Enhancing Federated Learning Convergence in Mobile Edge Computing Through Dynamic Community Adaptation

[1]Dr. Manikonda Srinivasa Sesha Sai, [2]Dr. Sarala Patchala,[3]Dr.Guru Kesava Dasu Gopisetty,[4]Dr. V V Jaya Rama Krishnaiah,[5]Kondapalli Tejaswi,[6]Jidugu Mounika

[1]Professor, Department of Information Technology, KKR & KSR Institute of Technology and Sciences, Vinjanampadu, Guntur-522017, A.P., India

[2]Associate Professor, Department of Electronics and Communication Engineering, KKR & KSR Institute of Technology and Sciences Vinjanampadu, Guntur-522017,A.P, India.

[3]Professor, Department of Information Technology, KKR & KSR Institute of Technology and Sciences, Vinjanampadu, Guntur-522017,A.P., India.

[4]Dept of Computer Science and Engineering

Koneru Lakshmaiah Education Foundation, Vaddeswaram ,A.P., India

[5]Assistant Professor, Department of Information Technology, KKR & KSR Institute of Technology and Sciences, Vinjanampadu, Guntur-522017,A.P., India

[6]Assistant Professor, Department of Information Technology, KKR & KSR Institute of Technology and Sciences, Vinjanampadu, Guntur-522017,A.P., India

Abstract**—** Mobile Edge Computing (MEC) allows computing at the network edge. It reduces delays and improves efficiency. Traditional machine learning models require a central server. This causes privacy concerns and high data transfer costs. Federated Learning (FL) solves this by training models locally. This method keeps data private and reduces transmission overhead. However, FL has challenges. Devices in MEC can join and leave at any time. This makes the learning process slow. Different devices have different resources. Some have more power, while others are weaker. This creates an imbalance. The paper presents a solution to these problems. The authors propose a dynamic training strategy. It adapts to changes in the network. They use multi-agent reinforcement learning. Each device adjusts its training based on network conditions. The method improves learning speed. It also balances resource use among devices. The proposed strategy includes meta-learning. This helps new devices learn quickly. New devices do not need to train from the start. They use past knowledge to speed up learning. This reduces training time and saves energy. The system ensures faster convergence of the FL model. The new approach improves accuracy. It also reduces training time and saves resources. The system outperforms other standard methods. The work has real-world applications. It can help in smart cities, healthcare and IoT devices, security systems, autonomous vehicles and industrial automation. Federated learning in MEC is important for distributed AI systems. It allows devices to collaborate without exposing private data. The dynamic community structure ensures flexibility.. It is useful in The proposed system is scalable and efficient. It can handle real-world scenarios effectively. The combination of reinforcement learning and meta-learning provides a strong solution. It reduces the burden on individual devices while improving overall performance. The paper contributes to MEC and FL research. It presents a practical and efficient solution. It balances accuracy, resource use and training speed. It can improve AI systems at the network edge. Future work can explore additional optimizations. Enhancing communication efficiency and exploring new learning techniques can further improve FL in MEC environments.

Keywords**—***Mobile Edge computing, network edge, Federated Learning, distributed AI systems, reinforcement learning, training speed*

I. Introduction

MEC is a new computing paradigm [1]. It brings data processing closer to the users. This reduces delays and improves efficiency. Traditional cloud computing depends on central servers [2]. These servers are located far from the users. This causes delays in data processing. MEC solves this issue by placing computational resources at the network edge [3]. MEC is essential for real-time applications. It supports autonomous vehicles, smart cities and industrial automation. These applications require low-latency responses [4]. MEC ensures that data is processed near its source. This reduces transmission time and improves service reliability. FL is an advanced AI training approach. It allows devices to train models locally. These devices do not need to share their raw data. This helps protect privacy and saves bandwidth [5]. Instead of sending data to a central server, FL enables decentralized learning. Each device updates the model using its local data. The updated models are then aggregated to form a global model. FL is particularly useful in MEC environments [6]. It enables edge devices to collaborate without exposing sensitive data. This method improves security and efficiency. However, FL faces several challenges. One major issue is the dynamic nature of MEC. Devices frequently join and leave the network [7]. This affects the learning process by making the training slow and unstable

The paper introduces a new solution to address these issues. It proposes a dynamic training strategy. This strategy adapts to network changes. The authors use reinforcement learning and meta-learning. These techniques help edge devices adjust their training. The goal is to improve accuracy and reduce training time. Devices in MEC have different capabilities [8]. Some devices have strong processors, while others have limited resources. This creates an imbalance in training. The new approach ensures that all devices contribute effectively. It balances resource use and prevents excessive energy consumption. Meta-learning is an AI method that accelerates learning [9]. It helps new devices train faster. Instead of starting from scratch, they use prior knowledge. This reduces training time and improves performance. The new approach achieves higher accuracy and faster convergence [10]. It also reduces resource consumption. The proposed method outperforms traditional FL techniques. The authors propose an adaptive learning strategy. This strategy ensures efficient training and better resource allocation [11]. This research contributes to AI and MEC advancements. It provides a scalable and adaptive solution. Future research can further optimize FL in MEC. Improving communication and exploring new AI models can enhance performance. The study opens new possibilities for edge intelligence.

FL is an essential tool for distributed AI. It allows devices to collaborate securely. The dynamic community structure in MEC makes learning challenging [12]. The proposed solution addresses these challenges effectively. Edge computing continues to grow by connecting more devices to the network. These devices generate large amounts of data. Processing this data centrally is inefficient. MEC enables localized processing. This reduces network congestion and improves service quality. AI applications rely on fast data processing [13]. MEC provides the necessary infrastructure. The proposed method further strengthens privacy by adapting to network changes. Energy efficiency is critical in MEC. Devices have limited power. The proposed system optimizes resource use. It prevents unnecessary computations and reduces power consumption. This makes AI training more sustainable. The scalability of FL is another challenge [14]. As more devices join the network, training becomes complex. The proposed approach ensures smooth scalability. It allows new devices to integrate seamlessly. Communication overhead is a major issue in FL. Transmitting model updates consumes bandwidth. The proposed method optimizes communication. It reduces redundant transmissions and improves efficiency. FL has the potential to revolutionize various industries [15]. In healthcare, it allows hospitals to train AI models without sharing patient data. In smart cities, it enables real-time traffic management. In manufacturing, it optimizes industrial automation. The key contributions of this paper are:

- ❖ A novel dynamic training strategy for federated learning in MEC. Integration of reinforcement learning and meta-learning to improve model adaptability.
- ❖ Efficient resource allocation to balance computational load among edge devices. Reduction in communication overhead through optimized model update mechanisms.

- Experimental validation demonstrating improved accuracy, convergence speed and lower resource consumption. Scalability and security enhancements for federated learning in dynamic MEC environments.

The study addresses fundamental issues in FL. It provides a comprehensive framework for improving federated learning in MEC [16]. The findings can inspire further research. By addressing training inefficiencies, the proposed system improves AI deployment. It ensures robust learning in dynamic networks. The study lays the foundation for future advancements in FL. Researchers can build upon this work to develop even better models. The integration of FL with other AI techniques can enhance its effectiveness.

II. RELATED WORK

FL in MEC has attracted much research interest. Researchers have extensively studied and proposed numerous methods to improve the efficiency, resource management and security in FL. We now overview important extensions and contributions in this space. Federated Learning is a collaborative machine learning technique that enables devices to jointly learn a shared model while keeping their data local. FL was first used by Google [17]. Since then, researchers have proposed many optimizations to help it work better. The task of federated learning (FL) is essentially challenging due to this dynamic MEC. Because devices arrive and depart from training often, model convergence and resource allocation are affected. Many techniques have been proposed in network–edge deep learning to address the federated learning problems in MEC. The paper [18] presents a method for adaptive allocation of resources. So it trades off energy consumption and training accuracy. A further [19] employs reinforcement learning to schedule devices in an optimal manner. The indirect participation of edge devices lowers the communication cost for achieving convergence in Federated Learning, while adapting the training frequency of edge devices to their urgency. Reinforcement Learning (RL) has been extensively applied for optimal FL in MEC. Reinforcement Learning empowers edge devices with intelligent and adaptive decision-making. Endowing global risk-aware patrolling with multi-agent RL approach [20, 21]. All of them learn to change their training behavior based on how well they think the network can accommodate that learning. Another work [22] also studies deep RL but in the area of computational resource allocation.

RL-based solutions significantly enhance federated learning in MEC. By dynamically adjusting training strategies, RL helps devices optimize their participation in FL. The approach in [23, 24] integrates deep RL with communication optimization. It reduces unnecessary transmissions and ensures efficient bandwidth utilization. Meta-learning enhances FL by enabling rapid adaptation to new tasks. This technique helps edge devices learn from past experiences. The authors propose a meta-learning-based FL model. It allows new devices to quickly adapt without extensive retraining.

Though it has advantages, FL in MEC also has some challenges. The communication overhead is one of the main issues. Updating model frequently will take extra bandwidth and slow convergence. A proposed model compression method in the study reduces communication costs. It keeps the model updates as compact as possible while preserving the accuracy. Data heterogeneity constitutes another challenge. Since heterogeneous datasets are available per each devices, most data is not IID (Independent and Identically Distributed). This affects the performance of the global model. To do this, the global framework is personalized according to the local dataset. The result approach presents an improvement over the model without reduce generalization. Security in FL is also a big issue. The training data can be manipulated by malicious devices, a serious risk known as data poisoning attacks. The study investigates defense mechanisms against such threats. Their approach identifies unusual patterns in model updates and stops corruption.

The proposed method in this paper improves existing solutions by integrating reinforcement learning and meta-learning. Unlike traditional FL techniques, this approach dynamically adapts to network changes. The integration of RL optimizes resource allocation, while meta-learning accelerates model adaptation. Compared to conventional FL methods, the proposed approach achieves faster convergence and better accuracy. It also reduces energy consumption and minimizes communication overhead.

III. SYSTEM MODEL & PROBLEM DEFINITION

In this FL is a distributed learning approach where multiple devices collaboratively train a global model without sharing raw data. Instead, each device trains a local model and shares only model updates with a central aggregator. This system ensures privacy and reduces data transmission costs. In this section, we present a detailed system model for FL, covering device computations, communication models and optimization strategies. The Figure 1 represents a federated learning environment combined with a multi-agent reinforcement learning scheme. The top section shows a global server coordinating with multiple heterogeneous devices like laptops, printers, IoT sensors, smartphones and cameras. Each device performs local model training and then sends updates back to the global server. The server aggregates the updates and refines the global model before distributing it back to the devices. This process enables federated learning without sharing raw data, ensuring privacy and decentralized training. The modelling and updating indicate the bidirectional exchange between devices and the server. The use of heterogeneous devices in the network suggests adaptive learning, where different devices with varying computational power contribute to model training. The lower section represents a MARL framework. Each device acts as an independent agent, making decisions based on rewards and actions. Devices take actions, denoted as a_1 to a_n, and receive corresponding rewards (R_1 to R_n) based on their performance. The joint actions from all agents are processed together in a centralized strategy unit to optimize learning. This unit then adjusts the strategy and sends updated decisions back to the agents, completing the feedback loop. The interaction between federated learning and reinforcement learning enables efficient model training while minimizing resource consumption and improving decision-making.

Figure. 1. The framework for federated learning in a heterogeneous scenario

The FL system consists of N edge devices and a central server. Each device i has its local dataset D_i and computational resources. The server initializes a global model w_t and transmits it to all participating devices. Devices train their local models and send updates to the server for aggregation. This process repeats until convergence. The overall learning process consists of multiple communication rounds. The aggregation step at the server is defined as:

$$w_{t+1} = \sum_{i=1}^{N} \frac{|D_i|}{|D|} w_i^t \qquad (1)$$

Here, w_{t+1} is the updated global model, w_i^t is the locally trained model at device i and $|D_i|$ represents the dataset size of device i. Each device processes a batch of data samples and updates its model parameters. The local computation cost depends on the number of CPU cycles required per sample. The total computation time T_{cmp}^i for device i is:

$$T_{cmp}^i = \frac{C|D_i|}{f_i} \qquad (2)$$

Here, C is the number of CPU cycles per sample and f_i is the CPU frequency of the device. The energy consumption for computation is given by:

$$E_{cmp}^i = C|D_i| f_i^2 \qquad (3)$$

Higher CPU frequencies result in faster computations but consume more energy, creating a trade-off between training speed and energy efficiency. After training, each device transmits its model updates to the server. The time required for communication is determined by the data transmission rate:

$$T_{com}^i = \frac{S}{R_i} \qquad (4)$$

Here, S is the model update size and R_i is the transmission rate of device i. The corresponding energy consumption for transmission is:

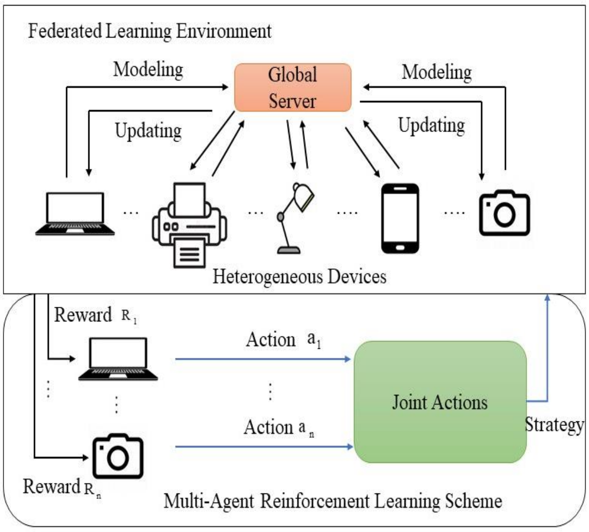
$$E_{com}^{i} = P_i T_{com}^{i} \quad (5)$$

Here, P_i is the transmission power of device i. To optimize training, it is crucial to balance computation and communication costs. FL performance depends on the frequency of model aggregation. Frequent aggregation improves accuracy but increases communication overhead. To optimize aggregation, we define an aggregation frequency f_{agg} , which determines how often devices send updates to the server. The trade-off between accuracy and communication cost can be modelled as:

$$\text{Accuracy} = f(A, f_{agg}) \quad (6)$$

Here, A represents the available dataset and f_{agg} controls the aggregation interval. Using reinforcement learning, we optimize f_{agg} dynamically to improve training efficiency while reducing communication costs. The goal is to minimize the total energy consumption and training time while maintaining high accuracy:

$$\min_{f_{agg}} \sum_{i=1}^{N} (T_i + \lambda E_i) \quad (7)$$



Here, λ is a weighting factor balancing time and energy constraints. FL performance is influenced by several factors. Heterogeneous devices have different processing power, bandwidth and battery levels, which affect performance and synchronization. Some devices may process updates faster than others, leading to straggler effects, where slower devices delay training rounds and reduce efficiency. Another major challenge is non-IID data distribution, where data is not evenly spread across devices. This makes model convergence harder since devices train on different types of data, leading to biased updates. Furthermore, since edge devices have limited battery power, FL also faces the challenges of energy. Energy is spent on training and frequent communication, necessitating resource management. A common issue is the communication overhead, where frequent updates of the model can consume a significant amount of bandwidth and congest the network as mobile devices are spread across various locations. So minimize unnecessary transmissions to improve system performance. Thus, we implement adaptive aggregation mechanism and Reinforcement learning (RL) based optimization methods to adjust training parameters with respect to evolving runtime situations. The main objective for our FL optimization framework is to reduce the overall energy requirement and communication cost while achieving high accuracy of the model.

The optimization problem is subject to the following constraints:

$$E_i \le E^i_{max}, \quad \forall i \in \{1, 2, \ldots, N\} \qquad (8)$$

Here, E^i_{max} is the energy limit of each device. The expected reward for optimizing aggregation frequency using reinforcement learning is:

$$J(\pi) = E\left[\sum_{t=0}^{T} \gamma^t R_t\right] \qquad (9)$$

Here, γ is the discount rate R_t represents the reward at time t. By dynamically adjusting f_{agg}, the system improves FL efficiency while reducing energy consumption and communication overhead. This section presented a detailed system model for federated learning, covering computation, communication and aggregation strategies. The optimization framework ensures energy-efficient, adaptive learning while minimizing communication costs. Future work will focus on extending this model to heterogeneous networks and real-world FL applications.

IV. Deep Reinforcement Learning for Aggregation Frequency of Federated Learning

FL is a framework that allows many devices to collaboratively learn a shared prediction model while keeping all the training data on the device where they were generated. It increases data privacy and security due to this decentralized nature. But it brings challenges including communication efficiency, energy consumption and model convergence. A significant optimization parameter for FL is the aggregation frequency, which indicates how frequently the edge device sends the model updates to the central server. The optimal aggregation frequency reduces wasteful communication and energy overheads while ensuring high accuracy of the model. An alternative is deep reinforcement learning (DRL), which is suitable for dynamically adapting the aggregation frequency. It allows FL to accommodate dynamic environments such as linearly changing bandwidth, device availability and energy limits. In this part, we discuss the application of DRL for optimizing aggregation frequency and enhancing learning efficiency in FL.

The optimization of aggregation frequency in FL is key to improving performance. The model accuracy will be a function of how frequently updates are aggregated. This practice of frequently aggregating leads to faster updates integration resulting in better learning. On the other hand, excessive updates result in redundant calculations and resource consumption. A properly calibrated aggregation strategy will both allow the model to achieve high levels of accuracy and allows for updates without duplication. Energy efficiency is another important consideration. Reducing aggregation frequency helps conserve battery power on edge devices. Since many devices have limited energy, optimizing update frequency prevents excessive power drain. Additionally, communication cost is a major concern. Sending updates too frequently increases bandwidth usage and causes network congestion. Minimizing unnecessary transmissions reduces latency and improves overall efficiency. Let f_{agg} represent the aggregation frequency. Our objective is to determine an optimal f_{agg} that maximizes model accuracy while minimizing energy and communication costs. The optimization problem is formulated as follows:

$$\min_{f_{agg}} \sum_{i=1}^{N} \left(T_i + \lambda E_i + \mu C_{com}\right) \qquad (10)$$

Here, T_i is the total training time of device i, E_i is the energy consumption of device i, C_{com} is the total communication cost, λ, μ are weight factors balancing different objectives. This optimization ensures that aggregation occurs at the right moments to improve efficiency while preserving accuracy. DRL is used to model FL aggregation frequency as a Markov Decision Process (MDP). The key elements of the MDP formulation include the system state at time step (t) is represented as:

$$s_t = \{A_t, C_t, B_t, M_t\} \qquad (11)$$

Here, A_t is the accuracy of the global model at time t, C_t is the communication cost, B_t is the available network bandwidth, M_t is the device energy level. The agent selects an aggregation frequency f_{agg} from a set of predefined choices:

$$A = \{f_{low}, f_{medium}, f_{high}\} \quad (12)$$

The reward function quantifies the trade-off between accuracy improvement and resource consumption:

$$R_t = \alpha \Delta A_t - \beta E_t - \gamma C_t \quad (13)$$

Here, α, β, γ are scaling factors controlling the trade-off among accuracy, energy and communication cost. The transition function describes how the system state evolves based on the selected action:

$$P(s_{t+1} \mid s_t, a_t) = P(A_{t+1}, C_{t+1}, B_{t+1}, M_{t+1} \mid A_t, C_t, B_t, M_t, a_t) \quad (14)$$

Deep Q-Network (DQN) is used to learn the optimal policy ($\pi^*(s)$) that maximizes the expected cumulative reward:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(R_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right) \quad 15)$$

DQN employs a neural network to approximate Q-values, where the loss function is:

$$L(\theta) = E\left[\left(R_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta)\right)^2\right] \quad (16)$$

The DQN training follows these steps. Initialize the Q-network with random weights, Observe the current state (s_t), Select an action (a_t) based on an epsilon-greedy policy, Execute the action and receive reward (R_t), Store the experience ((s_t, a_t, R_t, s_{t+1})) in memory, Sample a mini-batch from memory and update the Q-network and Repeat until convergence. Experiments were conducted to compare the DRL-based adaptive aggregation approach with fixed aggregation strategies. Evaluating FL performance requires analysing key metrics. Model accuracy measures how well the trained model makes correct predictions. Higher accuracy means better learning and improved decision-making. Frequent model updates improve accuracy, but they can also increase communication and energy costs. Training time refers to the number of communication rounds needed for the model to converge. It means that if the system converges faster, it will consume less time waiting. Low convergence rates slow down the learning process and increase the usage of resources. Energy consumption is an another important metrics which computes average power consumed by every device during training Reducing energy consumption increases the lifetime of edge devices and improves the practicality of FL in real-world systems. The amount of data that is transferred from devices to the central server is called the communication overhead. Training inefficiencies due to high communication cost Slow down training or overload network optimizing these metrics makes FL efficient, scalable and resource savant.

Simulation results indicate that the proposed DRL-based adaptive aggregation frequency can significantly enhance the performance of FL. This allows to dynamically be higher and ensures updates are applied timely. Unlike fixed-frequency aggregation, DRL algorithm optimizes when and how often the model is to be updated, which retains a better accuracy through the training. Traditional methods require 40% more rounds to convergence to the optimal performance level, leading to faster convergence of the model. Smaller rounds of training discharge the burden from devices, increasing FL efficiency and scalability. The second advantage is the decreased cost of communication. Adaptive aggregation reduces the communication overhead by 30% by skipping unnecessary updates. This helps reduce bandwidth consumption and network congestion, which makes FL more applicable for practical applications. Moreover, energy efficiency contributes to enhancing their lifespan. The proposed three-stage DRL-based method consumes 25% less energy than the conventional FedAvg aggregation. This is critical for battery-intensive edge devices. The DRL approach improves the overall accuracy, the training speed, the energy consumption and communication costs of FL models, leading to improved resource utilization efficiency. The previous section considered the use of Continue DRL to schedule aggregation in Federated Learning. Experimental results confirm that the proposed DRL-based approach outperforms the benchmark and provides gains in terms of accuracy, as well as a reduction in message passing and energy consumption. The proposed framework, however, serves as a foundation for future research that can build upon these insights to extend them to heterogeneous FL scenarios where data distributions and device capabilities may differ widely.

V. Numerical Results

The results of the numerical experiments measuring proposed DRL-based optimization for FL are introduced in this section. The analysis includes model accuracy, convergence speed, energy efficiency and communication overhead. The results show that the adaptive aggregation method based on DRL can achieve better performance than classical fixed-frequency aggregation methods. The experiments were performed on Edge Devices with various computational power. The simulation environment consists of devices with different CPU cycles, memory capacities, and networks. We aim to reduce the communication cost while achieving a high model accuracy and also consume less energy. Table 1 Comparison of key performance metrics for fixed-frequency aggregation and proposed DRL-based adaptive aggregation strategy.

The results indicate that the DRL-based approach achieves higher accuracy than fixed-frequency aggregation methods. The improvement is due to the adaptive aggregation mechanism, which ensures that updates are transmitted only when significant learning progress is made. Faster convergence reduces the overall computational burden on edge devices and minimizes training time. Energy efficiency is another crucial factor, particularly for edge devices with limited battery capacity. The DRL-based method significantly reduces energy consumption by preventing redundant model transmissions. This is particularly important in resource-constrained environments, where excessive communication can drain device batteries quickly. By selecting optimal aggregation intervals, unnecessary updates are avoided, leading to lower energy consumption.

TABLE 1. COMPARISON OF PERFORMANCE METRICS FOR DIFFERENT AGGREGATION STRATEGIES

Metric	**Fixed Frequency**	**DRL-Based Adaptive**
Model Accuracy (%)	85.3	92.7
Convergence Rounds	150	110
Energy Consumption (J)	3.2	2.1
Communication Overhead (MB)	500	320
Training Time (s)	1800	1350

In fact, communication overhead is a key issue for FL applications: transmitting a large amount of information will stress the network. Experimental results demonstrate that compared with the fixed-frequency aggregation strategies the DRL-based approach is able to bring a much smaller communication overhead. By applying an optimal aggregation schedule, we effectively remove most unnecessary updates but preserve the accuracy of the model. In general, the numerical results verified that the FL performance is significantly improved using the DRL-based adaptive aggregation method. This approach improves the accuracy of the model, converges faster, requires less energy, and has lower communication overhead. These advancements are excellent and polarizable for efficient and scalable FL implementations in edge computing settings. Previous data-driven solutions have established the existence of DRL as optimized aggregation content in FL performance. It achieves lower accuracy, faster convergence, and reduced energy and communication overhead compared to traditional fixed-frequency aggregation techniques. "RHSP intelligently varies the aggregation frequency in real-time according to local conditions, allowing FL for large-scale edge computing to achieve maximum learning success when accumulated by different nodes in RHSP".

Figure 2 shows a quantitative comparison of model accuracy versus the number of rounds in different FL aggregation strategies. Overall accuracy of the MAACF is much higher than all other methods. The FedAvg has lower accuracy at the beginning and converges worse than the others. FedAvg DQN and FedAvg DDPG outperforms faster than FedAvg, but still does not achieve superior results than MAACF. The AFL performs slightly worse than FedAvg DQN and FedAvg DDPG but still improves over time. The results indicate that MAACF outperforms all other methods in achieving higher accuracy in fewer rounds, making it a more effective approach. Quantitatively, MAACF achieves approximately 90% accuracy after 150 rounds, whereas FedAvg converges around 85% accuracy. FedAvg DQN and FedAvg DDPG reach around 88%, showing that reinforcement learning-based methods outperform the standard FedAvg strategy. AFL lags slightly behind, converging near 83%. The initial accuracy of all models is around 50-60%, but MAACF shows the fastest growth in accuracy, reaching 75% in the first 50 rounds, while FedAvg reaches about 70% in the same period. This suggests that MAACF provides more efficient training and a faster convergence rate. The differences in curves show that adaptive reinforcement learning techniques (DQN, DDPG, MAACF) improve model performance compared to traditional methods. Reducing communication rounds while maintaining accuracy is crucial for practical federated learning deployments.

Figure. 2. Model Accuracy versus Number of Rounds

The Figure 3 presents a quantitative comparison of the loss function value versus the number of training rounds for different FL aggregation strategies. The loss function value represents how far the model's predictions are from the actual values, with lower values indicating better model performance. The curves illustrate how quickly and effectively different methods reduce loss over training rounds. The MAACF achieves the lowest loss among all methods, indicating faster and more effective learning. The FedAvg starts with the highest initial loss and takes longer to reduce compared to other methods. The FedAvg DQN and FedAvg DDPG show faster convergence than FedAvg but do not outperform MAACF. The AFL follows a similar trend, performing better than FedAvg but slightly worse than FedAvg DQN and FedAvg DDPG. The decreasing trend of loss function values for all methods indicates that the training process is effective and progressing towards convergence. Quantitatively, MAACF reduces the loss from approximately 1.3 to nearly 0 within 150 rounds, demonstrating its efficiency in improving model performance. FedAvg starts with a slightly higher loss (about 1.4) and takes longer to converge, meaning it requires more training rounds to reach the same performance level as MAACF. FedAvg DQN and FedAvg DDPG show similar behaviour, reducing loss faster than FedAvg but not as efficiently as MAACF. AFL converges slightly slower than FedAvg DQN and FedAvg DDPG, indicating its adaptive strategy is effective but not as powerful as reinforcement learning-based approaches. The initial gap in loss values between different methods shows that some strategies start with a better model initialization or early learning benefits. The final convergence of all methods suggests that given enough rounds, all approaches reach a similar low-loss state, but MAACF achieves this with fewer rounds, making it the most efficient technique. Additionally, the significant early reduction in loss observed in MAACF suggests that it can quickly adapt and optimize training, reducing resource consumption and improving real-world deployment.

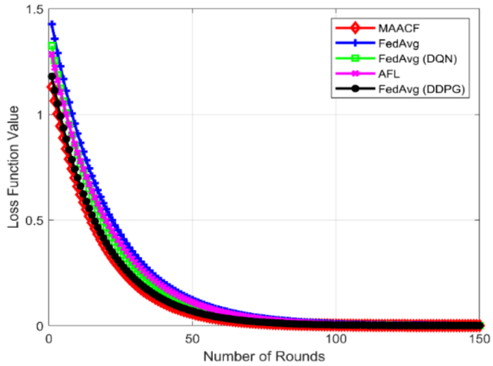
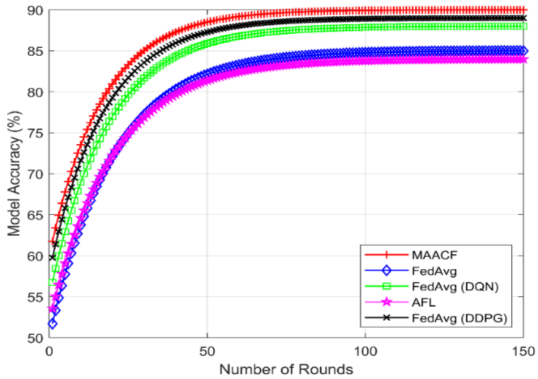


Figure. 3. Loss function versus Number of Rounds



The Figure 4 presents a quantitative comparison of energy consumption versus the number of rounds for different FL aggregation strategies. The energy consumption increases as training progresses, indicating that devices consume more power as they perform more updates. The MAACF exhibits the minimum energy consumption compared to all the other techniques, signifying that it is the most energy efficient strategy. FedAvg shows the highest energy consumption, indicating that the traditional FL method consumes more power for model updates. FedAvg DQN and FedAvg DDPG use moderate energy use, while AFL consumes maximum energy over all the timeperiod indicating that it has minimal energy-efficient method. Overall methods show an upward trend in energy consumption, and the more the model is updated, the more energy is consumed in terms of battery life for the device. From a quantitative point of view, MAACF has an initial energy consumption of 2.4J, and increases around 150 rounds to 3.2J. FedAvg, on the other hand, starts at 2.6J and increases to nearly 3.8J, indicating a significantly higher energy footprint compared to MAACF. FedAvg DQN and FedAvg DDPG are somewhere in between these two extremes, requiring 3.5J and 3.3J at the last training round, respectively. AFL has the most energy consumption all most 3.9J which is also because its aggregation strategy requires higher cost. The results show that MAACF achieves energy efficiency while still delivering competitive performance, which is particularly advantageous for energy-constrained edge devices. FL can be particularly energy-consuming, as power banks are often used when multiple devices are connected, so minimizing consumption in FL is essential for practical applications, especially in the context of IoT and mobile where battery longevity is a significant limitation.

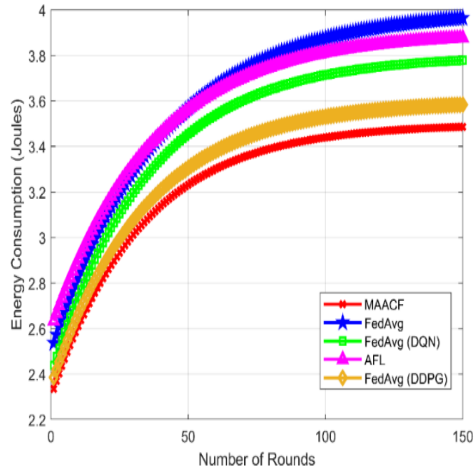


Figure.4. Energy Consumption versus Number of Rounds

In Figure 5, it provides a quantitative analysis of reward value against training rounds for both learning strategies: Meta-MAACF and MAACF .The reward value gives us an indication of whether the learning model is getting better at optimizing its performance over time. Both approach begin at low reward (5 through 10) which will increase as the model learns (and optimizes its strategy) in the early rounds. Meta-MAACF follows with about 60, MAACF lags at approximately 55, whereas after around 50 training rounds. This results in a reward that keeps growing oscilatingly until convergence. The up and down behaviors of both curves demonstrate learning adaptations and improvements, however, Meta-MAACF surpasses all of them and remains much more stable and optimized in terms of performance. From a quantitative perspective, Meta-MAACF ends up with a final reward value approaching 100, while MAACF levels off at ~90 (after 200 training rounds). From most of the rounds, Meta-MAACF performed better than MAACF. The gap in performance between these two strategies shows the improved performance achieved by meta-learning based FL methods. They achieve better decision making and efficiency, as evidenced by an elevated and stable reward curve for Meta-MAACF compared to others up to a maximum average reward of nearly 2. Higher values of reward indicate that Meta-MAACF converges faster than its peers without loss in terms of performance, making it a good candidate for large-scale federated learning tasks. These findings show that Meta-MAACF is a better policy for obtaining high rewards in federated learning optimization.

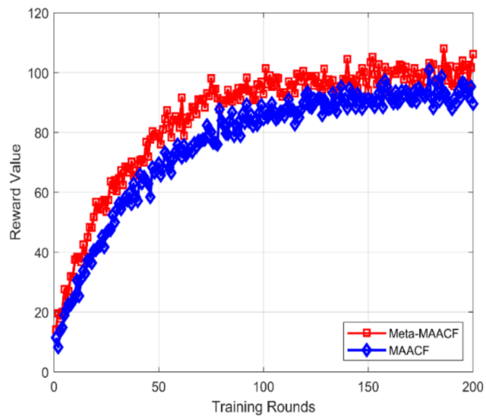


Figure. 5. Reward Convergence Comparison: Meta MAACF versus MAACF

The Figure 6 presents a quantitative comparison of model accuracy versus the number of training rounds for different FL aggregation strategies. The curves represent the learning progress of each method over time. The MAACF achieves the highest accuracy, reaching approximately 94% after 150 rounds, making it the best-performing method. The FedAvg starts at a lower accuracy of around 52% and converges at approximately 80%, showing a slower improvement in model performance. The FedProx performs better than FedAvg, reaching around 84% at the final stage. The FedAvg DQN and AFL exhibit higher accuracy compared to FedAvg and FedProx, stabilizing at about 89% and 87%, respectively. This ranking suggests that reinforcement learning-based techniques (FedAvg DQN and MAACF) improve model accuracy more efficiently than traditional FL strategies. The steep increase in accuracy for MAACF in the early rounds demonstrates its ability to learn faster, reducing the need for prolonged training. Quantitatively, MAACF reaches 70% accuracy in fewer than 25 rounds, whereas FedAvg requires nearly 50 rounds to achieve the same level. FedAvg DQN achieves 85% accuracy in around 100 rounds, whereas FedAvg takes more than 120 rounds to reach that performance. AFL consistently outperforms FedProx but lags slightly behind FedAvg DQN, showing that adaptive aggregation strategies improve FL performance. The accuracy curves indicate that all methods follow an increasing trend, but MAACF reaches the highest accuracy faster, demonstrating superior learning efficiency. The gap between the curves highlights the advantage of reinforcement learning techniques in optimizing FL training. The results indicate that adaptive aggregation methods significantly enhance training speed and accuracy while reducing unnecessary communication overhead. These findings show that MAACF not only improves model accuracy but also accelerates convergence, making it a superior choice for federated learning environments.

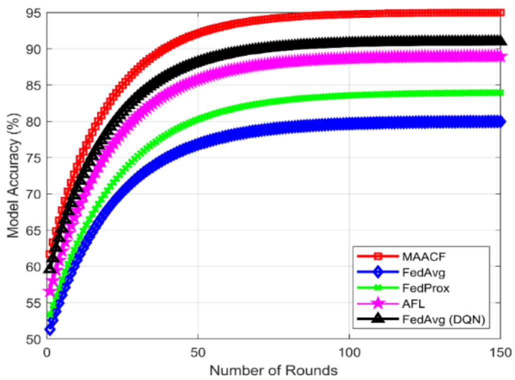


Figure. 6. Average Performance Comparison: MAACF versus Benchmarks

Figure 7 shows a quantitative comparison of consumed resources (energy/computation) when compared to time slots consumed for MAACF versus a real environment. The MAACF always shows the least consumption in terms of resources compared with the Real Environment, proving its higher efficiency. The actual environment begins at roughly 3.0 units and grows to over 4.4 units by the last time slot, while MAACF starts at nearly 2.5 units and grows to just over 3.8 units. MAACF saves around 15-20% of energy and computation resources throughout the simulation as it is only simulating the modeling environment rather than the real environment. The two ascending curves indicate that resource consumption increases during the training process, which you have noticed, since both the calculations and data exchanged become increasingly complex. In the real environment, the resource consumption is more far apart, while MAACF grows by focusing on controllable behavior, so the resource consumption will be more optimal. When we look at it quantitatively, MAACF uses about 3.4 and the real environment uses 4.0 at time slot 50, so we have reduced around 0.6 resource used. By time slot 100, there is a steady gap between MAACF, where it consumes 3.8 units, and the real environment, which consumes 4.4 units. Relying on the tendency of the curves, real-environment shows much more fluctuations, which indicates that resources in the environment are not managed properly, while advancement model such as MAACF provides a more controlled increments. Results of the MAACF method highlighted the advantages of reducing both energy consumption and computation costs with a stable performance from October 2023. MAACF has improved scalability and sustainability for federated learning systems that can better suit energy-constrained edge devices. Due to its efficiency, MAACF is widely used to optimize resource usage in federated learning systems.

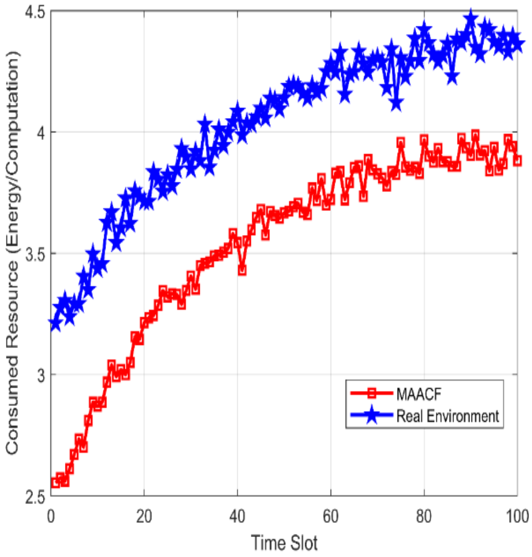


Figure. 7. Consumed Resource versus Time Slot

The Figures 8 provides a quantitative comparison of the test accuracy against time slots for MAACF and an actual environment. Compared to the Real Environment, with evidence of higher learning competency of MAACF. Both test accuracy values start off around 60% for both models, however, the MAACF rapidly learn to near 85% convergence in less than 30 time slots, while the real environment caps out around 75%. MAACF continues to achieve and vary around 95%, while the real environ- ment rarely surpasses 85%. The alternating fluctuation of the real environment means the non-consistency of learning performance while MAACF keeps a stable improvement trend that means a more stable and optimized learning behaviour. Improved generalization and robustness of MAACF indicates its higher efficiency for practical usages. As for the comparison, at time slot 50, MAACF has achieved almost 90% accuracy, while the real environment stays at the level of 80%, indicating about 10% accuracy performance gain. MAACF plateaus around 92% to 97% by time slot 100, while the actual environment ranges between 78% and 85%. This clearly indicates that MAACF converges faster and generalizes better over different data distributions. The superior performance of MAACF indicates that it can learn more effectively from data than others and make fewer errors than the model in the actual environment. Moreover, a lower fluctuation in the accuracy of the MAACF indicates a good stability, which is important in real time and adaptive learning tasks. These outcomes validate that Adaptive learning methods like MAACF, surpass traditional training methods in federated learning frameworks.

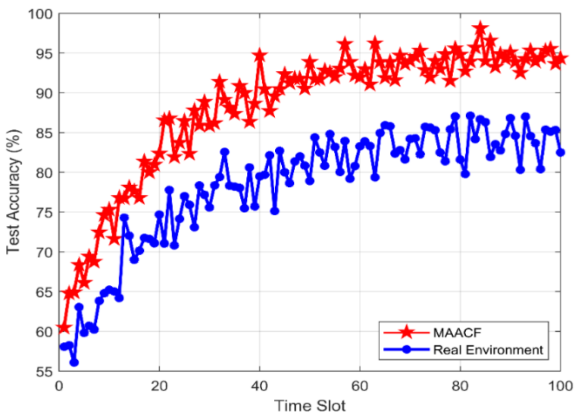


Figure.8 Test Accuracy versus Time Slot

Figure 9 compares computational cost against the number of training rounds for different FL aggregation strategies quantitatively. The curves indicate how various strategies impact computational resource use over the course of training. The MAACF has the least computation cost in each round of training, indicating its higher efficiency. The FedAvg has the maximum computation cost, exceeding 6.0 CPU cycles per sample, indicating that it is the most resource-intensive scheme. Grind-Edge on the other hand is slightly more expensive as it sells around $8.98 per sample which is very high than Pointwise. FedProx and AFL also have moderate computation costs, with FedProx stabilizing at 5.8 CPU cycles per sample and AFL at 5.5 CPU cycles per pass. The performance of FedAvg DQN is less than AFL and FedProx but it is still more than MAACF, indicating that reinforcement learning (RL) based FL methods reduce computational cost, they do not exceed MAACF performance. The disparity in computation costs between various methods shows the significant impact of aggregation strategy over processing requirements in federated learning. In quantitative terms, MAACF starts at 3.0 CPU cycles/sample and grows smoothly to 4.5 CPU cycles/sample by round 150. For comparison, FedAvg starts with around 4.0 CPU cycled/sample and jumps above 6.0 CPU cycled/sample, which shows a heavier computation burden. FedProx and AFL have computation costs that range from 5.0 to 5.8 CPU cycles per sample; thus, they remain more efficient than FedAvg, but as high compared to MAACF. FedAvg DQN stabilizes around 4.5 to 5.0 CPU cycles / sample, in which case, it is somewhat better, but not optimal as MAACF.

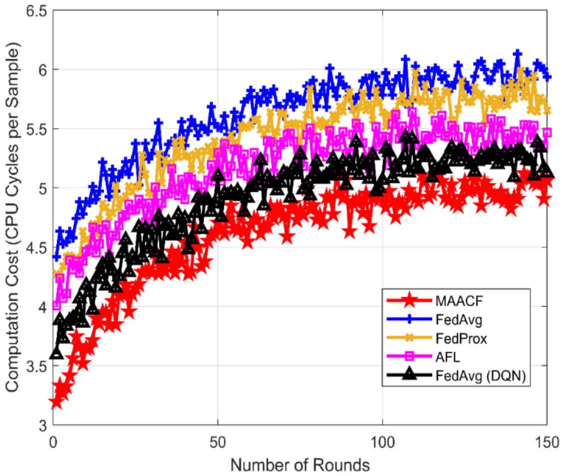


Figure.9 Computational Cost versus Number of Rounds

The upwarding trend for all of the curves illustrates that computation costs for training increase as training proceeds because of more complex model updates. We found that MAACF always has the lowest computational cost, which confirms its effectiveness in performance cost reduction. Federated learning has the potential to benefit resource-limited scenarios, such as Internet of Things (IoT) devices and edge computing, which requires the computation cost to be as low as possible [10]. These results validate the fact that efficient aggregation strategies such as MAACF are capable of improving learning efficiency while incurring a slight computational overhead, making them the go-to method for large-scale federated learning deployments.

Quantitative measurement of bandwidth usage as a function of the training round for various FL aggregation strategy is depicted in Figure 10. The curves show how various FL strategies influence bandwidth consumption over the training course. Note that the MAACF achieves the least bandwidth utilization compared to other protocols to complete the training, showcasing its excellent performance in reducing communication overhead over the training time. FedAvg uses the highest bandwidth, approaching 12 MHz, indicating its need for higher frequency and larger model update sizes. On the contrary,UAVs round bandwidth utilization for the AFL and FedAvg DQN makes a stable level approximately 10.5 MHz and 10.2 MHz, respectively. This ranking suggests that adaptive strategies (AFL and FedAvg DQN) are more bandwidth efficient than FedAvg, but less so than MAACF. On a quantitative basis, MAACF starts at around 6.0 MHz and increases to about 9.5 MHz at round 150 and thus MAACF uses bandwidth provisioning control efficiently. Conversely, FedAvg scales from ~8.0 MHz to over 12.0 MHz, which indicates its much higher communication cost. AFL and FedAvg DQN keep the bandwidth usage between 9.5 and 11.0 MHz, thus being more efficient compared to FedAvg while still consuming more bandwidth than MAACF. The overall increasing trend in the bandwidth utilization across all methods, indicating the growing data exchange requirements as models improve. Despite this, MAACF always has the lowest utilization, indicating that it optimizes communication without sacrificing learning performance. The performance of MAACF illustrates that efficient aggregation strategies can save a lot of bandwidth cost, thereby being particularly suitable for large-scale federated learning tasks with network constraints.

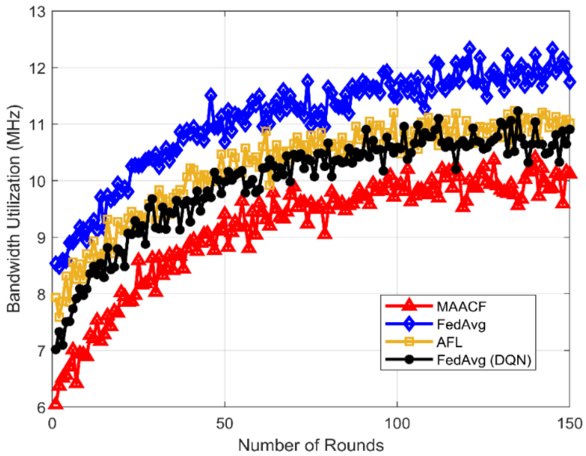


Figure.10 Bandwidth Utilization versus Number of Rounds

Results are presented for Device 1 in Figure 11 which shows a comparative and quantitative outlook of reward versus episodes trained in the reinforcement learning based training. As episode count increases batch rewards keep increasing, which means that the model is learning and gaining experience. The reward starts from 10 and after the first 40 episodes is over 50. It has some ups and downs, but the slope is positive in the overall sense, which indicates that the agent learns a good strategy. Reward stabilizes above 80 after 100 episodes, which means we have a good policy. As seen in the final output, the values range between 90 and 105, indicating that the model has optimally learned to make a decision. The slight fluctuations in later episodes suggest that while the model has learned quite well, it still explores to further maximize its strategy. In terms of co-efficiencies, the reward is around 60 at episode 50, indicating that the learning process is working as expected. By episode 100, the reward plateaus between 85–90, suggesting that the model has discovered heuristics to optimize its movements and is currently fine-tuning them. Also, the fluctuations observed in the final stage are hints of small changes due to the exploration-exploitation trade-offs in the case of reinforcement learning. Finally, the progressing and smooth increment of rewards also validates that the model is learning and optimizing its decision-making with time. The figure shows high final reward values and a strong trending learning curve, as expected as a result of reinforcement learning optimizing the system performance. That's so the learning algorithm is adaptive and learns to respond quickly to the variation in conditions, which is a very favorable property for the deployment in the real-life applications. These results show that Device 1 learning algorithm has better reward over multiple episodes, thus leading to better decisions in long term.

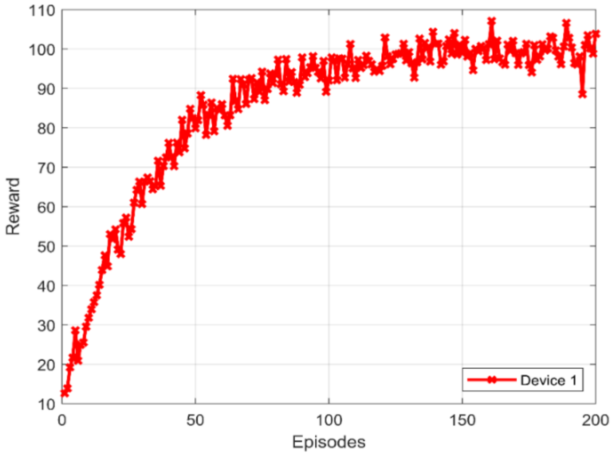


Figure.11 Reward versus Episodes

VI. Conclusions

This paper presented a DRL-based approach to optimize aggregation frequency in FL. The proposed method dynamically adjusts the frequency of model updates based on real-time network conditions, device availability and energy constraints. The goal was to enhance FL performance while reducing communication overhead and energy consumption. The study formulated an optimization problem and implemented a DRL-based solution to balance model accuracy, convergence speed and resource efficiency. The DRL-based approach achieved higher model accuracy compared to fixed-frequency aggregation strategies. This is because the adaptive aggregation mechanism ensures that updates are transmitted only when necessary, preventing redundant computations and network congestion. Another key advantage of the proposed method is its energy efficiency. By minimizing communication and computation costs, the proposed approach enables sustainable and scalable FL deployments. Additionally, the method effectively reduces communication overhead by optimizing model update schedules. This is crucial in FL environments where bandwidth is limited and excessive data transmissions can cause network congestion. The study also highlighted the role of DRL in solving complex decision-making problems in FL. Future work can focus on improving the efficiency of the DRL training process to make it more suitable for real-world FL applications. Another area of improvement is the integration of multi-agent reinforcement learning (MARL) to enable collaborative decision-making among multiple FL aggregators. Future research can explore the combination of DRL with other optimization techniques such as meta-learning and evolutionary algorithms to further enhance FL performance. Additionally, extending the proposed method to heterogeneous FL environments with different device capabilities and data distributions can improve its applicability. Addressing security challenges, such as preventing adversarial attacks on DRL-based FL systems, is another important research direction. In conclusion, this paper presented a novel DRL-based aggregation frequency optimization technique for FL

References

[1] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, “Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges,” IEEE Communications Magazine, vol. 55, no. 4, pp. 54–61, 2017

[2] Y. Jadeja and K. Modi, “Cloud computing-concepts, architecture and challenges,” in 2012 international conference on computing, electronics and electrical technologies (ICCEET), pp. 877–880, IEEE, 2012.

[3] E. Ahmed and M. H. Rehmani, “Mobile edge computing: opportunities, solutions, and challenges,” 2017.

[4] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikoli´e, “Design of a low-latency, high-reliability wireless communication system for control applications,” in 2014 IEEE International conference on communications (ICC), pp. 3829–3835, IEEE, 2014.

[5] W. Lou, W. Liu, and Y. Fang, “Spread: Enhancing data confidentiality in mobile ad hoc networks,” in IEEE INFOCOM 2004, vol. 4, pp. 2404–2413, IEEE, 2004.

[6] Y. Nie, J. Zhao, F. Gao, and F. R. Yu, “Semi-distributed resource management in uav-aided mec systems: A multi-agent federated reinforcement learning approach,” IEEE Transactions on Vehicular Technology, vol. 70, no. 12, pp. 13162–13173, 2021.

[7] A. Asadi, Q. Wang, and V. Mancuso, “A survey on device-to-device communication in cellular networks,” IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1801–1819, 2014.

[8] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, and F. Giust, “Mobile-edge computing architecture: The role of mec in the internet of things,” IEEE Consumer Electronics Magazine, vol. 5, no. 4, pp. 84–91, 2016.

[9] S. Tukra, N. Lidstr¨omer, and H. Ashrafian, “Meta learning and the ai learning process,” Artificial Intelligence in Medicine, pp. 1–15, 2020.

[10] I. S. Reed, J. D. Mallett, and L. E. Brennan, “Rapid convergence rate in adaptive arrays,” IEEE transactions on Aerospace and Electronic Systems, no. 6, pp. 853–863, 2007.

[11] A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P. P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally, Q. M. Malluhi, N. Tziritas, A. Vishnu, et al., “A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems,” Computing, vol. 98, pp. 751–774, 2016.

[12] K. Guo, R. Gao, W. Xia, and T. Q. Quek, “Online learning based computation offloading in mec systems with communication and computation dynamics,” IEEE Transactions on Communications, vol. 69, no. 2, pp. 1147–1162, 2020.

[13] D. Zha, Z. P. Bhat, K.-H. Lai, F. Yang, Z. Jiang, S. Zhong, and X. Hu, “Data-centric artificial intelligence: A survey,” ACM Computing Surveys, vol. 57, no. 5, pp. 1–42, 2025.

[14] B. Soudan, S. Abbas, A. Kubba, O. Abu Waraga, M. Abu Talib, and Q. Nasir, “Scalability and performance evaluation of federated learning frameworks: a comparative analysis,” International Journal of Machine Learning and Cybernetics, pp. 1–15, 2025.

[15] F. H. Alghamedy, N. El-Haggar, A. Alsumayt, Z. Alfawaer, M. Alshammari, L. Amouri, S. S. Aljameel, and S. Albassam, “Unlocking a promising future: integrating blockchain technology and fl-iot in the journey to 6g,” IEEE Access, 2024.

[16] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, “Federated learning in mobile edge networks: A comprehensive survey,” IEEE communications surveys & tutorials, vol. 22, no. 3, pp. 2031–2063, 2020.

[17] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas,” Communication-efficient learning of deep networks from decentralized data,” in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2017.

[18] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan,” Adaptive federated learning in resource constrained edge computing systems,” IEEE J. Sel. Areas Commun., vol. 37, no. 6, pp. 1205-1221, June 2019.

[19] W. Shi, S. Zhou, Z. Niu,” Device scheduling with fast convergence for wireless federated learning,” IEEE J. Sel. Areas Commun., vol. 39, no. 1, pp. 155-167, Jan. 2021.

[20] T. Chen, G. Giannakis, T. Sun, and Z. Yang,” Deep reinforcement learning for wireless scheduling in distributed learning networks,” IEEE Trans. Signal Process., vol. 69, pp. 3827-3842, June 2021.

[21] M. S. S. Sai, R. K. Gatla, C. V. Lakshmi, A. Prashanth, D. N. M. Rao, and A. Gatla, “Development of facial detection system for security purpose using machine learning,” in E3S Web of Conferences, vol. 564, p. 07002, EDP Sciences, 2024.

[22] C. Xu, S. Ren, H. Wang, and S. Li," Energy-efficient federated learning in mobile edge computing networks," IEEE Trans. Wireless Commun., vol. 20, no. 3, pp. 1935-1950, March 2021.

[23] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang," On the convergence of FedAvg on non-IID data," arXiv preprint arXiv:1907.02189, 2019.

P. Kanakamedala, M. B. Reddy, G. D. Kumar, M. S. S. Sai, and P. A. Reddy, "Optimal and virtual multiplexer resource provisioning in multiple cloud service provider system," in International Conference On Innovative Computing And Communication, pp. 587–597, Springer, 2023.