulation and practical deployment, thereby reinforcing the applicability of the proposed framework in next-generation 5G, 6G, and IoT scenarios.

Conclusion

This paper proposes a Deep Reinforcement Learning (DRL) framework based on Partially Observable Markov Decision Processes (POMDP) for the joint optimization of data offloading paths, relay node selection, and access point placement in wireless networks. The key innovation lies in combining the IEEE 802.11n cooperative MAC layer with DRL, where decisions are made utilizing both observed network parameters and real-time metrics such as Cooperation Gain, channel status, and user density. By accurately defining the state, action, and reward spaces, and leveraging Deep Neural Networks (DNNs) to approximate the Q-function, the model ensures scalable and consistent performance in dynamic environments. Unlike traditional methods such as Q-Learning, which suffer from slow convergence and limited generalizability, the proposed approach utilizes experience replay and gradient-based optimization for fast and stable learning. Simulation results in MATLAB demonstrate that the proposed DRL framework significantly (p-value <0.00005) outperforms baseline algorithms such as CoopMAC, threshold-based schemes, and classic Q-Learning in terms of energy efficiency, throughput, and cumulative rewards.

Reference

[1] R. Yan, Z. Guo, P. Liu, Q. Lan, X.-P. Zhang, and Y. Dong, "Multi-Agent Reinforcement Learning Based Channel Access Optimization for IEEE 802.11 bn," *IEEE Transactions on Green Communications and Networking*, 2024. DOI: https://doi.org/ 10.1109/TGCN.2024.3495236

[2] Y. Yang, and S. Yan, "Joint Throughput Maximization and Energy Management for Ultra-low Power Ambient Backscatter Communication in WBANs by Distributed Deep Reinforcement Learning," *IEEE Sensors Journal*, 2024. DOI: https://doi.org/10.1109/JSEN.2024.3487354.

[3] A. A. Chirani, "Network reconfiguration and optimal distributed generations allocation with whale optimizer algorithm," *Majlesi Journal of Electrical Engineering,* vol. 19, no. 1 (March 2025), pp. 1-12, 2025. DOI: https://doi.org/10.57647/j.mjee.2025.1901.04

[4] S. Song, Z. Zhang, Q. Wu, P. Fan, and Q. Fan, "Joint optimization of age of information and energy consumption in NR-V2X system based on deep reinforcement learning," *Sensors,* vol. 24, no. 13, pp. 4338, 2024, DOI: https://doi.org/10.3390/s24134338.

[5] M. Alizadeh Aliabadi, M. Karimi, Z. Karimi, and M. soheili Fard, "The Effect of Photoplethysmography Signal Denoising on Compression Quality," *IRANIAN JOURNAL OF ELECTRICAL AND ELECTRONIC ENGINEERING,* vol. 21, no. 1, pp. 3277-3277, 2025, DOI: https://doi.org/10.22068/IJEEE.21.1.3277

[6] M. J. Kadhim, R. Sadeghi, A. S. Abdalrada, B. Arandian, and R. Khorsand, "Performance improvement of data offloading using Krill herd optimization algorithm," *Majlesi Journal of Electrical Engineering,* vol. 19, no. 1 (March 2025), pp. 17-17, 2025, DOI: https://doi.org/10.57647/j.mjee.2025.1901.05

[7] A. Khoshnoudi, R. Sadeghi, and F. Faghani, "Performance Improvement of Data Offloading using Multi-rate IEEE 802.11 WLAN," *Majlesi Journal of Electrical Engineering,* vol. 13, no. 1, pp. 121-126, 2019.

[8] N. Dawar, K. N. Nguyen, A. Sehgal, Y. Zhu, B. L. Ng, and J. Choi, "Enhancing Wi-Fi 7: Traffic Flow Intelligence and Multi-Link Operation for Optimal Efficiency," *IEEE Access*, 2025, DOI: https://doi.org/10.1109/ACCESS.2025.3557435

[9] A. A. Alaidany, and M. M. Mahdi, "A Review of IoT-Based Wearable Sensor Systems for Healthcare Monitoring., DOI:No DOI

[10] M. Talebkhah, A. Sali, V. Khodamoradi, T. Khodadadi, and M. Gordan, "Task offloading for edge-IoV networks in the industry 4.0 era and beyond: a high-level view," *Engineering Science and Technology, an International Journal,* vol. 54, pp. 101699, 2024, DOI: https://doi.org/10.1016/j.jestch.2024.101699

[11] Z. Zabihi, A. M. Eftekhari Moghadam, and M. H. Rezvani, "Reinforcement learning methods for computation offloading: a systematic review," *ACM Computing Surveys,* vol. 56, no. 1, pp. 1-41, 2023, DOI: https://doi.org/10.1145/3603703

[12] M. Harouni, M. Karimi, A. Nasr, H. Mahmoudi, and Z. Arab Najafabadi, "Health monitoring methods in heart diseases based on data mining approach: A directional review," *Prognostic models in healthcare: Ai and statistical approaches*, pp. 115-159: Springer, 2022, DOI: https://doi.org/10.1007/978-981-19-2057-8_5

[13] E. T. Garmaserh, and M. Emadi, "Improving the criteria of electricity consumptionforecasting in petrochemical industrial units based ondeep learning," *Majlesi Journal of Electrical Engineering,* vol. 19, no. 2 (June 2025), 2025, DOI: https://doi.org/10.57647/j.mjee.2025.1902.41

[14] S. S. S. Abolghasemi, M. Emadi, and M. Karimi, "Accuracy improvement of breast tumor detection based on dimension reduction in the spatial and edge features and edge structure in the image," *Majlesi Journal of Electrical Engineering,* vol. 18, no. 1, 2024, DOI: https://doi.org/10.30486/mjee.2023.1991110.1174

[15] B. Kar, W. Yahya, Y.-D. Lin, and A. Ali, "Offloading using traditional optimization and machine learning in federated cloud–edge–fog systems: A survey," *IEEE Communications Surveys & Tutorials,* vol. 25, no. 2, pp. 1199-1226, 2023, DOI: https://doi.org/10.1109/COMST.2023.3239579

[16] S. Dong, J. Tang, K. Abbas, R. Hou, J. Kamruzzaman, L. Rutkowski, and R. Buyya, "Task offloading strategies for mobile edge computing: A survey," *Computer Networks*, pp. 110791, 2024, DOI: https://doi.org/10.1016/j.comnet.2024.110791

[17] R. Chaari, O. Cheikhrouhou, A. Koubâa, H. Youssef, and T. N. Gia, "Dynamic computation offloading for ground and flying robots: Taxonomy, state of art, and future directions," *Computer Science Review,* vol. 45, pp. 100488, 2022, DOI: https://doi.org/ 10.1016/j.cosrev.2022.100488

[18] Y. Li, G. Su, P .Hui, D. Jin, L. Su, and L. Zeng, "Multiple mobile data offloading through delay tolerant networks." pp. 43-48., 2011, DOI: https://doi.org/10.1145/2030652.2030665

[19] S. Andreev, A. Pyattaev, K. Johnsson, O. Galinina, and Y. Koucheryavy, "Cellular traffic offloading onto network-assisted device-to-device connections," *IEEE Communications Magazine,* vol. 52, no. 4, pp. 20-31, 2014, DOI: https://doi.org/10.1109/MCOM.2014.6807943

[20] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can WiFi deliver?," *IEEE/ACM Transactions on networking,* vol. 21, no. 2, pp. 536-550, 2012, DOI: https://doi.org/10.1109/TNET.2012.2218122

[21] C. P. Mayer, and O. P. Waldhorst, "Offloading infrastructure using delay tolerant networks and assurance of delivery." pp. 1-7, 2011, DOI: https://doi.org/10.1109/WirelessDays.2011.6134105

[22] X. Wang, M. Chen, Z. Han, D. O. Wu, and T. T. Kwon, "TOSS: Traffic offloading by social network service-based opportunistic sharing in mobile social networks." pp. 2346-2354, DOI: https://doi.org/10.1109/INFOCOM.2014.6848179

[23] A. Anagnostopoulos, R. Kumar, and M. Mahdian, "Influence and correlation in social networks." pp. 7-15 ,2008, DOI: https://doi.org/10.1145/1401890.1401897

[24] J. Y. Ryu, J. Lee, and T. Q. Quek, "Confidential cooperative communication with trust degree of potential eavesdroppers," *IEEE Transactions on Wireless Communications,* vol. 15, no. 6, pp. 3823-3836, 2016, DOI: https://doi.org/ 10.1109/TWC.2016.2530058

[25] N. Magaia, Z. Sheng, P. R. Pereira, and M. Correia, "REPSYS: A robust and distributed incentive scheme for collaborative caching and dissemination in content-centric cellular-based vehicular delay-tolerant networks," *IEEE Wireless Communications,* vol. 25, no. 3, pp. 65-71, 2018, DOI: https://doi.org/10.1109/MWC.2018.1700284

[26] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications,* vol. 11, no. 6, pp. 1991-1995, 2012, DOI: https://doi.org/10.1109/TWC.2012.041912.110912

[27] L. Valerio, R. Bruno, and A. Passarella, "Adaptive data offloading in opportunistic networks through an actor-critic learning method." pp. 31-36, DOI: https://doi.org/10.1145/2645672.2645676

[28] F. Rebecchi, M. D. De Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: A survey," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 2, pp. 580-603, 2014, DOI: https://doi.org/10.1109/COMST.2014.2369742

[29] G. Gao, M. Xiao, J. Wu, K. Han, and L. Huang, "Deadline-sensitive mobile data offloading via opportunistic communications." pp. 1-9, 2017, DOI: https://doi.org/10.1109/TPDS.2017.2720741

[30] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments." pp. 244-251, DOI: https://doi.org/10.1145/1080139.1080142

[31] S. G, K. K, and V. M, "A novel task offloading model for IoT: enhancing resource utilization with actor-critic-based reinforcement learning," *Earth Science Informatics,* vol. 18, no. 3, pp. 266, 2025/02/17, 2025, DOI: https://doi.org/ 10.1007/s12145-025-01773-5

[32] C. Liu, H. Wang, M. Zhao, J. Liu, X. Zhao, and P. Yuan, "Dependency-aware online task offloading based on deep reinforcement learning for IoV," *Journal of Cloud Computing,* vol. 13, no. 1, pp. 136, 2024, DOI: https://doi.org/ 10.1186/s13677-024-00701-0