Utility Maximization of Multi-Federated Learning in Edge Computing with Personalized Privacy Preservation

Qian Chen†, Zhiwei Ni†, Jing Li‡, and Weifa Liang‡
† School of Management, Hefei University of Technology, Hefei, P.R. China
‡ Department of Computer Science, City University of Hong Kong, Hong Kong, P. R. China

***Abstract*—Edge intelligence enables mobile users to benefit from real-time inference services based on deep neural networks (DNN). Federated learning (FL) provides a solution for using DNN training while protecting privacy. FL over mobile edge computing (MEC) can aggregate models at the edge and process them in parallel, providing far more real-time results for real-world applications. However, edge nodes have limited computing capacities and bandwidth, and not all user equipments (UEs) can be selected to upload their trained local models. In addition, private information of users can still be leaked while attackers analyze the uploaded model parameters, and users' privacy requirements vary. Thus, we proposed a novel optimization framework - Federated learning with personalized differential privacy over MEC based on deep reinforcement learning. We use deep reinforcement learning (DRL) to maximize the total utility, i.e., the overall accuracy of all global FL models, by choosing UEs for uploading their updated local models due to limited bandwidth on access points (APs) and computing resource capacities on cloudlets (edge servers). Then, we inject differential private noise into local models to enhance privacy and satisfy users' personalized privacy requirements while guaranteeing model accuracy. We finally evaluate the performance of the proposed approach through experiments. Experimental results show that the proposed approach outperforms the comparison counterparts significantly, using public accessible datasets.**

***Index Terms*—Multiple federated learning, differential privacy, mobile edge computing, deep reinforcement learning**

I. Introduction

As a promising framework, Federated learning (FL) provides a solution to utilizing the deep learning model while protecting user privacy. Mobile edge computing (MEC) has become a promising paradigm for federated learning in the Internet of Things [1], which can process models in parallel to provide real-time results for real-world applications.

As edge servers, multiple cloudlets in the federated learning (FL) framework combine trained local models from mobile user equipments (UEs), and cloudlets have their own limited computing resource capacities. Due to limited bandwidth on access points (APs) of edge server nodes, updating FL models will lead to a communication constraint on UEs chosen by cloudlets. In addition, it suffers the risk of leaking user data between UEs and APs by uploading model parameters wirelessly [2], although FL does not share UE data with others by uploading the trained local models only. To further enhance the privacy of user data, differential privacy (DP) [3] is a promising technique that can better protect user privacy through noise perturbation to the uploading models. Moreover, users have personalized privacy requirements. Some users have higher demands on privacy but lower chances of their UEs being selected for uploading, while others have higher participation chances but lower privacy protections. One such example is that within the FL-based e-commerce recommendation models, professionals like lawyers and doctors have stricter privacy requirements than other consumers, who are willing to receive more accurate recommendations instead.

Choosing different UEs for uploading will result in different model accuracies [4], implying it is challenging to choose which model source data (UEs) for uploading at each training round. Thus, we need to choose UEs for uploading their local models to cloudlets adaptively in order to maximize the accumulative accuracy of global FL models while maintaining the fairness among local models. The privacy budget allocation affects the intensity of DP noise. Therefore, it is also challenging to allocate privacy budgets of DP to satisfy various user privacy requirements while ensuring the accumulative accuracy of global FL models is maximized by UE uploading.

Although FL has been studied extensively in MEC [1], the utility optimization in FL based on DP in an MEC is still in an early stage, which needs to be further explored. There are several studies on performance optimization in FL based on DP [5], [6], which did not consider the resource constraints on servers. Thus, these methods cannot be directly applied to MEC. Choosing UEs for FL has attracted lots of attentions by many researchers, e.g., there are studies focusing on making UE-chosen decisions for a single global model in FL [7], [8]. However, none of them considered multiple global models of FLs over MEC and with personalized privacy requirements.

The novelty of this paper lies in considering computing capacities on cloudlets, bandwidth capacities on APs, and personalized privacy budgets on DP noise, to optimize the accumulative accuracy of multiple federated learning models. A DRL method is devised to choose the appropriate number of UEs in each round to upload their updated local models.

In this paper, we propose a novel optimization framework - multiple Federated Learning with personalized Differential Privacy in a Mobile edge computing environment based on Deep reinforcement learning (FLDPMD), with the aim to

ICC 2025 - IEEE International Conference on Communications | 979-8-3315-0521-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICC52391.2025.11161610

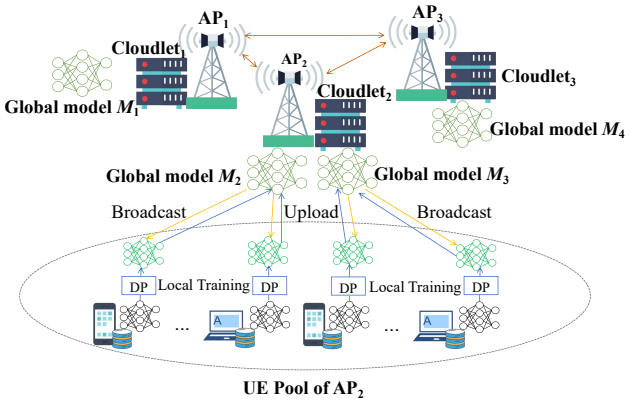


Fig. 1. An illustrative example of FL in an MEC network

maximize the accumulative accuracy of multiple FL models while satisfying personalized privacy requirements of users.

The main contributions of this paper are summarized as follows. We first propose a novel framework of multi-FL based on personalized DP over MEC. we also add the Gaussian noise to local models before uploading for global model training to protect user privacy, through allocating a proper noise volume to satisfy personalized privacy requirement. We then develop a DRL-based method for maximizing the total utility of multiple FL models by choosing UEs to upload their local models at each training round, under limited bandwidth capacity on each AP, limited computing capacity on each cloudlet, and the privacy budget for each local model uploading. We finally evaluate the performance of the proposed algorithm against its comparison benchmarks on public accessible datasets. Experimental results that demonstrate the proposed approach outperforms the comparison benchmarks.

II. Preliminaries

A. System model

We consider a mobile edge computing (MEC) network $G = (V, E)$, V is the set of Access Points (APs), and E is the set of links between APs. Each AP $v_j \in V$ has bandwidth capacity B_j, which is co-located with a cloudlet with computing capacity C_j. Each AP and its co-located cloudlet are connected via a high-speed optical cable, so the communication delay between them is negligible. An illustrative example of multiple FLs in an MEC network is shown in Fig. 1, where each AP provides wireless connections for UEs that are chosen to participate in multiple federated learning (MFL). Let each UE_i have an amount D_i of data stored at it, where i is the index of UEs with $1 \leq i \leq N$. Each UE_i uses its local dataset D_i for local training. The selection of UEs is achieved by maximizing the total utility of multiple FL systems while considering the resource constraints imposed by APs and Cloudlets. Each local model in chosen UEs is perturbed by the Gaussian noise of DP before uploading, which is detailed in Section II-C. Each cloudlet may contain different global models.

B. DP concepts and the FL paradigm

Definition 1 ((ϵ, δ)-differential privacy (DP)) [3]: Two datasets D and D' are neighbors (D and D' differ in at most one element). A privacy algorithm F gives (ϵ, δ)-DP if any outputs O of algorithm F on D and D' meet the following inequality.

$$Pr[F(D) = O] \leq exp(\epsilon) \times Pr[F(D') = O] + \delta. \quad (1)$$

where ϵ is the privacy budget and denotes the distinguishable bound of all outputs on adjacent datasets, and δ is the probability that cannot be bounded by ϵ.

Definition 2 (Sensitivity [3]): For F: $D \to \mathcal{R}^d$, the global sensitivity of function F is

$$\Delta f = \max_{D,D'} \parallel F(D) - F(D') \parallel_1, \quad (2)$$

where $\mathcal{R}$ represents the mapped real number space, and d denotes the query dimension of function F.

The noise addition mechanism is the primary technique to achieve differential privacy protection. We here adopt the Gaussian mechanism [9] that adds noise scaled to $\mathcal{N}(0, \sigma^2)$ to ensure (ϵ, δ)-DP, which is defined as follows.

Definition 3 (Gaussian mechanism [9]) Given a query $F : D \to \mathcal{R}^d$, the Gaussian mechanism is defined as

$$M(D) = F(D) + Y, \quad (3)$$

where $F(D)$ is a query function on dataset D, $M(D)$ is the returning result, and Y is the Gaussian noise term with $Y \sim \mathcal{N}(0, \sigma^2)$. The noise scale σ is the standard deviation of the Gaussian distribution with $\sigma \geq \frac{\sqrt{2\ln(1.25/\delta)} \cdot \Delta f}{\epsilon}$.

Multiple Federated Learning: We consider a FL system consisting of a set $M = \{M_1, M_2, \ldots, M_m\}$ of machine learning models to be trained, and the data set of each model is a subset of n UEs. In each training round t ($1 \leq t \leq T$) of a global FL model training, its local model at each chosen UE is trained by its local data, and all trained local models are then sent to the cloudlet hosting the global model for model aggregation.

Let $UE_M(k)$ be the set of UEs providing source data to model M_k, and let $UE_M(k, t)$ be the chosen subset of UEs to participate in model M_k training at round t, clearly $UE_M(k, t) \subseteq UE_M(k)$. We assume that each model $M_k \in M$ has been placed to a cloudlet $h(M_k)$. Given a chosen $UE_i \in UE_M(k, t)$, it trains the local model of each global model M_k in τ epochs for updating the local model, using its local data D_i and uploads its updated local model to cloudlet $h(M_k)$ for all models if UE_i is a source. For each model M_k, its hosting cloudlet $h(M_k)$ aggregates the updated local models at the cloudlet to form a new global model, and then the cloudlet $h(M_k)$ broadcasts the updated global model to every AP for the next round (i.e., $t+1$) of multiple FL training. This procedure continues until the last round T.

C. Local Model with personalized DP

To mitigate the privacy leakage of local users, adding the Gaussian noise to local model $w_{i,j,k,t}$ before uploading can provide privacy protection effectively. Thus, to achieve $(\epsilon_{i,j,k,t}, \delta_{i,j,k,t})$-DP, in a FL training at round t, the privacy budget $\epsilon_{i,j,k,t}$ $(0 < \epsilon_{i,j,k,t} < \epsilon^b_{i,j,k,t})$ allocated to $UE_i \in UE_M(k,t)$ by adding the Gaussian noise, a perturbed local model $\widetilde{w}_{i,j,t}$ at UE_i then is uploaded, where $\widetilde{w}_{i,j,t} = w_{i,j,k,t} + N_{i,j,k,t}$, where $N_{i,j,k,t} \sim \mathcal{N}(0, \sigma^2_{i,j,k,t})$. We denote $\sigma_{i,j,k,t} = \sqrt{2\ln(1.25/\delta_{i,j,k,t})}\Delta f/\epsilon_{i,j,k,t}$. $\delta_{i,j,k,t}$ is set to $\frac{1}{|D_i|}$. Δf is given by $\Delta f = \frac{2C}{|D_i|}$, where C is a predetermined clipping threshold [2].

The privacy budget in each training round is allocated equally [10]. Thus, the privacy budget $\epsilon_{i,j,k,t}$ is allocated to each local model at time slot t by the procedure as, $\sum_{i=1}^{|UE_M(k,t)|}\sum_{j=1}^{m}\epsilon_{i,j,k,t} = \epsilon/T$, where ϵ is the total privacy budget for the period T. The privacy budget for a local model can be set with respect to a user's personalized requirement when users have distinct privacy preferences [6]. In practice, it may be more reasonable to set appropriate values with user-friendly descriptors (e.g., low, medium, and high privacy) and users choose one from those categories. We here set three levels of privacy ϵ_L, ϵ_M, ϵ_H, i.e., $\epsilon_{i,j,k,t} = \{\epsilon_L, \epsilon_M, \epsilon_H\}$, which are calculated as, $\epsilon_L = \epsilon^b_{i,j,k,t}$, $\epsilon_M = \frac{\epsilon}{T\cdot|UE_M(k,t)|}$, $\epsilon_H = \frac{\frac{\epsilon}{T} - n_{\epsilon_L}\cdot\epsilon_H - n_{\epsilon_M}\cdot\epsilon_M}{n_{\epsilon_H}}$, where $\epsilon^b_{i,j,k,t}$ is the upper bound of the privacy budget for local model $w_{i,j,k,t}$. n_{ϵ_H}, n_{ϵ_M}, and n_{ϵ_L} are the numbers of users with personalized requirements. ϵ_H ensures the strictest privacy and ϵ_L has the least negative impact on the total utility (overall accuracy) of global models.

D. The subset of chosen UEs for model training at round t

Denote by $\mathbf{x}_{j,k,t} = [x_{1,j,k,t}, x_{2,j,k,t}, ..., x_{|UE(j,t)|,j,k,t}]$, $x_{i,j,k,t} \in \{0,1\}$, as the combination of UE-chosen decision for M_k, where $x_{i,j,k,t}$ is a binary variable indicating whether UE_i participating in model M_k training in cloudlet j at round t, if yes, $x_{i,j,k,t} = 1$, otherwise $x_{i,j,k,t} = 0$, and $\sum_{i=1}^{|UE(j,t)|} x_{i,j,k,t} \leq 1$. Not all UEs can be chosen because of limited computing capacity on each cloudlet and communication capacity on each AP. Different $UE_M(k,t)$ (i.e., different subsets of local models $w_{i,j,k,t}$) for training M_k result in a different accuracy of M_k at round t. Choosing proper local models for training M_k at each round t to obtain high accuracy of M_k can enhance the FL performance. Thus, we aim to choose a set $UE_M(k,t)$ at round t to maximize the total utility under limited resource capacities.

E. Problem Formulation

To improve the accuracy of model aggregation, we propose an algorithm for the UE choice problem, which is to maximize the accumulative utility of all global models within T rounds, subject to the computing resource capacity on each cloudlet, the communication resource capacity on each AP, and the given privacy budget. The related constraints on the optimization objective are (i) resource capacity: model uploading and aggregation consume computation and communication resources. The computing and bandwidth resource capacities on node V_j are C_j and B_j, which represent the computing resource capacity on cloudlet j and bandwidth resource capacity on AP j, respectively. (ii) privacy budget: the privacy budget ϵ can be set to control the noise volume to impact the privacy protection intensity. The upper bound on the privacy budget is the upper limit of privacy leakage. (iii) The optimization objective: we aim to maximize the accumulative utility gain of all models, i.e., we aim to maximize $\sum_{t=1}^{T}\sum_{k=1}^{m} u(k,t)$, where $u(k,t) \in [0,1]$ is the utility of global model M_k in round t, indicating the prediction accuracy of M_k in round t, which is defined as follows.

$$u(k,t) = v\left(\varphi_{k,t}\right) - \alpha_1 e^{-\alpha_2(\alpha_3|D(k)|)^{v\left(\varphi_{k,t}\right)}} \tag{4}$$

where $v\left(\varphi_{k,t}\right)$ reflects the performance deterioration of model M_k in round t with the increase on $\varphi_{k,t}$. $\varphi_{k,t}$ measures the model weight divergence. $v\left(\varphi_{k,t}\right) = \alpha_4 e^{-\left(\alpha_5 + \frac{\varphi_{k,t}}{\alpha_6}\right)^2} < 1$. $\alpha_1, \ldots, \alpha_6 > 0$ are positive curve fitting parameters. $|D(k)|$ is the size of data $D(k)$ for training model M_k in $UE_M(k,t)$. $D(k) = \sum_{j=1}^{|V|}\sum_{i=1}^{|UE(j,t)|} D_i \cdot x_{i,j,k,t}$ where $UE(j,t)$ is all available UEs that may be chosen to train model to cloudlet j at times lot t.

III. ILP FORMULATION

Let $y_{i,j,t}$ be a binary variable to indicate whether UE_i is chosen to upload its local model to cloudlet j at time slot t. If it does, then $y_{i,j,t} = 1$; otherwise zero. The problem optimization objective is to

$$\text{Maximize} \sum_{t=1}^{T}\sum_{k=1}^{m} u(k,t) \tag{5}$$

s.t. (4)

$$\epsilon_{i,j,k,t} \cdot y_{i,j,t} \leq \epsilon^b_{i,j,k,t}, \ \forall i,j,t \tag{6}$$

$$\sum_{i=1}^{|UE(j,t)|}\sum_{j=1}^{|V|}\sum_{k=1}^{m} \epsilon_{i,j,k,t} \cdot y_{i,j,t} \leq \epsilon/T, \ \forall t \tag{7}$$

$$\sum_{i=1}^{|UE(j,t)|}\sum_{k=1}^{m} comp(i,j,k,t) \cdot y_{i,j,t} \leq C_j, \quad \forall j,t \tag{8}$$

$$\sum_{i=1}^{|UE(j,t)|} b_{i,j,t} \cdot y_{i,j,t} \leq B_j, \quad \forall j,t \tag{9}$$

$$y_{i,j,t} \in \{0,1\}, \quad \forall i,j,t \tag{10}$$

$$y_{i,j,t} = 0 \quad \text{if } AP_j \text{ is not in the range of } UE_i \tag{11}$$

where $comp(i,j,k,t)$ is the amount of computing resource consumed by training local model in $UE_i \in UE_M(k,t)$ and sending the trained local model to cloudlet j for aggregation to form global model M_k. $b_{i,j,t}$ the amount of bandwidth allocated to UE_i if the UE is chosen to participate in FL for model training in cloudlet j at round t.

Constraint (6) ensures that the privacy budget allocated to each local model for generating a model in cloudlet j at any round is no greater than its upper bound. Constraint

(7) ensures that the privacy budget allocated to all local models for training in $(1, T)$ is no greater than the given ϵ. Constraint (8) ensures that the total computing resource demanded by all models at cloudlet j is no greater than its computing capacity C_j. Constraint (9) ensures that the total communication resource demanded by all UEs at AP j is no greater than its communication capacity B_j.

IV. Multi-Federated Learning with DP Requirements via Deep Reinforcement Learning

In this section, we propose a novel optimization framework - multiple federated learning with differential privacy requirements via deep reinforcement learning (FLDPMD), with the aim to maximize the accumulative accuracy of models while meeting individual user privacy requirements. To this end, we start with an overview of the framework. We then develop a DRL algorithm based on the proposed framework. We finally analyze the privacy reservation property of the proposed algorithm.

A. Overview of FLDPMD

The proposed federated learning framework FLDPMD trains multiple global models at the same time while ensuring user privacy. We utilize multiple cloudlets in an edge computing network to train multiple global models, where mobile users have chances to upload their local models to participate in different FLs. Then, we formulate the UE-choice problem and propose a deep reinforcement learning (DRL) algorithm for the accumulative utility (accuracy) maximization problem of all FL models under diverse resource constraints.

B. Differential private FL over MEC

In the following, we present the process of differential private multi-federated learning (FL) in a mobile edge computing (MEC) network in detail. As shown in `Algorithm` 1, Each cloudlet may host multiple FL models. In the initial training round $t = 0$, the parameters of each global model $M_{k,t}$ in its cloudlet have been initialized and broadcast to each chosen UE for updating its local model $w_{i,j,k,t}$. In each subsequent training round $t = 1, \cdots, T$, each local model is first trained and updated by clipping the gradients to the preset C in each local training epoch. Then, each chosen UE uploads its trained local model $w_{i,j,k,t}$ to the MEC network by adding the Gaussian noise to $w_{i,j,k,t}$ for privacy purposes. The uploaded local models are then used for the update of global model $M_{k,t}$. Finally, each local model at each UE is updated by the updated global model for the next round of training.

C. DRL algorithm for multiple FL model training

We formulate the accumulative utility maximization problem by a Markov Decision Process (MDP). We choose UEs for multi-FL model updating with the aim to maximize the accumulative utility of global models, subject to various resource constraints on the MEC network and the given privacy budget. We develop a deep reinforcement learning (DRL) algorithm for the problem based on a Deep-Q Network (DQN).

Algorithm 1 Differential private multi-FL over MEC

Input: An MEC network $G = (V, E)$, a given finite time horizon T, a local dataset D_i in each UE_i, preset clipping C, the local training epoch $l \in \tau$.

Output: A set of FL models $M = \{M_1, M_2, \ldots, M_m\}$ to be trained.

for $1 \le k \le m$ **do**
  UPDATE(UE_i, $M_{k,t}$):
  $t \leftarrow 0$;
  Initialize each global model parameters $M_{k,0}$;
  Update the local model: $w_{i,j,k,t} \leftarrow M_{k,0}$;
  for each round t with $1 \le t \le T$ **do**
    for each cloudlet j with $1 \le j \le |V|$ **do**
      for each UE_i with $1 \le i \le n$ **do**
        Obtain $UE_M(k,t)$ by `Algorithm` 2;
        if $UE_i \in UE_M(k,t)$ **then**
          for epoch l with $1 \le l \le \tau$ **do**
            Clip the local model parameters: $w_{i,j,k,t}(l) \leftarrow \frac{w_{i,j,k,t}(l)}{\max(1, \|w_{i,j,k,t}(l)\|/C)}$;
            Update the local model parameters: $w_{i,j,k,t}(l+1) \leftarrow w_{i,j,k,t}(l) - \eta \nabla f_{i,j,k,t}(l)$;
          end for
        end if;
      end for
      Produce the noise of DP to perturb the local model: $\overline{w}_{i,j,k,t} \leftarrow w_{i,j,k,t} + N(0, \sigma^2_{i,j,k,t})$;
      Update global models: $M_{k,t} \leftarrow \sum_{i=1}^{|UE_M(k,t)|} p_i \overline{w}_{i,j,k,t}$;
      Update local models: $w_{i,j,k,t+1} \leftarrow M_{k,t}, \quad \forall UE_i \in UE_M(k,t)$;
    end for
    $t \leftarrow t + 1$;
  end for
end for
return M.

1) MDP formalization: We formulate the problem to a MDP $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$. $\mathcal{S}$ is the state set of the environment, which represents global models, local models, available resources, and the allocated privacy budget at each training round t. $\mathcal{A}$ is the set of actions to choose UEs to participate in FL. $\mathcal{P}$ is the transition function, which is a probability from the current state to the next state. $\mathcal{R}$ is the reward function, which is the accumulative utility of global models. Given state $\mathbf{s}_t \in \mathcal{S}$ at time slot $t \in T$, an action $\mathbf{a}_t \in \mathcal{A}$ with $\mathbf{a}_t = \{a_{1,t}, a_{2,t}, \ldots, a_{n,t}\}$, that consists of UE choice decisions of n UEs. $\mathcal{P}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ is the probability of the state transition from $\mathbf{s}_t$ and its next state $\mathbf{s}_{t+1}$ through action $\mathbf{a}_t$. The following details the state, action, policy, and reward of an MDP for choosing UEs in federated learning.

State space:. According to the training process of multiple FLs, models will be updated at the end of each round of FL iterations. The state in round $t \in (0, T)$ is defined as $\mathbf{s}_t = \{\mathbf{M}_t, \mathbf{w}_t, \mathbf{C}_t, \mathbf{B}_t, \mathbf{e}_t, \mathbf{S}_{t-1}\}$, where $\mathbf{M}_t$ is the set of global models in cloudlets at round t, and $\mathbf{w}_t$ is the set of

local models that be chosen to participate in FL at round t. $\mathbf{C}_t$ is the set of available computation capacity of cloudlets at t. $\mathbf{B}_t$ is the set of available bandwidth of APs at t. $\mathbf{e}_t$ is the set of privacy budget for local models. $\mathbf{S}_{t-1}$ is the set of chosen UEs at round $t-1$ for FL of t.

Action space:. The action space is defined as $\mathbf{a}_t = [a_{1,t}, a_{2,t}, \ldots, a_{n,t}]$, where $a_{i,t} = \{y_{i,1,t}, y_{i,2,t}, \ldots, y_{i,j,t}, \ldots, y_{i,|V|,t}\}$, where $y_{i,j,t}$ is a binary variable to indicate whether UE_i sends its updating local model to cloudlet j at time slot t.

Policy: The policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ denotes the mapping between a state and an action, where π is a set of strategies, in which $\pi(a|s)$ represents the probability of taking action a for a certain state s, $\pi(a|s) = P(\mathbf{a}_t = a|\mathbf{s}_t = s)$.

Reward function:. If action $\mathbf{a}_t$ is applied in state $\mathbf{s}_t$, the RL agent receives a reward $\mathbf{r}_t$ for the global model M_k as $\mathbf{r}_t = \sum_{k=1}^{m} u(k,t)$, where $u(k,t)$ is calculated by Eq. (4).

2) DRL-based optimization: Traditional Reinforcement Learning (RL) is inefficient, due to calculating value functions for all possible pairs of states and actions. In contrast, the DRL leverages deep neural networks (DNNs) to solve the complex Markov Decision Process (MDP) model [4]. We address this MDP problem through a DRL framework - the Deep Q Network (DQN), which employs the replay buffer $\mathcal{B}$ to store the experience transformation, including the state $\mathbf{s}_t$, the action $\mathbf{a}_t$, the reward $\mathbf{r}_t$ at round t, and the next state $\mathbf{s}_{t+1}$.

Following the problem optimization objective in (5), we design a reward function to maximize the cumulative utility (accuracy) of all FL models, where the total reward of all models $\mathbf{r}$ for the period of $\mathbb{T}$ is defined as, $\mathbf{r} = \sum_{t=1}^{T} \sum_{k=1}^{m} \gamma \cdot u(k,t)$, where $\gamma \in (0,1]$ is constant, which is the reward discount factor with $0 < \gamma \leq 1$. The problem then is to choose a subset of UEs such that the total utility (accuracy) of multiple FL models in the given monitoring period T is maximized. A DRL agent is to maximize the cumulative reward, by exploring $\mathbf{a}$ as, $\mathbf{a} = \arg\max \mathbb{E}[\mathbf{r}]$. The proposed DRL algorithm is presented in `Algorithm 2`. The computational complexity is $O(|M| \cdot |UE_S|)$, where $|M|$ and $|UE_S|$ are the number of global models and selected UEs, respectively. Compared to decentralized DRL algorithms, the `Algorithm 2` has a higher computational overhead in large-scale networks, but has a superior global optimization capability.

Algorithm 2 DRL algorithm for the accumulative utility maximization problem

Input: An MEC network $G = (V, E)$, a given finite time horizon T, a DRL agent hyper-parameter π_θ.
Output: Optimize global models $M_1, \ldots, M_m$.

```
Initialize policy parameters θ, action-value function Q with θ, and buffer B;
t ← 1;
while t ≤ T do
    Extract state s_t
    Select action a_t = argMax_{a_t} Q(s_t, a_t; θ);
    Perform action a_t, and transform states from s_t to s_{t+1};
    Obtain Reward r_t;
    Store (s_t, a_t, r_t, s_{t+1}, e_t, d) to replay buffer B;
    Randomly sample a batch of transitions from B;
    Update θ
end while;
Obtain the action a_t in B;
if x_{i,j,k,t} = 1 then
    Add the current UE to set UE_M(k,t);
else if x_{i,j,k,t} = 0 then
    Exclude the current UE from set UE_M(k,t)
end if;
Obtain the privacy budget set e_t in B;
for all UE_i ∈ UE_M(k,t) in parallel do
    w_{i,j,k,t+1} ← UPDATE(UE_i, M_{k,t})
end for;
M_{k,t} ← Σ_{i=1}^{|UE_M(k,t)|} p_i w̄_{i,j,k,t}.
```

V. Performance evaluation

In this section, we evaluated the performance of the proposed algorithm FLDPMD against other benchmarks on popular datasets MNIST and CIFAR-10 , respectively. We mainly studied the utility of the mentioned approaches in training the above datasets (non-IID), and demonstrated the superiority of FLDPMD with the help of DP in an edge computing network.

A. Experimental environment settings

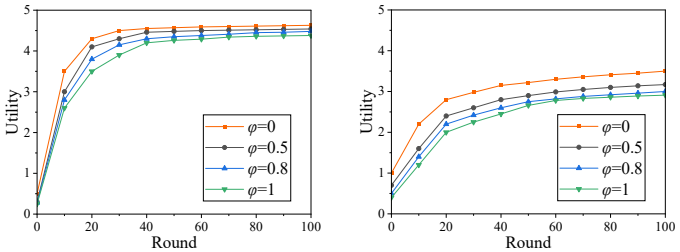
We set the number $|V|$ of APs at 5, and the privacy budget ϵ as 1, 5, and 10 respectively. Each dataset is partitioned into 100 segments with each being assigned to one UE, assuming that there are 100 UEs in the system. We set EMD φ with four different non-IID levels 0, 0.2, 0.4, and 0.8, repectively. We adopted two image classification datasets: MNIST and CIFAR-10. Two Concurrent Neural Network (CNN) models are used for training on the datasets. Each CNN model used consists of a softmax output layer and two 5 × 5 convolution layers. The Rectified Linear Unit (ReLU) activates each layer and normalizes the batch. The learning rates are set to 0.01 and 0.001, respectively. The batch sizes are set to 10 for two models. The reward discount factor γ is set to 0.9.

B. Performance of different algorithms

We compared the proposed algorithm FLDPMD with the following comparison algorithms. (1) `FedAvg-DP` [2] adopts the generic algorithm in FL $FedAvg$ by adding the perturbing DP noise to local models prior to uploading the local models, in which each edge server randomly selects UEs under its coverage for uploading local models. (2) `Kcenter-DP` selects UEs through the K-center clustering in FL [11] and perturbs DP noise to local models before uploading.

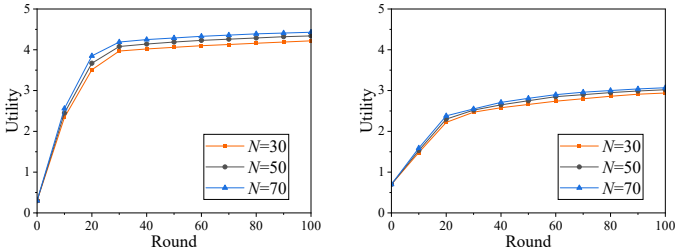
C. Results and Analysis

We investigated the impact of different levels of non-IID and the number of UEs on the performance of the FLDPMD algorithm FLDPMD. Then, we compared the performance of the three comparison algorithms by varying the privacy budget.



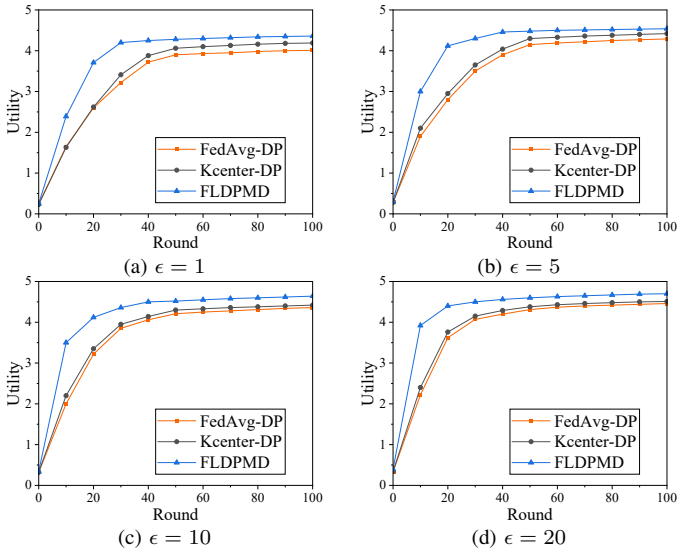
(a) MNIST (b) CIFAR-10

Fig. 2. The utility of FLDPMD by varying non-IID level φ



(a) MNIST (b) CIFAR-10

Fig. 3. The utility of FLDPMD by varying the number of UEs



(a) $\epsilon = 1$ (b) $\epsilon = 5$ (c) $\epsilon = 10$ (d) $\epsilon = 20$

Fig. 4. The utility of different algorithms by varying privacy budget ϵ

1) Performance evaluation: We evaluated the utility of the proposed algorithm FLDPMD in MNIST and CIFAR-10 data sets in each communication round, by varying φ when $\epsilon = 5$. Fig. 2 shows that FLDPMD converges well, and the utility of the global model decreases with the increase in value on φ. This is because the larger the value of φ, the higher the similarity of the data, providing a low-quality local model for global aggregations. We also studied the utility of FLDPMD in each communication round on MNIST and CIFAR-10, when the number of UEs is set at 30, 50, and 70, $\epsilon = 5$ and $\varphi = 0.5$, respectively. Fig. 3 shows that the utility of the FLDPMD algorithm increases slightly with more UE participations.

2) Performance comparison of different algorithms: We finally investigated the utility performance of FLDPMD against other comparison algorithms on the MNIST dataset, by varying the privacy budget ϵ at 1, 5, 10, and 20, respectively. As shown in Fig. 4, the proposed algorithm FLDPMD outperforms the other comparison algorithms with different privacy budgets ϵ. Furthermore, FLDPMD is also superior to others in convergence speed. In addition, compared with FedAvg-DP and Kcenter-DP, it exhibits a less negative impact on the utility but with higher privacy-preserving intensity (smaller ϵ).

VI. Conclusion

In this paper, we proposed a novel optimization framework of multi-federated Learning with personalized differential privacy in an MEC network via deep reinforcement learning. We considered multiple FL model training, using the DRL technique. We aimed to maximize the total utility by choosing UEs for uploading their trained local models due to limited bandwidth capacity on each AP. We also introduced DP noise into local models to meet user-personalized privacy requirements. Experimental results demonstrate that the proposed approach significantly outperforms these comparison baselines.

References

[1] Z. Xu, D. Zhao, W. Liang, O. F. Rana, P. Zhou, M. Li, W. Xu, H. Li, and Q. Xia, "Hierfedml: Aggregator placement and ue assignment for hierarchical federated learning in mobile edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 1, pp. 328–345, 2023.

[2] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[3] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, Berlin, Heidelberg, 2006, pp. 265–284.

[4] W. Yang, W. Xiang, Y. Yang, and P. Cheng, "Optimizing federated learning with deep reinforcement learning for digital twin empowered industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1884–1893, 2023.

[5] K. Wei, J. Li, M. Ding, C. Ma, H. Su, B. Zhang, and H. V. Poor, "User-level privacy-preserving federated learning: Analysis and performance optimization," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3388–3401, 2022.

[6] Y. Xu, M. Xiao, J. Wu, H. Tan, and G. Gao, "A personalized privacy preserving mechanism for crowdsourced federated learning," *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1568–1585, 2024.

[7] D. Kushwaha, S. Redhu, C. G. Brinton, and R. M. Hegde, "Optimal device selection in federated learning for resource-constrained edge networks," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10 845–10 856, 2023.

[8] W. Mao, X. Lu, Y. Jiang, and H. Zheng, "Joint client selection and bandwidth allocation of wireless federated learning by deep reinforcement learning," *IEEE Transactions on Services Computing*, vol. 17, no. 1, pp. 336–348, 2024.

[9] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[10] J. Zhou, N. Wu, Y. Wang, S. Gu, Z. Cao, X. Dong, and K.-K. R. Choo, "A differentially private federated learning model against poisoning attacks in edge computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 1941–1958, 2023.

[11] G. U. Talasso, A. M. de Souza, L. F. Bittencourt, E. Cerqueira, A. A. F. Loureiro, and L. A. Villas, "Fedsccs: Hierarchical clustering with multiple models for federated learning," in *IEEE International Conference on Communications*, 2024, pp. 3280–3285.