

“©2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

# Federated Learning Framework with Straggling Mitigation and Privacy-Awareness for AI-based Mobile Application Services

Yuris Mulya Saputra, Diep N. Nguyen, Dinh Thai Hoang, Quoc-Viet Pham, Eryk Dutkiewicz, and Won-Joo Hwang

**Abstract**—In this work, we propose a novel framework to address straggling and privacy issues for federated learning (FL)-based mobile application services, taking into account limited computing/communications resources at mobile users (MUs)/mobile application provider (MAP), privacy cost, the rationality and incentive competition among MUs in contributing data to the MAP. Particularly, the MAP first determines a set of the best MUs for the FL process based on the MUs' provided information/features. To mitigate straggling problems with privacy-awareness, each selected MU can then encrypt part of local data and upload the encrypted data to the MAP for an encrypted training process, in addition to the local training process. For that, each selected MU can propose a contract to the MAP according to its expected trainable local data and privacy-protected encrypted data. To find the optimal contracts that can maximize utilities of the MAP and all the participating MUs while maintaining high learning quality of the whole system, we first develop a multi-principal one-agent contract-based problem leveraging FL-based multiple utility functions. These utility functions account for the MUs' privacy cost, the MAP's limited computing resources, and asymmetric information between the MAP and MUs. Then, we transform the problem into an equivalent low-complexity problem and develop a light-weight iterative algorithm to effectively find the optimal solutions. Experiments with a real-world dataset show that our framework can speed up training time up to 49% and improve prediction accuracy up to 4.6 times while enhancing the network's social welfare, i.e., total utility of all participating entities, up to 114% under the privacy cost consideration compared with those of baseline methods.

**Index Terms**—Federated learning, privacy, encryption, straggling problem, contract theory.

## 1 INTRODUCTION

The ever-growing big data market between mobile users (MUs) and mobile application providers (MAPs) for emerging artificial intelligence (AI)-based mobile application services (e.g., healthcare, crowdsensing, and mobile social network applications) [1], [2] has recently attracted paramount interest from both industry and academia. Through utilizing the collected local data from many mobile devices, e.g., via embedded sensors, the AI-based mobile application services can extract the meaningful information leveraging machine learning (ML) approaches, e.g., centralized learning at the cloud server and local learning at mobile devices [3], [4]. Nonetheless, there exist two inherent challenges when applying such conventional ML approaches for AI-based mobile application services. First, MUs may not want to share their raw data due to the privacy risk of

storing/processing data at the centralized servers. Second, MUs with limited computing/communications resources as well as unreliable wireless channels can be stragglers, adversely impacting the learning/training quality/time.

Federated learning (FL) has been considered as a highly-effective learning approach to deal with the aforementioned problems. Like distributed learning, FL facilitates collaborative learning among multiple entities, e.g., MUs, without requiring them to share their raw data [5], [6]. In particular, each participating MU can first train its local data to produce a local trained model independently. Then, the MAP can collect the local models from all participating MUs to update the global model iteratively. Nonetheless, similar to the centralized learning above, MUs are usually limited in computing and communications resources. For that, the quality of the FL process may deteriorate when some of the MUs experience low computing resources to train their local dataset and/or unstable wireless communication links when sharing the local trained models to the MAP at each learning round (referred to as *straggling* MUs). Consequently, FL may likely suffer from significant delay because the MAP needs to wait for local trained models from all the participating MUs at each learning round to update the global model. To address the straggling problem, data from straggling MUs can be uploaded to an edge/a cloud server for the FL process (on behalf of the straggling MUs) as proposed in [7], [8], [9], [10]. However, this approach cannot be widely adopted as MUs usually do not want to send the raw data to the edge/cloud server, e.g., the MAP, for

- Y. M. Saputra is with the School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: yurismulya.saputra@student.uts.edu.au) and Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia.
- D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz are with the School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: {hoang.dinh, diep.nguyen, eryk.dutkiewicz}@uts.edu.au).
- Quoc-Viet Pham is with the Korean Southeast Center for the 4th Industrial Revolution Leader Education, Pusan National University, Busan 46241, Republic of Korea (e-mail: vietpq@pusan.ac.kr).
- Won-Joo Hwang is with the Department of Biomedical Convergence Engineering, Pusan National University, Yangsan 50612, Republic of Korea (e-mail: wjhwang@pusan.ac.kr).

processing due to privacy concerns.

To mitigate the straggling problems while preserving data privacy for the participating MUs under their limited computing resources, parts of local datasets from the participating MUs can be first encrypted to preserve their dataset privacy and then sent to the MAP for the additional encrypted training process. For example, the authors in [11], [12] first use homomorphic encryption (i.e., an encryption algorithm which allows operations to be executed directly on encrypted data/ciphertext without decryption) to protect the privacy of their local data and then send the encrypted data to the MAP for the encrypted training process, e.g., leveraging convolutional neural networks (CNN) and long-short term memory (LSTM). Nonetheless, both works only consider deep learning approaches to train the whole data at the MAP. In [13], [14], instead of encrypting the local datasets to share them with the MAP, MUs train their whole datasets locally then only encrypt their local models before sending them to the MAP. Note that [11]- [14] and other similar works in the literature do not account for the cryptography overhead/cost as well as limited computing and communications resources of the MAP and MUs. Moreover, they lack an incentive mechanism to motivate the MUs to participate in the FL processes, especially under the competition/conflict-of-interest among the MUs considering their diverse computing resources and data. To this end, how to optimally compensate all the participating MUs while maximizing the MAP's benefit under the privacy protection for the MUs and limited computing resources from the MAP and MUs remains as an open issue.

In this paper, we propose a novel framework to address straggling and privacy issues for an FL-based mobile application service leveraging data encryption and additional encrypted training process at the MAP given the limited computing/communications resources at MUs/MAP, privacy cost, the rationality and incentive competition among MUs in contributing data to the MAP. In particular, the MAP can first select the best active MUs in its considered area (e.g., a shopping mall or a residential area) based on their shared information, i.e., selection metric values. This aims to obtain reliable trained model updates for the global model accuracy of the FL process with minimum straggling issues [18]. After the MU selection process, the MAP can provide incentives/payments to encourage these MUs to contribute their data to the training process. In this way, the MAP can improve its learning quality to produce highly-accurate prediction model, thereby maximizing its utility correspondingly. Meanwhile, the participating MUs can maximize their utilities through acquiring certain incentives from the MAP in exchange for training the local datasets and/or sharing the encrypted datasets.

However, since there exists conflict-of-interest among the selected MUs due to their limited computing resources to train their local datasets, each participating MU may compete with other participating MUs in offloading encrypted dataset to the MAP for the additional training process under the MAP's limited computing resources (e.g., due to heavy workloads from other computationally-intensive services). To cope with this problem, Stackelberg game models can be used to study the interactions between the MUs and the MAP [19], [20], [21], [22]. Nonetheless, these game models

are only applicable when the MUs and the MAP have full information about each other (referred to as *information symmetry*), e.g., the MUs know the MAP's current computing resources. In practice, the MAP usually keeps its computing resources as private information (referred to as *information asymmetry*). Additionally, the MUs may not be interested in fully following the offers from the MAP due to the conflict of economic preferences between them. Hence, the above Stackelberg game models are not applicable to our considered problem.

To address the aforementioned problem, we develop a *multi-principal one-agent (MPOA)* contract-based framework [23], in which the participating MUs (as *principals*) can first offer contract items, i.e., the sizes of local and uploaded encrypted datasets as well as their requested payments, to the MAP. Then, the MAP (as *the agent*) can optimize the offered contracts on behalf of the participating MUs under its limited computing resources. Specifically, we formulate a privacy-preserving FL contract problem that can maximize the utilities of the MAP and the selected MUs for the entire FL process under the MAP's information-asymmetry and common constraints. These constraints guarantee that the MAP will always obtain a non-negative and maximum utility during the FL process. Unlike the contract problem for the conventional FL [6], in our framework, there exist two utility functions required to be addressed and optimized by each participating MU and the MAP, i.e., one for the encrypted (sharing) training data and the other for the unencrypted (local) training data.

To obtain the optimal contracts, the FL contract optimization problem is first transformed into an equivalent problem with a lower complexity (due to constraint reduction). Then, an iterative algorithm is developed to quickly find the equilibrium solution, i.e., optimal contracts, for the transformed problem. This equilibrium solution can achieve the performance gap of less than 1% compared with that of the information-symmetry FL contract, i.e., when each MU is fully aware of other MUs' available dataset sizes and the MAP's actual computing resources. Upon receiving the optimal contracts from the MAP, both the MAP and selected MUs can implement the proposed FL algorithm. We evaluate our FL-based framework using a real-world human activity recognition (HAR) dataset with 15M activity recognition samples from MUs' devices in 2019 [24]. While considering the privacy costs, the experimental results show that our privacy-preserving FL-based framework can speed up the training time up to 49% and improve the accuracy level up to 4.6 times compared with those of baseline methods, i.e., conventional FL methods without using the proposed framework. Furthermore, the proposed framework can enhance the social welfare of the mobile network up to 114% compared with that of the baseline methods. The main contributions of our work are as follows:

- Propose a novel FL-based framework for the privacy-aware mobile application service that considers additional encrypted training process, privacy protection costs, the limited computing/communications resources at MUs and the MAP, and incentive competition among MUs in contributing data to the MAP. This framework is resilient to the straggling

problems while preserving MUs' privacy.

- Introduce an MU selection scheme using the MUs' available dataset and device information. As such, we can choose the best MUs that can improve the overall privacy-preserving FL process with minimal stragglng problems.
- Formulate the MPOA-based FL contract problem to address the encrypted training process competition among the selected MUs under the MAP's information asymmetry. Then, we transform the original FL contract problem into a low-complexity equivalent problem and design the iterative algorithm that can quickly find the equilibrium solution for the selected MUs.
- Theoretically analyze the convergence of proposed FL with stragglng mitigation and privacy-awareness.
- Perform extensive experiments with a real-world HAR dataset. The results can provide useful information to help the MAP in designing the stable FL process with privacy-awareness for AI-based mobile application services.

The rest of this paper is organized as follows. Section 2 introduces the proposed FL-based framework. Section 3 presents the proposed MU selection method. Then, the proposed MPOA contract-based problem and solution are described in Section 4 and Section 5, respectively. Section 6 explains the detailed learning process of the proposed FL. The performance evaluation is provided in Section 7, and then the conclusion is summarized in Section 8.

## 2 PROPOSED FEDERATED LEARNING FRAMEWORK WITH STRAGGLING MITIGATION AND PRIVACY-AWARENESS

In this section, we first provide an overview of FL approach with stragglng mitigation and privacy-awareness. Then, we present the proposed FL-based framework for the AI-based mobile application service.

### 2.1 Federated Learning with Stragglng Mitigation and Privacy-Awareness Overview

The process of FL with stragglng mitigation and privacy-awareness is illustrated in Fig. 1 that consists of the MAP (as a master node) and participating MUs (as workers)  $\mathcal{N} = \{1, \dots, n, \dots, N\}$ . Each MU- $n$  has its own sensing dataset  $S_n = (\mathbf{F}_n, \mathbf{L}_n)$ , where  $\mathbf{F}_n$  and  $\mathbf{L}_n$  are the training feature and training label matrices of the dataset at MU- $n$  with the size  $D_n = |S_n|$ , respectively. Here,  $|S_n|$  is the number of samples at MU- $n$ . Unlike the conventional FL, the proposed FL first incorporates a *pre-FL process*, that is only executed once prior to implementing the FL process. The pre-FL aims to perform data encryption and offloading for an additional training process at the MAP, aiming at avoiding the stragglng problems, addressing the limited computing resources at MUs, and maintaining the learning quality. Specifically, as shown in Fig. 1(a), each MU- $n$  first encrypts part of its sensing dataset to protect its privacy. Then, the encrypted datasets from all MUs can be collected by the MAP for the additional encrypted training process. In

this case, the local/unencrypted dataset at each MU will be used for the local learning process as similarly implemented in the conventional FL approach. The mechanism on how to obtain optimal sizes of encrypted and unencrypted datasets is explained in Section 4.

Let  $D_n^o$  and  $D_n^l$  represent the sizes of encrypted and local (unencrypted) sensing datasets of MU- $n$ , respectively, such that  $D_n^o + D_n^l \leq D_n, \forall n \in \mathcal{N}$ . This condition indicates that some MUs may not want to train all their datasets in practice. To encrypt a specific dataset, each MU- $n$  can use fully homomorphic encryption (FHE) that is widely used for data encryption in untrusted environments, e.g., third parties, without masking or dropping any features to protect the data privacy [15]. To improve the performance of FHE, we adopt Brakerski/Fan-Vercauteren (BFV)-based FHE to support concurrent arbitrary operations on encrypted data [16]. Specifically, the BFV-based FHE includes the following polynomial-time algorithms during the encryption and decryption processes.

- $SKGen(n)$  to generate the secret key  $g_n^{sk}$  for MU- $n$ .
- $PKGen(g_n^{sk})$  to generate the public key  $g_n^{pk}$  as the function of  $g_n^{sk}$  for MU- $n$ .
- $Enc(g_n^{pk}, h)$  to encrypt data  $h$  utilizing  $g_n^{pk}$  with the output of encrypted data  $h^*$ .
- $Dec(g_n^{sk}, h^*)$  to decrypt data  $h^*$  utilizing  $g_n^{sk}$  with the output of original data  $h$ .
- $Add(h_1^*, h_2^*)$ ,  $Sub(h_1^*, h_2^*)$  and  $Mul(h_1^*, h_2^*)$  to respectively add, subtract, and multiply encrypted data  $h_1^*$  and  $h_2^*$  with the output of  $h_{add}^*$ ,  $h_{sub}^*$ , and  $h_{mul}^*$ .

Based on the aforementioned algorithms, each MU- $n$  first can generate the secret key  $g_n^{sk}$  using  $SKGen(n)$  and public key  $g_n^{pk}$  using  $PKGen(g_n^{sk})$ . The  $g_n^{sk}$  of MU- $n$  remains private to other MUs and the MAP. Meanwhile,  $g_n^{pk}$  of MU- $n$  is known by other MUs and the MAP. Then, each MU- $n$  can encrypt a part of the local dataset  $S_n$ , i.e.,  $S_n^\dagger$ , utilizing its  $g_n^{pk}$  to produce the encrypted dataset  $S_n^o = (\mathbf{F}_n^o, \mathbf{L}_n^o)$ , i.e.,  $Enc(g_n^{pk}, S_n^\dagger) = S_n^o$ , where  $\mathbf{F}_n^o$  and  $\mathbf{L}_n^o$  are the encrypted training feature and encrypted training label matrices of the dataset at MU- $n$  with the size  $D_n^o = |S_n^o|$ , respectively. All the  $S_n^o, \forall n \in \mathcal{N}$ , then can be collected by the MAP for the accumulation process prior to the encrypted training execution. As such, the training feature and label matrices of encrypted dataset from all MUs in  $\mathcal{N}$  can be respectively concatenated into

$$\mathbf{F}^o = \begin{pmatrix} \mathbf{F}_1^o \\ \mathbf{F}_2^o \\ \vdots \\ \mathbf{F}_N^o \end{pmatrix}, \text{ and } \mathbf{L}^o = \begin{pmatrix} \mathbf{L}_1^o \\ \mathbf{L}_2^o \\ \vdots \\ \mathbf{L}_N^o \end{pmatrix}. \quad (1)$$

To minimize extra privacy leakage when sharing the encrypted datasets  $S_n^o, \forall n \in \mathcal{N}$ , with the MAP, e.g., meta-data leakage of the encrypted datasets via their plaintext headers [25], a privacy protection level  $\varepsilon_n$  ( $\varepsilon_n \leq 1$ ) [26] with respect to the encrypted dataset size of MU- $n$  can be deployed. Specifically, the higher the privacy protection level (i.e., higher  $\varepsilon_n$ ) is, the higher the privacy cost for the MU- $n$  is. As such, the privacy cost for the encrypted dataset generation and sharing of MU- $n$  can be written by [17]

$$\xi_n = \frac{\beta}{2} \log_2 \left( 1 + \frac{\varepsilon_n D_n^o}{A_n^2} \right), \quad (2)$$

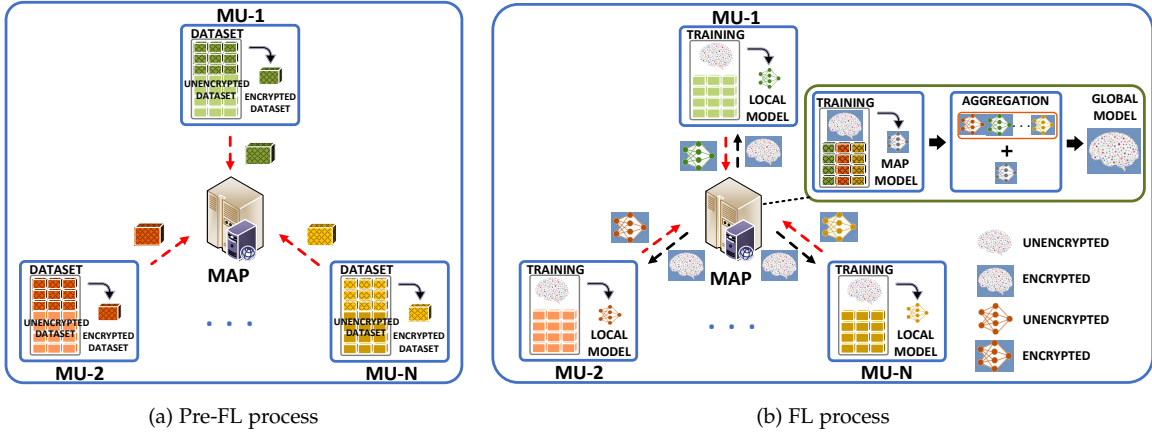


Fig. 1: The FL with straggling mitigation and privacy-awareness with (a) one-time pre-FL and (b) iterative FL processes.

where  $\beta$  is the unit cost of the privacy and  $A_n$  is a function to quantify the influence of raw dataset distribution with respect to the number of features at MU- $n$  defined in [17]. Equation (2) is later used for contract optimization in Section 4 (note that our framework can adopt any privacy cost metric for the contract optimization).

Upon completing the pre-FL process, the FL process between the MAP and all MUs in  $\mathcal{N}$  can be observed in Fig. 1(b). At each learning round, all MUs first train their local unencrypted datasets and then send the encrypted local trained models to the MAP (to preserve the privacy of the local trained models). Meanwhile, the MAP also trains all the collected encrypted datasets to produce another encrypted trained model. After completing the learning process within a training time threshold at each round (which is predefined by the MAP), the MAP can collect the encrypted local models from the MUs and aggregate them to obtain the aggregated encrypted local model. After that, the aggregated encrypted local model from MUs and encrypted model from the MAP can be combined together to update the encrypted global model, which is then used for the next FL process at the MUs and MAP. Note that the MUs are required to decrypt the encrypted global model before utilizing it to train the unencrypted local dataset [13]. Moreover, for each learning round, the MAP only requires to aggregate any encrypted models that are received within a pre-defined training time threshold, due to additional training at the MAP. The above processes aim to guarantee the learning convergence while minimizing the straggling problems from the straggling MUs.

## 2.2 Proposed Federated Learning-Based Framework for the Mobile Application Service

To effectively execute the above FL approach under the limited computing resources of the MAP as well as participating MUs, privacy protection costs in implementing encryption mechanism, and the conflict-of-interest among the MUs, we then describe the proposed FL-based framework for the mobile application service. Suppose that there exists a set of active MUs in the network (as the candidate workers that utilize smart devices, e.g., smartphones and smartwatches) denoted by  $\mathcal{M} = \{1, \dots, m, \dots, M\}$ . At a particular period, each

active MU- $m$  through the mobile device can capture motion sensing information through its embedded sensor devices, e.g., accelerometer and gyroscope. This sensing information can be stored in a log file at the mobile device's internal storage. To participate in the FL process, the interested MUs can first install a human activity service application established by the MAP at their mobile devices' built-in Android or iOS platforms, e.g., Samsung Health ([www.samsung.com/global/galaxy/apps/samsung-health](http://www.samsung.com/global/galaxy/apps/samsung-health)) and Apple Health ([www.apple.com/au/ios/health](http://www.apple.com/au/ios/health)). Then, the MUs can securely share their selection metric values with respect to sizes of datasets, device configurations, and communication connections in the network, for the entire FL process. Note that such sharing information is only used to help the MAP choose the best MUs without disclosing private sensing data that are collected by the selected MUs when they perform the sensing process. In particular, the set of participating MUs in  $\mathcal{N}$ , where  $\mathcal{N} \subset \mathcal{M}$ , is chosen by the MAP based on the above accessible information. The selection of  $N$  MUs ensures that they can provide useful sensing data for the FL process.

To reduce the straggling problem at MU- $n$ , the size of local sensing dataset must be less than or equal to the maximum size of dataset that can be processed at the MU- $n$ , i.e.,  $\hat{D}_n^l$ , for the entire FL process, which is  $D_n^l \leq \hat{D}_n^l$  [6]. For the collected encrypted dataset at the MAP, its size is constrained by maximum size of encrypted sensing samples  $D_{max}^o$  that can be trained at the MAP (based on the MAP's computing resources). In practice, the MAP retains its willingness to train the collected encrypted datasets as an unknown information for the MUs due to its economic benefit. This willingness is defined as the type of the MAP [28] and influenced by the MAP's current computing resources. Particularly, a higher type implies the willingness to train more encrypted sensing datasets from the MUs due to its higher available computing resources. In other words, the willingness to train more encrypted datasets can compensate the low computing resources and straggling problems more, and thus bring more benefits for the MUs and better learning quality for the MAP. To this end, a finite set of the MAP's types can be specified as  $\Pi = \{\pi_1, \dots, \pi_i, \dots, \pi_I\}$ , and  $\pi_1 < \pi_2 < \dots < \pi_i < \dots < \pi_{I-1} < \pi_I$ , where  $i \in \mathcal{I}$  with  $\mathcal{I} = \{1, \dots, i, \dots, I\}$ , represents the type index.

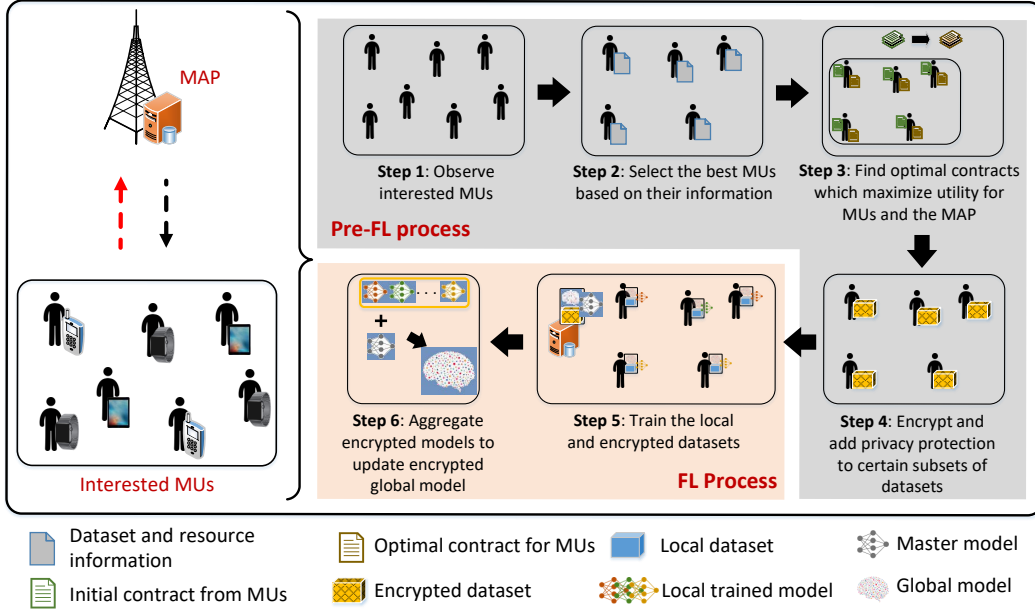


Fig. 2: The proposed FL-based framework for the mobile application service.

Despite that the MAP's type is private, the MUs can still observe the distribution of MAP's types, i.e.,  $\phi_i$ , where  $\sum_{i=1}^I \phi_i = 1, \forall i \in \mathcal{I}$  [26], e.g., by monitoring public workloads of the MAP in the previous FL processes. For the MAP with type  $\pi_i$ , it has the maximum trainable encrypted dataset size  $\hat{D}_i^o$ , where  $\hat{D}_i^o = \frac{\pi_i}{\pi_I} D_{max}^o, \forall i \in \mathcal{I}$ . From  $\hat{D}_i^o$ , the MAP can allocate the encrypted dataset proportion for each MU- $n$  correspondingly. Specifically, for the MAP with type  $\pi_i$ , the encrypted dataset proportion vector of all MUs can be denoted by  $\eta = [\eta_1, \dots, \eta_i, \dots, \eta_I]$ , where  $\eta_i = [\eta_i^1, \dots, \eta_i^n, \dots, \eta_i^N]$ , and  $0 \leq \eta_i^n \leq 1, \forall i \in \mathcal{I}, \forall n \in \mathcal{N}$ . We denote the expected size of encrypted dataset and corresponding payment vectors of all participating MUs for all MAP's types as  $D^o = [D_1^o, \dots, D_i^o, \dots, D_I^o]$  and  $\rho^o = [\rho_1^o, \dots, \rho_i^o, \dots, \rho_I^o]$ , respectively, where  $D_i^o = [D_{i,1}^o, \dots, D_{i,n}^o, \dots, D_{i,N}^o]$  and  $\rho_i^o = [\rho_{i,1}^o, \dots, \rho_{i,n}^o, \dots, \rho_{i,N}^o]$ . In this case, the actual size of encrypted dataset at MU- $n$  considering the MAP with type  $\pi_i$  will be  $\eta_i^n D_{i,n}^o$ . Moreover, each MU- $n$  can offer a bigger size of encrypted sensing dataset to the MAP when the MAP's type increases because the MU- $n$  can obtain higher payment from the MAP. Furthermore, we denote  $D^l = [D_1^l, \dots, D_n^l, \dots, D_N^l]$  and  $\rho^l = [\rho_1^l, \dots, \rho_n^l, \dots, \rho_N^l]$  to be the size of local dataset and corresponding payment vectors of all MUs, respectively. Overall, as illustrated in Fig. 2, the whole steps of proposed FL-based framework can be executed sequentially as follows:

- Step 1: The MAP broadcasts FL invitation to all MUs in the mobile network and observes the interested MUs therein. If an MU agrees to join the FL process, he/she can securely share his/her selection metric values to the MAP via the application.
- Step 2: The MAP chooses  $N$  active MUs based on their securely-shared information. More details are described in Section 3.
- Step 3: The selected MUs send initial contracts containing the sizes of required local training data and

encrypted data along with their requested payments to the MAP for local and additional training processes, respectively. Then, the MAP will optimize the contracts and send them back to the MUs.

- Step 4: Using the optimal sizes of encrypted data, all the MUs encrypt and add the privacy protection to specific subsets of dataset prior to sending them to the MAP.
- Step 5: All the MUs train the optimized local datasets to produce local trained models independently. Meanwhile, the MAP will use all the collected encrypted datasets to produce an encrypted trained model.
- Step 6: After reaching the training time threshold at each learning round, their encrypted local models are then collected by the MAP and combined with the MAP's encrypted trained model to update the current encrypted global model (before sending it back to the MUs).

In this case, Steps 1 to 4 are only implemented once before the FL process starts. Differently, Steps 5 and 6 are repeated for each learning round until the global model converges or after a pre-defined learning rounds is reached. More details are presented in the following sections.

### 3 MU SELECTION WITH AVAILABLE DATASET AND DEVICE INFORMATION

To select  $N$  participating MUs in the FL process, the MAP via the service application can first collect selection metric values from current active MUs in  $\mathcal{M}$  based on their available sensing dataset quantity, general device information, and communication connections. Specifically, each MU- $m$  can first observe the quantity of available sensing dataset as the size of the dataset  $D_m$ . Since we consider the time-series sensing dataset in this work, the larger the size of dataset is (due to longer sensing period), the better the quality of

dataset for the learning process is. For example, an MU that employs a sensing process for three hours usually obtains a higher number of training samples than those with less than one hour (considering that both MUs have the same sampling rate to do the sensing process continuously). As such, the MU with a larger dataset size can likely improve the learning process quality with a higher accuracy [29].

The active MUs also require to observe their general device configuration, e.g., computing resources, to execute the training process locally at the MUs efficiently. Here, the computing resources of each MU- $m$  can be denoted by  $\kappa_m$ . For the communication connections, each active MU- $m$  can observe the available transmission rate  $R_m$  (based on its hardware configuration).

From the observed  $D_m$ ,  $\kappa_m$ , and  $R_m$ , the MUs via the application can first normalize the sets of  $D_m$ ,  $\kappa_m$ , and  $R_m$ ,  $\forall m \in \mathcal{M}$ , within  $0 - 1$  range separately and obtain the normalized dataset size  $\tilde{D}_m$ , normalized computing resources  $\tilde{\kappa}_m$ , and normalized transmission rate  $\tilde{R}_m$ ,  $\forall m \in \mathcal{M}$ . Then, the MUs via the application can calculate their selection metric values, i.e.,  $\chi_m = \iota_1 \tilde{D}_m + \iota_2 \tilde{\kappa}_m + \iota_3 \tilde{R}_m$ . In this case,  $\iota_1$ ,  $\iota_2$ , and  $\iota_3$  are the selection weight parameters determined by the MAP to control  $\chi_m$ , where  $\iota_1 + \iota_2 + \iota_3 = 1$  [29], [30]. These  $\chi_m$ ,  $\forall m \in \mathcal{M}$ , are then shared by the MUs to the MAP. Using such selection metric values, the MAP can select  $N$  MUs that will execute the FL process based on their largest  $\chi_m$  values, i.e.,

$$N = \max_{[N]} \{\chi_1, \dots, \chi_M\}. \quad (3)$$

Equation (3) indicates that only MUs in  $\mathcal{N}$  can participate in the FL process, i.e., train local datasets and then send local trained models to the MAP for the global model update.

#### 4 MPOA-BASED FL CONTRACT OPTIMIZATION PROBLEM

Upon completing the MU selection process, the MPOA-based FL contract optimization problem can be formulated. Specifically, the selected MUs can send initial contracts including the sizes of local datasets  $D_n^l$ ,  $\forall n \in \mathcal{N}$  (where  $0 \leq D_n^l \leq D_n$ ), and requested payments  $\rho_n^l = \alpha_l D_n^l$ ,  $\forall n \in \mathcal{N}$ , as well as the sizes of encrypted datasets  $D_{i,n}^o$ ,  $\forall n \in \mathcal{N}$ ,  $\forall i \in \mathcal{I}$  (where  $0 \leq D_{i,n}^o \leq D_n$ ), and requested payments  $\rho_{i,n}^o = \alpha_o D_{i,n}^o$ ,  $\forall n \in \mathcal{N}$ ,  $\forall i \in \mathcal{I}$ . These  $\alpha_l$  and  $\alpha_o$  specify unit prices of using a local sensing sample for local training at an MU and an encrypted sensing sample for training process at the MAP, respectively. Using these initial contracts, the MAP can help the MUs find the optimal contracts under the competition among the MUs in the proposed FL process.

##### 4.1 Utility Optimization for the MAP

Accounting for the encrypted dataset proportion vector for the MAP with type  $\pi_i$ , i.e.,  $\eta_i$ , the size of offered local dataset, i.e.,  $D^l$ , and its payment vector, i.e.,  $\rho^l$ , the size of offered encrypted dataset for the MAP with type  $\pi_i$ , i.e.,  $D_i^o$ , and its payment vector for the MAP with type  $\pi_i$ , i.e.,  $\rho_i^o$ , the utility of the MAP with type  $\pi_i$  (for executing the

proposed FL process along with the MU- $n$ ,  $\forall n \in \mathcal{N}$ ) can be expressed as follows:

$$U_{MAP}^i = \underbrace{\pi_i G_o(\eta_i, D_i^o) - C_o(\eta_i, \rho_i^o, D_i^o)}_{\text{Utility from training encrypted data}} + \underbrace{G_l(D^l) - C_l(\rho^l)}_{\text{Utility from local training at MUs}}. \quad (4)$$

Specifically, the utility of the MAP with type  $\pi_i$  can be decomposed into two components. The first component is the difference between the gain function  $G_o(\eta_i, D_i^o)$  and cost function  $C_o(\eta_i, \rho_i^o, D_i^o)$  for collecting and training the encrypted dataset at the MAP, where  $\pi_i$  represents the weight of  $G_o(\eta_i, D_i^o)$  for the MAP with type  $\pi_i$ . Meanwhile, the second component is the difference between the gain function  $G_l(D^l)$  and cost function  $C_l(\rho^l)$  for local training process at the MUs.

For both gain functions, we use a squared-root function as that of [26] since in terms of economic perspective, the gain increases when a dataset with larger size is used in the FL process. However, the MAP may have less interest to further enhance the gain when much larger size of dataset incurs less global model accuracy improvement [31]. To this end, the gain functions for training encrypted and local datasets can be respectively formulated by

$$G_o(\eta_i, D_i^o) = v_o \sqrt{\sum_{n \in \mathcal{N}} \eta_i^n D_{i,n}^o}, \text{ and } G_l(D^l) = v_l \sqrt{\sum_{n \in \mathcal{N}} D_n^l}, \quad (5)$$

where  $v_o > 0$  and  $v_l > 0$  are the conversion parameters indicating the monetary unit of utilizing the encrypted and local datasets, respectively, according to the current data trading market [26]. The MAP's cost function  $C_o(\eta_i, \rho_i^o, D_i^o)$  is the sum of total payments for participating MUs in  $\mathcal{N}$  (in regards to their encrypted dataset sharing) and energy consumption cost for training the combined encrypted dataset, i.e.,

$$C_o(\eta_i, \rho_i^o, D_i^o) = \sum_{n \in \mathcal{N}} \eta_i^n \rho_{i,n}^o + \zeta c f^2 \sum_{n \in \mathcal{N}} \eta_i^n D_{i,n}^o, \quad (6)$$

where  $\zeta$ ,  $c$ , and  $f$  are the efficient capacitance constant of computing chipset, the number of CPU cycles to execute one sample of encrypted dataset, and utilized computing resources for the MAP, respectively. Moreover, the MAP's cost function  $C_l(\rho^l)$  is the total payments for the participating MUs with respect to the local training process at the MUs, i.e.,

$$C_l(\rho^l) = \sum_{n \in \mathcal{N}} \rho_n^l. \quad (7)$$

Based on (4)-(7), the MAP's utility maximization problem for type  $\pi_i$  can be written as:

$$(\mathbf{P}_1) \quad \max_{\eta_i} U_{MAP}^i, \quad (8)$$

$$\text{s.t.} \quad \sum_{n \in \mathcal{N}} \eta_i^n D_{i,n}^o \leq \hat{D}_i^o, \quad (9)$$

$$0 \leq \eta_i^n \leq 1, \forall n \in \mathcal{N}, \quad (10)$$

where constraint (9) ensures that the total encrypted dataset collected by the MAP cannot exceed the maximum trainable encrypted dataset at the MAP with type  $\pi_i$ , i.e.,  $\hat{D}_i^o$ . As the objective function (8) is a convex function, i.e., the gain and cost functions are respectively concave and linear, and the constraints (9)-(10) in  $(\mathbf{P}_1)$  are linear, the optimal  $\hat{\eta}_i$ ,  $\forall i \in \mathcal{I}$ , can be found using popular convex solvers.

## 4.2 Utility Optimization for Participating MUs

Using the optimal encrypted dataset proportion  $\hat{\eta}_i, \forall i \in \mathcal{I}$ , each MU- $n$  can derive its expected utility by considering all the MAP's possible types, which is

$$U_n(\mathbf{D}^o, \boldsymbol{\rho}^o, \mathbf{D}^l, \boldsymbol{\rho}^l) = \sum_{i=1}^I \left( \underbrace{\hat{\eta}_i^n \rho_{i,n}^o - \xi_{i,n} - \gamma \hat{\eta}_i^n D_{i,n}^o}_{\text{Utility of training encrypted data at MAP}} + \underbrace{\rho_n^l - \zeta_n c_n f_n^2 D_n^l}_{\text{Utility from local training}} \right) \phi_i. \quad (11)$$

Similar to the MAP's utility, the utility of MU- $n$  can be divided into two components. The first component is the difference between the received payment  $\hat{\eta}_i^n \rho_{i,n}^o$  for sharing the encrypted dataset and the costs for encrypted dataset privacy protection and transmission, where  $\xi_{i,n} = \frac{\beta}{2} \log_2 \left( 1 + \frac{\varepsilon_n \hat{\eta}_i^n D_{i,n}^o}{A_n^2} \right)$  indicates the privacy cost for the encrypted dataset generation of MU- $n$  when the MAP has type  $\pi_i$ . Additionally,  $\gamma \hat{\eta}_i^n D_{i,n}^o$  denotes the transmission cost of encrypted dataset, where  $\gamma$  is the unit cost of sending an encrypted sample to the MAP. Meanwhile, the second component is the difference between the received payment and the cost of energy consumption  $\zeta_n c_n f_n^2 D_n^l$  for training the local dataset, where  $\zeta_n$ ,  $c_n$ , and  $f_n$  are the efficient capacitance constant of computing chipset, the number of CPU cycles to execute one sample of local dataset, and utilized computing resources for the MU- $n$ , respectively. In addition, the use of  $\sum_{i=1}^I (\cdot) \phi_i$  means that the expected utility of each MU- $n$  depends on the type distribution of the MAP.

In this utility optimization, we can find optimal contracts  $(\mathbf{D}^o, \boldsymbol{\rho}^o, \mathbf{D}^l, \boldsymbol{\rho}^l)$  that satisfy the MAP's individual rationality (IR) and incentive compatibility (IC) constraints. Particularly, the IR constraints ensure that the MAP with all possible types will produce non-negative utilities as stated in Definition 1.

**Definition 1.** *IR constraint: The MAP with type  $\pi_i$  must achieve a positive utility, i.e.,*

$$\pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \boldsymbol{\rho}_i^o, \mathbf{D}_i^o) + G_l(\mathbf{D}^l) - C_l(\boldsymbol{\rho}^l) \geq 0, \quad \forall i \in \mathcal{I}, \quad (12)$$

to join in the contract optimization.

Using the IR constraints, the utility maximization problem for each MU- $n$  when all MUs have full information of the MAP with true type  $\pi_i$  (information-symmetry contract) can be derived as follows:

$$(\mathbf{P}_2) \quad \max_{\mathbf{D}^o, \boldsymbol{\rho}^o, \mathbf{D}^l, \boldsymbol{\rho}^l} \hat{\eta}_i^n \rho_{i,n}^o - \xi_{i,n} - \gamma \hat{\eta}_i^n D_{i,n}^o + \rho_n^l - \zeta_n c_n f_n^2 D_n^l, \quad \forall n \in \mathcal{N}, \quad (13)$$

$$\text{s.t.} \quad \sum_{n \in \mathcal{N}} \hat{\eta}_i^n D_{i,n}^o \leq \hat{D}_i^o, \quad (14)$$

$$D_n^l \leq \hat{D}_n^l, \quad \forall n \in \mathcal{N}, \quad (15)$$

$$\hat{\eta}_i^n D_{i,n}^o + D_n^l \leq D_n, \quad \forall n \in \mathcal{N}, \quad (16)$$

$$\pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \boldsymbol{\rho}_i^o, \mathbf{D}_i^o) + G_l(\mathbf{D}^l) - C_l(\boldsymbol{\rho}^l) \geq 0, \quad (17)$$

where the constraint (15) represents that the dataset which will be trained locally cannot exceed the maximum trainable local dataset at the MU- $n$ . Then, the constraint (16) specifies that the total trainable datasets of MU- $n$ , i.e., encrypted and

local datasets, must be less than or equal to the available data at MU- $n$ .

Since there exists the information asymmetry between the MAP and MUs in practice, the optimal contracts must also satisfy the IC constraints to guarantee that the contracts are feasible. The IC constraint ensures that the MAP with a certain type can achieve the maximum utility when the appropriate contract originated for that type of the MAP is applied. This condition can be described in the following Definition 2.

**Definition 2.** *IC constraint: The MAP with current type  $\pi_i$  will select a contract designed for its current type  $\pi_i$  rather than with another type  $\pi_{i^*}$  to maximize its utility, i.e.,*

$$\begin{aligned} \pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \boldsymbol{\rho}_i^o, \mathbf{D}_i^o) + G_l(\mathbf{D}^l) - C_l(\boldsymbol{\rho}^l) &\geq \\ \pi_i G_o(\hat{\eta}_{i^*}, \mathbf{D}_{i^*}^o) - C_o(\hat{\eta}_{i^*}, \boldsymbol{\rho}_{i^*}^o, \mathbf{D}_{i^*}^o) + G_l(\mathbf{D}^l) - C_l(\boldsymbol{\rho}^l), & \\ i \neq i^*, \forall i, i^* \in \mathcal{I}, & \end{aligned} \quad (18)$$

$$\begin{aligned} \pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \boldsymbol{\rho}_i^o, \mathbf{D}_i^o) &\geq \\ \pi_i G_o(\hat{\eta}_{i^*}, \mathbf{D}_{i^*}^o) - C_o(\hat{\eta}_{i^*}, \boldsymbol{\rho}_{i^*}^o, \mathbf{D}_{i^*}^o), & \\ i \neq i^*, \forall i, i^* \in \mathcal{I}. & \end{aligned} \quad (19)$$

From the IR and IC constraint definitions, we can formulate the optimization problem for each MU- $n$  which can maximize its expected utility individually at the MAP under the competition with other MUs in  $\mathcal{N}$  and the MAP's unknown limited computing resources as follows:

$$(\mathbf{P}_3) \quad \max_{\mathbf{D}^o, \boldsymbol{\rho}^o, \mathbf{D}^l, \boldsymbol{\rho}^l} U_n(\mathbf{D}^o, \boldsymbol{\rho}^o, \mathbf{D}^l, \boldsymbol{\rho}^l), \quad \forall n \in \mathcal{N}, \quad (20)$$

$$\text{s.t.} \quad \sum_{n \in \mathcal{N}} \hat{\eta}_i^n D_{i,n}^o \leq \hat{D}_i^o, \quad \forall i \in \mathcal{I}, \quad (21)$$

$$D_n^l \leq \hat{D}_n^l, \quad \forall n \in \mathcal{N}, \quad (22)$$

$$\hat{\eta}_i^n D_{i,n}^o + D_n^l \leq D_n, \quad \forall i \in \mathcal{I}, \forall n \in \mathcal{N}, \quad (23)$$

$$\begin{aligned} \pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \boldsymbol{\rho}_i^o, \mathbf{D}_i^o) + G_l(\mathbf{D}^l) - C_l(\boldsymbol{\rho}^l) &\geq 0, \\ \forall i \in \mathcal{I}, & \end{aligned} \quad (24)$$

$$\begin{aligned} \pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \boldsymbol{\rho}_i^o, \mathbf{D}_i^o) &\geq \\ \pi_i G_o(\hat{\eta}_{i^*}, \mathbf{D}_{i^*}^o) - C_o(\hat{\eta}_{i^*}, \boldsymbol{\rho}_{i^*}^o, \mathbf{D}_{i^*}^o), & \\ i \neq i^*, \forall i, i^* \in \mathcal{I}. & \end{aligned} \quad (25)$$

## 4.3 Social Welfare of the Mobile Network

To devise the social welfare, i.e., the total actual utilities of the MAP and all participating MUs in the proposed FL process, we can first obtain the actual utility of each MU- $n$  when the MAP has type  $\pi_i$  as  $\hat{\eta}_i^n \rho_{i,n}^o - \frac{\beta}{2} \log_2 \left( 1 + \frac{\varepsilon_n \hat{\eta}_i^n D_{i,n}^o}{A_n^2} \right) - \gamma \hat{\eta}_i^n D_{i,n}^o + \rho_n^l - \zeta_n c_n f_n^2 D_n^l$ . Given the MAP with type  $\pi_i$ , the total actual utility of all participating MUs can be derived by

$$U_{MU_{tot}}^i = \sum_{n \in \mathcal{N}} \left( \hat{\eta}_i^n \rho_{i,n}^o - \frac{\beta}{2} \log_2 \left( 1 + \frac{\varepsilon_n \hat{\eta}_i^n D_{i,n}^o}{A_n^2} \right) - \gamma \hat{\eta}_i^n D_{i,n}^o + \rho_n^l - \zeta_n c_n f_n^2 D_n^l \right). \quad (26)$$

Thus, the social welfare of the network when the MAP has type  $\pi_i$  based on (4) and (26) can be computed by

$$U_{SW}^i = U_{MAP}^i + U_{MU_{tot}}^i. \quad (27)$$



## 5 MPOA-BASED FL CONTRACT OPTIMIZATION SOLUTION

### 5.1 Contract Problem Transformation

The following Proposition 1 shows that the computational complexity of solving  $(\mathbf{P}_3)$  increases quadratically when the number of MAP's possible types increases, i.e.,  $O(I^2)$ , leading to a high computing resources to solve the  $(\mathbf{P}_3)$  at the MAP.

**Proposition 1.** *The computational complexity of solving  $(\mathbf{P}_3)$  is  $O(I^2)$ .*

*Proof.* See Appendix A.  $\square$

To reduce the complexity of solving  $(\mathbf{P}_3)$ , we transform  $(\mathbf{P}_3)$  into an equivalent problem using IR and IC constraint transformation. In particular, the Lemma 1 below shows that all the participating MUs will offer larger encrypted datasets to the MAP when the MAP's type is higher, i.e., the higher willingness to train more encrypted dataset because of higher computing resources.

**Lemma 1.** *Given  $(\mathbf{D}^o, \rho^o, \mathbf{D}^l, \rho^l)$  to be any feasible contracts from the MUs to the MAP such that if  $\pi_i \geq \pi_{i^*}$ , then  $\mathbf{D}_i^o \geq \mathbf{D}_{i^*}^o$ , where  $i \neq i^*, i, i^* \in \mathcal{I}$ .*

*Proof.* See Appendix B.  $\square$

As a result, the MUs will ask for higher payments to the MAP when the MAP has type  $\pi_i$  since  $\mathbf{D}_i^o \geq \mathbf{D}_{i^*}^o$  from Lemma 1. This condition is stated in Proposition 2.

**Proposition 2.** *If  $\mathbf{D}_i^o \geq \mathbf{D}_{i^*}^o$ , then  $\rho_i^o \geq \rho_{i^*}^o$ , where  $i \neq i^*, i, i^* \in \mathcal{I}$ .*

*Proof.* See Appendix C.  $\square$

From Lemma 1 and Proposition 2, the following Proposition 3 shows that the utility of the MAP is monotonically increasing with its type.

**Proposition 3.** *For any feasible contract  $(\mathbf{D}^o, \rho^o, \mathbf{D}^l, \rho^l)$ , the MAP's utility must hold*

$$\begin{aligned} & \pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \rho_i^o, \mathbf{D}_i^o) + G_l(\mathbf{D}^l) - C_l(\rho^l) \geq \\ & \pi_{i^*} G_o(\hat{\eta}_{i^*}, \mathbf{D}_{i^*}^o) - C_o(\hat{\eta}_{i^*}, \rho_{i^*}^o, \mathbf{D}_{i^*}^o) + G_l(\mathbf{D}^l) - C_l(\rho^l) \\ & \pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \rho_i^o, \mathbf{D}_i^o) \geq \\ & \pi_{i^*} G_o(\hat{\eta}_{i^*}, \mathbf{D}_{i^*}^o) - C_o(\hat{\eta}_{i^*}, \rho_{i^*}^o, \mathbf{D}_{i^*}^o), \end{aligned} \quad (28)$$

where  $\pi_i \geq \pi_{i^*}, i \neq i^*, \forall i, i^* \in \mathcal{I}$ .

*Proof.* See Appendix D.  $\square$

The Proposition 3 helps reducing the number of IR constraints through applying the MAP's minimum type, i.e.,  $\pi_1$ , such that  $\pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \rho_i^o, \mathbf{D}_i^o) + G_l(\mathbf{D}^l) - C_l(\rho^l) \geq \pi_i G_o(\hat{\eta}_1, \mathbf{D}_1^o) - C_o(\hat{\eta}_1, \rho_1^o, \mathbf{D}_1^o) + G_l(\mathbf{D}^l) - C_l(\rho^l) \geq \pi_1 G_o(\hat{\eta}_1, \mathbf{D}_1^o) - C_o(\hat{\eta}_1, \rho_1^o, \mathbf{D}_1^o) + G_l(\mathbf{D}^l) - C_l(\rho^l) \geq 0$ . To this end, the IR constraints for other types  $\pi_i, i > 1$ , are satisfied if and only if the IR constraint for  $\pi_1$  is held. Thus, the IR constraints in (22) can be transformed as follows:

$$\pi_1 G_o(\hat{\eta}_1, \mathbf{D}_1^o) - C_o(\hat{\eta}_1, \rho_1^o, \mathbf{D}_1^o) + G_l(\mathbf{D}^l) - C_l(\rho^l) \geq 0. \quad (29)$$

Based on the following Lemma 2, we can also reduce the number of IC constraints in (23).

**Lemma 2.** *The IC constraints in (23) of  $(\mathbf{P}_3)$  can be transformed into the local downward incentive constraints (LDIC) as follows:*

$$\pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \rho_i^o, \mathbf{D}_i^o) \geq \quad (30)$$

$$\pi_i G_o(\hat{\eta}_{i-1}, \mathbf{D}_{i-1}^o) - C_o(\hat{\eta}_{i-1}, \rho_{i-1}^o, \mathbf{D}_{i-1}^o), \forall i \in \{2, \dots, I\},$$

where  $\mathbf{D}_i^o \geq \mathbf{D}_{i-1}^o, \forall i \in \{2, \dots, I\}$ .

*Proof.* See Appendix E.  $\square$

Equation (30) indicates that if the IC constraint for type  $\pi_{i-1}$  holds, then all other IC constraints are also satisfied as long as the conditions in Lemma 1 hold. Finally, using (29) and (30), the optimization problem  $(\mathbf{P}_3)$  can be transformed into the problem  $(\mathbf{P}_4)$  as follows:

$$(\mathbf{P}_4) \quad \max_{\mathbf{D}^o, \rho^o, \mathbf{D}^l, \rho^l} U_n(\mathbf{D}^o, \rho^o, \mathbf{D}^l, \rho^l), \forall n \in \mathcal{N}, \quad (31)$$

s.t. (21)-(23), and,

$$\pi_1 G_o(\hat{\eta}_1, \mathbf{D}_1^o) - C_o(\hat{\eta}_1, \rho_1^o, \mathbf{D}_1^o) + G_l(\mathbf{D}^l) - C_l(\rho^l) \geq 0, \quad (32)$$

$$\pi_i G_o(\hat{\eta}_i, \mathbf{D}_i^o) - C_o(\hat{\eta}_i, \rho_i^o, \mathbf{D}_i^o) \geq \quad (33)$$

$$\pi_i G_o(\hat{\eta}_{i-1}, \mathbf{D}_{i-1}^o) - C_o(\hat{\eta}_{i-1}, \rho_{i-1}^o, \mathbf{D}_{i-1}^o), \forall i \in \{2, \dots, I\},$$

$$\mathbf{D}_i^o \geq \mathbf{D}_{i-1}^o, \forall i \in \{2, \dots, I\}. \quad (34)$$

This  $(\mathbf{P}_4)$  has the complexity  $O(I)$  as stated in Proposition 4.

**Proposition 4.** *The computational complexity of solving  $(\mathbf{P}_4)$  is  $O(I)$ .*

*Proof.* See Appendix F.  $\square$

### 5.2 Proposed FL Contract Iterative Algorithm

To obtain the optimal contracts  $(\hat{\mathbf{D}}^o, \hat{\rho}^o, \hat{\mathbf{D}}^l, \hat{\rho}^l)$  from problem  $(\mathbf{P}_4)$ , we implement an iterative process as shown in Algorithm 1. In this case, the participating MUs in  $\mathcal{N}$  first offer the initial contracts prior to the iterative process, i.e.,  $(\mathbf{D}_n^{o,(\tau)}, \rho_n^{o,(\tau)}, \mathbf{D}_n^{l,(\tau)}, \rho_n^{l,(\tau)})$ , where  $\mathbf{D}_n^{o,(\tau)} = [D_{1,n}^{o,(\tau)}, \dots, D_{i,n}^{o,(\tau)}, \dots, D_{I,n}^{o,(\tau)}]$ ,  $\rho_n^{o,(\tau)} = [\rho_{1,n}^{o,(\tau)}, \dots, \rho_{i,n}^{o,(\tau)}, \dots, \rho_{I,n}^{o,(\tau)}]$ , and  $\tau = 0$ . Using the initial contracts, the MAP can find the optimal encrypted dataset proportion values  $\hat{\eta}^{(\tau)}$  that maximize the objective function in  $(\mathbf{P}_1)$ . Then, the MAP can find a new contract for the MU- $n, n \in \mathcal{N}$ , iteratively using the  $\hat{\eta}^{(\tau)}$  and other MUs' current contracts  $(\mathbf{D}_{-n}^{o,(\tau)}, \rho_{-n}^{o,(\tau)}, \mathbf{D}_{-n}^{l,(\tau)}, \rho_{-n}^{l,(\tau)})$  [32], aiming at maximizing the objective function in  $(\mathbf{P}_4)$  at iteration  $\tau + 1$ .

The above process continues iteratively and then terminates when the gaps between the expected utilities of MU- $n, \forall n \in \mathcal{N}$ , at iteration  $\tau$  and  $\tau + 1$  are equal or less than the optimality tolerance  $\sigma$ . In this way, the algorithm converges and the equilibrium contract solution can be found (as described in Theorems 1 and 2). Alternatively, considering the equilibrium contract solution of other MUs  $(\hat{\mathbf{D}}_{-n}^o, \hat{\rho}_{-n}^o, \hat{\mathbf{D}}_{-n}^l, \hat{\rho}_{-n}^l)$ , the expected utility of the MU- $n$  at the equilibrium contract solution  $(\hat{\mathbf{D}}_n^o, \hat{\rho}_n^o, \hat{\mathbf{D}}_n^l, \hat{\rho}_n^l)$  will produce the highest value compared with the ones using all other contract solutions. As such, all the MU- $n, \forall n \in \mathcal{N}$ , cannot enhance their expected utilities through deviating from its equilibrium solution unilaterally. This condition is formally described in Definition 3.

**Definition 3.** *Equilibrium contract solution for (P<sub>4</sub>):* The optimal contracts  $(\hat{D}^o, \hat{\rho}^o, \hat{D}^l, \hat{\rho}^l)$  are the equilibrium solution of the (P<sub>4</sub>) if and only if the conditions

$$U_n(\hat{D}_n^o, \hat{\rho}_n^o, \hat{D}_n^l, \hat{\rho}_n^l, \hat{D}_{-n}^o, \hat{\rho}_{-n}^o, \hat{D}_{-n}^l, \hat{\rho}_{-n}^l) \geq U_n(D_n^o, \rho_n^o, D_n^l, \rho_n^l, D_{-n}^o, \rho_{-n}^o, D_{-n}^l, \rho_{-n}^l), \quad (35)$$

$\forall n \in \mathcal{N}$ , satisfy and the optimal contracts  $(\hat{D}^o, \hat{\rho}^o, \hat{D}^l, \hat{\rho}^l)$  still hold the constraints (32)-(34).

### 5.3 Convergence, Equilibrium, and Complexity Analysis of the Proposed FL Contract Algorithm

In this section, we first can show that the Algorithm 1 converges to the equilibrium contract solution by adopting the best response method [32]. In this case, we can define the best response of MU- $n$  at iteration  $\tau + 1$  using  $(D_{-n}^{o,(\tau)}, \rho_{-n}^{o,(\tau)}, D_{-n}^{l,(\tau)}, \rho_{-n}^{l,(\tau)})$  as follows:

$$\Theta_n^{(\tau+1)}(D_{-n}^{o,(\tau)}, \rho_{-n}^{o,(\tau)}, D_{-n}^{l,(\tau)}, \rho_{-n}^{l,(\tau)}) = \arg \max_{\{D_n^{o,new}, \rho_n^{o,new}, D_n^{l,new}, \rho_n^{l,new}\} \in \mathbb{Q}_n} U_n(\cdot), \quad (36)$$

where

$$U_n(\cdot) = U_n(D_n^{o,new}, \rho_n^{o,new}, D_n^{l,new}, \rho_n^{l,new}, D_{-n}^{o,(\tau)}, \rho_{-n}^{o,(\tau)}, D_{-n}^{l,(\tau)}, \rho_{-n}^{l,(\tau)}), \quad (37)$$

and  $\mathbb{Q}_n$  is the non-empty contract space [34] for MU- $n$  with  $\mathbb{Q} = \prod_{n \in \mathcal{N}} \mathbb{Q}_n$ . According to the Algorithm 1, we can apply the current contract  $(D_n^{o,new}, \rho_n^{o,new}, D_n^{l,new}, \rho_n^{l,new}) \in \Theta_n^{(\tau+1)}$  to be the new contract at  $\tau + 1$ , i.e.,  $(D_n^{o,(\tau+1)}, \rho_n^{o,(\tau+1)}, D_n^{l,(\tau+1)}, \rho_n^{l,(\tau+1)})$ , when the following condition exists

$$\left[ U_n(D_n^{o,new}, \rho_n^{o,new}, D_n^{l,new}, \rho_n^{l,new}, D_{-n}^{o,(\tau)}, \rho_{-n}^{o,(\tau)}, D_{-n}^{l,(\tau)}, \rho_{-n}^{l,(\tau)}) - U_n(D_n^{o,(\tau)}, \rho_n^{o,(\tau)}, D_n^{l,(\tau)}, \rho_n^{l,(\tau)}, D_{-n}^{o,(\tau)}, \rho_{-n}^{o,(\tau)}, D_{-n}^{l,(\tau)}, \rho_{-n}^{l,(\tau)}) \right] > \sigma. \quad (38)$$

The algorithm stops when the condition in (38) does not hold for all MU- $n$ ,  $\forall n \in \mathcal{N}$ . For that, the algorithm converges as stated in the following Theorem 1.

**THEOREM 1.** *The Algorithm 1 converges under the optimality tolerance  $\sigma$ .*

*Proof.* See Appendix G.  $\square$

Upon proving that the Algorithm 1 converges under  $\sigma$ , we need to guarantee that the Algorithm 1 also converges to the equilibrium contract solution  $(\hat{D}^o, \hat{\rho}^o, \hat{D}^l, \hat{\rho}^l)$ . In particular, we first analyze that the equilibrium solution exists by finding a fixed point in a set-valued function  $\Theta$ ,  $\Theta: \mathbb{Q} \rightarrow 2^{\mathbb{Q}}$ , i.e.,

$$\Theta = \left[ \Theta_n(D_{-n}^o, \rho_{-n}^o, D_{-n}^l, \rho_{-n}^l), \Theta_{-n}(D_n^o, \rho_n^o, D_n^l, \rho_n^l) \right]. \quad (39)$$

The finding of this fixed point is equivalent to the equilibrium solution [23], [34], where all the MUs in  $\mathcal{N}$  obtain the maximum expected utilities where  $(\hat{D}^o, \hat{\rho}^o, \hat{D}^l, \hat{\rho}^l)$  are found. As a result, the Algorithm 1 converges to the equilibrium contract solution as formally stated in Theorem 2.

### Algorithm 1 Proposed FL Contract Iterative Algorithm

---

```

1: Initialize  $\tau = 0$  and  $\sigma$ 
2: All MU- $n$  send initial contracts  $(D_n^{o,(\tau)}, \rho_n^{o,(\tau)}, D_n^{l,(\tau)}, \rho_n^{l,(\tau)})$ ,  $\forall n \in \mathcal{N}$ , to the MAP
3: repeat
4: Find  $\hat{\eta}^{(\tau)}$  values which maximize (P1) using  $(D_{-n}^{o,(\tau)}, \rho_{-n}^{o,(\tau)}, D_{-n}^{l,(\tau)}, \rho_{-n}^{l,(\tau)})$ 
5: for  $\forall n \in \mathcal{N}$  do
6: Produce a new contract  $(D_n^{o,new}, \rho_n^{o,new}, D_n^{l,new}, \rho_n^{l,new})$ , which maximizes (P4) given  $\hat{\eta}^{(\tau)}$  and  $(D_{-n}^{o,(\tau)}, \rho_{-n}^{o,(\tau)}, D_{-n}^{l,(\tau)}, \rho_{-n}^{l,(\tau)})$ 
7: if the condition in (38) is satisfied then
8: Set  $(D_n^{o,(\tau+1)}, \rho_n^{o,(\tau+1)}, D_n^{l,(\tau+1)}, \rho_n^{l,(\tau+1)}) = (D_n^{o,new}, \rho_n^{o,new}, D_n^{l,new}, \rho_n^{l,new})$ 
9: else
10: Set  $(D_n^{o,(\tau+1)}, \rho_n^{o,(\tau+1)}, D_n^{l,(\tau+1)}, \rho_n^{l,(\tau+1)}) = (D_n^{o,(\tau)}, \rho_n^{o,(\tau)}, D_n^{l,(\tau)}, \rho_n^{l,(\tau)})$ 
11: end if
12: end for
13:  $\tau = \tau + 1$ 
14: until  $U_n(D_n^{o,(\tau)}, \rho_n^{o,(\tau)}, D_n^{l,(\tau)}, \rho_n^{l,(\tau)}, D_{-n}^{o,(\tau)}, \rho_{-n}^{o,(\tau)}, D_{-n}^{l,(\tau)}, \rho_{-n}^{l,(\tau)})$ ,  $\forall n \in \mathcal{N}$ , remain unchanged
15: Obtain the optimal contracts  $(\hat{D}^o, \hat{\rho}^o, \hat{D}^l, \hat{\rho}^l)$ 

```

---

**THEOREM 2.** *The best response iterative process in Algorithm 1 converges to its equilibrium contract  $(\hat{D}^o, \hat{\rho}^o, \hat{D}^l, \hat{\rho}^l)$  if and only if  $(\hat{D}^o, \hat{\rho}^o, \hat{D}^l, \hat{\rho}^l)$  is a fixed point in  $\Theta$ .*

*Proof.* See Appendix H.  $\square$

Next, the complexity of Algorithm 1 can be observed by considering  $N$  MUs and the iterative process to find the optimal contract for each MU in  $\mathcal{N}$ . In this case, the complexity can be bounded above by polynomial complexity  $\log(N) + e^\epsilon + O(1/N)$ , where  $\epsilon$  is the Euler constant, as formally derived in Theorem 3.

**THEOREM 3.** *Algorithm 1 has polynomial complexity  $\log(N) + e^\epsilon + O(1/N)$ .*

*Proof.* See Appendix I.  $\square$

## 6 FEDERATED LEARNING WITH STRAGGLING MITIGATION AND PRIVACY-AWARENESS IMPLEMENTATION

Using the vectors of optimal encrypted dataset size  $\hat{D}^o$  and optimal local dataset size  $\hat{D}^l$  of the selected MUs in  $\mathcal{N}$  according to the optimal contracts  $(\hat{D}^o, \hat{\rho}^o, \hat{D}^l, \hat{\rho}^l)$ , we then can execute the entire FL process at the MAP and MUs. Specifically, we apply a deep learning approach utilizing CNN along with bi-directional LSTM (BLSTM) for a classification prediction model. These two approaches are proved to perform well for time-series problems [3]. Let  $S_i^o = (\mathbf{F}_i^o, \mathbf{L}_i^o)$  with optimal size  $\sum_{n \in \mathcal{N}} \hat{D}_{i,n}^o$  denote the encrypted dataset of all MUs in  $\mathcal{N}$  at the MAP with type  $\pi_i$ , where  $\mathbf{F}_i^o$  and  $\mathbf{L}_i^o$  are the training feature and training label matrices of encrypted dataset at the MAP with type  $\pi_i$ , respectively. Additionally, we denote  $S_n^l = (\mathbf{F}_n^l, \mathbf{L}_n^l)$  with

optimal size  $\hat{D}_n^l$  to be the local dataset at each MU- $n$ , where  $\mathbf{F}_n^l$  and  $\mathbf{L}_n^l$  are the training feature and label matrices of local dataset at each MU- $n$ , respectively.

For the CNN, we have  $\mathbf{F}_i^{o,a}$  and  $\mathbf{F}_n^{l,a}$  such that  $\mathbf{F}_i^{o,1} = \mathbf{F}_i^o$  and  $\mathbf{F}_n^{l,1} = \mathbf{F}_n^l$ , respectively, where  $a$  is the training layer,  $a \in [1, 2, \dots, a_{max}]$ . For each convolutional layer, the training outputs  $\mathbf{F}_i^{o,(a+1)}$  and  $\mathbf{F}_n^{l,(a+1)}$  can be respectively produced as expressed by [3]

$$\begin{aligned} \mathbf{F}_i^{o,(a+1)} &= \alpha_i^a \left( \sum \mathbf{F}_i^{o,a} * \mathbf{Q} \right), \text{ and} \\ \mathbf{F}_n^{l,(a+1)} &= \alpha_n^a \left( \sum \mathbf{F}_n^{l,a} * \mathbf{Q} \right), \end{aligned} \quad (40)$$

where  $\mathbf{Q}$  is the squared kernel matrix with a certain number of filters and  $\alpha_i^a(\cdot), \alpha_n^a(\cdot)$  are the activation functions [3]. These outputs are then fed into a max pooling layer to obtain  $\mathbf{F}_i^{o,(a+2)} = \omega(\mathbf{F}_i^{o,(a+1)})$  and  $\mathbf{F}_n^{l,(a+2)} = \omega(\mathbf{F}_n^{l,(a+1)})$ , where  $\omega$  is the pooling function representing the maximum element values of  $\mathbf{F}_i^{o,(a+1)}$  and  $\mathbf{F}_n^{l,(a+1)}$  based on the pre-defined pooling size. The above process between convolutional and max pooling layers is repeated for certain times, and then the training outputs from the last max pooling layer can be considered as the inputs of the BLSTM layer, i.e.,  $\mathbf{F}_i^{o,a^*}$  and  $\mathbf{F}_n^{l,a^*}$ . Using  $\mathbf{F}_i^{o,a^*}$ ,  $\mathbf{F}_n^{l,a^*}$ , and the BLSTM method in [36], we can generate the output matrices  $\mathbf{F}_i^{o,(a^*+1)}$  and  $\mathbf{F}_n^{l,(a^*+1)}$ .

Upon passing  $\mathbf{F}_i^{o,(a^*+1)}$  and  $\mathbf{F}_n^{l,(a^*+1)}$  to a dropout layer, and obtaining  $\mathbf{F}_i^{o,(a_{drop})}$  and  $\mathbf{F}_n^{l,(a_{drop})}$ , both matrices can enter the fully connected layers to obtain the final output matrix of layer  $a_{max}$ , i.e.,  $\mathbf{L}_i^{o,a_{max}}$  at the MAP and  $\mathbf{L}_n^{l,(a_{max})}$  at each MU- $n$ , which can be expressed by [3]

$$\begin{aligned} \mathbf{L}_i^{o,a_{max}} &= \alpha_i^{a_{max}} \left( \mathbf{F}_i^{o,(a_{max}-1)} \mathbf{\Upsilon}_i^{a_{max}} \right), \text{ and} \\ \mathbf{L}_n^{l,a_{max}} &= \alpha_n^{a_{max}} \left( \mathbf{F}_n^{l,(a_{max}-1)} \mathbf{\Upsilon}_n^{a_{max}} \right), \end{aligned} \quad (41)$$

where  $\alpha_i^{a_{max}}(\cdot)$  and  $\alpha_n^{a_{max}}(\cdot)$  are the *softmax* activation functions used to provide a probability distribution for the classification results [3]. Moreover,  $\mathbf{F}_i^{o,(a_{max}-1)}$  and  $\mathbf{F}_n^{l,(a_{max}-1)}$  are the output matrices from the previous fully connected layer  $a_{max} - 1$ , while  $\mathbf{\Upsilon}_i^{a_{max}}$  and  $\mathbf{\Upsilon}_n^{a_{max}}$  are the weight matrices at the final layer  $a_{max}$ .

Based on  $\mathbf{L}_i^o, \mathbf{L}_n^l, \mathbf{L}_i^{o,a_{max}}$ , and  $\mathbf{L}_n^{l,a_{max}}$ , the loss functions of the MAP and each MU- $n, n \in \mathcal{N}$ , at learning round  $\theta$  can be obtained through using a squared Frobenius norm, i.e.,

$$\begin{aligned} \varphi_i(\tilde{\mathbf{Y}}^{(\theta)}) &= \frac{1}{2 \sum_{n \in \mathcal{N}} \hat{D}_{i,n}^o} \left\| \mathbf{L}_i^{o,a_{max}} - \mathbf{L}_i^o \right\|_F^2 \\ &= \frac{1}{2 \sum_{n \in \mathcal{N}} \hat{D}_{i,n}^o} \sum_{j=1}^{\sum_{n \in \mathcal{N}} \hat{D}_{i,n}^o} (\hat{\ell}_{i,j} - \ell_{i,j})^2, \end{aligned} \quad (42)$$

and

$$\begin{aligned} \varphi_n(\mathbf{Y}^{(\theta)}) &= \frac{1}{2 \hat{D}_n^l} \left\| \mathbf{L}_n^{l,a_{max}} - \mathbf{L}_n^l \right\|_F^2 \\ &= \frac{1}{2 \hat{D}_n^l} \sum_{j=1}^{\hat{D}_n^l} (\hat{\ell}_{n,j} - \ell_{n,j})^2, \end{aligned} \quad (43)$$

respectively, where  $\tilde{\mathbf{Y}}^{(\theta)}$  and  $\mathbf{Y}^{(\theta)}$  are the encrypted and decrypted global model matrices (i.e.,  $\mathbf{Y}^{(\theta)} = \text{Dec}(g_{sk}, \tilde{\mathbf{Y}}^{(\theta)})$ ), respectively, containing weights for the BLSTM and fully connected layers at round  $\theta$ ,  $\ell_{i,j}$  and  $\ell_{n,j}$  are the sample points of ground-truth label data  $\mathbf{L}_i^o$  and  $\mathbf{L}_n^l$ , respectively,

while  $\hat{\ell}_{i,j}$  and  $\hat{\ell}_{n,j}$  are the elements of predicted label data  $\mathbf{L}_i^{o,a_{max}}$ , and  $\mathbf{L}_n^{l,a_{max}}$ , respectively.

From (42) and (43), the encrypted gradients at the MAP and local gradient at each MU- $n$  can be respectively expressed by

$$\nabla \tilde{\mathbf{Y}}_i^{(\theta)} = \frac{\partial \varphi_i(\tilde{\mathbf{Y}}^{(\theta)})}{\partial \tilde{\mathbf{Y}}^{(\theta)}}, \text{ and } \nabla \mathbf{Y}_n^{(\theta)} = \frac{\partial \varphi_n(\mathbf{Y}^{(\theta)})}{\partial \mathbf{Y}^{(\theta)}}. \quad (44)$$

Suppose that the local loss gradient at MU- $n$  when we use baseline FL without encrypted training process can be simplified as

$$\begin{aligned} \nabla \hat{\mathbf{Y}}_n^{(\theta)} &= \frac{\partial \varphi_n(\mathbf{Y}^{(\theta)})}{\partial \mathbf{Y}^{(\theta)}} = \frac{\partial \left[ \frac{1}{2 \hat{D}_n^l} \left\| \hat{\mathbf{L}}_n - \mathbf{L}_n \right\|_F^2 \right]}{\partial \mathbf{Y}^{(\theta)}} \\ &= \frac{1}{\hat{D}_n^l} \mathbf{F}_n^T (\mathbf{F}_n \mathbf{Y}^{(\theta)} - \mathbf{L}_n), \end{aligned} \quad (45)$$

where  $\mathbf{F}_n, \mathbf{L}_n$ , and  $\hat{\mathbf{L}}_n$  are the training feature data, ground truth label data, and predicted label data at MU- $n$ , respectively. When the encrypted training is used, the encrypted gradient of encrypted dataset at the MAP with type  $\pi_i$  in (44) can be derived by

$$\nabla \tilde{\mathbf{Y}}_i^{(\theta)} = \frac{1}{\sum_{n \in \mathcal{N}} \hat{D}_{i,n}^o} (\mathbf{F}_i^o)^T (\mathbf{F}_i^o \tilde{\mathbf{Y}}^{(\theta)} - \mathbf{L}_i^o). \quad (46)$$

Likewise, the local gradient of the local dataset at each MU- $n$  considering the encrypted training can be written as

$$\nabla \mathbf{Y}_n^{(\theta)} = \frac{1}{\hat{D}_n^l} (\mathbf{F}_n^l)^T (\mathbf{F}_n^l \mathbf{Y}^{(\theta)} - \mathbf{L}_n^l). \quad (47)$$

Using (44), each MU- $n$  can encrypt  $\nabla \mathbf{Y}_n^{(\theta)}$  into  $\nabla \tilde{\mathbf{Y}}_n^{(\theta)} = \text{Enc}(g_{pk}, \nabla \mathbf{Y}_n^{(\theta)})$  and share its  $\nabla \tilde{\mathbf{Y}}_n^{(\theta)}$  to the MAP for the encrypted model aggregation such that the MAP has

$$\nabla \tilde{\mathbf{Y}}_{\mathcal{N}}^{(\theta)} = \sum_{n \in \mathcal{N}} \hat{D}_n^l \nabla \tilde{\mathbf{Y}}_n^{(\theta)}. \quad (48)$$

As the MAP also has  $\nabla \tilde{\mathbf{Y}}_i^{(\theta)}$  from the encrypted training, the encrypted global gradient can be updated as follows:

$$\nabla \tilde{\mathbf{Y}}^{(\theta)} = \frac{1}{\sum_{n \in \mathcal{N}} \hat{D}_n^l + \sum_{n \in \mathcal{N}} \hat{D}_{i,n}^o} (\nabla \tilde{\mathbf{Y}}_{\mathcal{N}}^{(\theta)} + \nabla \tilde{\mathbf{Y}}_i^{(\theta)}). \quad (49)$$

Hence, the MAP can renew the encrypted global model for the next learning round as described by

$$\tilde{\mathbf{Y}}^{(\theta+1)} = \tilde{\mathbf{Y}}^{(\theta)} - \lambda^{(\theta+1)} \Omega(\nabla \tilde{\mathbf{Y}}^{(\theta)}), \quad (50)$$

where  $\lambda^{(\theta+1)}$  is the encrypted learning step function of the adaptive learning rate *Adam* optimizer [33]. Furthermore,  $\Omega(\nabla \tilde{\mathbf{Y}}^{(\theta)})$  indicates the local update rules as a function of encrypted gradient  $\nabla \tilde{\mathbf{Y}}^{(\theta)}$  described in [33]. To this end, we can also determine the global loss function at learning round  $\theta + 1$  as

$$\varphi(\tilde{\mathbf{Y}}^{(\theta+1)}) = \frac{1}{N+1} \left( \varphi_i(\tilde{\mathbf{Y}}^{(\theta)}) + \sum_{n \in \mathcal{N}(\theta)} \varphi_n(\tilde{\mathbf{Y}}^{(\theta)}) \right). \quad (51)$$

The above global process terminates when the global loss converges or the learning rounds achieve a given threshold  $\theta_{thr}$ , and thus the final encrypted global model  $\tilde{\mathbf{Y}}^*$  and the final global loss  $\varphi^*(\tilde{\mathbf{Y}}^*)$  are produced. The whole process of proposed FL-based framework is summarized in Algorithm 2 and the convergence analysis is discussed in Appendix J.

---

**Algorithm 2** Proposed FL-Based Framework Algorithm
 

---

- 1: Set  $\theta_{th}$ ,  $\tilde{\Upsilon}^{(0)}$ , and  $\theta = 0$
  - 2: The MAP determines MUs in  $\mathcal{N} \subset \mathcal{M}$  using (3)
  - 3: Perform Algorithm 1 for all MUs in  $\mathcal{N}$  to obtain optimal contracts  $(\hat{D}_i^o, \hat{\rho}_i^o, \hat{D}_i^l, \hat{\rho}_i^l)$
  - 4: **for**  $\forall n \in \mathcal{N}$  **do**
  - 5:   Implement the data encryption with the optimal size  $\hat{D}_{i,n}^o$  considering the MAP with type  $\pi_i$
  - 6:   Send the encrypted dataset to the MAP
  - 7:   Set  $\mathbf{F}_n^l$  and  $\mathbf{L}_n^l$  based on the optimal size  $\hat{D}_n^l$
  - 8: **end for**
  - 9: The MAP combines the received encrypted datasets into  $S_i^o$
  - 10: The MAP sets  $\mathbf{F}_i^o$  and  $\mathbf{L}_i^o$  from  $S_i^o$
  - 11: **while**  $\theta \leq \theta_{th}$  **and**  $\varphi(\tilde{\Upsilon}^{(\theta)})$  does not converge **do**
  - 12:   **for**  $\forall n \in \mathcal{N}$  **do**
  - 13:     Compute  $\mathbf{L}_n^{l,amax}$  using  $\mathbf{F}_n^l$  and  $\Upsilon^{(\theta)}$
  - 14:     Decrypt  $\nabla \tilde{\Upsilon}_n^{(\theta)}$  into  $\nabla \Upsilon_n^{(\theta)}$
  - 15:     Find  $\varphi_n(\Upsilon_n^{(\theta)})$  and  $\nabla \Upsilon_n^{(\theta)}$
  - 16:     Encrypt  $\nabla \Upsilon_n^{(\theta)}$  into  $\nabla \tilde{\Upsilon}_n^{(\theta)}$  and send it to the MAP
  - 17:   **end for**
  - 18: The MAP calculates  $\mathbf{L}_i^{o,amax}$  using  $\mathbf{F}_i^o$  and  $\Upsilon^{(\theta)}$
  - 19: The MAP obtains  $\varphi_i(\tilde{\Upsilon}^{(\theta)})$  and  $\nabla \tilde{\Upsilon}_i^{(\theta)}$
  - 20: Aggregate  $\nabla \tilde{\Upsilon}_n^{(\theta)}, \forall n \in \mathcal{N}$ , using (48) and the encrypted global gradient  $\nabla \tilde{\Upsilon}^{(\theta)}$  using (49)
  - 21: Update the encrypted global model  $\tilde{\Upsilon}^{(\theta+1)}$  using (50)
  - 22: Obtain the global loss  $\varphi(\tilde{\Upsilon}^{(\theta+1)})$  using (51)
  - 23:  $\theta = \theta + 1$
  - 24: **end while**
  - 25: Finalize the encrypted global model  $\tilde{\Upsilon}^*$  and global loss  $\varphi^*(\tilde{\Upsilon}^*)$
- 

## 7 PERFORMANCE EVALUATION

### 7.1 Dataset Pre-processing

We investigate the performance of the proposed FL-based framework utilizing an actual HAR dataset in 2019 [24]. This dataset contains 15M raw sensor samples divided into accelerometer and gyroscope data of smartphones and smartwatches from MUs. To evaluate the efficiency of the FL-based framework, we first combine all  $x, y, z$  direction features from data of all smartphones and smartwatches into a single dataset to obtain 12 features with 1M samples (by removing samples with incomplete features). Then, we extract the first six activity labels from total 18 labels including walking, jogging, stairs, sitting, standing, and typing. Using the sampling rate 20 Hz, we convert the raw data into 10-second time-series data to obtain  $\sim 5K$  time-series samples with 200 time periods for each sample.

### 7.2 Experiment Setup

For the contract optimization, we use one agent, i.e., the MAP, and  $N$  principals (referring to  $N$  selected MUs). Specifically, we set 10 types of the MAP with uniform distribution of the types. Considering the number of HAR samples prior to time-series conversion, we set  $D_{max}^o = 5 \times 10^5$  samples. We set  $v_o$  and  $v_l$  at 0.125 and 3, respectively, as well as  $\alpha_o$  and  $\alpha_l$  at 0.001 and 0.005, respectively. We also denote  $\beta = 1$  and  $\gamma = 0.0001$ . We use  $\zeta_n = 0.5 \times 10^{-26}, \forall n \in \mathcal{N}$  [21], and  $f_n = 2\text{GHz}$  for practical CPU frequency of today's smartphone. Moreover, we use an average transmission rate 293 Mbps of 802.11ac WLAN with

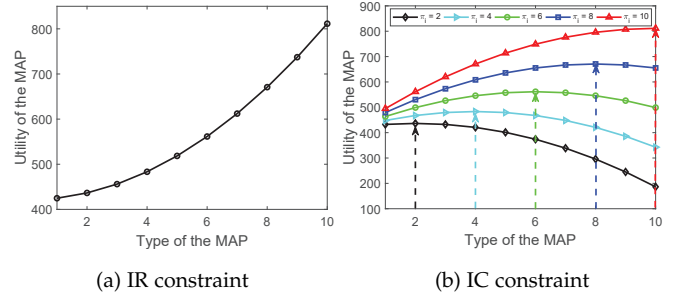


Fig. 3: The contract feasibility of the MAP.

bandwidth 80 MHz [37]. Given that each sample occupies 187 bytes, we use  $c_n = 44880$  cycles/sample considering 30 cycles/bit. Furthermore, we assume that the cycles per bit at the MAP is much lower than that of the participating MUs due to its much faster computing capability [17], [18]. We then compare our proposed contract solution with other baseline and information-symmetry methods. For the baseline method, each MU can offer the proportional size of encrypted dataset to the MAP without using contract policy [27]. For the information-symmetry method, each MU completely knows the MAP's true type and other MUs' contracts to obtain the optimal contract policy. In this way, we consider this method as the upper bound solution.

To implement the proposed FL process, we utilize *TensorFlow CPU 2.2* containing *TensorFlow Federated* in a shared cluster. Specifically, we consider 100 active MUs which are then reduced (using the MU selection in Section 3) to 10 best MUs for the FL process to compare with the conventional FL algorithm (conv-FL), i.e., the MAP waits for a specified number of participating MUs at each round to upload their encrypted local models without using encrypted data training, and proposed framework. For the proposed FL, each MU obtains the size of its optimal encrypted dataset based on the MAP's type. In particular, 10%, 30%, and 50% encrypted datasets of the whole dataset (from all MUs in  $\mathcal{N}$ ), i.e., Enc 10%, Enc 30%, and Enc 50%, represent the MAP with type 2, 6, and 10, respectively. Next, we split the pre-processed dataset into 80% training set and 20% testing set. To represent a practical scenario where each mobile device may capture different activities from its MU, we consider a non-i.i.d dataset distribution scenario. As such, we first sort the training set according to the training labels in ascending order. Then, a certain number of samples from the sorted training set is equally distributed to the selected MUs sequentially. For the CNN, we use three-stacked 1D-convolutional layers with 64 filters and 3, 5, and 11 kernel sizes for each corresponding layer. We also utilize a BLSTM and two hidden DNN layers with 64 neurons for each layer followed by a dropout layer with fraction rate 0.25. For the optimizer, we use the Adam optimizer with initial step size 0.01 [27].

### 7.3 Proposed FL Contract Performance

#### 7.3.1 IR and IC constraint feasibility of the MAP

We first demonstrate in Fig. 3(a) that the MAP always obtains a non-negative utility to ensure the IR constraint validity as mathematically formulated in (12). This utility

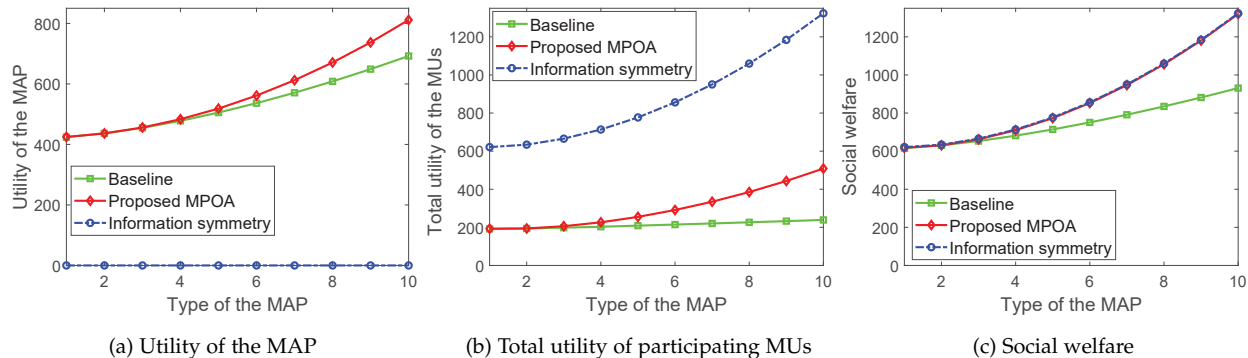


Fig. 4: The learning contract performance vs type of the MAP.

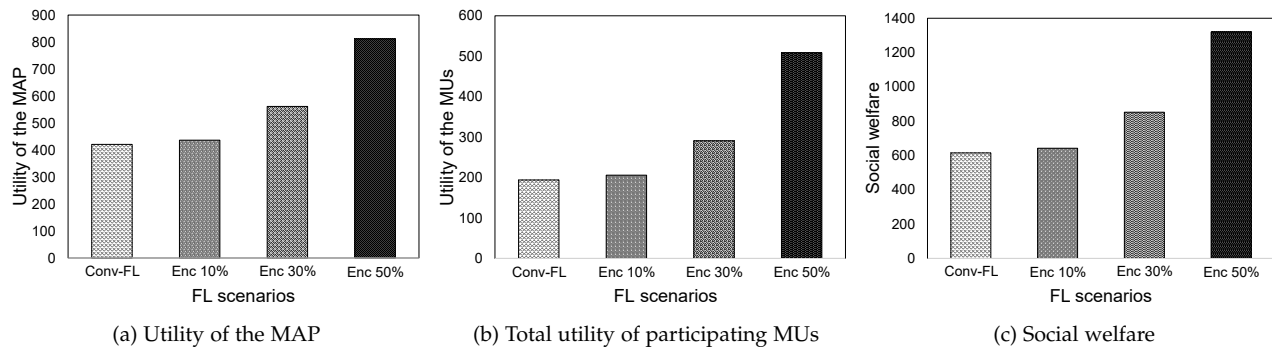


Fig. 5: The learning contract performance for various FL scenarios.

follows a monotonic increasing function with respect to the MAP's type because the MAP is willing to train more encrypted datasets due to higher benefit for the MAP in terms of the global model accuracy. Moreover, the MAP can also produce the highest utility when it applies the right contract for its true type, i.e., the IC constraints in (12) are satisfied, as shown in Fig. 3(b). For example, the MAP with types 2, 4, 6, 8, and 10 will generate the highest utility when corresponding contracts for the respective types are applied. Since both IR and IC constraints are satisfied, we can obtain the feasible contracts for all participating MUs to maximize their utilities in the proposed FL process.

### 7.3.2 The utility of the MAP/participating MUs and the social welfare

We then compare the MAP's utility of the proposed framework with those of the baseline and information-symmetry methods. As can be observed in Fig. 4(a), the proposed framework can achieve the highest utility for all types of the MAP especially when the MAP's type gets higher. Particularly, the proposed framework can obtain the utility up to 17% higher than that of the baseline method. The reason is that the baseline method cannot optimize the encrypted dataset proportion from the MAP since each participating MU only can send the proportional amount of encrypted dataset under the MAP's current computing resources (to train the whole encrypted datasets from all participating MUs). Additionally, the information-symmetry method suffers from zero utility for all types because all participating MUs can collaborate together to obtain maximum utility

through completely perceiving the current true type, i.e., the available computing resources, of the MAP. Consequently, as shown in Fig. 4(b), the information-symmetry method can maximize the total utility of participating MUs. In this case, the proposed framework can still outperform the baseline method up to 113% in terms of the total utility of participating MUs. An interesting point can be observed in Fig. 4(c). Particularly, although the information-symmetry can obtain the total utility of the MUs up to 2.6 times higher than that of the proposed framework, our proposed framework can achieve the social welfare within 0.57% of the information-symmetry method (which acts as the upper bound solution). Moreover, the proposed framework can obtain the social welfare up to 42% higher than that of the baseline method. To this end, we can demonstrate that our proposed framework is applicable to implement in the proposed FL process through stabilizing the utility performance of the MAP and participating MUs efficiently. Then, we observe the contract performance of our FL-based framework compared with that of the conventional FL method. As shown in Fig. 5, the use of encrypted training in our FL-based framework, i.e., 10%, 30%, and 50% encrypted datasets, can improve the utility of the MAP, the total utility of the participating MUs, and social welfare of the network up to 93%, 162%, and 114%, respectively, compared with those of the conventional FL method. The reason is that the MUs can reduce remarkable training tasks through offloading their encrypted datasets to the MAP for the encrypted training, which then leads to better learning quality, especially when some MUs suffer from straggling computing resources. These results reflect

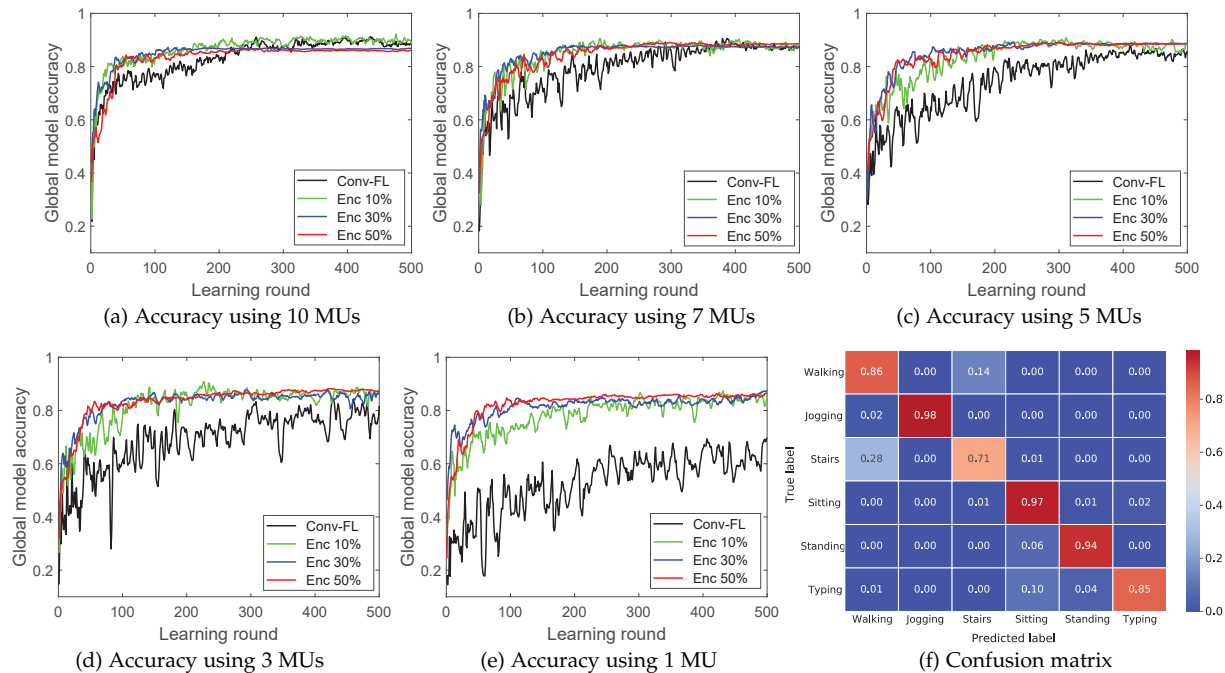


Fig. 6: The performance of different FL scenarios using various participating MUs.

that a higher proportion of encrypted dataset at the MAP may result in higher utilities and social welfare for both the participating MUs and the MAP at the expense of a higher privacy protection cost (which is further discussed in Section 7.5).

## 7.4 Proposed FL Accuracy Performance

### 7.4.1 Accuracy with various number of participating MUs

In this section, we first analyze performance of the proposed FL with straggling mitigation and privacy-awareness when the MAP updates the encrypted global model upon successfully receiving the encrypted local models from a fixed number of participating MUs at each learning round. As shown in Fig. 6, the proposed FL generally can maintain its accuracy performance for all participating MU scenarios regardless the encrypted dataset proportion of the participating MUs. Specifically, when all local models from 10 participating MUs are applied to update the encrypted global model at each round in Fig. 6(a), the accuracy performance for the conv-FL and proposed FL approaches can achieve the best results due to the equal size of training samples at each participating MU (including at the MAP for the proposed FL with 10% encrypted dataset). Although the conv-FL can eventually obtain the same final accuracy level at 88% as the proposed FL with 10% encrypted dataset proportion, the proposed FL provides more stable accuracy performance at each learning round and achieves the convergence rate 28.5% faster than that of the conv-FL method. Note that the proposed FL with 30% and 50% encrypted datasets converge to 2% lower accuracy level (i.e., at level 86%) compared with that of the conv-FL due to high imbalanced size of non-i.i.d training samples at the participating MUs and the MAP, i.e., the number of training samples at the MAP is much larger than that at each MU. As such, the

aggregation of encrypted local models from participating MUs and the MAP will slightly degrade final accuracy of the converged global model [6].

When less encrypted local models, i.e., 7 participating MUs to 1 participating MU, are used to update the encrypted global model (representing worse straggling issues), the proposed FL for all encrypted dataset proportions can maintain their accuracy performance at the same final accuracy level, thanks to the additional training at the MAP. Meanwhile, the performance of conv-FL suffers from the accuracy degradation due to insufficient number of non-i.i.d samples to boost the accuracy level. To this end, the proposed FL for all encrypted dataset proportions can obtain the accuracy gap with the conv-FL up to 1.72 times, 1.72 times, 2.9 times, and 4.6 times using 7 MUs, 5 MUs, 3 MUs, and 1 MU scenarios, respectively. Additionally, in Fig. 6(e), we can observe that the use of higher encrypted dataset proportion sent to the MAP, i.e., 50% encrypted dataset, can bring more stable accuracy performance at each learning round and reach the convergence 70% (70 rounds) and 127% (140 rounds) faster than those of 10% and 30% encrypted dataset scenarios, respectively. This means that when more devices have straggling problems (i.e., only very few participating MUs are used as the FL learners) at each learning round, the use of larger encrypted training at the MAP can greatly compensate the insufficient training samples at the MUs. In this case, within 500 learning rounds, the conv-FL even cannot reach the convergence due to high fluctuation of the accuracy performance. Then, the confusion matrix to show how good the prediction model towards each activity label for the proposed FL with 10% encrypted dataset and 10 participating MUs can be seen in Fig. 6(f). Based on the above performances, it can be implied that a higher type of the MAP (which corresponds to a larger encrypted dataset proportion at the MAP) generally can speed up

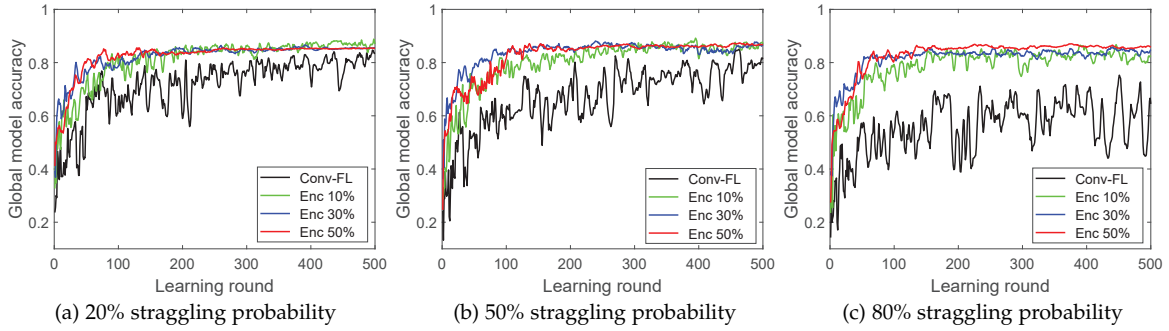


Fig. 7: The performance of various FL scenarios using different straggling probabilities.

the convergence rate, improve global model accuracy, and provide more stable performance especially when a small fixed number of MUs participate in the FL process.

#### 7.4.2 Accuracy with various straggling probabilities

To further show the superiority of proposed FL, we consider various straggling probabilities, i.e., the probabilities that participating MUs experience the straggling problems, e.g., due to low computing resources and/or bad communication resources, such that they cannot send the encrypted local models to the MAP at a certain learning round. Using fixed 5 participating MUs at each learning round, we consider 20%, 50%, and 80% straggling probabilities. In this case, the higher the straggling probability is, the lower the number of participating MUs that can send the encrypted local models to the MAP at each round. As shown in Fig. 7, the proposed FL can preserve its accuracy approximately at level 86% for all straggling probability scenarios. In contrast, the conv-FL cannot maintain its accuracy when the straggling probability gets worse. As such, in terms of the converged accuracy performance, the proposed FL can outperform the conv-FL up to 1.84 times, 3.35 times, and 3.44 times for 20%, 50%, and 80% straggling probability scenarios, respectively. The interesting point is that the proposed FL with 50% encrypted dataset can fully keep its accuracy at the same level with stable values for most of learning rounds, while the ones with 10% and 30% encrypted datasets degrade their accuracy performances when the straggling probability gets higher. The reason is that the higher encrypted training at the MAP can compensate the straggling problems at the participating MUs more due to its learning process improvement with respect to the larger size of encrypted dataset.

#### 7.5 Trade-off Between Training Time and Privacy Cost

In addition to the accuracy and convergence rate performance, the proposed FL outperforms the conv-FL in terms of the training time to complete 500 learning rounds. Specifically, as observed in Fig. 8(a), the conv-FL requires the longest training time, i.e., 1702 seconds, to reach 500 learning rounds. Meanwhile, for the proposed FL, the larger the encrypted dataset proportion is trained at the MAP (i.e., the higher the type of the MAP is), the faster the training time to complete 500 learning rounds. This is due to the smaller number of local datasets to be trained at the participating MUs such that the encrypted local models can be aggregated faster at the MAP to update the encrypted

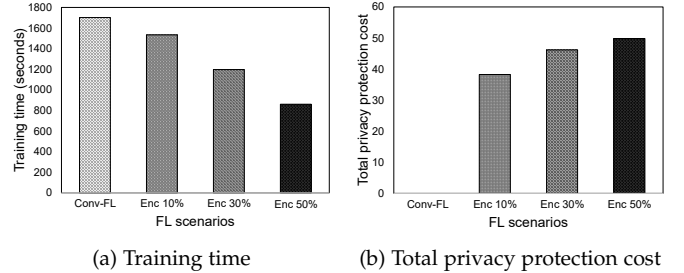


Fig. 8: The training time to achieve 500 learning rounds and the total privacy protection cost of encrypted dataset sharing for various FL scenarios.

global model. In this case, the proposed FL with 10%, 30%, 50% encrypted datasets can reduce the training time up to 10%, 30%, and 49%, respectively, compared with that of the conv-FL. The training time result contradicts with the total privacy protection cost of all participating MUs for sharing encrypted datasets to the MAP. As observed in Fig. 8(b), the performance of total privacy cost follows a logarithmic function (as defined in (2)) with respect to the proportion of total uploaded encrypted dataset. As such, although more encrypted dataset offloading comes with higher privacy protection cost, the privacy of offloaded encrypted datasets can be still protected efficiently. This interesting trade-off provides an insightful information to selectively determine the best FL scenarios which can improve the learning quality, i.e., straggling problem reduction with higher global model accuracy and faster training time/convergence rate, while minimizing the privacy protection cost of encrypted dataset sharing in the FL process.

## 8 CONCLUSION

In this paper, we have proposed the novel FL-based framework with straggling mitigation and privacy-awareness for the mobile application service to maximize the utilities for the MAP and participating MUs while improving the global model accuracy with additional encrypted training. Particularly, we have designed the MU selection scheme to determine the best MUs based on the abstract of dataset and device information. Using the selected MUs, we have developed the MPOA-based FL contract problem under the MAP's common constraints (i.e., the IR and IC constraints), the competition among participating MUs, and unknown

computing resources from the MAP. To obtain the optimal contracts containing the MUs' optimal sizes of encrypted and local datasets for the FL process, we have transformed the problem and implemented the iterative FL contract algorithm that can achieve the equilibrium solution for all the participating MUs. The experiments have demonstrated that our proposed framework can significantly improve the training time, prediction accuracy, utilities, and social welfare of the MAP and participating MUs (while considering the privacy protection cost of encrypted datasets) compared with those of other baseline methods.

## REFERENCES

- [1] I. H. Sarker, *et al.*, "Mobile data science and intelligent apps: concepts, ai-based modeling and research directions," *Mobile Networks and Applications*, vol. 26, no. 1, pp.285-303, Feb. 2021.
- [2] Big Data and Business Analytics Market 2027, Sept. 2020. Available Online: <https://www.alliedmarketresearch.com/big-data-and-business-analytics-market>. Last Accessed on: Mar. 2021.
- [3] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: a survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224-2287, Thirdquarter 2019.
- [4] Y. Sun, *et al.*, "Application of machine learning in wireless networks: key techniques and open issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072-3108, Fourthquarter 2019.
- [5] Q. Yang, *et al.*, "Federated learning: synthesis lectures on artificial intelligence and machine learning," vol. 13, no. 3, pp. 1-207, Dec. 2019.
- [6] 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.
- [7] X. Wang, *et al.*, "In-Edge AI: intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Network*, vol. 33, no. 5, pp. 156-165, Sept.-Oct. 2019.
- [8] N. Yoshida, *et al.*, "Hybrid-FL: cooperative learning mechanism using non-iid data in wireless networks," arXiv:1905.07210v3 [cs.LG], Mar. 2020.
- [9] 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.
- [10] 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.
- [11] N. J. Hernandez Marcano, *et al.*, "On fully homomorphic encryption for privacy-preserving deep learning," in *IEEE Globecom Workshops*, Dec. 2019, pp. 1-6.
- [12] 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.
- [13] L. T. Phong, *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1333-1345, May 2018.
- [14] H. Fang and Q. Qian. "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, pp. 1-20, Apr. 2021.
- [15] C. Gentry, "A fully homomorphic encryption scheme," Stanford University, 2009.
- [16] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *IACR Cryptol. ePrint Arch.*, Mar. 2012, pp. 1-19.
- [17] 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.
- [18] 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.
- [19] D. Yang, *et al.*, "Incentive mechanisms for crowdsensing: crowdsourcing with smartphones," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1732-1744, Jun. 2016.
- [20] S. Feng, *et al.*, "Joint service pricing and cooperative relay communication for federated learning," in *IEEE iThings, GreenCom, CPSCoM, and SmartData*, Jul. 2019, pp. 815-820.
- [21] N. Kim, *et al.*, "Incentive-based coded distributed computing management for latency reduction in IoT services - a game theoretic approach," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8259-8278, May 2021.
- [22] N. Ding, *et al.*, "Incentive mechanism design for distributed coded machine learning," arXiv:2012.08715 [cs.GT], Dec. 2020.
- [23] B. D. Bernheim and M. D. Whinston, "Common agency," *Econometrica*, vol. 54, no. 4, pp. 923-942, Jul. 1986.
- [24] WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set, Oct. 2019. Available Online: <https://archive.ics.uci.edu/ml/machine-learning-databases/00507/>. Last Accessed on: Mar. 2021.
- [25] K. Nikitin, *et al.*, "Reducing metadata leakage from encrypted files and communication with PURBs," *Proceedings on Privacy Enhancing Technologies*, vol. 4, pp. 6-33, Jun. 2019.
- [26] L. Xu, *et al.*, "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.
- [27] Y. M. Saputra, *et al.*, "Federated learning meets contract theory: economic-efficiency framework for electric vehicle networks," *IEEE Trans. Mobile Comput.*, early access, Dec. 2020.
- [28] P. Bolton and M. Dewatripont, *Contract Theory*. Cambridge, MA, USA: MIT Press, 2005.
- [29] Z. Song, *et al.*, "QoI-aware multitask-oriented dynamic participant selection with budget constraints," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4618-4632, Nov. 2014.
- [30] W. Wang, *et al.*, "Credible and energy-aware participant selection with limited task budget for mobile crowd sensing," *Ad Hoc Networks*, vol. 43, pp. 56-70, Jun. 2016.
- [31] P. A. Samuelson and W. D. Nordhaus, *Microeconomics*, 18th ed. Boston, MA, USA: McGraw-Hill, 2005.
- [32] Z. Zhu, *et al.*, "A game theoretic optimization framework for home demand management incorporating local energy resources," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 353-362, Apr. 2015.
- [33] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *ICLR 2015*, May 2015, pp. 1-15.
- [34] J. Fraysse, "Common agency: existence of an equilibrium in the case of two outcomes," *Econometrica*, vol. 61, no. 5, pp. 1225-1229, Sept. 1993.
- [35] S. Durand and B. Gaujal, "Complexity and optimality of the best response algorithm in random potential games," in *SAGT*, Sept. 2016, pp. 40-51.
- [36] Z. Yu, *et al.*, "Using bidirectional LSTM recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech," in *IEEE ASRU*, Dec. 2015, pp. 338-345.
- [37] 802.11ac: The Fifth Generation of Wi-Fi Technical White Paper, 2018. Available Online: <https://www.cisco.com/c/dam/en/us/products/collateral/wireless/aironet-3600-series/white-paper-c11-713103.pdf>. Last Accessed on: Feb. 2021.
- [38] J. Ren, *et al.*, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690-7703, Nov. 2020.
- [39] M. M. Amiri, *et al.*, "Convergence of update aware device scheduling for federated learning at the wireless edge," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3643-3658, Jun. 2021.