Federated Learning Algorithm to Suppress Occurrence of Low-Accuracy Devices

Original Scientific Paper

Koudai Sakaida
Department of Informatics,
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
Tokyo, Japan
sakaida.koudai@ohsuga.lab.uec.ac.jp

Akihiko Ohsuga
Department of Informatics,
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
Tokyo, Japan
ohsuga@uec.ac.jp

Keiichiro Oishi
Department of Computer Science,
Faculty of Environmental and Life Science,
Okayama University,
Okayama, Japan
oishi@okayama-u.ac.jp

Andrew J
Department of Computer Science and Engineering,
Manipal Institute of Technology,
Manipal Academy of Higher Education,
Manipal, India
andrew.j@manipal.edu

Yasuyuki Tahara*
Department of Informatics,
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
Tokyo, Japan
tahara@uec.ac.jp

Yuichi Sei*
Department of Informatics,
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
Tokyo, Japan
seiuny@uec.ac.jp

*Corresponding author

Abstract *– In recent years, federated learning (FL), a decentralized machine learning approach, has garnered significant attention. FL enables multiple devices to collaboratively train a model without sharing their data. However, when the data across devices are non-independent and identically distributed (non-IID), performance degradation issues such as reduced accuracy, slower convergence speed, and decreased performance fairness are known to occur. Under non-IID data environments, the trained model tends to exhibit varying accuracies across different devices, often overfitting on some devices while achieving lower accuracy on others. To address these challenges, this study proposes a novel approach that integrates reinforcement learning into FL under Non-IID conditions. By employing a reinforcement learning agent to select the optimal devices in each round, the proposed method effectively suppresses the emergence of low-accuracy devices compared to existing methods. Specifically, the proposed method improved the average accuracy of the bottom 10% devices by up to 4%, without compromising the overall average accuracy. Furthermore, the device selection patterns revealed that devices with more diverse local data tend to be chosen more frequently.*

Keywords*: FL, Non-IID, Performance Fairness, Device Selection, RL, DDQN*

Received: December 20, 2024; Received in revised form: May 11, 2025; Accepted: May 12, 2025

1. INTRODUCTION

In recent years, with advancements in the Internet of Things (IoT) and artificial intelligence, machine learning technologies have been utilized in various aspects of daily life, bringing significant convenience to people. Concurrently, the explosive increase in data volume has led to privacy breaches, heightening concerns regarding privacy and security. Traditional machine learning methods require the aggregation of data in a single location. For example, many smartphones contain private data that must be integrated for training. However, aggregating data in one place not only results in high communication costs and significant battery consumption on devices but also increases the risk of compromising user data privacy and security.

Federated learning (FL), introduced by Google in 2016 [1], has garnered attention as a decentralized machine learning approach that addresses these issues. FL has

demonstrated its efficacy in enabling global-scale collaborative training, as evidenced by its successful application in rare cancer boundary detection. This initiative aggregated insights from 71 hospitals spanning six continents while rigorously preserving patient data privacy [2]. However, it is crucial to recognize that despite its inherent privacy-preserving advantages, FL is not impervious to privacy leakage stemming from shared model updates. Recent scholarly work, such as the differentially private knowledge transfer paradigm proposed by Qi et al. [3], underscores the necessity of integrating supplementary privacy-enhancing mechanisms to bolster FL's resilience against inference attacks. Furthermore, Boscarino et al. [4] highlighted FL's pivotal role in supporting indigenous data sovereignty, illustrating its potential to empower communities in maintaining control over sensitive genomic information.

Communication efficiency constitutes another significant impediment to the widespread adoption of FL. Wu et al. [5] introduced FedKD, an adaptive knowledge distillation strategy coupled with gradient compression techniques, which substantially curtails communication overhead, thereby tackling a critical scalability bottleneck. Similarly, the comprehensive survey by Asad et al. [6] meticulously examined existing methodologies and prospective avenues for alleviating FL's communication costs, reinforcing the urgency and multifaceted nature of this challenge in practical deployments.

A further salient obstacle in federated learning arises from the Non-Independent and Identically Distributed (Non-IID) nature of local datasets across participating devices. This inherent data heterogeneity not only diminishes model accuracy but also adversely affects the active engagement of users, thereby complicating model convergence and the reliable evaluation of performance [7, 8]. Personalized federated learning frameworks, such as the one proposed by Lin et al. [9], have been developed to address these non-IID issues by tailoring local models with a focus on communication efficiency, robustness, and fairness concurrently, representing a notable trajectory in contemporary FL research.

The issue of fairness in federated learning has emerged as a particularly pressing concern, primarily due to the intrinsic heterogeneity among participating clients. Chaudhury et al. [10] emphasized the importance of explicitly addressing fairness, proposing solutions grounded in cooperative game theory to ensure equitable model performance across diverse client populations. Moreover, recent innovations like FedFed, introduced by Yang et al. [11], prioritize the mitigation of non-IID effects through selective feature distillation, carefully balancing the inherent trade-offs between model accuracy and privacy preservation.

These recent advancements collectively underscore the imperative for federated learning to continue its evolution by comprehensively addressing the intertwined challenges of privacy, communication efficiency, fairness, and data heterogeneity. Such holistic approaches are essential to ensure the deployment of robust, scalable, and equitable FL systems in diverse real-world settings, aligning closely with the practical motivations and ongoing challenges elaborated upon within this study.

In FL, the process of sharing and updating models is repeatedly performed while maintaining the data on each device, thereby enabling training while protecting privacy. FL randomly selects a subset of devices to participate in each update, rather than having all devices participate each time, which improves scalability and reduces communication costs.

However, FL has several limitations. The first is that data across devices may be non-independent and identically distributed (non-IID). This implies that the data distribution varies across devices, which differ in the labels they hold or the amount of data they possess. Therefore, the nature of non-IID data complicates FL training and evaluation.

Another challenge in FL is fairness, as discussed in Section 3, "Heterogeneity and Performance Fairness." Fairness issues arise from various perspectives, including the fairness of machine learning algorithms, as described by Pessach et al. [12] and in FL device selection, as raised by Vucinich et al. [13]. This study focuses on fairness in performance, particularly in devices with lower accuracy. Specifically, under non-IID data conditions, the differing data distributions on each device tend to cause high variance in model test accuracies across devices. In such situations, performance fairness in FL is likely to be compromised, leading to an increase in low-accuracy devices.

This paper proposes a novel approach that applies reinforcement learning to address the issue of low-accuracy devices in FL. Conventional methods typically employ random device selection and enhance aggregation to improve performance. However, these methods tend to prioritize reducing the variance in accuracy over improving average accuracy, without sufficiently considering the performance enhancement of low-accuracy devices. This study aims to suppress low-accuracy devices more effectively than other methods while maintaining the performance of high-accuracy devices. The proposed method utilizes a reinforcement learning agent to learn how to improve the accuracy of lower-performing devices in each round, with the aim of enhancing the performance of low-accuracy devices without compromising average accuracy compared to existing methods. The contributions of this study are as follows:

- A novel algorithm that applies reinforcement learning is designed to address the issue of low-accuracy device occurrence in FL. This algorithm enables effective device selection during the FL training process, thereby suppressing the emergence of low-accuracy devices.
- Compared to existing methods, the proposed method significantly improves the average accuracy of the bottom 10% of devices in a non-IID data

environment without reducing the overall average accuracy. This result demonstrates that the proposed method contributes to model fairness and performance enhancement, even under non-IID data conditions.

- We confirmed that the proposed method can effectively suppress the impact of low-accuracy devices on complex datasets with more than 10 classes. This validates the effectiveness of the proposed method across a wide range of datasets.

The structure of this paper is as follows: Section 2 presents background information on FL. Section 3 reviews related research. Section 4 describes the details of the proposed FL algorithm that utilizes reinforcement learning. Section 5 presents the evaluation results of the proposed method using real-world datasets. Finally, Section 6 concludes the paper.

2. BACKGROUND

FL is a method for training models through iterative communication between a central server and multiple devices. Each round consists of the following steps, which form a continuous flow referred to as a "round." The learning process progresses by repeating these rounds.

i. Initialization: Before starting the first round, the central server initializes the weights of the global model.

ii. Device Selection: At the beginning of each round, the central server randomly selects the devices according to a specified ratio. Subsequently, the current global model weights are sent to the selected devices.

iii. Update: In this phase, each device trains the global model based on its local dataset and sends the updated local model weights back to the central server.

iv. Aggregation: The central server aggregates the received updated local model weights to update the global model.

v. Termination: This process is terminated when the global model converges and reaches a specific threshold. If convergence is not achieved, the process returns to device selection and proceeds with local updates and weight aggregation.

FedAVG [14], a fundamental FL framework, conducts learning as described in Equation (1):

$$w_k^{t+1} = w_k^t - \eta \nabla \mathcal{L}_k(w_k^t; D_k) \quad (1)$$

where w_k^t represents the weights of the local model of device k at round t, with w_k^{t+1} denoting the updated weights of the local model at round t+1; $\nabla L_k(w_k^t; D_k)$ indicates the gradient of the loss function with respect to the local dataset D_k. In this manner, each device updates the model weights w_k^t using its local dataset D_k and learning rate η.

Next, the central server updates the global model by aggregating the weights w_k^{t+1} collected from each device according to Equation (2).

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1} \quad (2)$$

where w^t is the weight of the global model at round t; K is the number of devices selected in round t; n_k is the number of data samples on device k; and n is the total number of data samples across all devices.

3. RELATED WORK

3.1. DEVICE SELECTION TECHNIQUES FOR FL

Recent investigations have explored diverse methodologies for optimizing device selection within federated learning frameworks, primarily focusing on enhancing overall model performance and training efficiency. For instance, Tian et al. [15] introduced FedRank, a client selection method predicated on ranking that leverages imitation learning to mitigate cold-start issues frequently encountered with reinforcement learning-based techniques. By employing a pairwise ranking strategy, FedRank effectively selects clients based on system and data heterogeneity, demonstrating significant improvements in convergence speed and energy efficiency. Furthermore,

Pan et al. [16] developed a contextual client selection framework utilizing a Neural Contextual Combinatorial Bandit (NCCB) algorithm. This framework extracts client features through locality-sensitive hashing and exploits correlations among datasets, resulting in reduced training duration and enhanced model accuracy, approaching performance levels observed in IID scenarios.

In a related vein, Zhang et al. [17] proposed an approach integrating spectrum allocation optimization with device selection for federated learning in wireless networks. Their method aims to minimize training delay and energy consumption by selecting devices according to the divergence between local and global model weights, thereby facilitating faster convergence under non-IID conditions. While these methodologies offer considerable advancements in device selection strategies and overall system efficiency, it is crucial to acknowledge that none of these explicitly address fairness among devices, such as ensuring balanced accuracy or equitable participation across heterogeneous data distributions.

3.2. FEDERATED REINFORCEMENT LEARNING (FRL)

FRL is an approach that combines FL with reinforcement learning (RL). FL focuses on collaborative training of models across multiple devices while preserving privacy, whereas FRL introduces reinforcement learning techniques to enable optimal device selection and parameter tuning. In FRL, the elements of RL (en-

vironment, state, and action) are applied within the FL framework to potentially address complex issues [18]. Thus, FRL holds promise for overcoming the limitations of FL and is expected to have applications in various fields. Research on the use of reinforcement learning for device and client selection in FL has been active [19-22], with selections directly impacting the quality and utility of the model, which makes this a highly important area.

Wang et al. [19] proposed FAVOR, which utilizes the double deep-Q-network (DDQN) algorithm for client selection. This method allows device and client selection, which enhances convergence speed of the model under non-IID conditions, thereby saving on computational resources. However, because DDQN model training is limited to a single client, the agent may not rapidly converge.

Additionally, Bouaziz et al. [22] proposed FL to address system and static heterogeneity using reinforcement learning (FLASH-RL), which employs the DDQN model to perform client selection, aimed at reducing computational and communication costs. By enabling multi-action selection and learning, their approach accelerates the learning process. Furthermore, FLASH-RL contributes to latency reduction by individually evaluating each client using a proprietary evaluation function.

Yu et al. [23] introduced DDPG-AdaptConfig, a deep reinforcement learning framework based on Deep Deterministic Policy Gradient (DDPG), which adaptively selects devices and configures local training hyperparameters such as batch size and epoch count. This method incorporates a transformer-based actor network to capture heterogeneous information from model parameters and applies clustering-based aggregation to further accommodate system and data diversity.

3.3. HETEROGENEITY AND PERFORMANCE FAIRNESS

Shi et al. [24] argue that many current FL frameworks are designed with a central server-centric perspective, prioritizing metrics such as convergence speed and overall model accuracy, often at the expense of individual client needs. This imbalance can disincentivize participation from less capable clients and potentially compromise the global model's representativeness. Their work proposes a taxonomy of fairness-aware FL methodologies, identifying critical stages where fairness considerations are paramount, including client selection, optimization processes, and incentive mechanisms.

Furthermore, Rafi et al. [25] emphasize that fairness issues in FL extend beyond client selection to encompass reward allocation strategies. They contend that the uniform distribution of global models to all clients, irrespective of their individual contributions to the training process, can be perceived as unfair, particularly by clients who have invested more resources or data. The authors also highlight the potential for demographic biases, such as those related to gender or ethnicity, to compound these fairness challenges within FL systems.

Chen et al. [26] investigate the inherent trade-off between privacy preservation and fairness in FL. Their analysis suggests that privacy-enhancing techniques, such as the introduction of noise or limitations on data sharing, can disproportionately impact disadvantaged groups by causing a greater degradation in their model performance compared to others. Conversely, efforts to enhance fairness might necessitate increased data transparency, potentially leading to heightened privacy risks.

Huang et al. [27] categorize fairness in FL into two primary dimensions: collaboration fairness and performance fairness. Collaboration fairness addresses the equitable distribution of rewards and the provision of adequate incentives for client participation. Performance fairness, on the other hand, focuses on ensuring consistent model accuracy across all clients. The authors assert that the simultaneous achievement of both collaboration and performance fairness is crucial for the development of sustainable and robust FL systems, particularly in real-world applications characterized by significant client heterogeneity.

These perspectives collectively demonstrate that fairness in FL is a multifaceted issue that intersects with client heterogeneity, privacy concerns, and system sustainability. Addressing fairness effectively requires comprehensive strategies that go beyond mere accuracy optimization. In real-world scenarios, the data on devices are often non-IID, which accelerates imbalanced learning across devices. Data heterogeneity poses a significant challenge in FL, leading to variations in learning outcomes and substantial differences in model accuracy among devices.

The conventional FedAVG method [14] is known to exhibit unstable performance under non-IID conditions, with some devices demonstrating significantly higher or lower accuracy than others. In such situations, not only overall model accuracy enhancement but also performance fairness across devices should be considered. Performance fairness ensures that the model performs uniformly across all participating devices, preventing scenarios in which low-performance devices are disproportionately affected, thereby improving overall fairness.

Huang et al. [28] successfully increased the convergence speed of the model while maintaining performance fairness by employing a dual-momentum descent method and weighted aggregation that accounts for client accuracy and participation frequency. Wentao et al. [29] introduced federated fairness and effectiveness (FedFE), which integrates momentum gradient descent into the FL process and performs accuracy-based weighted aggregation, thereby achieving improvements in both fairness and convergence speed. Despite these advancements, a sufficient num-

ber of studies have not been conducted on complex datasets with more than 10 classes, leading to a lack of validation regarding their adaptability to multiclass environments.

These studies presented effective approaches for addressing imbalances caused by non-IID data while enhancing performance fairness. This study focuses on performance fairness, specifically aiming to construct models for non-IID data environments.

Performance fairness refers to the uniformity of model performance across all devices participating in FL. In this paper, we define performance fairness as "achieving as equal accuracy as possible for all devices within FL," with the objective of enhancing this fairness while suppressing the emergence of low-accuracy devices.

Li et al. [30] proposed q-fair federated learning (q-FFL), which improves performance fairness by placing greater emphasis on devices with larger losses. Specifically, q-FFL mitigates performance disparities by weighting devices' losses using a parameter q (where $q \geq 0$), which controls the emphasis on high-loss clients. A larger q leads to a stronger focus on fairness across clients. However, the appropriate value of q must be determined empirically, as it depends on the dataset characteristics and involves a trade-off between fairness and overall accuracy.

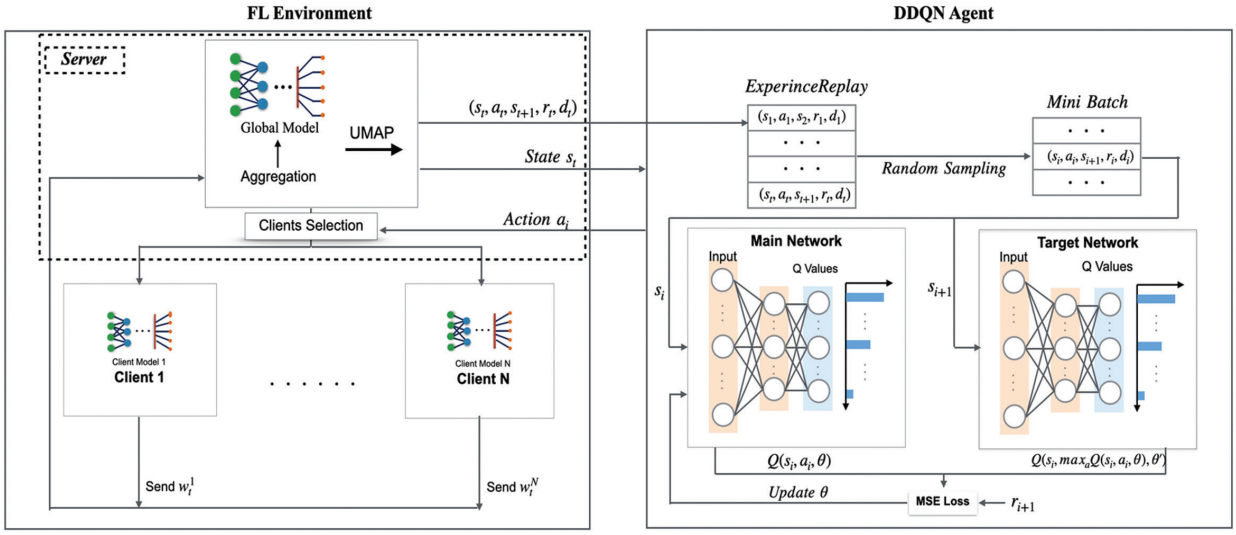


Fig. 1. Overview of the proposed method

Although the aforementioned methods represent noteworthy progress in addressing fairness and performance challenges in federated learning, they are predominantly constrained by their reliance on pre-defined parameters and their limited adaptability in highly heterogeneous and multiclass scenarios. In contrast, our approach introduces a dynamic device selection mechanism guided by reinforcement learning, which specifically prioritizes devices with lower predictive accuracy and incrementally enhances their performance over the course of the federated training process. A comprehensive comparative analysis, presented in the experimental evaluation section, benchmarks our method against the techniques described in [14, 29–30]. Although we did not directly compare our method with [28], we confirmed that it outperforms [29]. As [29] has been shown to achieve better performance than [28], we considered a comparison with [29] sufficient. The results consistently indicate that our method achieves superior performance fairness, particularly in environments characterized by pronounced non-IID conditions and complex multiclass data distributions.

FedHEAL [31] is a recently developed FL algorithm designed to address fairness issues in environments characterized by domain bias. It leverages the consistency of parameter updates to mitigate the impact of noisy or low-quality updates by masking the updates of unimportant parameters. Additionally, FedHEAL promotes fair model aggregation by utilizing the Euclidean distance, thereby preventing convergence bias often observed in conventional FL approaches. As a generic method, FedHEAL can be integrated with various existing FL algorithms.

3.4. ENHANCING PRIVACY FOR FL

Recent research has increasingly emphasized the need to fortify privacy safeguards within FL. While the distributed architecture of FL, which retains raw data on local devices, provides a baseline of privacy, it remains vulnerable to sophisticated inference and poisoning attacks.

Bietti et al. [32] introduced a personalized federated learning framework grounded in differential privacy [33]. Their study illustrates how personalized models can refine the trade-off between privacy and accuracy. However, they also acknowledge that tightening privacy guarantees inevitably results in diminished model performance.

Addressing the challenge of intermittent client participation, Jiang et al. [34] proposed Dordis, a distributed differential privacy framework resilient to client dropout. This approach achieves robust privacy protection without relying on a trusted central server, although the noise required for differential privacy introduces an unavoidable computational overhead.

Naseri et al. [35] explored the complementary use of local differential privacy (LDP) [36-37] and central differential privacy to mitigate both backdoor and membership inference attacks in FL. Their findings confirm that while these privacy techniques can enhance system resilience, they do so at the cost of reduced utility in the trained models.

In a related vein, Qi et al. [38] examined the susceptibility of differentially private FL (DPFL) to poisoning attacks. To counter this, they developed Robust-DPFL, which augments resilience to poisoned gradients. While their method successfully improves robustness, it introduces added complexity into the FL pipeline.

Collectively, these studies underscore that although FL inherently offers a foundational level of privacy, augmenting it with advanced privacy-preserving techniques frequently entails a trade-off with model accuracy and system complexity. The method proposed in this study is compatible with such techniques and can be integrated where stronger privacy assurances are necessary. Nonetheless, the empirical validation of this integration remains an open avenue for future investigation.

4. METHODOLOGY

In this section, we describe the proposed method for improving the performance of the bottom B% of devices in FL by integrating reinforcement learning. Here, B is a tunable parameter that specifies the proportion of devices with the lowest individual accuracies, which we particularly aim to support. This metric serves as an indicator of fairness, emphasizing performance improvement for underperforming clients. As illustrated in Fig. 1, the proposed method incorporates device selection using DDQN within an FL framework. Unlike existing methods, our approach adopts reinforcement learning to enhance device selection. Specifically, we employed uniform manifold approximation and projection (UMAP) for dimensionality reduction, transforming high-dimensional model weights into lower-dimensional representations while retaining essential information. In addition, we designed a reward mechanism based on the distance from the global model to discourage the selection of low-accuracy devices. This strategy enables efficient model construction, even in environments with significant disparities in data distribution across devices.

The workflow of the proposed method is presented in Algorithm 1. In each round, the reinforcement learning agent selects the optimal devices and transmits the global model to these devices. The selected devices then perform training on their local datasets. Finally, the central server aggregates the models sent by the selected devices to update the global model. The agent updates its parameters based on the received rewards, which are designed to minimize the selection of low-accuracy devices.

Algorithm 1: FL with DDQN for Device Selection

Initialize:

for each device k **do**

Device k trains local model w_0^k for 1 epoch with local dataset.

Send updated weights w_0^k to the server.

end for

Server performs dimensionality reduction on $\{w_0^k\}$ using UMAP to obtain initial state s_0.

Initialize DDQN agent with initial state s_0.

for each communication round $t = 1$ to T **do**

DDQN agent selects K devices based on the Q-values.

for each selected device k **do**

Send w_t to the device k.

Device k trains local model w_{tk} for E epochs with local dataset.

Send updated weights $w^k_{\{t+1\}}$ to the server.

Aggregate global model: $w_{\{t+1\}} = 1/C_t \sum_{\{k \in C_t\}} w^k_{\{t+1\}}$

Select all devices to calculate rewards.

for each device k **do**

Send $w_{\{t+1\}}$ to the device k.

Device k tests model $w_{\{t+1\}}$ on local test dataset and calculates accuracy acc_t^k.

Send accuracy acc_t^k back to the server.

end for

Aggregate global model: $w_{\{t+1\}} \leftarrow 1/|C_t| \sum_{k \in C_t}$

Select all devices to calculate rewards

for each device k **do**

Send $w_{\{t+1\}}$ to the device k.

Device k tests model $w_{\{t+1\}}$ on local test dataset and calculates accuracy acc_t^k.

Send accuracy acc_t^k back to the server.

end for

Calculate rewards r_t using accuracy acc_t^k equation (6).

DDQN Agent Do:

Update the DDQN parameters θ_t by minimizing the loss $L_t(\theta_t)$.

end for

Table 1 lists the symbols and descriptions used in this study.

Table 1. Notation

Symbol	Definition	Description
N	Total number of devices	The total number of devices
K	Number of selected devices	The number of devices selected in each round
C_t	Set of devices selected in round t	The set of K devices selected in round t
a_i	Action i	The action of selecting device i
A	Action space	The set of possible device selections
r_t^k	Reward	The reward for selecting device k in round t
γ	Discount factor	The importance of future rewards
θ	Parameters of the main network	The weights of the neural network being trained
θ'	Parameters of the target network	The fixed weights of the target network
w_t^k	Weight in round t	The weights of device k's model in round t

4.1. DDQN-BASED DEVICE SELECTION

To apply reinforcement learning to device selection, we formulated the Markov decision process.

- **State:**

The state at round t, namely s_t, is represented as a vector

$s_t = (w_t, wt^{(1)} \ldots, w_t^{(N)}$ where w_t represents the global model weights after round t, and $wt^{(1)}, \ldots, w_t^{(N)}$ represent the local model weights of all N devices. The agent is colocated with the FL server and holds a list of weights. A specific $w_t^{(k)}$ is updated only in round t if device k is selected for training and the resulting $\Delta t^{(k)}$ is received by the FL server. Consequently, the state space can become very large, making learning in such a space difficult. Therefore, we applied UMAP to compress the weights of each model into a 10-dimensional space, reducing the size of the state to 10×(N+1) dimensions.

- **Action:**

Actions (a) are represented as vectors of N Boolean values, where a value of 1 indicates a selection:

$$a_t = \{i\} \times N, \ where \ \ i \in \{0,1\} \tag{3}$$

As described in the subsequent section "Application of DDQN," in standard reinforcement learning, a subset of size K must be selected at each round t, resulting in $\binom{N}{K}$ possible combinations. However, in FL, this makes the action space enormous, causing computational costs to skyrocket, thus making the application infeasible. Therefore, we utilized the multi-action selection approach proposed by Bouaziz et al. [22], to allow multiple actions to be selected and learned simultaneously. This method treats each device selection as an independent action, significantly improving computational efficiency.

- **Reward:**

Rewards (r) are represented as a set of length $|C_t|$:

$$r_t = \{r_t^k \mid k \in C_t\} \tag{4}$$

where ζ_t^k measures the contribution of the local model of device k in round t relative to the global model:

$$\zeta_t^k = \frac{1}{\sqrt{|w_t - w_t^k|_2^2 + 1}} \tag{5}$$

$$r_t^k = M^{(TargetACC - BottomACC)} \cdot \zeta_t^k \tag{6}$$

A small value of Equation (5) indicates a large difference (Euclidean distance) between the weights of the device's local model and server's global model. In such cases, the device is considered unimportant, resulting in smaller ζ_t^k and reward r_t^k. This is because the local model weights of the selected device are generated through an aggregation process that combines the weighted sums. Devices with small differences between their local and global model weights are assumed to make significant contributions in a round, thereby substantially affecting the performance of the global model. M is a constant; $TargetACC$ represents the target average accuracy of the bottom B% of devices within a specified number of communication rounds, and $BottomACC$ denotes the average accuracy of the bottom B% of devices when all devices perform testing using the aggregated global model in each round t. This process provides an important metric for the agent to learn actions that improve the accuracy of the bottom devices. As the average accuracy of the bottom B% of devices increases, the agent is more likely to receive rewards, encouraging actions that enhance the fairness among devices. Calculating the test accuracy for all the devices introduces additional computational costs, with each device potentially being selected for up to twice the number of rounds. However, because the test accuracy calculation is not part of the learning process, the load on the devices is relatively small. This method allows the agent to effectively improve the average accuracy of the bottom B% of devices.

4.2. APPLICATION OF DDQN

In this study, following multiple existing studies, we employed the DDQN algorithm [39], which consisted of two neural networks: the main and target networks. The main network is used for training, whereas the target network evaluates the actions in the next state and is updated every P steps. The DDQN agent incorporates a replay memory mechanism to eliminate correlations between consecutive experiences, specifically between(s_i, a_i, s_{i+1}, r_i, d_i) and (s_{i+1}, a_{i+1}, s_{i+2}, r_{i+1}, d_{i+1}); d_i is Boolean and indicates whether the terminal state has been reached.

The RL learning problem can be formulated by minimizing the mean squared error (MSE) loss between the target value and the approximated value, expressed by the following equation:

$$L_t^k(\theta_t) = \left(Y_t^k - Q(s_t, a_k; \theta_t)\right)^2 \quad (7)$$

where $L_t^k(\theta_t)$ represents the loss function for action a_k; Y_t^k is the target value for action a_k, and $Q(s_t, a_k; \theta_t)$ is the approximated Q-value for action a_k in state s_t. Target value Y_t^k is defined as follows:

$$Y_t^k = r_t^k + \gamma\, Q\left(s_t, \arg\max_{a_i \in \mathcal{A}} Q(s_t, a_i; \theta_t); \theta_t'\right) \quad (8)$$

The action space A is defined as:

$$\mathcal{A} = \prod_{i=1}^{N} 0,1 \quad (9)$$

Each element a_i indicates selection 1 or non-selection 0. However, in each round, the following constraints must be satisfied:

$$\| a_t \|_1 = K \quad (10)$$

where r_t^k is the reward associated with the selection of device k; θ and θ' represent the parameters of the main and target networks, respectively, and γ is the discount factor, with $0 \leq \gamma \leq 1$, determining the importance of future rewards compared to current rewards. A value closer to 1 place more emphasis on future rewards, whereas a value closer to 0 prioritizes current rewards.

In traditional reinforcement learning, the goal is to select a single optimal action for a given state. However, in this study, we adopted a multi-action selection approach to select multiple devices. Specifically, the selection of each device was treated as an independent action, and the loss for each device was calculated using Equation (7). This approach eliminates the need to explore all possible device combinations.

This method allows for efficient identification of optimal devices while considering the cooperative relationships and interactions among the devices. Furthermore, by evaluating the impact of each device selection on the overall learning outcomes, more effective learning is expected.

5. EVALUATION

5.1. EXPERIMENTAL SETUP

The datasets and models used in this study are as follows.

- **MNIST:**

The dataset consists of 60,000 grayscale images of handwritten digits for training and 10,000 images for testing. Each image has a resolution of 28 × 28 pixels and is classified into one of 10-digit classes (0–9). Due to its simplicity, balanced class distribution, and ease of implementation, MNIST is one of the most used benchmark datasets in FL research. In this study, it was adopted to enable comparison with existing methods and to validate the effectiveness of the proposed method under standard and relatively simple experimental conditions.

For the model, we used a simple multilayer perceptron (MLP) consisting of one hidden layer with 100 units and ReLU activation. The input layer had 784 dimensions (28 × 28), and the output layer had 10 units corresponding to the number of classes.

- **CIFAR-10:**

The CIFAR-10 dataset is a widely used standard benchmark consisting of 60,000 32 × 32 pixel color images classified into 10 classes. Each class contains 6,000 images. This dataset is extensively used in machine learning research, including comparative methods, and was thus adopted in this study. In addition, to introduce heterogeneity, we performed non-IID partitioning following a Dirichlet distribution. The parameter values used were $Dir(0.1)$ and $Dir(0.5)$.

The model used for this dataset was a simple convolutional neural network composed of two convolutional layers and three fully connected layers. Specifically, the first convolutional layer used 16 filters with a kernel size of 3×3, followed by a 2×2 max pooling layer. The second convolutional layer had 32 filters with a kernel size of 3×3, followed by another 2×2 max pooling layer. The fully connected layers had 120, 84, and 10 units respectively.

- **GTSRB:**

The German Traffic Sign Recognition Benchmark (GTSRB) is one of the primary datasets for traffic sign recognition and classification tasks and contains approximately 50,000 images classified into 43 different traffic sign classes. Each image was captured in a real road environment, encompassing variations in lighting conditions and viewpoints. The GTSRB is commonly used in FL research [40-42]. In this study, in using a dataset with 43 classes, which exceeds the 10 classes of CIFAR-10, we aimed to evaluate the model's classification ability and its adaptability to heterogeneity more thoroughly. This enabled a multifaceted validation of the versatility and performance of the proposed method. Additionally, non-IID partitioning was performed following a Dirichlet distribution to introduce heterogeneity. The parameter values used were Dir(0.1) and Dir(0.5).

For this dataset, we used a simple multilayer perceptron consisting of an input layer with 3072 dimensions (32 × 32 × 3), a 128-dimensional hidden layer with ReLU activation, and an output layer with 43 units corresponding to the number of traffic sign classes. This model design follows the experimental settings of Li et al. [30] and Jialuo et al. [43].

- **Synthetic:**

The synthetic dataset is generated using the method inspired by Li et al. [30] and Shamir et al. [44], denoted as $SYNTHETIC(\alpha, \beta)$.

Specifically, the data samples (X_k, Y_k) for device k (with sample size n_k) were generated as follows: The model is defined by the following equation:

$$y = argmax\left(softmax(W_k x + b_k)\right) \quad (11)$$

where $x \in R^{60}$, $W_k \in R^{10\times60}$, and $b_k \in R^{10}$. The weight matrix W_k and bias vector b_k were sampled from a normal distribution with mean u_k and variance 1:

$$W_k \sim \mathcal{N}(u_k, 1), \quad b_k \sim \mathcal{N}(u_k, 1) \tag{12}$$

The mean vector u_k was sampled from a normal distribution with mean 0 and variance α:

$$U_k \sim \mathcal{N}(0, \alpha) \tag{13}$$

Each element of the input data x_k, denoted by $(x_k)_j$, was sampled from a normal distribution with mean v_k and variance $j^{-1.2}$:

$$x_{kj} \sim \mathcal{N}(v_k, j^{-1.2}) \tag{14}$$

where v_k is sampled from a normal distribution with mean μ_k and variance 1, and μ_k followed a normal distribution with mean 0 and variance β:

$$v_k \sim \mathcal{N}(\mu_k, 1), \quad \mu_k \sim \mathcal{N}(0, \beta) \tag{15}$$

This method allows controlling the heterogeneity of models and data across devices by adjusting parameters α and β. *SYNTHETIC*(0,0) and *SYNTHETIC*(1,1) were used in the experiments. Both had 10 classes and a data size of approximately 50,000. In this study, synthetic data were generated and used to control for heterogeneity and evaluate the changes in the performance of the proposed method by varying the degrees of heterogeneity. For this dataset, we used a logistic regression model following the experimental settings of Li et al. [30] and Jialuo et al. [43]. The model consisted of a single fully connected (linear) layer that takes a 100-dimensional input vector and outputs scores for 10 classes.

5.2. COMPARISON METHODS

We selected the following methods as baselines:

- **FedAVG [14]**: This is adopted as the basic method to evaluate the baseline performance against Non-IID data.
- **FedFE [29]**: This is a method that uses momentum gradient descent to improve convergence speed while considering fairness. The parameter settings used (α, β)=(0.5,0.5), were based on the optimal values in the experiments of Wentao et al. [29].
- **q-FFL [30]**: This is adopted to reduce performance disparities among devices, using $q = 1$ for the synthetic dataset and $q = 0.1$ for other datasets. These settings were determined based on the optimal values in the experiments of Li et al. [30].
- **FedHEAL [31]**: We adopted this method, a state-of-the-art FL method aimed at improving fairness. Following the experimental settings reported by Chen et al. [31], we set the parameters to (β, τ) = (0.4, 0.1), which demonstrated the best performance in their experiments.

The proposed method was trained using the hyperparameters listed in Table 2. These values were chosen based on common practices in the federated learning literature [29-31]. In particular, the number of local epochs was selected within the typical range of 1 to 10, which is widely adopted in prior studies [29-31]. The value of B was determined based on the experimental results reported by Wentao et al. [29].

Table 2. Hyperparameters of Experiments

Hyperparameters	MNIST	CIFAR10/GTSRB	SYNTHETIC
N (number of devices)	100	100	100
K (size of selected devices)	10	10	10
E (local epochs)	5	10	5
B (batch size)	16	32	32
Learning rate	0.01	0.01	0.1
Momentum	0.9	0.9	0.9
RL batch size	50	50	50
P (number of steps)	10	10	10
RL learning rate	$10e^{-5}$	$10e^{-5}$	$10e^{-5}$
γ (discount factor)	0.99	0.99	0.99
M	1.01	1.01	1.01

For each dataset, the target accuracy (*TargetACC*) was set based on existing FedFE [29] and q-FFL [30] methods. Specifically, experiments were conducted using these methods, and the average accuracy of the bottom B% of devices (*BottomACC*) was measured. Based on these results, *TargetACC* was set to a value that exceeded the *BottomACC* achieved by q-FFL and FedFE by a few percent.

- MNIST (0.1): 85%
- MNIST (0.5): 90%
- CIFAR-10(0.1): 37%
- CIFAR-10(0.5): 47%
- GTSRB (0.1): 77%
- GTSRB (0.5): 8%
- SYNTHETIC (0,0): 15%
- SYNTHETIC (1,1): 10%

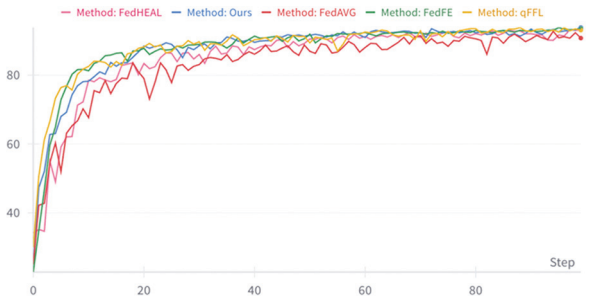
5.3. RESULTS & DISCUSSION

Fig. 2 and Table 3 present the progression and final outcomes of the accuracy of each method across the datasets used in this study. The evaluation metrics employed included the average accuracy of each device's local test data, variance and average accuracy of the bottom 10% of the devices, as well as top 10% of the devices.

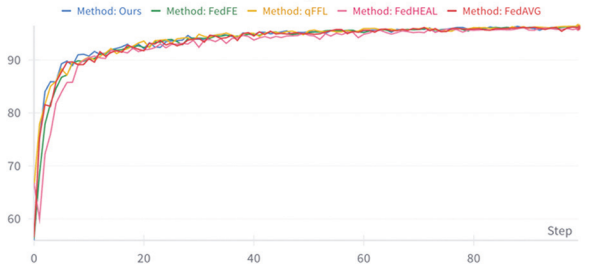
The proposed method successfully enhanced the accuracy of the bottom 10% of the devices across all datasets, without compromising the overall average accuracy. In some datasets, variance of the accuracy was reduced compared with those of existing methods (Table 3). Notably, on CIFAR-10 with a Dirichlet parameter of 0.1 CIFAR-10(0.1), which represents a highly non-IID environment, the improvement in the accuracy of the bottom 10% of the devices was particularly significant.

Dataset	Method	Variance	Worst 10%	Accuracy	Best 10%
MNIST(0.1)	FedAVG	36.1 ±18.9	79.5±3.3	90.7 ±2.6	99.1+.8
	q-FFL	30.5 ±16.9	81.4±4.8	93.1+.8	99.4±.4
	FedFE	17.7+3.5	85.2±1.8	93.5 ±.3	99.4±.4
	FedHEAL	**14.7 ±2.2**	86.1 ±.7	93.7±.3	99.3±.3
	Ours	15.1±3.9	**86.2 ±1.0**	**93.9 ±.4**	**99.4±.1**
MNIST(0.5)	FedAVG	5.8+1.8	91.1+1.3	96.1 ±.4	99.3±.1
	q-FFL	**4.0 ±.8**	**92.4 ±.6**	**96.4 ±.2**	**99.2±.1**
	FedFE	4.1 ±.5	92.3 ±.4	96.2±.2	99.1+.1
	FedHEAL	5.6+1.5	91.2±1.0	95.9±.2	99.0±.2
	Ours	4.2 ±.6	92.3±.5	96.3 ±.3	99.2±.2
CIFAR10(0.1)	FedAVG	152.7+32.5	30.4±3.2	51.3+1.0	72.0±2.0
	q-FFL	191.9+14.5	30.6 ±1.2	52.8 ±1.5	**78.7±2.3**
	FedFE	**123.4+3.7**	34.7 ±1.3	53.5+1.7	73.0±1.2
	FedHEAL	145.3+35.3	29.4 ±3.9	48.8±2.8	71.0±2.9
	Ours	130.8+11.3	**38.7 ±1.0***	**54.7+.9**	78.4±1.7
CIFAR10(0.5)	FedAVG	63.3+2.4	42.5 ±.7	56.9 ±.5	**70.7 ±.6**
	q-FFL	56.9+3.7	43.0±1.0	57.0+1.0	69.8 ±.9
	FedFE	**51.5+5.6**	43.8±.8	56.6±1.0	69.1 + 1.3
	FedHEAL	52.6+8.8	41.7±1.2	54.5+.8	66.8±1.8
	Ours	51.4+3.8	**44.8 ±.9**	**57.6 ±1.0**	69.7+1.5
GTSRB(0.1)	FedAVG	82.1+24.8	68.0±5.0	86.9+1.5	97.7 ±.4
	q-FFL	70.6+20.1	69.4 ±4.0	88.0±.5	97.9+.4
	FedFE	59.6+13.5	70.3 ±2.2	86.9+1.6	96.5±1.6
	FedHEAL	78.4+23.4	66.0 ±4.6	84.9±2.5	95.9+1.4
	Ours	**44.7 ±6.5**	**75.5 ±1.4**	**89.9+.4**	**98.1 ±.4**
GTSRB(0.5)	FedAVG	38.5+20.1	70.4+11.1	81.2±8.5	90.7+5.4
	q-FFL	**9.0 ±1.1**	87.7±.9	93.5+ .5	97.9 ±.7
	FedFE	17.9+2.7	81.6+2.7	89.6±1.5	96.1 + 1.1
	FedHEAL	16.4±1.4	82.5 ±1.2	90.4+1.0	96.2±.6
	Ours	9.2+1.2	**87.9+.8**	**93.6 ±.4**	**98.3±.4**
SYNTHETIC(0,0)	FedAVG	1429.7±35.3	o.o±.o	34.3+1.8	99.9±.1
	q-FFL	849.6 +42.0	11.0+1.0	69.2±1.0	**100.0 ±.0**
	FedFE	893.1+26.1	12.1±1.1	71.4+.7	**100.0 ±.0**
	FedHEAL	1075.8±70.8	0.1+.2	48.3±2.1	99.5±.7
	Ours	**824.6 + 55.4**	**15.1 ±.7**	**73.5+2.1**	**100.0 ±.0**
SYNTHETIC(1.1)	FedAVG	1405.5 ±74.1	0.0+.0	34.0 ±1.4	**100.0 ±.0**
	q-FFL	1024.6 ±46.3	7.8 ±2.0	68.4+2.1	**100.0 ±.0**
	FedFE	1044.5 ±46.3	6.7+1.1	70.4±2.4	**100.0 ±.0**
	FedHEAL	1423.1 ±36.3	o.o±.o	48.0+3.7	**100.0 ±.0**
	Ours	**926.5 ±71.8**	**10.5+0.5**	**73.8±2.0**	**100.0 ±.0**

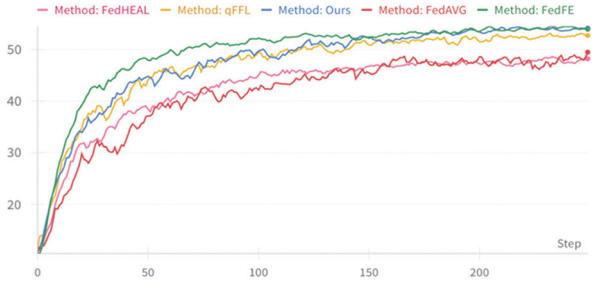
*Indicates statistically significant differences (p<0.05) between the proposed method and other methods for the corresponding metric



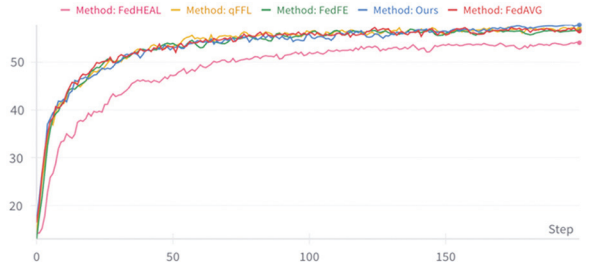
(a)



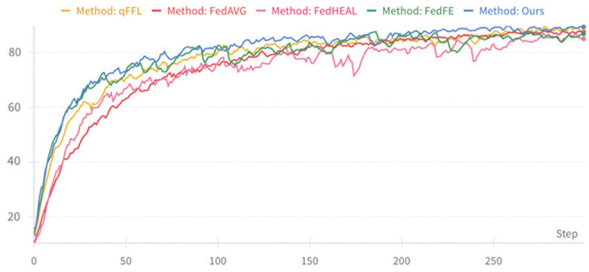
(b)



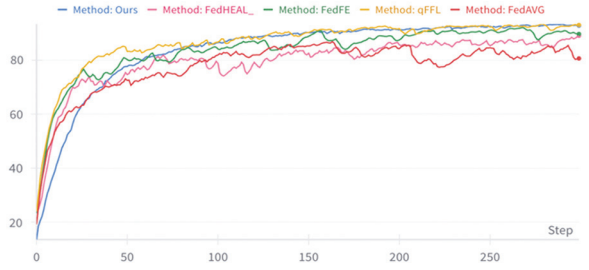
(c)



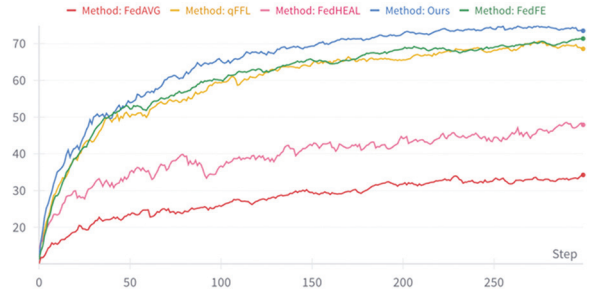
(d)



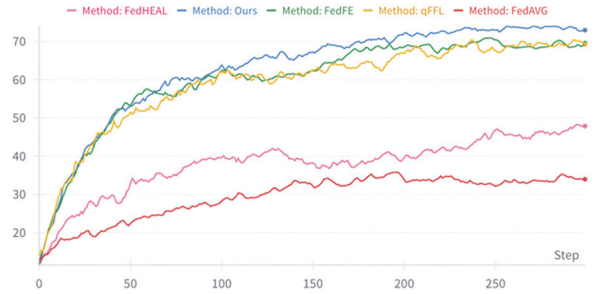
(e)



(f)



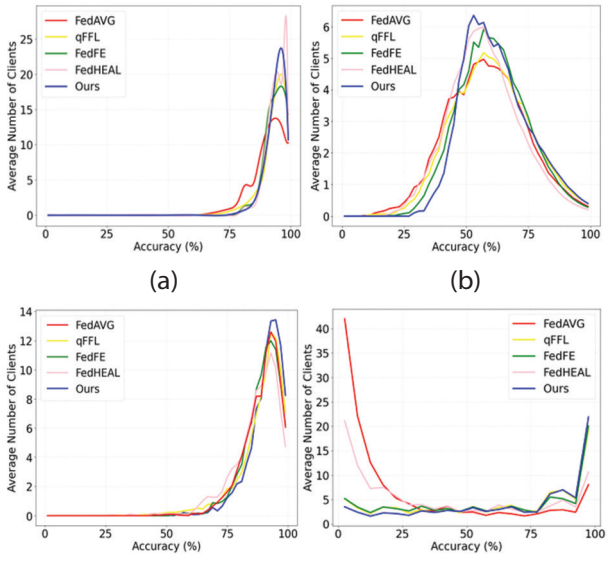
(g)



(h)

Fig. 2. Accuracy progression charts of average accuracy for each of the six datasets. **(a)** MNIST (0.1), **(b)** MNIST (0.5), **(c)** CIFAR-10(0.1), **(d)** CIFAR-10(0.5), **(e)** GTSRB (0.1), **(f)** GTSRB (0.5), **(g)** SYNTHETIC (0,0), **(h)** SYNTHETIC (1,1)

A partial distribution of the accuracy is shown in Fig. 3. This figure was derived from datasets with stronger non-IID characteristics, displaying the most pronounced improvements. Visually, the number of low-performing devices has clearly decreased compared with respect to the baseline methods.



(a) (b)

(c) (d)

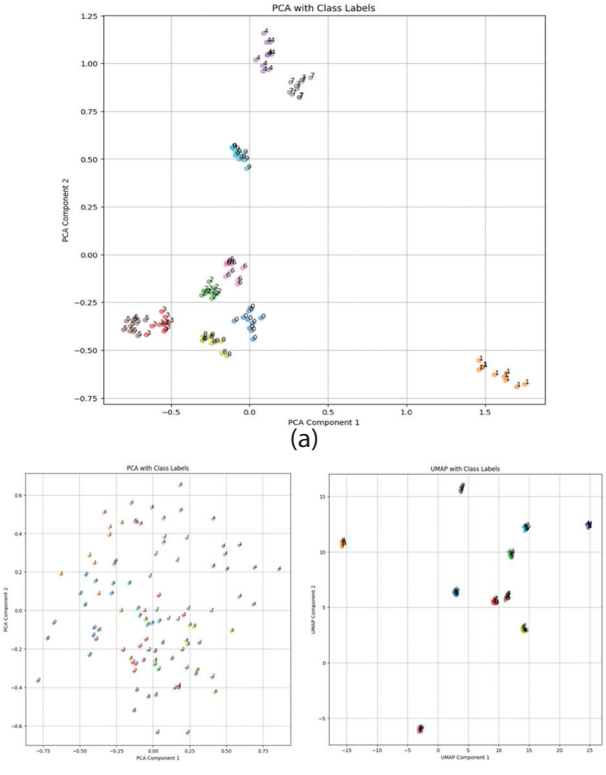
Fig. 3. Accuracy distribution map **(a)** MNIST(0.1), **(b)** CIFAR-10(0.1), **(c)** GTSRB(0.1), **(d)**SYNTHETIC(1,1)

In this study, we used UMAP for dimensionality reduction of the model weights of each device to better capture the underlying distribution among devices.

To evaluate its effectiveness, we used the MNIST dataset and introduced varying degrees of label imbalance among devices by controlling a parameter Z, which denotes the proportion of a single dominant label in each device's data. For instance, Z=80 indicates that 80% of the data within a device belong to one specific label, while the remaining 20% are uniformly distributed among the other labels. A setting of Z=100 represents extreme label concentration (single-label scenario), whereas Z =10 corresponds to a fully IID scenario, with all ten MNIST labels evenly represented.

We visualized the model weights after one epoch of local training and reduced them to two dimensions using both PCA and UMAP. While PCA was able to reveal some cluster structure under highly imbalanced settings (Fig. 4 (a)), it struggled to clearly separate clusters when the distribution became more subtle (Fig. 4 (b)). In contrast, UMAP consistently provided clearer and more distinct cluster formations, even in moderately complex distributions (Fig. 4 (c)). This suggests that UMAP captured the latent structures in the model weights more effectively than PCA.

By reducing the weights to 10 dimensions and using them as state representations for reinforcement learning, our method allowed the agent to more accurately distinguish between devices with different underlying data characteristics. This contributed to more effective device selection and, ultimately, better performance under heterogeneous data distributions.



(a)

(b) (c)

Fig. 4. Dimensionality Reduction of Local Model Weights. **(a)**with PCA (Z = 80), **(b)** with PCA (Z=20), **(c)** with UMAP (Z=20)

Additionally, Fig. 5 shows the number of selections for each device in CIFAR-10(0.1). As observed, the reinforcement learning agent intentionally selected devices that would increase accuracy. When the devices are randomly selected, the number of selections X for a single device follows $X \sim Binomial(n=250, p=0.1)$ with expected mean $\mu = np = 25$ and $\sigma = \sqrt{(np(1-p))} \approx 4.74$ as standard deviation. Typically, assuming a normal distribution, approximately 99.7% of the data would lie within the range [10.78,39.22]. However, in Fig. 5, approximately ten devices fall outside this range, suggesting that the reinforcement learning agent intentionally increased the selection frequency of these devices. In addition, upon examining the data distribution of the most frequently selected devices in Fig. 5, these devices were observed to have the largest amounts of data among those possessing more than five classes. As illustrated, because the number of devices selected per round is limited to K, devices with more diverse and abundant data were chosen more frequently.

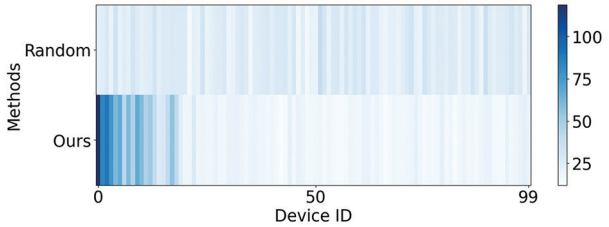


Fig. 5. Average number of device selections in CIFAR-10 (0.1)

6. CONCLUSION

This study introduces a novel approach designed to suppress the occurrence of low-accuracy devices in FL. The proposed method integrates reinforcement learning-based device selection using a DDQN and incorporates a reward mechanism based on the distance from the global model. Furthermore, it employs multi-action selection to choose multiple devices simultaneously, thereby ensuring an efficient selection process. By utilizing UMAP for the state representation, this method achieves both dimensionality reduction and enhanced representational capabilities.

The results indicate that the proposed approach effectively improves the average accuracy of the bottom 10% of the devices by up to approximately 4% without diminishing the overall average accuracy compared to existing methods. In addition, beyond the 10-class CIFAR-10 dataset, the method successfully suppressed low-accuracy devices in the GTSRB dataset, containing a greater number of classes. This demonstrates the versatility and effectiveness of the proposed method across diverse datasets.

In future research, we plan to extend the application of reinforcement learning beyond device selection to include weighted aggregation, with particular attention paid to the potential of multi-agent reinforcement learning. Moreover, addressing real-world challenges such as data heterogeneity, communication costs, and privacy concerns remains essential. Developing new algorithms that specifically aim to suppress low-accuracy devices in non-IID environments, while considering these practical constraints, is critical for ongoing and future investigations.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI, Grant Numbers JP22K12157, JP23K28377, and JP24H00714.

6. REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data", arXiv:1602.05629, 2016.

[2] S. Pati, U. Baid, B. Edwards, M. Sheller, S. H. Wang, G. A. Reina, L. Poisson, "Federated Learning Enables Big Data for Rare Cancer Boundary Detection", Nature Communications, Vol. 13, No. 1, 2022.

[3] T. Qi, F. Wu, C. Wu, L. He, Y. Huang, X. Xie, "Differentially Private Knowledge Transfer for Federated Learning", Nature Communications, Vol. 14, No. 1, 2023, p. 3785.

[4] N. Boscarino, R. A. Cartwright, K. Fox, K. S. Tsosie, "Federated Learning and Indigenous Genomic Data Sovereignty", Nature Machine Intelligence, Vol. 4, No. 11, 2022, pp. 909-911.

[5] C. Wu, F. Wu, L. Lyu, Y. Huang, X. Xie, "Communication-Efficient Federated Learning via Knowledge Distillation", Nature Communications, Vol. 13, No. 1, 2022, p. 2032.

[6] M. Asad, S. Shaukat, D. Hu, Z. Wang, E. Javanmardi, J. Nakazato, M. Tsukada, "Limitations and Future Aspects of Communication Costs in Federated Learning: A Survey", Sensors, Vol. 23, No. 17, 2023, p. 7358.

[7] X. Ma, J. Zhu, Z. Lin, S. Chen, Y. Qin, "A State-of-the-Art Survey on Solving Non-IID Data in Federated Learning", Future Generation Computer Systems, Vol. 135, 2022, pp. 244-258.

[8] K. Oishi, Y. Sei, Y. Tahara, A. Ohsuga, "Federated Learning Algorithm Handling Missing Attributes", Proceedings of the IEEE International Conference on Internet of Things and Intelligence Systems, Bali, Indonesia, 28-30 November 2023, pp. 146-151.

[9] S. Lin, Y. Han, X. Li, Z. Zhang, "Personalized Federated Learning Towards Communication Efficiency, Robustness and Fairness", in Advances in Neural Information Processing Systems, Vol. 35, 2022, pp. 30471-30485.

[10] B. R. Chaudhury, L. Li, M. Kang, B. Li, R. Mehta, "Fairness in Federated Learning via Core-Stability", in Advances in Neural Information Processing Systems, Vol. 35, 2022, pp. 5738-5750.

[11] Z. Yang, Y. Zhang, Y. Zheng, X. Tian, H. Peng, T. Liu, B. Han, "FedFed: Feature Distillation Against Data Heterogeneity in Federated Learning", Advances in Neural Information Processing Systems, Vol. 36, 2023, pp. 60397-60428.

[12] D. Pessach, E. Shmueli, "A review on fairness in machine learning", ACM Computing Surveys, Vol. 55, No. 3, 2022, pp. 1-44.

[13] S. Vucinich, Q. Zhu, "The Current State and Challenges of Fairness in Federated Learning", IEEE Access, Vol. 11, 2023, pp. 80903-80914.

[14] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data", Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, Vol. 54, 2017, pp. 1273-1282.

[15] C. Tian, Z. Shi, X. Qin, L. Li, C. Xu, "Ranking-Based Client Selection with Imitation Learning for Efficient Federated Learning", arXiv:2405.04122, 2024.

[16] Q. Pan, H. Cao, Y. Zhu, J. Liu, B. Li, "Contextual Client Selection for Efficient Federated Learning over Edge Devices", IEEE Transactions on Mobile Computing, Vol. 23, No. 6, 2023, pp. 6538-6548.

[17] T. Zhang, K. Y. Lam, J. Zhao, F. Li, H. Han, N. Jamil, "Enhancing Federated Learning with Spectrum Allocation Optimization and Device Selection", IEEE/ACM Transactions on Networking, Vol. 31, No. 5, 2023, pp. 1981-1996.

[18] J. Qi, Q. Zhou, L. Lei, K. Zheng, "Federated Reinforcement Learning: Techniques, Applications and Open Challenges", arXiv:2108.11887, 2021.

[19] H. Wang, Z. Kaplan, D. Niu, B. Li, "Optimizing Federated Learning on Non-IID Data with Reinforcement Learning", Proceedings of IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, Toronto, ON, Canada, 6-9 July 2020, pp. 1698-1707.

[20] W. Chen, J. Du, Y. Shao, J. Wang, Y. Zhou, "Dynamic fair federated learning based on reinforcement learning", Proceedings of the 5th International Conference on Data-driven Optimization of Complex Systems, Tianjin, China, 22-24 September 2023, pp. 1-8.

[21] H. Zhang, Z. Xie, R. Zarei, T. Wu, K. Chen, "Adaptive Client Selection in Resource Constrained Federated Learning Systems: A Deep Reinforcement Learning Approach", IEEE Access, Vol. 9, 2021, pp. 98423-98432.

[22] S. Bouaziz, H. Benmeziane, Y. Imine, L. Hamdad, S. Niar, H. Ouarnoughi, "FLASH-RL: Federated Learning Addressing System and Static Heterogeneity using Reinforcement Learning", Proceedings of the IEEE 41st International Conference on Computer Design, Washington, DC, USA, October 16-18, 2023, pp. 444-447.

[23] X. Yu, Z. Gao, Z. Xiong, C. Zhao, Y. Yang, "Ddpg-AdaptConfig: A Deep Reinforcement Learning Framework for Adaptive Device Selection and Training Configuration in Heterogeneity Federated Learning", Future Generation Computer Systems, Vol. 163, 2025, p. 107528.

[24] Y. Shi, H. Yu, C. Leung, "Towards Fairness-Aware Federated Learning", IEEE Transactions on Neural Networks and Learning Systems, Vol. 35, No. 9, 2024, pp. 11922-11938.

[25] T. H. Rafi, F. A. Noor, T. Hussain, D. K. Chae, "Fairness and Privacy Preserving in Federated Learning: A Survey", Information Fusion, Vol. 105, 2024, p. 102198.

[26] H. Chen, T. Zhu, T. Zhang, W. Zhou, P. S. Yu, "Privacy and Fairness in Federated Learning: On the Perspective of Tradeoff", ACM Computing Surveys, Vol. 56, No. 2, 2023, pp. 1-37.

[27] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, Q. Yang, "Federated Learning for Generalization, Robustness, Fairness: A Survey and Benchmark", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 46, No. 12, 2024, pp. 9387-9406.

[28] W. Huang, T. Li, D. Wang, S. Du, J. Zhang, "Fairness and Accuracy in Federated Learning", arXiv:2012.10069, 2020.

[29] P. Wentao, H. Zhou, "Fairness and Effectiveness in Federated Learning on Non-independent and Identically Distributed Data", Proceedings of the IEEE 3rd International Conference on Computer Communication and Artificial Intelligence, Taiyuan, China, 26-28 May 2023, pp. 97-102.

[30] T. Li, M. Sanjabi, A. Beirami, V. Smith, "Fair Resource Allocation in Federated Learning", arXiv:1905.10497, 2019.

[31] Y. Chen, W. Huang, and M. Ye, "Fair Federated Learning under Domain Skew with Local Consistency and Domain Diversity", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16-22 June 2024, pp. 12077-12086.

[32] A. Bietti, C. Y. Wei, M. Dudík, J. Langford, S. Wu, "Personalization Improves Privacy-Accuracy Tradeoffs in Federated Learning", Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 2022, pp. 1945-1962.

[33] C. Dwork, "Differential Privacy", Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, Venice, Italy, 10-14 July 2006, pp. 1-12.

[34] Z. Jiang, W. Wang, R. Chen, "Dordis: Efficient Federated Learning with Dropout-Resilient Differential Privacy", Proceedings of the Nineteenth European Conference on Computer Systems, Athens, Greece, 22-25 April 2024, pp. 472-488.

[35] M. Naseri, J. Hayes, E. De Cristofaro, "Local and Central Differential Privacy for Robustness and Privacy in Federated Learning", arXiv:2009.03561, 2020.

[36] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized Aggregatable Privacy-Preserving Ordinal Response", Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3-7 November 2014, pp. 1054-1067.

[37] Y. Sei, J. A. Onesimu, A. Ohsuga, "Machine Learning Model Generation with Copula-Based Synthetic Dataset for Local Differentially Private Numerical Data", IEEE Access, Vol. 10, 2022, pp. 101656-101671.

[38] T. Qi, H. Wang, Y. Huang, "Towards the Robustness of Differentially Private Federated Learning", Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20-27 February 2024, pp. 19911-19919.

[39] S. Latif, H. Cuayáhuitl, F. Pervez, F. Shamshad, H. S. Ali, E. Cambria, "A Survey on Deep Reinforcement Learning for Audio-Based Applications", Artificial Intelligence Review, Vol. 56, No. 3, 2023, pp. 2193-2240.

[40] M. Hasumi, T. Azumi, "Federated Learning Platform on Embedded Many-core Processor with Flower", Proceedings of the IEEE 3rd Real-Time and Intelligent Edge Computing Workshop, Hong Kong, Hong Kong, 13 May 2024, pp. 1-6.

[41] J. Lai, X. Huang, X. Gao, C. Xia, J. Hua, "GAN-Based Information Leakage Attack Detection in Federated Learning", Security and Communication Networks, Vol. 2022, No. 3, 2022, pp. 1-10.

[42] S. A. Khowaja, P. Khuwaja, K. Dev, A. Antonopoulos, "SPIN: Simulated Poisoning and Inversion Network for Federated Learning-Based 6G Vehicular Networks", Proceedings of the IEEE International Conference on Communications, Rome, Italy, 28 May - 1 June 2023, pp. 6205-6210.

[43] H. Jialuo, W. Chen, X. Zhang, "Reinforcement Learning as a Catalyst for Robust and Fair Federated Learning: Deciphering the Dynamics of Client Contributions", arXiv:2402.05541, 2024.

[44] O. Shamir, N. Srebro, T. Zhang, "Communication-Efficient Distributed Optimization Using an Approximate Newton-Type Method", Proceedings of the 31st International Conference on Machine Learning, 2014, Vol. 32, No. 2, pp. 1000-1008.