Federated Deep Reinforcement Learning for Optimal Resource Allocation in Vehicular Edge Computing

Reyadath Ullah[1,*], Ittehad Chowdhury[1,*], Md. Abdur Razzaque[1], Palash Roy[1,2], and Mohammad Mehedi Hassan[3]

[1] Green Networking Research Group, Department of Computer Science and Engineering, University of Dhaka, Dhaka 1000, Bangladesh
[2] Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka, Bangladesh
[3] Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

Abstract. Federated Learning (FL) in Vehicular Edge Computing (VEC) plays a key role in the development of Intelligent Transportation Systems (ITS), enhancing real-time decision-making and preserving data privacy in connected vehicle networks. However, the dynamic and resource constrained nature of VEC, coupled with non-IID data distributions, presents significant challenges to the effective deployment of FL. Current solutions often use static or greedy scheduling methods that lack adaptability, resulting in suboptimal learning performance, inefficient energy utilization, and increased latency. This study proposes a novel Data Aware Deep Reinforcement Learning (DRL) algorithm, namely DA-FeDRL, which utilizes an improved Twin Delayed Deep Deterministic Policy Gradient (TD3) method to optimize device scheduling while achieving optimal resource allocation by balancing data diversity, energy efficiency, and latency optimization. A key contribution of this work is the development of a reward function that jointly minimizes energy consumption and time delay while integrating Data Diversity metrics to enhance learning performance. By expanding the state representation to include crucial factors like device energy, data diversity, and network conditions, the proposed algorithm dynamically adapts to varying network conditions, providing a scalable and robust solution for FL in VEC.

Keywords: Federated learning, Vehicular edge computing, Deep reinforcement learning, Intelligent transportation systems, Data diversity

1 Introduction

The proliferation of data-generating devices has driven the need for innovative distributed machine-learning approaches. Traditional centralized learning, which aggregates data at a central server, faces challenges like data privacy, network

[*] These authors contributed equally to this work

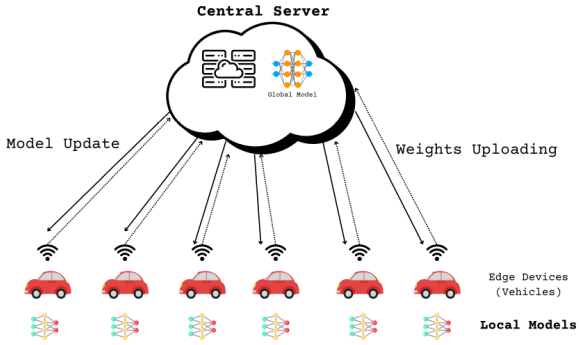


Fig. 1. Federated learning framework

congestion, and latency [1, 2]. Federated Learning (FL) addresses these issues by enabling decentralized model training through locally computed updates, reducing data transfer, and preserving privacy [3]. FL has gained traction across domains where privacy, security, and efficient distributed computation are critical.

In the context of Vehicular Edge Computing (VEC), FL holds significant promise for advancing Intelligent Transportation Systems by facilitating real-time decision-making within connected vehicle networks. VEC environments, characterized by high mobility and distributed data generation on numerous edge devices, present unique opportunities and complex challenges for FL. Unlike conventional static edge scenarios, VEC involves dynamic and resource-constrained conditions, leading to fluctuating network states, heterogeneous data distributions, and constrained energy and bandwidth resources among edge devices. The non-independent and identically distributed (non-IID) nature of data in VEC networks further complicates the learning process, impacting model convergence and overall system performance. As illustrated in Fig. 1, a federated learning framework allows edge devices to collaboratively train models by exchanging model updates instead of raw data, thereby enhancing data privacy and reducing network congestion. Existing approaches in FL for VEC often focus on optimizing communication efficiency [4] and energy consumption [5, 6] but fall short of addressing the dynamic nature of vehicular networks. Many rely on static or simplistic scheduling techniques that fail to adapt to changing conditions, resulting in inefficiencies and suboptimal learning performance. The challenge lies in designing a system that optimizes resource allocation and adapts to varying states of data distribution, energy availability, and network conditions.

To this end, we propose a novel approach leveraging Deep Reinforcement Learning (DRL) with an improved Twin Delayed Deep Deterministic Policy Gradient (TD3) framework to optimize device scheduling and resource allocation in federated VEC systems. By incorporating an expanded state representation, including data diversity, remaining energy, and network conditions, the proposed

solution dynamically adapts to network variability. A key contribution of this work is formulating a reward function that jointly minimizes energy consumption and latency while integrating data diversity metrics to enhance the Data Diversity Factor. This adaptive approach aims to improve the scalability and robustness of FL in complex, real-world VEC scenarios, thereby enhancing its potential for deployment in next-generation Intelligent Transportation Systems.

2 Related Works

Task offloading serves as a foundational concept in the evolution of Multi-access Edge Computing (MEC), facilitating the efficient distribution of computation-intensive tasks from resource-constrained devices to more capable edge servers [7] [8]. One of the earliest studies in this area demonstrated the potential of offloading tasks to improve performance and energy efficiency for large-scale Internet of Things (IoT) setups by introducing an innovative offloading framework with adaptive resource scaling [9]. Similarly, intelligent task offloading strategies have been tailored for specific domains, such as the Maritime Internet of Things (M-IoT), where a reinforcement learning-based approach was employed to optimize task allocation across integrated edge, terminal, and cloud resources [10]. To address latency-sensitive environments with fluctuating channel conditions, model-free reinforcement learning techniques have also been explored for dynamic task offloading.

Building on the advancements of task offloading, Edge Computing (EC) emerged as a critical paradigm to bring computation closer to data sources, thereby reducing latency, conserving bandwidth, and enhancing the responsiveness of real-time applications [11], [12]. The integration of MEC with wireless power transfer technologies has further expanded the potential of edge computing by enabling energy sustainability in low-power IoT networks through optimized resource allocation schemes. Such studies highlight the importance of efficient computation and resource management in edge environments, especially in scenarios constrained by energy and latency requirements.

Within the context of Vehicular Edge Computing (VEC), Federated Learning (FL) has been recognized as a transformative approach to decentralized model training, allowing vehicles and roadside units (RSUs) to collaboratively train models using their local data while preserving privacy and minimizing communication overhead [13]. The decentralized nature of FL in VEC not only mitigates data privacy concerns but also reduces the burden on centralized servers, making it highly suitable for intelligent transportation applications. However, the effectiveness of FL in VEC is often constrained by the dynamic and resource-limited nature of vehicular networks, characterized by non-IID data distributions and varying network states.

Recent works have focused on optimizing communication and resource allocation in federated VEC systems. For example, novel algorithms have been developed to jointly manage bandwidth allocation and user scheduling, minimizing energy consumption and communication delays while ensuring effective

model training [14]. Other studies have emphasized the importance of data-aware scheduling to enhance learning performance by prioritizing devices with more informative and diverse datasets [15]. These approaches highlight the need for adaptive and efficient strategies to manage resource-constrained vehicular networks, where heterogeneity in device capabilities and network conditions presents unique challenges.

In addressing these challenges, the integration of deep reinforcement learning (DRL) into federated learning (FL) frameworks has gained traction as a promising solution for dynamic resource optimization. The TD3 algorithm has been utilized to optimize task offloading in vehicular edge environments, enhancing performance by reducing latency and energy consumption [20]. DRL-based methods, such as the FL-TD3 framework, leverage continuous state and action spaces to optimize device scheduling and resource allocation in real-time, adapting to changing network conditions to maximize learning performance while minimizing energy consumption [16]. Another application of the TD3 algorithm is in urban air mobility (UAM) systems, where it has been used to optimize computation offloading and resource allocation, achieving notable reductions in delay and energy consumption through the integration of federated learning and reinforcement learning approaches [21]. However, DRL methods often struggle with non-IID data distributions prevalent in vehicular networks and typically do not account for communication time delays, which can impact model convergence and system efficiency. Furthermore, incorporating active learning concepts into FL has proven effective in improving data diversity, allowing more efficient and representative model training across diverse vehicular data sets [17]. By balancing resource allocation, communication efficiency, and learning performance, these advanced FL strategies pave the way for robust, scalable solutions in complex VEC environments.

3 System Models

3.1 Time Model

The total time consumption for each edge device in a communication round consists of the time for local model training and the time for uploading the trained model. Specifically, the time required by an edge device i for local model training is given by:

$$\tau_{t,i}^{\text{train}} = \frac{Lc_i D_{t,i}}{f_i}, \tag{1}$$

where, L denotes the number of epochs for Federated Learning (FL) local model training, c_i is the number of CPU cycles required to process one unit of data, $D_{t,i}$ represents the dataset size of device i at time t, and f_i is the computation capacity of the edge device in CPU cycles per second.

For communication with the edge server, each edge device has an uplink transmit rate given by:

$$r_{t,i}^{\text{up}} = b_{t,i} \log_2 \left(1 + \frac{P_{t,i} G_{t,i}}{N_0 b_{t,i}}\right), \tag{2}$$

where, $b_{t,i}$ is the bandwidth allocated to the device, $P_{t,i}$ denotes the transmit power, $G_{t,i}$ is the channel gain, and N_0 represents the Gaussian noise power. Using this rate, the time required to upload local model parameters of size $\mathcal{S}$ is:

$$\tau_{t,i}^{\text{up}} = \frac{\mathcal{S}}{r_{t,i}^{\text{up}}}. \tag{3}$$

Thus, the total time for one communication round for edge device i is:

$$\tau_{t,i}^{\text{total}} = \tau_{t,i}^{\text{train}} + \tau_{t,i}^{\text{up}}. \tag{4}$$

This total time encapsulates the local computation time and the communication delay, both of which are crucial for optimizing the overall FL system efficiency in dynamic edge environments.

3.2 Energy Model

The energy consumption of an edge device during a communication round comprises the energy used for local model training and the energy required for data transmission. The energy consumption for local model training at edge device i is expressed as:

$$E_{t,i}^{\text{cmp}} = L\zeta_i c_i D_{t,i} f_i^2, \tag{5}$$

where, ζ_i is the effective capacitance coefficient of the device's computing chipset, reflecting the energy efficiency of the hardware. For data transmission, the energy consumed by edge device i is given by:

$$E_{t,i}^{\text{up}} = P_{t,i}\tau_{t,i}^{\text{up}} = \frac{P_{t,i}\mathcal{S}}{b_{t,i}\log_2\left(1 + \frac{P_{t,i}G_{t,i}}{N_0 b_{t,i}}\right)}. \tag{6}$$

Therefore, the total energy consumption of edge device i during the t-th communication round is:

$$E_{t,i}^{\text{c}} = E_{t,i}^{\text{cmp}} + E_{t,i}^{\text{up}}. \tag{7}$$

Minimizing this total energy consumption is critical for prolonging the operational lifetime of edge devices and achieving energy-efficient federated learning.

3.3 Data Diversity Factor

The non-IID nature of data in Federated Learning (FL) poses significant challenges in ensuring robust and generalized model training. As highlighted in [15], addressing this requires prioritizing devices with diverse and informative datasets while maintaining fairness in device selection. To achieve this, we incorporate a Data Diversity Factor into our reward function, ensuring that selected devices contribute heterogeneous and representative data to the training process.

The Data Diversity Index ($I_{t,i}$) for a device is computed locally, as outlined in [15], considering factors such as dataset size, data heterogeneity, and age of

updates. This index evaluates the informativeness and diversity of datasets while aligning with FL principles by preserving data privacy. The computed $I_{t,i}$ is then used to guide the selection of devices in each training round.

To integrate this measure into our Deep Reinforcement Learning (DRL) framework, the Data Diversity Factor for a given communication round is defined as:

$$\Gamma(a_{t,i}) = \log\left(1 + \sum_{i=1}^{I} \psi_i a_{t,i} I_{t,i}\right) \quad \forall t \in \mathcal{T}, \tag{8}$$

where, $\psi_i > 0$ represents the contribution weight of device i, $a_{t,i}$ is a binary indicator (1 if device i is selected for training in round t, 0 otherwise), and $I_{t,i}$ is the Data Diversity Index of the device. This formulation incentivizes the selection of devices that contribute maximally to diversity, improving the generalization and efficiency of the global model in dynamic edge environments.

4 Federated DRL-based Scheduling

We formulate the problem as a Markov Decision Process (MDP), where the goal is to optimize the scheduling of edge devices by balancing data diversity, energy efficiency, and latency. The proposed DA-FeDRL algorithm leverages this MDP framework to make informed decisions based on the dynamic characteristics of the devices and the network environment.

State Space. The state space for the proposed federated reinforcement learning framework, denoted by S_ϕ, captures the dynamic characteristics of the edge devices and their interactions with the network environment. For each edge device i, the state is described as:

$$S_{\phi,i} = \{I_{\phi,i}, E_{\phi,i}, B_{\phi,i}, G_{\phi,i}, H_{\phi,i}\}, \tag{9}$$

where $I_{\phi,i}$ denotes the data diversity index of the device, $E_{\phi,i}$ represents the remaining energy of the device, $B_{\phi,i}$ indicates the bandwidth of edge device i, $G_{\phi,i}$ and $H_{\phi,i}$ represent the channel gains of both the edge devices and the edge server, reflecting the quality of their respective communication links. These two parameters are used to capture the dynamic environmental conditions of Vehicular Edge Computing (VEC). This state representation captures crucial parameters that impact the learning and resource allocation processes, including data diversity, energy status, and network conditions.

Action Space. The action space governs the decisions related to the selection of devices and the allocation of resources. For each edge device i, the action is defined as:

$$A_{\phi,i} = \{s_{\phi,i}, r^*_{\phi,i}\}, \tag{10}$$

where $s_{\phi,i} \in \{0,1\}$ is a binary variable indicating whether device i is selected for participation in the current communication round ($s_{\phi,i} = 1$) or not ($s_{\phi,i} = 0$), and $r^*_{\phi,i}$ is the bandwidth score output by the action.

The bandwidth scores $r^*_{\phi,i}$ are normalized across all devices to compute the bandwidth ratio $r_{\phi,i}$:

$$r_{\phi,i} = \frac{\exp(r^*_{\phi,i})}{\sum_{j=1}^{N} \exp(r^*_{\phi,j})}. \tag{11}$$

The normalized bandwidth ratio $r_{\phi,i}$ is then used to compute the allocated bandwidth $b_{\phi,i}$ for each device:

$$b_{\phi,i} = r_{\phi,i} \cdot B, \tag{12}$$

where B represents the total available bandwidth.

This process ensures that the allocated bandwidth is proportional to the normalized scores while satisfying the constraint:

$$\sum_{i=1}^{N} r_{\phi,i} = 1. \tag{13}$$

where B is the total available bandwidth. This approach ensures that the total allocated bandwidth does not exceed the system's bandwidth capacity B, while maintaining fairness and efficiency in resource allocation.

Reward Function. The reward function for optimizing the Federated Learning (FL) framework in vehicular edge computing is defined as:

$$R_\phi = \frac{\Gamma(a_{t,i})}{\sum_{i=1}^{I} a_{t,i} \left(\alpha E^{\mathrm{c}}_{t,i} + \beta \tau^{\mathrm{total}}_{t,i}\right)}, \tag{14}$$

where $\Gamma(a_{t,i})$, as defined in Eq. 8, represents the Data Diversity Factor based on the data diversity of selected devices. $E^{\mathrm{c}}_{t,i}$ denotes the total energy consumption of device i, and $\tau^{\mathrm{total}}_{t,i}$ indicates the total time delay for device i. The parameters α and β are scaling factors that balance the influence of energy consumption and time delay.

The Data Aware Federated Deep Reinforcement Learning (DA-FeDRL) algorithm is designed to optimize resource allocation in vehicular edge computing environments. DA-FeDRL algorithm uses the TD3 algorithm which maintains an actor network and two critic networks on each edge device. The actor-network determines the optimal action $a_{\phi,i}$, while the critic networks evaluate the value of these actions.

The actor-network, parameterized by ϕ, outputs actions $a_{\phi,i}$ that aim to maximize the expected reward:

$$a_{\phi,i} = \pi_\phi(S_{\phi,i}), \tag{15}$$

where, $S_{\phi,i}$ represents the state of edge device i.

The action space $A_{\phi,i}$ includes all possible actions that the actor-network can output for device i as indicated by (10). The critic networks Q_{θ_1} and Q_{θ_2}, parameterized by θ_1 and θ_2, estimate the value of the actor's actions:

$$Q_{\theta_j}(S_{\phi,i}, a_{\phi,i}), \quad j = 1, 2. \tag{16}$$

Using two critic networks mitigates overestimation bias, which is common in single-critic architectures.

The critic networks are trained using the Bellman equation, which minimizes the temporal difference error. The target value for training is computed as:

$$y_t = R_\phi + \gamma \min_{j=1,2} Q_{\theta'_j}(S'_{\phi,i}, \pi_{\phi'}(S'_{\phi,i})), \tag{17}$$

where $Q_{\theta'_j}$ and $\pi_{\phi'}$ represent the target critic and actor networks, γ is the discount factor, and $S'_{\phi,i}$ is the next state.

The critic loss is defined as:

$$\mathcal{L}_{\text{critic}} = \mathbb{E}\left[\left(Q_{\theta_j}(S_{\phi,i}, a_{\phi,i}) - y_t\right)^2\right]. \tag{18}$$

This loss is minimized by updating the parameters of the critic networks.

The actor-network is updated less frequently than the critic networks to ensure stability in the training process. Its objective is to maximize the Q-value estimated by the first critic network:

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}\left[Q_{\theta_1}(S_{\phi,i}, \pi_\phi(S_{\phi,i}))\right]. \tag{19}$$

The target networks are updated using a soft update mechanism to ensure stability during training. This approach gradually blends the parameters of the current networks with the target networks, controlled by a factor τ, avoiding abrupt changes that can destabilize the learning process.

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi', \quad \theta'_j \leftarrow \tau\theta_j + (1-\tau)\theta'_j, \quad j = 1, 2, \tag{20}$$

The global model parameters in the federated learning process are obtained by averaging the local model parameters from all participating edge devices. This ensures the aggregation of diverse knowledge while maintaining computational efficiency:

$$w_{\text{global}} = \frac{1}{N}\sum_{i=1}^{N} w_i, \tag{21}$$

where w_i denotes the local model parameters of edge device i, and N is the number of participating devices. The pseudocode for the DA-FeDRL algorithm is presented in Algorithm 1.

Algorithm 1 DA-FeDRL: Data Aware Federated Deep Reinforcement Learning

Initialize: Actor network π_ϕ, critic networks $Q_{\theta_1}, Q_{\theta_2}$, target networks $\pi_{\phi'}, Q_{\theta'_1}, Q_{\theta'_2}$, replay buffer $\mathcal{B}$, noise $\mathcal{N}(0,\sigma)$, total bandwidth B, policy update delay d.
Input: Total communication rounds T, number of devices N, batch size M.

for $t = 1, 2, \ldots, T$ **do** ▷ Communication rounds
  Observe states for all edge devices $S_{\phi,i},\ i = 1, 2, \ldots, N$ (Eq. (9)).
  for $i = 1, 2, \ldots, N$ **do** ▷ Predict unnormalized actions for all devices
    Predict actions:

$$A_{\phi,i} = \{s_{\phi,i}, r^*_{\phi,i}\}, \quad r^*_{\phi,i} \leftarrow \pi_\phi(S_{\phi,i}) + \mathcal{N}(0,\sigma).$$

  end for
  for $i = 1, 2, \ldots, N$ **do** ▷ Normalize bandwidth scores and execute actions
    Normalize bandwidth using Eq. (11)
    if $s_{\phi,i} = 1$ **then** ▷ Check if the device is selected
      Compute allocated bandwidth:

$$b_{\phi,i} = r_{\phi,i} \cdot B \quad \text{(Eq. (12))}$$

      Observe reward R_ϕ (Eq. (14))
    end if
  end for
  Store transition $(S_{\phi,i}, s_{\phi,i}, r_{\phi,i}, R_\phi, S'_{\phi,i})$ in replay buffer $\mathcal{B}$.
  Sample mini-batch $\{(S, A, R, S')\}$ of size M from $\mathcal{B}$.
  Compute target Q-value using Eq. (17)
  Update critic networks by minimizing loss using Eq. (18)
  if $t \mod d = 0$ **then** ▷ Delayed Policy Update
    Update actor network using Eq. (19)
    Update target networks using soft update:

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi$$

$$\theta'_j \leftarrow \tau\theta_j + (1-\tau)\theta_j, \quad j = 1, 2$$

  end if
  Aggregate global FL model parameters at the server using Eq. (21)
  Distribute global FL parameters w_{global} back to edge devices.
end for

5 Simulation and Results

5.1 System Specifications

The simulations were executed on two hardware setups. The first setup was a desktop configuration equipped with an AMD Ryzen 5600 processor clocked at 4.28 GHz and an AMD 6600 GPU, utilizing the AMD ROCm software stack for high-performance computation. The second setup was a MacBook Air featuring the Apple M2 chip with an 8-core CPU (4 high-performance and 4 high-efficiency cores) and a 10-core GPU supported by Metal Performance Shaders

Table 1. Simulation parameters

Parameter	**Value Range**
Local Epochs	1
Energy Coefficient (ζ)	1.2×10^{-28}
Computation Capacity	[2, 4] GHz
Cycles per Bit	[10, 30] cycles/bit
Gini-Simpson Index	[0, 1]
Bandwidth	2 GHz
Transmit Power	[0.1, 60] W
Time Delay	[0.1, 0.9] s
Channel Gain	$[10^{-3}, 10^{-1}]$
Remaining Energy	[0.1, 1.0]
Alpha (α)	[0.2, 0.8]
Beta (β)	[0.2, 0.8]

(MPS) for model training, complemented by 8GB of unified LPDDR5 memory. These configurations provided diverse computational environments to validate the simulation's robustness and efficiency.

5.2 Simulation Environment

The federated learning simulations used the Flower (FLWR) framework, configured for 15 rounds with 100 clients. In each round, 15, 20, or 25 clients participated in the fitting phase, and 3-5 in the evaluation phase. Each client processed the non-IID MNIST dataset (28x28 grayscale images, 10 classes) with a batch size of 64, a learning rate of 0.01, a momentum of 0.9, and one local epoch. The dataset was partitioned into shards to ensure data heterogeneity. Data loading utilized 6 workers, and the server employed random client selection with an aggregation factor of 10.

The MNIST dataset was chosen because its image-based nature aligns with applications in vehicular edge computing environments, such as autonomous driving, where tasks like traffic sign recognition and navigation often involve processing image data. It provides a simplified yet effective benchmark to evaluate federated learning algorithms under these conditions. Table 1 presents the range of generated parameters used for simulations, including energy coefficients, computation capacity, transmit power, and other system metrics. Alpha (α) and Beta (β) are scaling factors used for reward calculations.

5.3 Model Architecture

The model employed in this simulation was a lightweight convolutional neural network (CNN) designed for the MNIST dataset. It featured two convolutional layers: the first with 6 filters of size 5×5 followed by max pooling, and the second with 16 filters of size 5×5, followed by max pooling. This was succeeded by three fully connected layers, comprising 120 neurons, 84 neurons, and an

output layer with 10 neurons corresponding to the MNIST classes. The model was trained using CrossEntropyLoss as the loss function and evaluated based on accuracy metrics.

5.4 Performance Analysis

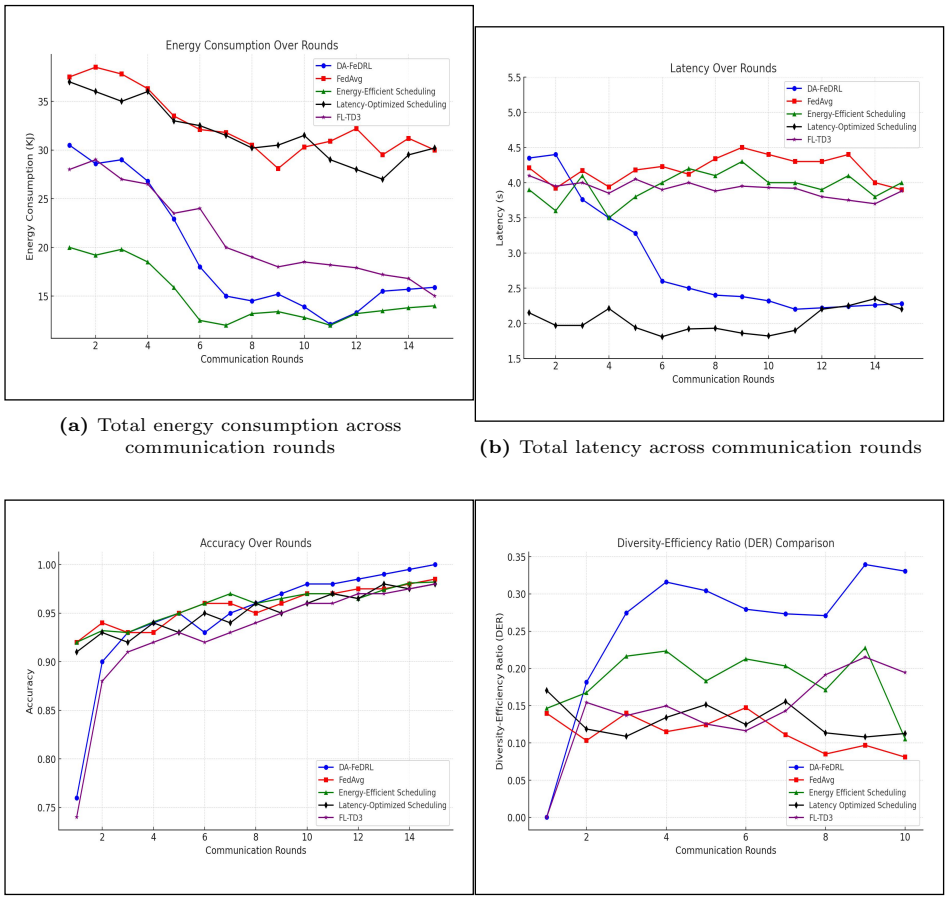
To evaluate the performance of the proposed DA-FeDRL algorithm, we compare it against several baseline approaches. FedAvg [18], a widely-used federated learning method, effectively aggregates models but does not account for critical factors such as energy efficiency, latency, or data diversity. Energy-Efficient Scheduling [6] focuses exclusively on minimizing energy consumption, neglecting data quality and other essential aspects such as latency and accuracy. Similarly, Latency-Optimized Scheduling [19] aims to reduce latency but overlooks energy efficiency and data quality. Lastly, FL-TD3 [16] optimally balances accuracy and energy consumption by leveraging data size in MEC systems, aiming to maximize the accuracy-to-energy ratio in federated learning using the TD3 algorithm.

To evaluate the effectiveness of the proposed DA-FeDRL algorithm, the following key performance metrics are considered:

1. **Energy Consumption:** The total energy consumed by all edge devices in a specific communication round, as defined in Eq. 7.
2. **Latency:** The total time taken for the completion of federated learning operations in a specific round, including communication and computation delays, as given in Eq. 4.
3. **Accuracy:** The effectiveness of the global model in correctly classifying test data, reflecting the model's generalization performance in federated learning.
4. **Diversity-Efficiency Ratio (DER):** Introduced in this work and derived from the reward function (Eq. 14), DER represents the trade-off between data diversity (Eq. 8) and the weighted sum of total energy (Eq. 7) and latency (Eq. 4). It evaluates the balance of these factors across all devices in a communication round, utilizing efficient resource utilization while maintaining diverse data.

In this section, we compare the performance of DA-FeDRL against established algorithms in terms of energy consumption, latency, accuracy, and Diversity-Efficiency Ratio (DER). The evaluation is conducted in two phases: first, by varying the number of communication rounds, and second, by varying the number of active devices. For the DA-FeDRL algorithm, we set the parameters $\alpha = 0.5$ and $\beta = 0.5$ to balance the trade-off between data diversity and resource efficiency in the reward function.

Impacts of Varying Communication Rounds. Fig. 2(a) and 2(b) demonstrate DA-FeDRL's ability to balance energy consumption and latency effec-



(a) Total energy consumption across communication rounds

(b) Total latency across communication rounds

(c) Learning accuracy over communication rounds

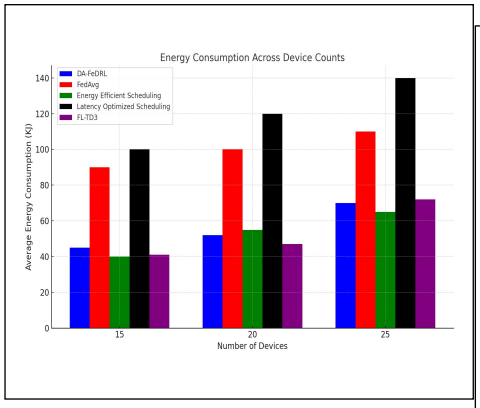
(d) Diversity-Efficiency Ratio (DER) comparison

Fig. 2. Impacts of varying number of communication rounds

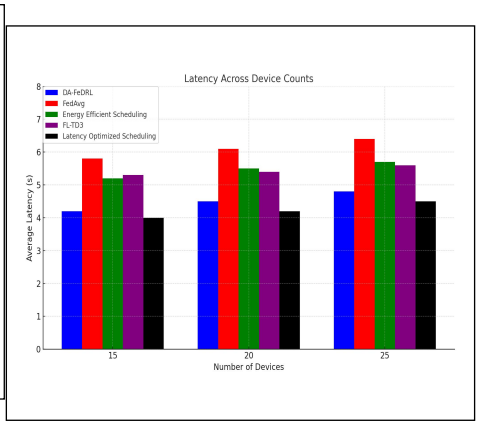
tively. While Energy-Efficient Scheduling achieved the lowest energy consumption, its sole focus on energy minimization led to underperformance in both accuracy and latency. Similarly, Latency-Optimized Scheduling achieved minimal latency but neglected energy efficiency and accuracy. In contrast, DA-FeDRL consistently maintained low energy consumption and latency while ensuring robust accuracy, showcasing its ability to optimize multiple metrics simultaneously. On the other hand, FL-TD3 focuses on maximizing the Accuracy-to-Energy ratio but does not account for Data Diversity or latency. As a result, while it shows promising energy consumption outcomes, it performs poorly in terms of latency.

As shown in Fig. 2(c), DA-FeDRL achieved consistently high accuracy, steadily improving over communication rounds. This highlights its effectiveness in optimizing scheduling decisions while maintaining data diversity and ensuring ro-

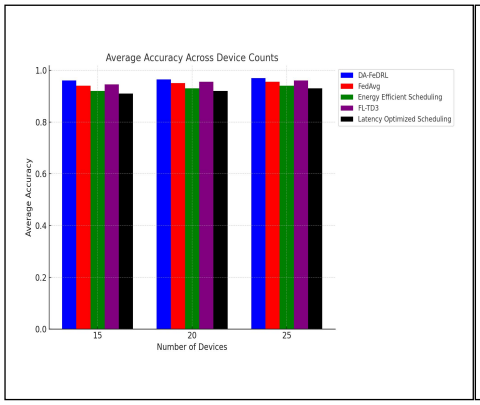
bust performance. Fig. 2(d) presents the DER values for DA-FeDRL and other established algorithms. DA-FeDRL achieved consistently higher DER, demonstrating its ability to optimize data diversity without significant resource overhead. Energy-efficient scheduling performed well in minimizing resource usage but lacked consistency due to its singular focus on energy. Latency-optimized scheduling showed occasional improvements but struggled to maintain stable DER, while FedAvg remained steady but lower overall, reflecting its lack of optimization for resource efficiency and diversity. FL-TD3, on the other hand, focuses on maximizing the Accuracy-to-Energy ratio without considering Data Diversity or latency, resulting in suboptimal DER performance.



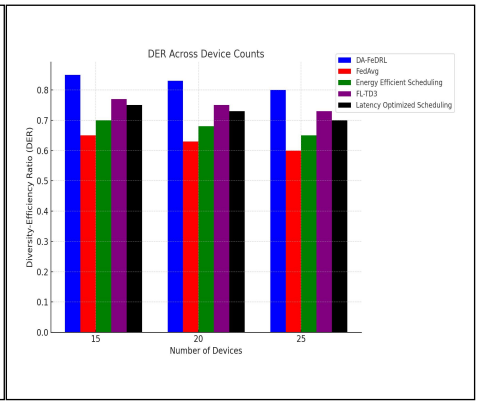
(a) Total energy consumption across communication rounds



(b) Total latency across communication rounds



(c) Learning accuracy across communication rounds



(d) Diversity-Efficiency Ratio (DER) comparison

Fig. 3. Impacts of varying numbers of active devices

The DA-FeDRL algorithm demonstrates balanced performance across accuracy, energy efficiency, and latency. Unlike state-of-the-art algorithms that excel in specific metrics at the expense of others, DA-FeDRL consistently achieves an optimal balance across all metrics. This superior performance stems from careful problem formulation and the innovative design of the state space and reward function. The state space effectively captures key system dynamics, including data diversity, energy consumption, and latency, comprehensively representing the scheduling problem. The reward function further enhances this by jointly maximizing the Diversity-Efficiency Ratio (DER), encouraging data diversity while simultaneously minimizing energy consumption and latency. This synergy between problem formulation, state space, and reward design enables DA-FeDRL to make intelligent scheduling decisions, ensuring robust and efficient performance in dynamic VEC environments. DA-FeDRL establishes itself as a versatile and well-rounded solution for real-world applications by addressing the core trade-offs in federated learning.

Impacts of Varying Number of Active Devices. The results in Fig. 3 illustrate the performance of DA-FeDRL compared to other state-of-the-art algorithms across energy consumption, latency, accuracy, and Diversity-Efficiency Ratio (DER) for varying numbers of devices. In Fig. 3(a), DA-FeDRL maintains consistently lower energy consumption compared to most algorithms and performs on par with Energy-Efficient Scheduling in certain configurations, such as with 20 devices. FL-TD3, which focuses on maximizing the Accuracy-to-Energy ratio, also demonstrates strong energy conservation capabilities, performing comparably to DA-FeDRL in terms of energy efficiency. However, its focus on energy and accuracy comes at the cost of overlooking other factors, such as latency and data diversity, which limits its overall performance balance.

Fig. 3(b) shows that while latency increases slightly with more devices for all algorithms, DA-FeDRL achieves latency levels comparable to latency-oriented scheduling, balancing efficiency without sacrificing energy or accuracy. FL-TD3, on the other hand, focuses on maximizing the Accuracy-to-Energy ratio and does not prioritize latency optimization. As a result, while it performs well in conserving energy and maintaining accuracy, it exhibits higher latency compared to DA-FeDRL and Latency-Optimized Scheduling, highlighting its limitation in achieving a balanced trade-off across metrics.

Fig. 3(c) highlights DA-FeDRL's superior accuracy across all device counts, outperforming other approaches as the number of devices increases. In Fig. 3(d), DA-FeDRL consistently achieves the highest DER values, reflecting its ability to balance data diversity with resource efficiency. This strong performance is driven by the inclusion of a reward function that optimizes data diversity alongside energy and latency, enabling DA-FeDRL to excel across multiple metrics simultaneously, unlike algorithms with narrower optimization objectives. In contrast, FL-TD3 only leverages dataset size as a measure of accuracy, whereas DA-FeDRL incorporates data diversity, which significantly contributes to its higher accuracy and better overall performance.

Our model shows promising performance across various metrics, demonstrating its effectiveness in balancing accuracy, energy efficiency, and latency. Superior performance can be further achieved by fine-tuning the reward function, optimizing model parameters, and improving the adaptability of the algorithm to dynamic VEC conditions. These adjustments can enhance the system's ability to deliver even more robust and efficient results in diverse scenarios.

6 Conclusion

In this paper, we proposed the DA-FeDRL algorithm, a deep reinforcement learning algorithm for federated learning in vehicular edge computing. The algorithm ensured balanced and efficient scheduling by modeling the problem as a Markov Decision Process (MDP) and incorporating data diversity, energy efficiency, and latency into the reward function. The results showed that DA-FeDRL achieved up to 5% higher accuracy, reduced energy consumption by up to 38%, and lowered latency by up to 32% compared to the state-of-the-art algorithms. These improvements were attributed to the inclusion of data diversity in the optimization process and the ability to maintain a trade-off between efficiency and diversity. Furthermore, as the number of devices increased, DA-FeDRL demonstrated scalability and robust performance, effectively balancing critical metrics in federated learning systems. However, the current model focuses on balancing these factors but does not specialize in excelling at any one metric. Future work includes fine-tuning the DA-FeDRL algorithm to cater to specific applications, enabling it to prioritize the optimization of certain metrics, such as energy consumption or latency, based on the requirements of the application.

Acknowledgment

The authors are grateful to King Saud University, Riyadh, Saudi Arabia for funding this work through Researchers Supporting Project Number (RSP2025R18).

References

1. S. Sakib, M. M. Fouda, Z. M. Fadlullah, and N. Nasser, "Migrating Intelligence from Cloud to Ultra-Edge Smart IoT Sensor Based on Deep Learning: An Arrhythmia Monitoring Use-Case," in 2020 International Wireless Communications and Mobile Computing (IWCMC), Jun. 2020, pp. 595–600.
2. M. H. u. Rehman, C. S. Liew, T. Y. Wah, M. Imran, K. Salah, N. Nasser, and D. Svetinovic, "Device-centric adaptive data stream management and offloading for analytics applications in future Internet architectures," Future Generation Computer Systems, vol. 114, pp. 155–168, Jan. 2021.
3. G. Zhu, Y. Wang, and K. Huang, "Broadband Analog Aggregation for Low-Latency Federated Edge Learning," IEEE Transactions on Wireless Communications, vol. 19, no. 1, pp. 491–506, Jan. 2020.

4. T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," International Conference on Communications 2019, pp. 1–7, May 2019.
5. N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated Learning over Wireless Networks: Optimization Model Design and Analysis," in IEEE INFOCOM 2019, Apr. 2019, pp. 1387–1395.
6. Q. Zeng, Y. Du, K. Huang and K. K. Leung, "Energy-Efficient Radio Resource Allocation for Federated Edge Learning," 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 2020.
7. T. -Y. Kan, Y. Chiang, and H. -Y. Wei, "Task offloading and resource allocation in mobile-edge computing system," 2018 27th Wireless and Optical Communication Conference (WOCC), Hualien, Taiwan, 2018, pp. 1-4.
8. Nandi, P. K., Reaj, M. R. I., Sarker, S., Razzaque, M. A., Mamun-or-Rashid, M., & Roy, P. (2024). Task offloading to edge cloud balancing utility and cost for energy harvesting internet of things. Journal of Network and Computer Applications, 221, 103766.
9. H. Flores et al., "Large-scale offloading in the Internet of Things," 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 2017, pp. 479-484,
10. Z. Wang, B. Lin, L. Sun, and Y. Wang, "Intelligent Task Offloading for 6G-Enabled Maritime IoT Based on Reinforcement Learning," 2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Chengdu, China, 2021, pp. 566-570.
11. Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," in Proceedings of the IEEE, vol. 107, no. 8, pp. 1738-1762, Aug. 2019.
12. Nujhat, N., Haque Shanta, F., Sarker, S., Roy, P., Razzaque, M.A., Mamun-Or-Rashid, M., Hassan, M.M. and Fortino, G., 2024. Task offloading exploiting grey wolf optimization in collaborative edge computing. Journal of Cloud Computing, 13(1), p.23.
13. D. Ye, R. Yu, M. Pan, and Z. Han, "Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach," in IEEE Access, vol. 8, pp. 23920-23935, 2020.
14. S. Shinde, A. Bozorgchenani, D. Tarchi, and Q. Ni, "On the Design of Federated Learning in Latency and Energy Constrained Computation Offloading Operations in Vehicular Edge Computing Systems," IEEE Transactions on Vehicular Technology, vol. 71, pp. 2041-2057, 2022.
15. Taïk, Afaf & Mlika, Zoubeir & Cherkaoui, Soumaya. (2021). Data-Aware Device Scheduling for Federated Edge Learning. IEEE Transactions on Cognitive Communications and Networking.
16. C. Sun, X. Li, J. Wen, X. Wang, Z. Han, and V. C. M. Leung, "Federated Deep Reinforcement Learning for Recommendation-Enabled Edge Caching in Mobile Edge-Cloud Computing Networks," in IEEE Journal on Selected Areas in Communications, vol. 41, no. 3, pp. 690-705, March 2023.
17. J. -H. Ahn, Y. Ma, S. Park, and C. You, "Federated Active Learning (F-AL): An Efficient Annotation Strategy for Federated Learning," in IEEE Access, vol. 12, pp. 39261-39269, 2024.
18. S. Xing, Z. Ning, J. Zhou, X. Liao, J. Xu and W. Zou, "N-FedAvg: Novel Federated Average Algorithm Based on FedAvg," 2022 14th International Conference on Communication Software and Networks (ICCSN), Chongqing, China, 2022, pp. 187-196

19. W. Gao, Z. Zhao, G. Min, Q. Ni and Y. Jiang, "Resource Allocation for Latency-Aware Federated Learning in Industrial Internet of Things," in IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 8505-8513, Dec. 2021.
20. X. Chen, "Federated Reinforcement Learning-Driven Offloading Strategy for Edge Computing," 2024 6th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT), Guangzhou, China, 2024.
21. Z. Luo, C. Pan, S. Wang and L. Guo, "Federated-TD3:Joint Computing Offloading and Resource Allocation in Urban Air Mobility," 2024 IEEE/CIC International Conference on Communications in China (ICCC Workshops), Hangzhou, China, 2024.