In-Network Caching and Learning Optimization for Federated Learning in Mobile Edge Networks

Yuris Mulya Saputra[1,2], Diep N. Nguyen[1], Dinh Thai Hoang[1], and Eryk Dutkiewicz[1]
[1]School of Electrical and Data Engineering, University of Technology Sydney, Australia
[2]Department of Electrical Engineering and Informatics, Universitas Gadjah Mada, Indonesia

Abstract—**In this paper, we develop a novel privacy-aware framework to address straggling problem in a federated learning (FL)-based mobile edge network through maximizing profit for the mobile service provider (MSP). In particular, unlike the conventional FL process when participating mobile users (MUs) have to train their all data locally, we propose a highly-effective solution that allows MUs to encrypt parts of local data and upload/cache the encrypted data to nearby mobile edge nodes (MENs) and/or a cloud server (CS) to perform additional training processes. In this way, we can not only mitigate the straggling problem caused by limited computing/communications resources at MUs but also enhance the usage efficiency of learning data from all MUs in the FL process. To optimize portions of encrypted data cached and trained at MENs/CS given constraints from MUs and the MSP while considering data privacy and training costs, we first formulate the profit maximization problem for the MSP as an optimal in-network encrypted data caching and learning optimization. We then prove that the objective function is concave, and thus an interior-point method algorithm can be effectively adopted to quickly find the optimal solution. The numerical results demonstrate that our proposed framework can enhance the profit of the MSP up to 5.39 times compared with other FL methods.**

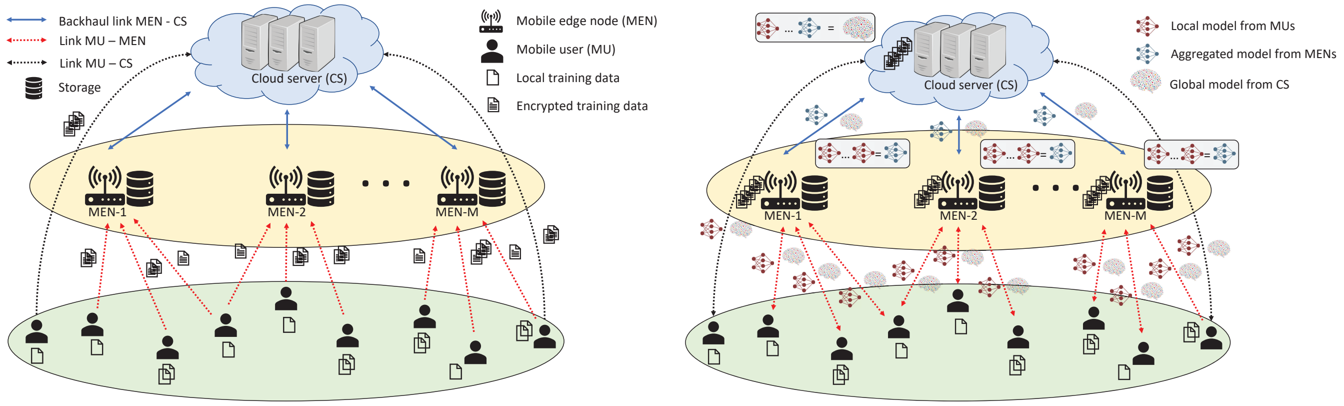
Keywords- Federated learning, privacy, encryption, straggling problem, profit optimization.

I. Introduction

Recently, the ever-increasing development of federated learning (FL) to address the limitations of conventional cloud-based learning paradigm in mobile edge computing (MEC) networks has engaged huge interests from both academia and industry. The FL approach can support mobile service providers (MSPs) to develop highly-accurate mobile applications through performing effective and privacy-protected collaborative learning between the MSPs and mobile users (MUs) in MEC networks [1]. In a conventional FL-based MEC network, all the interested MUs have to participate in the FL process at each learning round to improve the global model accuracy of the service applications. Nonetheless, this process is ineffective and impractical to be adopted in real-world MEC networks. The reason is that participating MUs may not efficiently train all their local data due to insufficient computing resources and/or experience unreliable wireless communication links when the local trained models are uploaded to the MSP at each leaning round (referred to as *straggling problem*) [2]. Consequently, the MSP may produce a low-accurate prediction model (especially when there exists a limited time to train the data at each learning round), thereby deteriorating the learning quality of the FL process.

To cope with the straggling problem, participating MUs who have limited computing and communication resources can upload their local data to a nearby mobile edge node (MEN) or a cloud server (CS). For example, the works in [3]–[5] propose local data sharing from MUs to an MEN or a CS to perform the entire FL process, aiming at maintaining the performance of FL in terms of global model accuracy. However, the above works consider that each MU may only offload its whole data to a specific MEN or the CS. Moreover, such an approach may break the key benefit of using FL, i.e., data privacy protection. To further address the privacy concern of local data sharing while mitigating the straggling problem, encryption methods can be performed prior to uploading and training them at the MEN/CS. Such an approach is motivated from centralized deep learning-based works in [6] and [7]. However, both works are limited to one learning node, i.e., an MEN or a CS, to help the MUs execute the encrypted training process. Consequently, the MEN may not cache and learn their encrypted data efficiently due to inherent limitation of the MEN's computing resources to train all encrypted data from many MUs. Meanwhile, some MUs may also suffer from high communication cost when the encrypted data are only cached directly to the CS [8].

In this paper, we propose a novel privacy-aware framework which can mitigate the straggling problem in an FL-based MEC network through maximizing the profit for the MSP. Specifically, we first propose an effective method that enables participating MUs to encrypt parts of their local data and then cache them at multiple nearby MENs and/or the CS for additional training processes. To optimize portions of MUs' local data that can be cached at MENs and/or the CS, we first formulate the profit maximization problem as an in-network encrypted data caching and learning optimization under constraints from both MUs and the MSP (i.e., computing resources at MENs and MUs, the MSP's fixed budget to perform the FL process, and deadline time for each learning round) while accounting for data encryption/caching and training costs. Then, we prove that the objective function of the problem is concave, and thus an interior-point method algorithm can be effectively used to find the optimal solution. To the best of our knowledge, this is the first work that considers concurrent use of multiple MENs and the CS for privacy-aware FL-based MEC networks with the additional encrypted training processes. Through numerical results, we show that our proposed framework can improve the MSP's profit up to 5.39 times compared with other FL methods, i.e.,



(a) Pre-FL process

(b) FL process

Fig. 1: The privacy-aware FL process with (a) one-time pre-FL process and (b) iterative FL process.

conventional FL without additional encrypted training process and privacy-aware FL with the CS only.

II. System Model

A. Proposed Privacy-Aware Federated Learning Overview

The proposed privacy-aware FL architecture in an MEC network is shown in Fig. 1 in which a CS and multiple MENs (owned by an MSP) serve multiple MUs. Denote the set of MENs and the CS as $\mathcal{M} = \{1, \dots, m, \dots, M\}$, where MEN-1 to MEN-$(M-1)$ have finite storage capacity and limited computing resources, and the MEN-M (referred to as the CS which collocates with a macro base station) has abundant storage and computing resources [8], [9]. We also define the total set of participating MUs in the FL process of the network to be $\mathcal{J} = \{1, \dots, j, \dots, J\}$. Each MEN serves a set of MUs within its coverage area via wireless links, e.g., Wi-Fi, where some MUs may be connected to multiple MENs concurrently. Some MUs may also be connected directly to the CS via celullar networks, e.g., 4G/5G networks.

Prior to the FL process, each participating MU can cache a part of local data to nearby MENs or the CS to reduce local computation load and speed up the learning process as illustrated in Fig. 1(a). To protect the privacy of data to be cached, the MU can encrypt the data using a fully homomorphic encryption with the Brakerski/Fan-Vercauteren (BFV) method [10]. Such kind of encryption method allows MENs/the CS to train encrypted data directly without decrypting the data. The above process is considered as the one-time pre-FL process.

After completing the pre-FL process, the iterative FL process is performed as illustrated in Fig. 1(b). In this case, each MEN works as an intermediate node which can aggregate the trained models from the corresponding MUs and forward the aggregated model to the CS (in addition to the encrypted training process). Meanwhile, the CS acts as the master node which can aggregate all trained models from participating entities, i.e., MUs and MENs, as well as update the global model (in addition to the encrypted training process). In particular, at each learning round, all the MUs first train their remaining local (unencrypted) datasets and then send the local trained models to the corresponding MEN(s)/CS. At the same time, MENs and the CS perform the encrypted training processes to produce other trained models. Upon completing the learning process within a training time threshold at each learning round (which is predefined by the MSP), each MEN can simultaneously obtain its own trained model and collect the local models from the MUs to aggregate them for aggregated local model generation. Afterwards, the aggregated local model from each MEN can be forwarded to the CS for final aggregation and global model update. Note that the MENs and the CS only need to aggregate trained models from MUs that are received within a pre-defined training time threshold for each learning round.

Let D_j denote the whole local dataset at MU-j with the size $s_j = \xi|D_j|$, respectively. Here, ξ is the size (in bits) per one data sample and $|D_j|$ is the number of samples at MU-j. We define the portions of local dataset at MU-j and encrypted dataset of MU-j at MEN-m as d^l_j and $d^c_{j,m}$, respectively, where $d^l_j, d^c_{j,m} \in [0,1], \forall j \in \mathcal{J}, \forall m \in \mathcal{M}$. Here, we define $d^l_j = \Big(1 - \sum_{m=1}^{M} d^c_{j,m}\Big)$, and denote $\mathbf{d} = [d^c_{1,1}, \dots, d^c_{1,m}, \dots, d^c_{1,M}, d^c_{2,1} \dots, d^c_{j,m}, \dots, d^c_{J,M}]$ as the vector of continuous variables between "0" and "1" to determine how much portion of each MU's local dataset that will be trained at an MEN or the CS. We define r^M_m (where $m \neq M$), r^m_j (where $m \neq M$), and r^M_j as the bandwidth between MEN-m and the CS, the bandwidth between MU-j and MEN-m, and the bandwidth between MU-j and the CS, respectively. For each participating MU, due to its computing resources, it can only process dataset up to size $\hat{s}^l_j$ for the entire FL process [1], [2]. As such, if $s_j > \hat{s}^l_j$, then the MU has to cache a part of s_j to corresponding MENs or the CS. For each MEN-m, $m \neq M, m \in \mathcal{M}$, also due to its limited computing resources, it can only train encrypted dataset from all MUs up to size $\hat{s}^c_m$.

B. Computing and Communication Model

For the computing model of proposed FL, an MU-j has size of local dataset $d^l_j s_j$ and can implement the local training process using its η_j (cycles/bit) CPU cycles with computing resource f_j (Hz). The computing time of the training process at the MU-j can be derived by

$$t_j^{cmp} = \frac{\eta_j d_j^l s_j}{f_j}. \tag{1}$$

For each MEN-m, the encrypted training process can be executed using its η_m (cycles/bit) CPU cycles with computing resource f_m (Hz) and the total collected encrypted datasets from MUs, i.e., $\sum_{j=1}^{J} d_{j,m}^c s_j$. As such, the computing time of an MEN-m can be expressed by

$$t_m^{cmp} = \frac{\eta_m \sum_{j=1}^{J} d_{j,m}^c s_j}{f_m}. \tag{2}$$

For the communication model, let $s_{\mathbf{\Gamma}^{(\tau)}}$, $s_{\nabla\mathbf{\Gamma}_j^{(\tau)}}$, and $s_{\nabla\mathbf{\Gamma}_m^{(\tau)}}$ denote the sizes of global model, trained model of MU-j, and aggregated model of MEN-m in bits, respectively, in which $s_{\mathbf{\Gamma}^{(\tau)}} = s_{\nabla\mathbf{\Gamma}_j^{(\tau)}} = s_{\nabla\mathbf{\Gamma}_m^{(\tau)}}$ [2], [11]. We also define $\nu_{m,M}^{down}, \nu_{j,M}^{down}, \nu_{j,m}^{down}$ and $\nu_{m,M}^{up}, \nu_{j,M}^{up}, \nu_{j,m}^{up}$ as the numbers of successful transmissions for downlink and uplink processes among the CS, MEN-m, and MU-j, respectively. As such, the time required to download a global model $\mathbf{\Gamma}^{(\tau)}$ at learning round τ from the CS to an MEN-m/MU-j and from an MEN-m to an MU-j can be respectively written as follows:

$$t_{m,M}^{com-down} = \nu_{m,M}^{down}\frac{s_{\mathbf{\Gamma}^{(\tau)}}}{r_m^M}, t_{j,M}^{com-down} = \nu_{j,M}^{down}\frac{s_{\mathbf{\Gamma}^{(\tau)}}}{r_j^M}, \text{ and}$$
$$t_{j,m}^{com-down} = \nu_{j,m}^{down}\frac{s_{\mathbf{\Gamma}^{(\tau)}}}{r_j^m}, m \neq M. \tag{3}$$

Moreover, the time to upload a local trained model $\nabla\mathbf{\Gamma}_j^{(\tau)}$ (from an MU-j to an MEN-m/the CS) and an aggregated model $\nabla\mathbf{\Gamma}_m^{(\tau)}$ (from an MEN-m to the CS) are expressed by

$$\begin{aligned} t_{j,m}^{com-up} &= \nu_{j,m}^{up}\frac{s_{\nabla\mathbf{\Gamma}_j^{(\tau)}}}{r_j^m}, t_{j,M}^{com-up} = \nu_{j,M}^{up}\frac{s_{\nabla\mathbf{\Gamma}_j^{(\tau)}}}{r_j^M}, \text{ and} \\ t_{m,M}^{com-up} &= \nu_{m,M}^{up}\frac{s_{\nabla\mathbf{\Gamma}_m^{(\tau)}}}{r_m^M}, m \neq M. \end{aligned} \tag{4}$$

Note that, in practice, the time for trained model aggregation at MEN-m, $\forall m \in \mathcal{M}$, is very fast, and thus it can be ignored [2].

From (1) to (4), if an MU-j is indirectly connected to the CS via MEN(s), then the total time from the global model sharing of the CS to the local model collection of MU-j at MEN-m can be derived as

$$t_j^{m,\dagger} = t_{m,M}^{com-down} + t_{j,m}^{com-down} + t_j^{cmp} + t_{j,m}^{com-up}. \tag{5}$$

Nonetheless, each MEN-m, where $m \neq M$, has a deadline t_m^{max}. Here, we assume that the MSP sets the same deadline for all the MENs such that $t_1^{max} = \ldots = t_m^{max} = \ldots = t_{M-1}^{max} = t^{max}$. After the deadline t_m^{max}, all MENs can simultaneously aggregate their models and forward them to the CS. For that, we express the above condition as $0 \leq t_j^{m,\dagger} \leq t^{max}, \forall j \in \mathcal{J}, \forall m \in \mathcal{M}, m \neq M$. Then, the total learning time for one learning round is $t^{max} + t_M^{com-up}$. We set that $t_{m,M}^{com-up} = t_M^{com-up}, m \neq M, \forall m \in \mathcal{M}$, when the aggregated models are shared to the CS through MENs (under the consideration that the MSP typically can set the same $\nu_{m,M}^{up}$ and r_m^M for all its MENs). Finally, the total time from the global model sharing of the CS to the aggregated model collection of MEN-m at the CS is

$$t_j^m = t_j^{m,\dagger} + t_{m,M}^{com-up}, \forall j \in \mathcal{J}, \tag{6}$$

where $0 \leq t_j^m \leq t^{max} + t_M^{com-up}, \forall j \in \mathcal{J}, \forall m \in \mathcal{M}, m \neq M$.

Similarly, if an MU-j is directly connected to the CS, the total time from the global model sharing of the CS to the local model collection of MU-j at the CS directly is written by

$$t_j^M = t_{j,M}^{com-down} + t_j^{cmp} + t_{j,M}^{com-up}. \tag{7}$$

This t_j^M is also upper-bounded by the learning round time $t^{max} + t_M^{com-up}$, i.e., $0 \leq t_j^M \leq t^{max} + t_M^{com-up}, \forall j \in \mathcal{J}$. Additionally, the training time at MEN-m, $\forall m \in \mathcal{M}$ (including the CS), can be respectively obtained as follows:

$$t_m^* = \begin{cases} t_{m,M}^{com-down} + t_m^{cmp} + t_{m,M}^{com-up}, \\ \text{if } m \neq M, \\ t_M^{cmp}, \text{ otherwise}, \end{cases} \tag{8}$$

where $t_m^* \leq t^{max} + t_M^{com-up}$. Note that the use of total learning time constraint containing the deadline t_{max} may influence the portions of local and encrypted data to be trained at MUs and MENs/the CS, respectively. As such, if t_{max} is very small, then most of local data from the MUs may be encrypted and trained at MENs/the CS (due to their higher computing resources) to compensate the straggling problem (under incentive budget consideration for data encryption, caching, and training).

III. Problem Formulation

To minimize the straggling problem under the presence of data encryption/caching and training incentives, we require to maximize the MSP's profit through finding the optimal $\mathbf{d}$ in the proposed privacy-aware FL process. Particularly, the MSP's profits for the encrypted training process at MEN-m and the local training process at MU-j can be respectively derived by

$$P_m = \lambda_m \sqrt{\sum_{j=1}^{J} d_{j,m}^c s_j} - (\zeta_m \eta_m f_m^2 \alpha_m + \beta_m) \sum_{j=1}^{J} d_{j,m}^c s_j, \tag{9}$$

and

$$P_j = \lambda_j \sqrt{\left(1 - \sum_{m=1}^{M} d_{j,m}^c\right) s_j} - \rho_j \left(1 - \sum_{m=1}^{M} d_{j,m}^c\right) s_j, \tag{10}$$

where λ_m and λ_j are the conversion parameters representing the monetary unit of using encrypted dataset at MEN-m and the remaining local dataset at MU-j, respectively, based on the current data trading market [12]. Moreover, α_m is the energy consumption cost unit to train an encrypted data sample, β_m is the incentive unit per one data sample for MU-j in encrypting and uploading the part of local dataset for the encrypted training process at the MEN/CS, and ρ_j is the incentive unit per one data sample for MU-j in joining the local training process. The use of square root function [12] in the first terms of P_m and P_j specifies that the gain values increase when a larger dataset is trained in the FL process. However, the MSP may have less interest to further improve the value when dataset with much larger size leads to less global model accuracy improvement [13]. For that, the optimization problem that can maximize the MSP's profit can be expressed by

$$(\mathbf{P_d}) \quad \max_{\mathbf{d}} \sum_{j=1}^{J} P_j + \sum_{m=1}^{M} P_m, \tag{11}$$

$$\text{s.t. } 1 - \sum_{m=1}^{M} d^c_{j,m} \geq 0, \forall j \in \mathcal{J}, \tag{12}$$

$$\left(1 - \sum_{m=1}^{M} d^c_{j,m}\right) s_j \leq \hat{s}^l_j, \forall j \in \mathcal{J}, \tag{13}$$

$$\sum_{j=1}^{J} d^c_{j,m} s_j \leq \hat{s}^c_m, m \neq M, \forall m \in \mathcal{M}, \tag{14}$$

$$0 \leq t^m_j \leq t^{max} + t^{com-up}_M, \forall j \in \mathcal{J}, \forall m \in \mathcal{M}, \tag{15}$$

$$0 \leq t^*_m \leq t^{max} + t^{com-up}_M, \forall m \in \mathcal{M}, \tag{16}$$

$$\sum_{j=1}^{J} \rho_j \left(1 - \sum_{m=1}^{M} d^c_{j,m}\right) s_j + \sum_{m=1}^{M} \left(\beta_m \sum_{j=1}^{J} d^c_{j,m} s_j\right) \leq B, \tag{17}$$

$$0 \leq d^c_{j,m} \leq 1, \forall j \in \mathcal{J}, \forall m \in \mathcal{M}, \tag{18}$$

where constraints (12) specify the portion of local unencrypted dataset to be trained at each MU-j (in which the maximum sum of all portions of the MU-j's dataset, i.e., local and encrypted datasets, is equal to 1). Constraints (13) and (14) imply that the unencrypted dataset that is trained locally cannot exceed the maximum trainable local dataset at MU-j, and the collected encrypted dataset that is trained at each MEN cannot exceed the maximum trainable encrypted dataset at the MEN-m, where $m \neq M$, respectively (due to their limited computing resources). Meanwhile, constraints (15) and (16) indicate that the training time for each learning round cannot exceed the pre-defined deadline time $t_{max} + t^{com-up}_M$ (to avoid straggling problem). Furthermore, constraint (17) represents that the incentives for participating MUs cannot exceed the incentive budget of the MSP, i.e., B.

IV. Optimal Encrypted Data Caching and Learning Solution

To find the optimal solution $\mathbf{d}$ from the optimization problem $(\mathbf{P_d})$, we first prove that the objective function in (11) is a concave function.

Theorem 1. *The objective function* $\left[\sum_{j=1}^{J} P_j + \sum_{m=1}^{M} P_m\right]$ *in (11) is a concave function for all* $d^c_{j,m}, \forall j \in \mathcal{J}, \forall m \in \mathcal{M}$, *that satisfies constraints (12)-(18).*

Proof. We first define $\gamma_m = \zeta_m \eta_m f^2_m \alpha_m + \beta_m$. Then, we modify (9) and (10) respectively into

$$P_m = \lambda_m \left[\sum_{j=1}^{J} d^c_{j,m} s_j\right]^{\frac{1}{2}} - \gamma_m \sum_{j=1}^{J} d^c_{j,m} s_j, \tag{19}$$

and

$$P_j = \lambda_j \left[\left(1 - \sum_{m=1}^{M} d^c_{j,m}\right) s_j\right]^{\frac{1}{2}} - \rho_j \left(1 - \sum_{m=1}^{M} d^c_{j,m}\right) s_j. \tag{20}$$

Next, we can compute the first partial derivative of P_m and P_j in regards to $\mathbf{d}$ respectively as follows:

$$\begin{aligned}\nabla P_m &= \left[\frac{\partial P_m}{\partial d^c_{1,1}}, \dots, \frac{\partial P_m}{\partial d^c_{1,M}}, \dots, \frac{\partial P_m}{\partial d^c_{j,m}}, \dots, \frac{\partial P_m}{\partial d^c_{J,M}}\right] \\ &= \left[0, \dots, \frac{\partial P_m}{\partial d^c_{j,m}}, \dots, 0\right],\end{aligned} \tag{21}$$

and

$$\begin{aligned}\nabla P_j &= \left[\frac{\partial P_j}{\partial d^c_{1,1}}, \dots, \frac{\partial P_j}{\partial d^c_{1,M}}, \dots, \frac{\partial P_j}{\partial d^c_{j,m}}, \dots, \frac{\partial P_j}{\partial d^c_{J,M}}\right] \\ &= \left[0, \dots, \frac{\partial P_j}{\partial d^c_{j,m}}, \dots, 0\right],\end{aligned} \tag{22}$$

where

$$\frac{\partial P_m}{\partial d^c_{j,m}} = \frac{1}{2} \lambda_m s_j \left[\sum_{j=1}^{J} d^c_{j,m} s_j\right]^{-\frac{1}{2}} - \gamma_m s_j, \tag{23}$$

$$\frac{\partial P_j}{\partial d^c_{j,m}} = -\frac{1}{2} \lambda_j s_j \left[\left(1 - \sum_{m=1}^{M} d^c_{j,m}\right) s_j\right]^{-\frac{1}{2}} + \rho_j s_j. \tag{24}$$

We can also compute the second partial derivative of P_m, i.e., $\mathbf{H}_m = \nabla^2 P_m$, as follows:

$$\mathbf{H}_m = \begin{bmatrix} \frac{\partial^2 P_m}{\partial^2 d^c_{1,1}} & \cdots & \frac{\partial^2 P_m}{\partial d^c_{1,1} \partial d^c_{j,m}} & \cdots & \frac{\partial^2 P_m}{\partial d^c_{1,1} \partial d^c_{J,M}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial^2 P_m}{\partial d^c_{j,m} \partial d^c_{1,1}} & \cdots & \frac{\partial^2 P_m}{\partial^2 d^c_{j,m}} & \cdots & \frac{\partial^2 P_m}{\partial d^c_{j,m} \partial d^c_{J,M}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial^2 P_m}{\partial d^c_{J,M} \partial d^c_{1,1}} & \cdots & \frac{\partial^2 P_m}{\partial d^c_{J,M} \partial d^c_{j,m}} & \cdots & \frac{\partial^2 P_m}{\partial^2 d^c_{J,M}} \end{bmatrix}. \tag{25}$$

From (25), the general expressions of second derivative elements can be determined by

$$\frac{\partial^2 P_m}{\partial^2 d^c_{j,m}} = -\frac{1}{4} \lambda_m s^2_j \left[\sum_{j=1}^{J} d^c_{j,m} s_j\right]^{-\frac{3}{2}}, \tag{26}$$

$$\frac{\partial^2 P_m}{\partial d^c_{j,m} \partial d^c_{j^\dagger,m}} = -\frac{1}{4} \lambda_m s_j s_{j^\dagger} \left[\sum_{j=1}^{J} d^c_{j,m} s_j\right]^{-\frac{3}{2}}, \forall j^\dagger \neq j, \tag{27}$$

$$\frac{\partial^2 P_m}{\partial d^c_{j,m} \partial d^c_{j,m^\dagger}} = \frac{\partial^2 P_m}{\partial d^c_{j,m} \partial d^c_{j^\dagger,m^\dagger}} = 0, \forall j^\dagger \neq j, \forall m^\dagger \neq m. \tag{28}$$

Likewise, we can derive $\mathbf{H}_j = \nabla^2 P_j$, and obtain that

$$\begin{aligned}\frac{\partial^2 P_j}{\partial^2 d^c_{j,m}} &= \frac{\partial^2 P_m}{\partial d^c_{j,m} \partial d^c_{j,m^\dagger}} \\ &= -\frac{1}{4} \lambda_j s^2_j \left[\left(1 - \sum_{m=1}^{M} d^c_{j,m}\right) s_j\right]^{-\frac{3}{2}}, \forall m^\dagger \neq m,\end{aligned} \tag{29}$$

$$\frac{\partial^2 P_m}{\partial d^c_{j,m} \partial d^c_{j^\dagger,m}} = \frac{\partial^2 P_m}{\partial d^c_{j,m} \partial d^c_{j^\dagger,m^\dagger}} = 0, \forall j^\dagger \neq j, \forall m^\dagger \neq m. \tag{30}$$

Given an arbitrary real vector $\mathbf{v} \in \mathbb{R}^{(J^* M) \times 1}$ and $0 \leq d^c_{j,m} \leq 1, \forall j \in \mathcal{J}, \forall m \in \mathcal{M}$, we obtain that $\mathbf{v}^T \mathbf{H}_m \mathbf{v} \leq 0$ and $\mathbf{v}^T \mathbf{H}_j \mathbf{v} \leq 0$, where $\mathbf{H}_m, \mathbf{H}_j, \forall j \in \mathcal{J}, \forall m \in \mathcal{M}$, are negative semi-definite matrices. Hence, $P_m, \forall m \in \mathcal{M}$, and $P_j, \forall j \in \mathcal{J}$, are concave functions with respect to vector $\mathbf{d}$. As the objective function is $\left[\sum_{j=1}^{J} P_j + \sum_{m=1}^{M} P_m\right]$, we can summarize that the objective function is also a concave function [14]. □

To address the optimization problem $(\mathbf{P_d})$, popular optimization tools explained in [14] can be utilized since the objective function is concave with linear constraints. For that, we adopt the *interior-point method (IPM)* algorithm which can effectively solve convex function evaluations with constraints and second derivative equations for large-scale and sparse nonlinear problem [15].

V. Performance Evaluation

A. Simulation Setup

We consider 5 MENs (including the CS) and 50 participating MUs in the MEC network. We denote the whole dataset size of each participating MU-j, i.e., $|D_j|$, is uniformly random between 1M and 10M number of samples with $\zeta = 1496$ bits. We set $r_m^M, \forall m \in \mathcal{M}, m \neq M$, $r_j^m, \forall j \in \mathcal{J}, \forall m \in \mathcal{M}, m \neq M$, and $r_j^M, \forall j \in \mathcal{J}$, at 60Mbps, 30Mbps, and 10Mbps, respectively [16]. We also define $\lambda_m = 0.3, \forall m \in \mathcal{M}, m \neq M$, $\lambda_M = 0.01$, and $\lambda_j = 0.05, \forall j \in \mathcal{J}$. We utilize $\xi_m = 0.5 \times 10^{-26}$ and $f_m = f_j = 2$GHz, $\forall j \in \mathcal{J}, \forall m \in \mathcal{M}$. We set $\alpha_m = 0.01, \forall m \in \mathcal{M}$, $\rho_j = 0.0001, \forall j \in \mathcal{J}$, and $t_{max} = 0.5$ seconds. Moreover, we specify $\beta_m = 0.0005, \forall m \in \mathcal{M}, m \neq M$, and $\beta_M = 0.0001$ to show that encrypting and caching the dataset at the CS has a higher cost than those at MENs. We compare the proposed FL, i.e., Pro-FL, with the scenarios when the encrypted data from all participating MUs only can be cached at the CS, i.e., Pro-FL CS-only, and all the datasets are trained locally at MUs without encryption, i.e., Conv-FL.

B. Numerical Results

Let's first evaluate the MSP's profit of the proposed FL when the computing resources (i.e., the total data sizes that can be trained) at MENs increase between 10 and 80Gbit equally. In this case, we fix the computing resources at MUs to be very small, i.e., 0.1Gbit, to highlight the straggling problem in the FL process. As observed in Fig. 2, the proposed FL can outperform up to 5.39 times and 1.77 times in terms of total profit of the MSP compared with Conv-FL and Pro-FL CS-only, respectively. The reason is that the MSP can help participating MUs to train their datasets securely via encryption and caching at multiple MENs, and thus can minimize the straggling problem and may speed up the training time with high accuracy. Additionally, by training the encrypted dataset at MENs which are closer to the MUs, the higher encryption and caching costs at the CS can be avoided. There exists a condition when the MSP's profit remains the same even though we further increase the computing resources at MENs. This is due to the diminishing return characteristic of the gain function in the profit function [13] in which the global model accuracy cannot be improved anymore. This implies that most data samples of an MU are encrypted, cached and trained at MENs to maximize the profits (as shown in Fig. 3). To this end, the MU still can train approximately 14.5% of the whole dataset locally without straggling problem, and thus the encrypted training process at the CS with higher cost can be minimized.

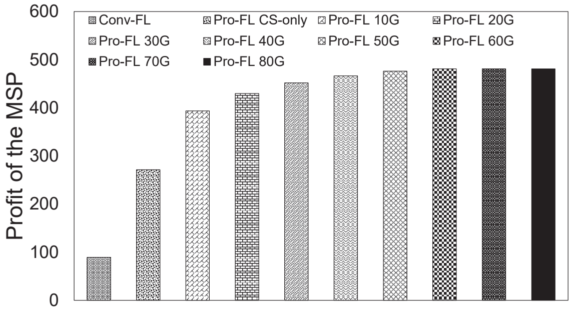


Fig. 2: The MSP's profit for various FL scenarios.

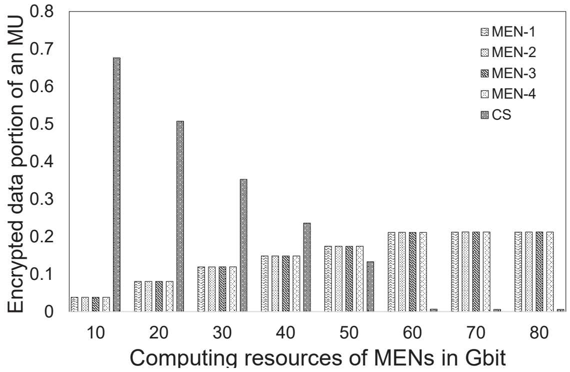
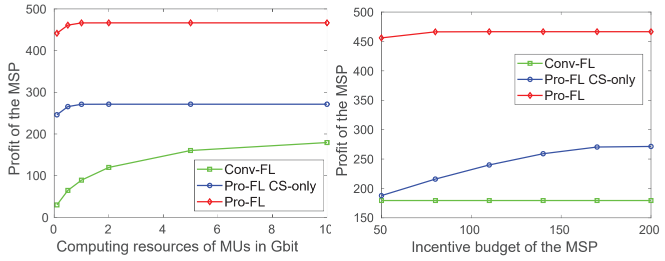


Fig. 3: The encrypted data portion of an MU when computing resources of MENs increase.

Next, we observe the superiority of Pro-FL performance when the computing resources at MUs increase from 0.1Gb to 10Gb with fixed computing resources at MENs. As expected from Fig. 4(a), the MSP's profit for Conv-FL increases gradually when the computing resources at MUs get larger, aiming at reducing the straggling problem. This aligns with the higher portion of local datasets that can be trained at MUs without any straggling problem shown in TABLE I. When an MU can train all the whole local dataset in the Conv-FL, i.e., 100% dataset is trained at the MU, the MSP's profit will not improve anymore. To this end, the Pro-FL can still obtain the highest profit regardless the computing resources of MUs (up to 14.8 times and 1.79 times compared with those of Conv-FL and Pro-FL CS-only, respectively). Specifically, although the portion of the MU's encrypted dataset trained at MENs/the CS decrease due to its higher computing resource shown in TABLE I, the Pro-FL can still slightly improve the profit, thanks to the additional profit of training more local dataset at the MU. Then, the MSP's profit of the Pro-FL will not increase anymore since the portions of encrypted datasets at MENs reach the optimality. The same trend with lower profit can be observed for Pro-FL CS-only.



(a) MUs' computing resources vary (b) The MSP's budget varies

Fig. 4: Profit performances when MUs' computing resources and MSP's incentive budget increase.

TABLE I: The encrypted and local data portions of an MU for the training process when MUs' computing resources increase.

$\hat{s}^l_j$ (Gb)	Conv-FL	Pro-FL CS-only		Pro-FL	
	Local	Encrypted	Local	Encrypted	Local
0.1	0.0170	0.9830	0.0170	0.9830	0.0170
0.5	0.0851	0.9149	0.0851	0.9149	0.0851
1	0.1701	0.8299	0.1701	0.8299	0.1701
2	0.3403	0.8081	0.1919	0.8123	0.1877
5	0.8507	0.8081	0.1919	0.8123	0.1877
10	1.0000	0.8081	0.1919	0.8123	0.1877

TABLE II: The encrypted and local data portions of an MU for the training process when the MSP's incentive budget increases.

B	Conv-FL	Pro-FL CS-only		Pro-FL	
	Local	Encrypted	Local	Encrypted	Local
50	1.0000	0.0000	1.0000	0.5900	0.4100
80	1.0000	0.0157	0.9843	0.7848	0.2152
110	1.0000	0.3090	0.6907	0.8123	0.1877
140	1.0000	0.5481	0.4519	0.8123	0.1877
170	1.0000	0.7508	0.2492	0.8123	0.1877
200	1.0000	0.8081	0.1919	0.8123	0.1877

To further show the efficacy of Pro-FL, we increase the incentive budget of MSP from 50 to 200 monetary unit in Fig. 4(b). In particular, when the incentive budget is very small at 50 monetary units, the MSP's profit of the Pro-FL is the lowest one due to less incentives for MUs to train unencrypted dataset and protect data privacy (by encrypting and uploading the dataset at MENs/the CS). As a result, the portion of an MU's encrypted dataset is the smallest one (as seen in TABLE II). Nonetheless, the MSP's profit and portion of the MU's encrypted dataset will increase and remain the same for the rest of incentive budget. Here, the Pro-FL can obtain the profit 2.54 times and 2.43 times higher than those of Conv-FL and Pro-FL CS-only, respectively. For the Conv-FL, the MSP's profit does not change because 50 monetary units are sufficient to incentivise the MUs to train the whole datasets locally (as seen in TABLE II). Meanwhile, the MSP's profit for the Pro-FL CS-only increases gradually from 50 to 200 monetary units due to the fact that more incentives given to the MUs (for encryption and uploading) will increase the portions of encrypted datasets to be trained at the CSs. The above results indicate that the MSP can obtain more profits when more encrypted datasets from MUs are trained at MENs with sufficient computing resources and incentive budget as well as minimum encryption and caching costs.

VI. Conclusion

In this paper, we have proposed the novel privacy-aware FL-based framework for the MEC network to mitigate straggling problem by maximizing the MSP's profit in the FL process. Specifically, we have formulated the optimal in-network encrypted data caching and learning optimization for the FL process under computing resource constraints of MUs and MENs, training time deadline, and incentive budget for MUs. To obtain the optimal portions of MUs' encrypted datasets trained at MENs/the CS, we first have proven the convexity of objective function and then adopted the interior-point method algorithm to find the optimal solution. The numerical results have demonstrated that our proposed framework can significantly improve the MSP's profit compared with those of other baseline FL methods (by training more encrypted datasets at MENs while considering the privacy of encrypted datasets).

References

[1] W. Y. B. Lim, *et al.*, "Federated learning in mobile edge networks: a comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031-2063, Apr. 2020.

[2] S. Prakash, *et al.*, "Coded computing for low-latency federated learning over wireless edge networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 233-250, Jan. 2021.

[3] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL: cooperative learning mechanism using non-iid data in wireless networks," arXiv:1905.07210v3 [cs.LG], Mar. 2020.

[4] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986-5994, Jul. 2020.

[5] L. U. Khan, *et al.*, "Federated learning for edge networks: resource optimization and incentive mechanism," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 88-93, Oct. 2020.

[6] N. J. Hernandez Marcano, *et al.*, "On fully homomorphic encryption for privacy-preserving deep learning," in *IEEE Globecom Workshops*, Dec. 2019, pp. 1-6.

[7] Z. Yue, *et al.*, "Privacy-preserving time-series medical images analysis using a hybrid deep learning framework," *ACM Trans. Internet Technol.*, vol. 21, no. 3, pp. 1-21, Jun. 2021.

[8] Y. Mao, *et al.*, "A survey on mobile edge computing: the communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[9] A. u. R. Khan, *et al.*, "A survey of mobile cloud computing application models," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 393-413, Firstquarter 2014.

[10] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *IACR Cryptol. ePrint Arch.*, Mar. 2012, pp. 1-19.

[11] J. Kang, *et al.*, "Incentive mechanism for reliable federated learning: a joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700-10714, Dec. 2019.

[12] L. Xu, C. Jiang, Y. Chen, Y. Ren, and K. J. R. Liu, "Privacy or utility in data collection? a contract theoretic approach" *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 7, pp. 1256-1269, Oct. 2015.

[13] P. A. Samuelson and W. D. Nordhaus, *Microeconomics*, 18th ed. Boston, MA, USA: McGraw-Hill, 2005.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[15] R. Byrd, N. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM J. Optimization*, vol. 9, no. 4, pp. 877–900, Sep. 1999.

[16] Y. M. Saputra, *et al.*, "A novel mobile edge network architecture with joint caching-delivering and horizontal cooperation," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 19-31, Jan. 2021.