Federated Learning Framework With Straggling Mitigation and Privacy-Awareness for AI-Based Mobile Application Services

Yuris Mulya Saputra \textsuperscript{\textdegree}, Member, IEEE, Diep N. Nguyen \textsuperscript{\textdegree}, Senior Member, IEEE, Dinh Thai Hoang \textsuperscript{\textdegree}, Senior Member, IEEE, Quoc-Viet Pham \textsuperscript{\textdegree}, Member, IEEE, Eryk Dutkiewicz \textsuperscript{\textdegree}, Senior Member, IEEE, and Won-Joo Hwang \textsuperscript{\textdegree}, Senior Member, IEEE

Abstract—In this article, 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—Contract theory, encryption, federated learning, privacy, straggling problem

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) \cite{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 \cite{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 \cite{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

---

Received 3 November 2021; revised 5 April 2022; accepted 24 May 2022. Date of publication 30 May 2022; date of current version 4 August 2023. The work of Dinh Thai Hoang was supported by the Australian Government through the Australian Research Council’s Discovery Projects funding scheme under Project DE210100651. The work of Quoc-Viet Pham and Won-Joo Hwang was supported in part by the National Research Foundation of Korea (NRF), Korean Government (MSTD), under Grant NRF-2019R1C1C1006143 and Grant NRF-2019R1I1A3A0401060518, in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP), Korea Government (MSIT), under Grant 2020-0-01450, in part by the Artificial Intelligence Convergence Research Center, Pusan National University, and in part by BK21 Four, Korean Southeast Center for the 4th Industrial Revolution Leader Education. (Corresponding author: Quoc-Viet Pham.)
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/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 processing due to privacy concerns. Furthermore, MUs may provide unreliable data which can lead to frauds in the FL process [11]. These challenges can limit the development of FL in 6G networks when a wide range of network connections with ubiquitous services and system heterogeneity among MUs exist [12].

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 [22]. 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 [23], [24], [25], [26]. Nonetheless, these game models are only applicable when the MUs and the MAP have full information about each other (referred to as information asymmetry), e.g., the MUs know the MAP’s current computing resources. Moreover, these models only consider a non-negotiable mechanism due to the characteristics of the leader and followers. In this case, the MAP first informs the fixed incentive to the MUs. Then, the MUs need to adjust the amount of encrypted dataset to be offloaded based on the given information without any incentive negotiation. 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 [30], in which the participating MUs (as principals) can first propose a contract including 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. Although a few recent works, e.g., [22], [27], [28], have studied the contract-based incentive mechanism in FL, they do not account...
for the FL with data encryption and the additional training processes at the MAP. They also do not use multi-principal one-agent-based contract policy, and thus unable to capture the competition among principals. Moreover, unlike the contract problem for the conventional FL [6] and the coded FL [29], 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. In this case, each MU requires to optimize its utility for both encrypted data sharing and the local training process. Meanwhile, the MAP needs to optimize its utility for the encrypted training process and incentive management for the local training process at the MUs.

However, it is intractable to directly solve the original MPOA-based contract problem due to a strong interdependency between the optimizations of the MAP and the participating MUs with multiple objectives and constraints. Then, 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 15 M activity recognition samples from MUs’ devices in 2019 [31]. 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/communication 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 a 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 straggling problems. To the best of our knowledge, our proposed selection method is the first work considered in the mobile application service.
- 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. To the best of our knowledge, this is the first work which adopts the MPOA-based contract approach for the FL-based mobile application service with encrypted training process.
- Theoretically analyze the convergence of proposed FL with straggling 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 straggling mitigation and privacy-awareness. Then, we present the proposed FL-based framework for the AI-based mobile application service.

2.1 Federated Learning With Straggling Mitigation and Privacy-Awareness Overview

The process of FL with straggling mitigation and privacy-awareness is illustrated in Fig. 1 that consists of the MAP (as a master node) and participating MUs (as workers) \( N = \{1, \ldots, n, \ldots, N\} \). Each MU-\( n \) has its own sensing dataset \( S_n = (F_n, L_n) \), where \( F_n \) and \( L_n \) 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 straggling 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 \( D_n^e \) and \( D_n^l \) represent the sizes of encrypted and local (unencrypted) sensing datasets of MU-\( n \), respectively, such that \( D_n^e + D_n^l \leq D_n, \forall n \in N \). 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 [32]. To improve the performance of FHE, we adopt Brakerski/Fan-Vercauteren (BFV)-based FHE to support concurrent arbitrary operations on encrypted data [33]. Specifically, the BFV-based FHE includes the following polynomial-time algorithms during the encryption and decryption processes.

- **SKGen** to generate the secret key $g_{sk}^{n_k}$ for MU-$n$.
- **PKGen** to generate the public key $g_{pk}^{n_k}$ as the function of $g_{sk}^{n_k}$ for MU-$n$.
- **Enc**($g_{sk}^{n_k}$, $h$) to encrypt data $h$ utilizing $g_{sk}^{n_k}$ with the output of encrypted data $h^*$.  
- **Dec**($g_{sk}^{n_k}$, $h^*$) to decrypt data $h^*$ utilizing $g_{sk}^{n_k}$ 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 $g_{sk}^{n_k}$ using **SKGen**($n$) and public key $g_{sk}^{n_k}$ using **PKGen**($g_{sk}^{n_k}$). The $g_{sk}^{n_k}$ of MU-$n$ remains private to other MUs and the MAP. Meanwhile, $g_{sk}^{n_k}$ 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., $S_{n}^*$, utilizing its $g_{sk}^{n_k}$ to produce the encrypted dataset $S_{n}^r = (F_0^n, L_0^n)$, i.e., $Enc(g_{sk}^{n_k}, S_{n}) = S_{n}^r$, where $F_0^n$ and $L_0^n$ are the encrypted training feature and encrypted training label matrices of the dataset at MU-$n$ with the size $D_0^n = |S_{n}^r|$, respectively. All the $S_{n}^r, \forall n \in \mathcal{N}$, 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

$$ F^r = \begin{pmatrix} F_0^n & \cdots & F_N^n \end{pmatrix}, \quad \text{and} \quad L^r = \begin{pmatrix} L_0^n & \cdots & L_N^n \end{pmatrix}. $$

To minimize extra privacy leakage when sharing the encrypted datasets $S_{n}^r, \forall n \in \mathcal{N}$, with the MAP, e.g., metadata leakage of the encrypted datasets via their plaintext headers [35], a privacy protection level $\varepsilon_n$ ($\varepsilon_n \leq 1$) [36] 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 [34]

$$ \varepsilon_n = \frac{\beta}{2} \log_2 \left( 1 + \frac{\varepsilon_n D_0^n}{A_n^2} \right), $$

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 [34]. 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 [17]. 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

---

**Fig. 1.** The FL with straggling mitigation and privacy-awareness with (a) one-time pre-FL and (b) iterative FL processes.
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 \( \mathcal{M} = \{1, \ldots, m, \ldots, M\} \). At a particular period, each active MU-\( n \) 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) and Apple Health (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 \( \mathcal{N} \) MUs ensures that they can provide useful sensing data for the FL process. This can be determined from the active MUs considering the maximum payment budget allocation of the MAP [38].

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., \( D^o_n \), for the entire FL process, which is \( D^o_n \leq D^o \) [6]. For the collected encrypted dataset at the MAP, its size is constrained by maximum size of encrypted sensing samples \( D^o_{\text{max}} \) 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 [39] 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 \( \Pi = \{ \pi_1, \ldots, \pi_i, \ldots, \pi_I \} \), and \( \pi_1 < \pi_2 < \ldots < \pi_i < \ldots < \pi_I \), where \( i \in I \) with \( I = \{1, \ldots, I\} \) represents the type index.

Despite that the MAP’s type is private, the MUs can still observe the distribution of MAP’s types, i.e., \( \phi_i \), where \( \sum_{i=1}^{I} \phi_i = 1, \forall i \in I \) [36], 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 \( \tilde{D}^i_n \), where \( \tilde{D}^i_n = \frac{\pi_i}{\pi_I} D_{\text{max}} \), \( \forall i \in I \). From \( \tilde{D}^i \), the MAP can allocate the encrypted dataset proportion for each MU-\( n \) corresponding. Specifically, for the MAP with type \( \pi_i \), the encrypted dataset proportion vector of all MUs can be denoted by \( \eta^i_n = [\eta^i_1, \ldots, \eta^i_\mathcal{M}] \), where \( \eta^i_n = [\eta^i_1, \ldots, \eta^i_\mathcal{M}] \), and \( 0 \leq \eta^i_n \leq 1, \forall i \in I, \forall n \in \mathcal{N} \). Here, the proportion vector for the encrypted dataset is required to comply with the limited computing resources of the MAP based on its type. In addition, this ensures that each participating MU can compete with other participating MUs fairly in the proposed contract-based framework for the additional encrypted FL process at the MAP. We denote the expected size of encrypted dataset and corresponding payment vectors of all participating MUs for all MAP’s types as \( D^i = [D^i_1, \ldots, D^i_N] \) and \( \rho^i = [\rho^i_1, \ldots, \rho^i_m, \ldots, \rho^i_M] \), respectively, where \( D^i = [D^i_1, \ldots, D^i_N] \) and \( \rho^i_n = [\rho^i_1, \ldots, \rho^i_m, \ldots, \rho^i_M] \).

In this case, the actual size of encrypted dataset at MU-\( n \) considering the MAP with type \( \pi_i \) will be \( \eta^i_n D^i_n \). 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 \( D^i = [D^i_1, \ldots, D^i_N] \) and \( \rho^i_n = [\rho^i_1, \ldots, \rho^i_m, \ldots, \rho^i_M] \) 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 [40].

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 $R_m$, $\forall m \in M$, within $0 - 1$ range separately and obtain the normalized dataset size $\tilde{D}_m$, normalized computing resources $\tilde{\kappa}_m$, and normalized transmission rate $\tilde{R}_m$, $\forall m \in M$. Then, the MUs via the application can calculate their selection metric values, i.e., $\chi_m = t_1\tilde{D}_m + t_2\tilde{\kappa}_m + t_3\tilde{R}_m$. In this case, $t_1, t_2$, and $t_3$ are the selection weight parameters determined by the MAP to control $\chi_m$, where $t_1 + t_2 + t_3 = 1$ [40], [41]. These $\chi_m, \forall m \in M$, 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.,

$$N = \max\{\chi_1, \ldots, \chi_M\}. \quad (3)$$

Equation (3) indicates that only MUs in $N$ 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 $D'_n, \forall n \in N$ (where $0 \leq D'_n \leq D_n$), and requested payments $\rho_n = \alpha_n D'_n, \forall n \in N$, as well as the sizes of encrypted datasets $E'_n, \forall n \in N, \forall i \in I$ (where $0 \leq E'_i,n \leq E_i,n$) and requested payments $\rho'_n = \alpha_o D'_n, \forall n \in N, \forall i \in I$. These $\alpha_n$ 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 $\pi_m$, i.e., $\eta_m$, 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 $\pi_l$, i.e., $D'^{o}_l$, and its payment vector for the MAP with type $\pi_l$, i.e., $\rho'^{o}_l$, the utility of the MAP with type $\pi_l$ (for executing the proposed FL process along with the MU-$n, \forall n \in N$) can be expressed as follows:

$$U^l_{MAP} = \pi_l G_o(\eta_l, D'^{o}_l) - C_o(\eta_l, \rho'^{o}_l, D'^{o}_l) + G_l(D'_l) - C_l(\rho'_l). \quad (4)$$

Utility from training encrypted data
Utility from local training at MUs

Specifically, the utility of the MAP with type $\pi_l$ can be decomposed into two components. The first component is the difference between the gain function $G_o(\eta_l, D'^{o}_l)$ and cost function $C_o(\eta_l, \rho'^{o}_l, D'^{o}_l)$ for collecting and training the encrypted dataset at the MAP, where $\pi_l$ represents the weight of $G_o(\eta_l, D'^{o}_l)$ for the MAP with type $\pi_l$. 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 [36] 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 [42]. To this end, the gain functions for training encrypted and local datasets can be respectively formulated by

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

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 [36]. 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.,

\[
C_o(\eta_i, \rho_i^o, D_i^o) = \sum_{n \in N} \eta_i^n \rho_i^{o,n} + \xi c f^2 \sum_{n \in N} \eta_i^n D_{i,n}^o,
\]

where \(\xi, c,\) 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.,

\[
C_l(\rho^l) = \sum_{n \in N} \rho_i^l.
\]

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

\[
(P_1) \quad \max_{\eta_i} U_{\text{MAP}},
\]

s.t. \(\sum_{n \in N} \eta_i^n D_{i,n}^o \leq \tilde{D}_i^o\),

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

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., \(\tilde{D}_i^o\). 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 (P1) 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

\[
U_n(D^o, \rho^o, D^l, \rho^l) = \sum_{i=1}^{I} \left( \frac{\hat{\eta}_i^o \rho_i^{o,n} - \xi_i - \gamma_n D_{i,n}^o + \rho_i^l - \xi c n \gamma_n D_{i,n}^l}{\text{Utility from training encrypted data at MAP}} \right) \phi_i.
\]

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 \(\hat{\eta}_i^o \rho_i^{o,n}\) for sharing the encrypted dataset and the costs for encrypted dataset privacy protection and transmission, where \(\xi_i = \frac{\log_2(1 + \gamma_n D_{i,n}^o / \Delta_i)}{2}\) indicates the privacy cost for the encrypted dataset generation of MU-\(n\) when the MAP has type \(\pi_i\). Additionally, \(\gamma_n D_{i,n}^o\) 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 \(\xi_i c n \gamma_n D_{i,n}^l\) for training the local dataset, where \(\xi, c\), and \(\gamma\) 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 \(\sum_{i=1}^{I} \phi_i\) 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 \((D^o, \rho^o, D^l, \rho^l)\) 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.,

\[
\pi_i G_o(\hat{\eta}_i, D_i^o) - C_o(\hat{\eta}_i, \rho_i^o, D_i^o) + G_l(D^l) - C_l(\rho^l) \geq 0,
\]

\(\forall i \in \mathcal{I}\),

\(\text{to join in the contract optimization.}\)

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 constraint) can be derived as follows:

\[
(P_2) \quad \max_{D^o, \rho^o, D^l, \rho^l} \sum_{i=1}^{I} \left( \frac{\hat{\eta}_i^o \rho_i^{o,n} - \xi_i - \gamma_n D_{i,n}^o + \rho_i^l - \xi c n \gamma_n D_{i,n}^l}{\text{Utility from training encrypted data at MAP}} \right) \phi_i.
\]

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 \(\hat{\eta}_i^o \rho_i^{o,n}\) for sharing the encrypted dataset and the costs for encrypted dataset privacy protection and transmission, where \(\xi_i = \frac{\log_2(1 + \gamma_n D_{i,n}^o / \Delta_i)}{2}\) indicates the privacy cost for the encrypted dataset generation of MU-\(n\) when the MAP has type \(\pi_i\). Additionally, \(\gamma_n D_{i,n}^o\) 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 \(\xi_i c n \gamma_n D_{i,n}^l\) for training the local dataset, where \(\xi, c\), and \(\gamma\) 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 \(\sum_{i=1}^{I} \phi_i\) 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 \((D^o, \rho^o, D^l, \rho^l)\) 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 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.,

$$\pi_i G_s(\bar{\eta}_i, D_i^R) - C_0(\bar{\eta}_i, \rho_i^c, D_i^R) + G_l(D_i) - C_l(\rho_i') \geq \pi_{i'} G_s(\bar{\eta}_{i'}, D_{i'}^R) - C_{0}(\bar{\eta}_{i'}, \rho_{i'}^c, D_{i'}^R) + G_l(D_{i'}) - C_l(\rho_{i'}'), \quad i \neq i', \forall i, i' \in \mathcal{I}.$$ 

$$(18)$$

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:

$$\left(P_3 \right) \max_{D^R, \rho^c, D^I, \rho^i} U_n(D^R, \rho^c, D^I, \rho^i), \forall n \in \mathcal{N},$$

s.t.

$$\sum_{n \in \mathcal{N}} \eta_n^R D_n^R \leq \bar{D}^R_i, \forall i \in \mathcal{I},$$

$$(20) \quad (21)$$

$$D_n^R \leq D_n^d, \forall n \in \mathcal{N},$$

$$\eta_n^c D_n^c + D_n^I \leq D_n, \forall i \in \mathcal{I}, \forall n \in \mathcal{N},$$

$$\forall i \in \mathcal{I},$$

$$\forall i \in \mathcal{I},$$

$$(22) \quad (23) \quad (24) \quad (25)$$

Here, the constraints (24) and (25) guarantee that the MAP benefits from the $P_3$ optimization.

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 only obtain the actual utility of each MU-n when the MAP has type $\pi_1$, as $\eta_n^R \rho_n^c + \eta_n^c D_n^c + \rho_n^i - \xi_n c_n f_n D_n^R$. Given the MAP with type $\pi_{i}$, the total actual utility of all participating MUs can be derived by

$$U_{\text{SW}}^n = \sum_{n \in \mathcal{N}} \left( \eta_n^R \rho_n^c + \frac{\beta}{2} \log_2 \left( 1 + \frac{\eta_n^c \rho_n^c D_n^c}{A_n^2} \right) \right) - \eta_n^c D_n^c + \rho_n^i - \xi_n c_n f_n D_n^R.$$ 

$$(26)$$

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

$$U_{\text{SW}}^{i} = U_{\text{MAP}}^{i} + U_{\text{MU} \text{tot}}^{i}.$$ 

$$(27)$$

In this case, the social welfare can reflect the goal of the FL process, i.e., producing the global model with improved accuracy, through aggregating reliable local models from the best participating MUs and using higher types of the MAP.

5 MPOA-BASED FL CONTRACT OPTIMIZATION SOLUTION

5.1 Contract Problem Transformation

The following Proposition 1 shows that the computational complexity of solving $(P_3)$ increases quadratically with the number of MAP’s possible types increases, i.e., $O(I^2)$, leading to a high computing resources to solve the $(P_3)$ at the MAP.

Proposition 1. The computational complexity of solving $(P_3)$ is $O(I^2)$. 

Proof. See Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputer.org/10.1109/TMC.2022.3178949.

To reduce the complexity of solving $(P_1)$, we transform $(P_3)$ 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 $(D^R, \rho^c, D^I, \rho^i)$ to be any feasible contracts from the MUs to the MAP such that if $\pi_i \geq \pi_{i'}$, then $D_i^R \geq D_{i'}^R$, where $i \neq i', i, i' \in I$.

Proof. See Appendix B.

As a result, the MUs will ask for higher payments to the MAP when the MAP has type $\pi_i$ since $D_i^R \geq D_{i'}^R$ from Lemma 1. This condition is stated in Proposition 2.

Proposition 2. If $D_i^R \geq D_{i'}^R$, then $\rho^c_i \geq \rho^c_{i'}$, where $i \neq i', i, i' \in I$.

Proof. See Appendix C.

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 $(D^R, \rho^c, D^I, \rho^i)$, the MAP’s utility must hold

$$\pi_i G_s(\bar{\eta}_i, D_i^R) - C_0(\bar{\eta}_i, \rho_i^c, D_i^R) + G_l(D_i) - C_l(\rho_i') \geq \pi_{i'} G_s(\bar{\eta}_{i'}, D_{i'}^R) - C_{0}(\bar{\eta}_{i'}, \rho_{i'}^c, D_{i'}^R) + G_l(D_{i'}) - C_l(\rho_{i'}'),$$

$$(28)$$

where $\pi_i \geq \pi_{i'}$, $i \neq i', i, i' \in I$.

Proof. See Appendix D.

The Proposition 3 helps reducing the number of IR constraints through applying the MAP’s minimum type, i.e., $\pi_1$, such that $\pi_1 G_s(\bar{\eta}_1, D_1^R) - C_0(\bar{\eta}_1, \rho_1^c, D_1^R) + G_l(D_1) - C_l(\rho_1') \geq \pi_{i'} G_s(\bar{\eta}_{i'}, D_{i'}^R) - C_{0}(\bar{\eta}_{i'}, \rho_{i'}^c, D_{i'}^R) + G_l(D_{i'}) - C_l(\rho_{i'}'), \quad i \neq i', i, i' \in I.$

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:

$$\pi_1 G_s(\bar{\eta}_1, D_1^R) - C_0(\bar{\eta}_1, \rho_1^c, D_1^R) + G_l(D_1) - C_l(\rho_1') \geq 0.$$ 

$$(29)$$
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 (P₃) can be transformed into the local downward incentive constraints (LDIC) as follows:

\[ \pi, G_o(h_i, D^o_i) = \pi, G_o(h_i, D^o_i) - C_o(h_i, \rho^o_i, D^o_i), \forall i \in \{2, \ldots, I\}, (30) \]

where \( D^o_i \geq D^{e_i}_{i-1}, \forall i \in \{2, \ldots, I\} \).

**Proof.** See Appendix E.

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 (P₃) can be transformed into the problem (P₄) as follows:

**Definition 3.** Equilibrium contract solution for (P₄): The optimal contracts \((\hat{D}^o, \hat{\rho}^o, \hat{D}_i, \hat{\rho}_i)\) are the equilibrium solution of the (P₄) if and only if the conditions

\[ U_n(\hat{D}^o_n, \hat{\rho}^o_n, \hat{D}^o_i, \hat{\rho}^o_i, \hat{D}^l_i, \hat{\rho}^l_i) \geq U_n(D^o_n, \rho^o_n, D^o_i, \rho^o_i, D^l_i, \rho^l_i, \hat{D}^o_n, \hat{\rho}^o_n) \]

\[ \forall n \in N, \text{ satisfy and the optimal contracts } (\hat{D}^o, \hat{\rho}^o, \hat{D}_i, \hat{\rho}_i) \text{ still hold the constraints (32)-(34)}. \]

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 [43]. In this case, we can define the best response of MU-n at iteration \( \tau + 1 \) using \((D^{o(t)}_n, \rho^{o(t)}_n, D^{l(t)}_n, \rho^{l(t)}_n)\) as follows:

\[ \Theta^{(t+1)}(D^{o(t)}_n, \rho^{o(t)}_n, D^{l(t)}_n, \rho^{l(t)}_n) = \text{ arg max} \]

\[ (D^{o(n+1)}, \rho^{o(n+1)}, D^{l(n+1)}, \rho^{l(n+1)}) \in Q_n \]

where

\[ U_n(\tau + 1) = U_n(D^{o(n+1)}, \rho^{o(n+1)}, D^{l(n+1)}, \rho^{l(n+1)}) \]

and \( Q_n \) is the non-empty contract space [45] for MU-n with \( \Omega = \prod_{n \in N} Q_n \). According to the Algorithm 1, we can apply the current contract \((D^{o(n+1)}, \rho^{o(n+1)}, D^{l(n+1)}, \rho^{l(n+1)}) \in \Theta^{(t+1)}(\tau + 1)\) to be the new contract at \( \tau + 1 \), i.e., \((D^{o(t+1)}, \rho^{o(t+1)}, D^{l(t+1)}, \rho^{l(t+1)})\), when the following condition exists

\[ \left[ U_n(D^{o(n+1)}, \rho^{o(n+1)}, D^{l(n+1)}, \rho^{l(n+1)}) \right] > \sigma. \]

The algorithm stops when the condition in (38) does not hold for all MU-n, \( \forall n \in N \). 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.

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

\[ \Theta = \{ \Theta_{n}(D^{o(n)}, \rho^{o(n)}, D^{l(n)}, \rho^{l(n)}), \Theta_{n}(D^{o(n)}, \rho^{o(n)}, D^{l(n)}, \rho^{l(n)}) \} \]

\[ \Theta_{n}(D^{o(n)}, \rho^{o(n)}, D^{l(n)}, \rho^{l(n)}) \]

\[ \Theta_{n}(D^{o(n)}, \rho^{o(n)}, D^{l(n)}, \rho^{l(n)}) \]

\[ \Theta_{n}(D^{o(n)}, \rho^{o(n)}, D^{l(n)}, \rho^{l(n)}) \]

\[ \Theta_{n}(D^{o(n)}, \rho^{o(n)}, D^{l(n)}, \rho^{l(n)}) \]

The finding of this fixed point is equivalent to the equilibrium solution [30], [45], where all the MUs in \( N \) obtain the maximum expected utilities where \((\hat{D}^o, \hat{\rho}^o, \hat{D}_i, \hat{\rho}_i)\) 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 $(D_n^{(0)}, \rho_n^{(0)}, D_n^{(0)}, \rho_n^{(0)})$, $\forall n \in \mathcal{N}$, to the MAP
3: repeat
4: Find $\hat{\eta}^{(l)}$ values which maximize $(P_l)$ using $(D_l^{(l)}, \rho_l^{(l)}, D_l^{(l)}, \rho_l^{(l)})$
5: for $\forall n \in \mathcal{N}$ do
6: Produce a new contract $(D_n^{(new)}, \rho_n^{(new)}, D_n^{(new)}, \rho_n^{(new)})$, which maximizes $(P_l)$ given $\hat{\eta}^{(l)}$ and $(D_l^{(l)}, \rho_l^{(l)}, D_l^{(l)}, \rho_l^{(l)})$
7: if the condition in (38) is satisfied then
8: Set $(D_n^{(l+1)}, \rho_n^{(l+1)}, D_n^{(l+1)}, \rho_n^{(l+1)}) = (D_n^{(new)}, \rho_n^{(new)}, D_n^{(new)}, \rho_n^{(new)})$
9: else
10: Set $(D_n^{(l+1)}, \rho_n^{(l+1)}, D_n^{(l+1)}, \rho_n^{(l+1)}) = (D_n^{(l)}, \rho_n^{(l)}, D_n^{(l)}, \rho_n^{(l)})$
11: end if
12: end for
13: $\tau = \tau + 1$
14: until $U_n(D_n^{(l)}, \rho_n^{(l)}, D_n^{(l)}, \rho_n^{(l)}, D_n^{(l)}, \rho_n^{(l)}, D_n^{(l)}, \rho_n^{(l)}) \forall n \in \mathcal{N}$ remain unchanged
15: Obtain the optimal contracts $(\tilde{D}^o, \tilde{\rho}^o, \tilde{D}^f, \tilde{\rho}^f)$

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

Proof. See Appendix H.

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 $\log(N) + \varepsilon^* + O(1/N)$, where $\varepsilon$ is the Euler constant, as formally derived in Theorem 3.

Theorem 3. Algorithm 1 has polynomial complexity $\log(N) + \varepsilon^* + O(1/N)$.

Proof. See Appendix I.

6 Federated Learning with Straggling Mitigation and Privacy-Awareness

Implementation

Using the vectors of optimal encrypted dataset size $\tilde{D}^o$ and optimal local dataset size $\tilde{D}$ of the selected MUs in $\mathcal{N}$ according to the optimal contracts $(\tilde{D}^o, \tilde{\rho}^o, \tilde{D}^f, \tilde{\rho}^f)$, 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 $S^o = (F^o, L^o)$ with optimal size $\sum_n \tilde{D}^o_n$ denote the encrypted dataset of all MUs in $\mathcal{N}$ at the MAP with type $\pi_n$, where $F^o$ and $L^o$ are the training feature and training label matrices of encrypted dataset at the MAP with type $\pi_n$, respectively. Additionally, we denote $S^f = (F^f, L^f)$ with optimal size $\tilde{D}^f_n$ to be the local dataset at each MU-n, where $F^o_n$ and $L^o_n$ are the training feature and label matrices of local dataset at each MU-n, respectively.

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

$$F^o_{i(a+1)} = \alpha^o_a \left( \sum F^o_{i,a} * Q \right),$$

$$F^f_{(a+1)} = \alpha^o_a \sum F^f_{i,a} * Q,$$

where $Q$ is the squared kernel matrix with a certain number of filters and $\alpha^o_a(\cdot), \alpha^o_a(\cdot)$ are the activation functions [3]. These outputs are then fed into a max pooling layer to obtain $F^o_{i(a+2)} = \omega(F^o_{(a+1)})$ and $F^f_{(a+2)} = \omega(F^f_{(a+1)})$, where $\omega$ is the pooling function representing the maximum element values of $F^o_{i(a+1)}$ and $F^f_{(a+1)}$ 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., $F^o_{i:a}^*$ and $F^f_{a}^*$. Using $F^o_{i:a}^*$, $F^f_{a}^*$, and the BLSTM method in [47], we can generate the output matrices $F^o_{i(a+1)}^*$ and $F^f_{(a+1)}^*$.

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

$$L^o_{i,a_{max}} = \alpha_{a_{max}} \left( F^o_{i(a_{max}-1)}* Y_{i,a_{max}} \right),$$

$$L^f_{a_{max}} = \alpha_{a_{max}} \left( F^f_{(a_{max}-1)}* Y_{a_{max}} \right),$$

where $\alpha_{a_{max}}(\cdot)$ and $\alpha_{a_{max}}(\cdot)$ are the softmax activation functions used to provide a probability distribution for the classification results [3]. Moreover, $F^o_{i(a_{max}-1)}$ and $F^f_{(a_{max}-1)}$ are the output matrices from the previous fully connected layer $a_{max} - 1$, while $Y_{i,a_{max}}$ and $Y_{a_{max}}$ are the weight matrices at the final layer $a_{max}$.

Based on $L^o_{i,a_{max}}$, $L^f_{a_{max}}$, $L^o_{i,a_{max}}$, and $L^f_{a_{max}}$, 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.,

$$q_{i}(\hat{T}^{(\theta)}) = \frac{1}{2} \sum_{n \in \mathcal{N}} \left\| L^o_{i,a_{max}} - L_{i}^{(\theta)} \right\|_F^2,$$

$$q_{n}(\hat{T}^{(\theta)}) = \frac{1}{2} \sum_{n \in \mathcal{N}} \sum_{j=1}^{\tilde{D}^o_n} \left( \hat{e}_{i,j} - \hat{e}_{i,j} \right)^2,$$

and

$$q_{n}(\hat{T}^{(\theta)}) = \frac{1}{2} \sum_{n \in \mathcal{N}} \sum_{j=1}^{\tilde{D}^f_n} \left( \hat{e}_{i,j} - \hat{e}_{i,j} \right)^2,$$

respectively, where $\hat{T}^{(\theta)}$ and $\hat{T}^{(\theta)}$ are the encrypted and decrypted global model matrices (i.e., $\hat{T}^{(\theta)} = Dec(d_k, \hat{T}^{(\theta)})$),
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 \( L^{i}_{\text{true}} \) and \( L^{n}_{\text{true}} \) respectively, while \( \ell_{i,j} \) and \( \ell_{n,j} \) are the elements of predicted label data \( L^{i}_{\text{max}} \) and \( L^{n}_{\text{max}} \), respectively.

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

\[
\nabla \mathbf{T}^{(\theta)}_i = \frac{\partial \varphi_i(\hat{\mathbf{T}}^{(\theta)})}{\partial \hat{\mathbf{T}}^{(\theta)}}, \quad \text{and} \quad \nabla \mathbf{T}^{(\theta)}_n = \frac{\partial \varphi_\theta(\hat{\mathbf{T}}^{(\theta)})}{\partial \hat{\mathbf{T}}^{(\theta)}}. \quad (44)
\]

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

\[
\nabla \mathbf{T}^{(\theta)}_n = \frac{\partial \varphi_\theta(\hat{\mathbf{T}}^{(\theta)})}{\partial \hat{\mathbf{T}}^{(\theta)}} = \frac{1}{\partial \hat{\mathbf{T}}^{(\theta)}} \left( \sum_{n \in \mathcal{N}} L^{i}_{\text{true}} - L^{n}_{\text{true}} \right)^2.
\]

Likewise, the local gradient of the local dataset at each MU-\( n \) when we use encrypted training process can be expressed by

\[
\nabla \mathbf{T}^{(\theta)}_n = \frac{1}{\partial \hat{\mathbf{T}}^{(\theta)}} \left( \sum_{n \in \mathcal{N}} L^{i}_{\text{true}} - L^{n}_{\text{true}} \right)^2.
\]

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

\[
\nabla \hat{\mathbf{T}}^{(\theta)}_N = \sum_{n \in \mathcal{N}} \hat{D}^{i}_{n} \nabla \hat{\mathbf{T}}^{(\theta)}_n. \quad (48)
\]

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

\[
\hat{\mathbf{T}}^{(\theta+1)} = \frac{1}{\sum_{n \in \mathcal{N}} \hat{D}^{i}_{n} + \sum_{n \in \mathcal{N}} \hat{D}^{o}_{n}} \left( \nabla \hat{\mathbf{T}}^{(\theta)}_N + \nabla \hat{\mathbf{T}}^{(\theta)}_i \right). \quad (49)
\]

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

\[
\hat{\mathbf{T}}^{(\theta+1)} = \hat{\mathbf{T}}^{(\theta)} - \lambda^{(\theta+1)} \Omega \left( \nabla \hat{\mathbf{T}}^{(\theta)} \right). \quad (50)
\]

where \( \lambda^{(\theta+1)} \) is the encrypted learning step function of the adaptive learning rate Adam optimizer [44]. Furthermore, \( \Omega (\nabla \hat{\mathbf{T}}^{(\theta)}) \) indicates the local update rules as a function of encrypted gradient \( \nabla \hat{\mathbf{T}}^{(\theta)} \) described in [44]. To this end, we can also determine the global loss function at learning round \( \theta + 1 \) as

\[
\varphi (\hat{\mathbf{T}}^{(\theta+1)}) = \frac{1}{N+1} \left( \varphi_i (\hat{\mathbf{T}}^{(\theta)}) + \sum_{n \in \mathcal{N}(t)} \varphi_n (\hat{\mathbf{T}}^{(\theta)}) \right). \quad (51)
\]

The above global process terminates when the global loss converges or the learning rounds achieve a given threshold \( \theta_{\text{th}} \), and thus the final encrypted global model \( \hat{\mathbf{T}} \) and the final global loss \( \varphi (\hat{\mathbf{T}}) \) 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 \( \theta_{\text{th}} \) and \( \varphi (\hat{\mathbf{T}}) \), and \( \theta = 0 \)
2. The MAP determines MUs in \( \mathcal{N} \subset M \) using (32)
3. Perform Algorithm 1 for all MUs in \( \mathcal{N} \) to obtain optimal contracts \( D \), \( \rho \)
4. for all \( n \in \mathcal{N} \) do
5. Implement the data encryption with the optimal size \( D^{i}_{n} \) considering the MAP with type \( \pi \)
6. Send the encrypted dataset to the MAP
7. Set \( F^{i}_{n} \) and \( L^{i}_{n} \) based on the optimal size \( D^{i}_{n} \)
8. end for
9. The MAP combines the received encrypted datasets into \( S^{i}_{o} \)
10. The MAP sets \( F^{o}_{n} \) and \( L^{o}_{n} \) from \( S^{i}_{o} \)
11. while \( \theta \leq \theta_{\text{th}} \) and \( \varphi (\hat{\mathbf{T}}^{(\theta)}) \) does not converge do
12. for all \( n \in \mathcal{N} \) do
13. Compute \( L^{i}_{\text{max}} \) using \( F^{i}_{n} \) and \( L^{i}_{n} \)
14. Decrypt \( \nabla \hat{\mathbf{T}}^{(\theta)} \) into \( \nabla \mathbf{T}^{(\theta)} \)
15. Find \( \varphi_i (\mathbf{T}^{(\theta)}) \) and \( \nabla \mathbf{T}^{(\theta)}_n \)
16. Encrypt \( \nabla \mathbf{T}^{(\theta)}_n \) into \( \nabla \hat{\mathbf{T}}^{(\theta)}_n \) and send it to the MAP
17. end for
18. The MAP calculates \( L^{i}_{\text{max}} \) using \( F^{i}_{n} \) and \( L^{i}_{o} \)
19. The MAP obtains \( \varphi_i (\hat{\mathbf{T}}^{(\theta)}) \) and \( \nabla \hat{\mathbf{T}}^{(\theta)}_n \)
20. Aggregate \( \nabla \hat{\mathbf{T}}^{(\theta)}_n \) for all \( n \in \mathcal{N} \), using (48) and the encrypted gradient \( \nabla \hat{\mathbf{T}}^{(\theta)}_n \) using (49)
21. Update the encrypted global model \( \hat{\mathbf{T}}^{(\theta+1)} \) using (50)
22. Obtain the global loss \( \varphi (\hat{\mathbf{T}}^{(\theta+1)}) \) using (51)
23. \( \theta = \theta + 1 \)
24. end while
25. Finalize the encrypted global model \( \hat{\mathbf{T}}^{*} \) and global loss \( \varphi (\hat{\mathbf{T}}^{*}) \)

**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 [31]. This dataset contains 15 M 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 1 M 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 ~5 K 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 \( D_{\text{max}}^n = 5 \times 10^5 \) samples. We set \( v_i \) and \( v_f \) at 0.125 and 3, respectively, as well as \( a_i \) and \( a_l \) at 0.001 and 0.005, respectively. The reason is that the encrypted training process at the MAP consumes the MAP’s computing resources and does not consume the MUs’ computing resources. We also denote \( \beta = 1 \) and \( \gamma = 0.0001 \). We use \( \zeta = 5 \times 10^{-2}, \forall u \in N \) [25], and \( f_n = 2 \) GHz for practical CPU frequency of today’s smartphone. Moreover, we use an average transmission rate 293 Mbps of 802.11ac WLAN with bandwidth 80 MHz [48]. 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 [22], [34]. 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 [37]. 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. We also compare the accuracy performance of our proposed framework with red-FL, i.e., FL with straggling mitigation by reducing the training data size at MUs to comply with the MUs’ computing resources [6], and async-FL, i.e., FL with straggling mitigation by asynchronously sending MUs’ trained models to the MAP [49]. For the proposed FL, each MU obtains the size of its optimal encrypted dataset based on the MAP’s type.

### 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 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 it applies the right contract for its corresponding type.

#### 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 is limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
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 problems. These results reflect that a higher proportion of encrypted dataset at the MAP may result in higher utilities and social welfare for 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]. Moreover, although other baseline FL methods with straggling mitigation, i.e., red-FL and async-FL, can mitigate the straggling problems, they cannot improve the accuracy performance at high learning rounds. This is due to reduction of training data sizes at participating MUs and asynchronous trained model updates from non-IID dataset of each participating MU, respectively, which then affect the quality of the FL process.

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 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 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. Nonetheless, we would like to clarify that the actual training time for the proposed framework may be higher than that of the conv-FL if the MAP has the same computing capability as the MUs. The reason is that the sizes of encrypted data and encrypted model gradient updates are higher than those of unencrypted data and unencrypted model gradient updates, thereby requiring more training time at each learning round [16], [18]. In our framework, the encrypted data are only trained at the MAP with high computing power, and thus the above problem can be minimized. 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 Netw. Appl., vol. 26, no. 1, pp. 285–303, Feb. 2021.

[2] Big Data and Business Analytics Market 2027, pp. 1–417, Sep. 2020. Last Accessed on: Mar. 2021. [Online]. Available: https://www.alliedmarketresearch.com/big-data-and-business-analytics-market
Z. Xiong, J. Zhao, Y. Zhang, D. Niyato, and J. Zhang, “Contract design in hierarchical game for sponsored content service market,” IEEE Trans. Mobile Comput., vol. 20, no. 9, pp. 2763–2778, Sep. 2021.

Z. Xiong, J. Kang, D. Niyato, P. Wang, H. V. Poor, and S. Xie, “A multi-dimensional contract approach for data rewarding in mobile networks,” IEEE Trans. Wireless Commun., vol. 19, no. 9, pp. 5779–5793, Sep. 2020.

J. S. Ng et al., “A hierarchical incentive design toward motivating participation in coded federated learning,” IEEE J. Sel. Areas Commun., vol. 40, no. 1, pp. 359–375, Jan. 2022.

B. D. Bernheim and M. D. Whinston, “Common agency,” Econometrica, vol. 54, no. 4, pp. 923–942, Jul. 1986.

WISDM smartphone and smartwatch activity and biometrics dataset - data set 2019. Last Accessed on Mar. 2021. [Online]. Available: https://archive.ics.uci.edu/ml/machine-learning-databases/00507/

C. Gentry, A Fully Homomorphic Encryption Scheme. Stanford, CA, USA: Stanford Univ., 2009.

J. Fan and F. Vercauteren, “Somewhat practical fully homomorphic encryption,” in Proc. IACR Cryptol. ePrint Arch., 2012, pp. 1–18.

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.

K. Nikitin et al., “Reducing metadata leakage from encrypted files and communication with PURBs,” Proc. Privacy Enhancing Techn., vol. 4, pp. 6–33, Jun. 2019.

L. Xu, C. Jiang, Y. Chen, Y. Ren, and K. J. R. Liu, “Privacy or utility in data collection? A multiobjective theoretic approach,” IEEE Trans. Signal Process., vol. 9, no. 7, pp. 1256–1269, Oct. 2015.

Y. M. Saputra, D. Nguyen, H. T. Dinh, T. X. Vu, E. Dutkiewicz, and S. Chatzinotas, “Federated learning meets contract theory: Economic-efficiency framework for electric vehicle networks,” IEEE Trans. Mobile Comput., early access, Dec. 21, 2020, doi: 10.1109/TMC.2020.3049867.

Z. Zheng, F. Wu, X. Gao, H. Zhu, S. Tang, and G. Chen, “A budget feasible incentive mechanism for users of connected Vehicular Ad Hoc Networks,” IEEE Trans. Veh. Technol., vol. 63, no. 9, pp. 5986–5994, Jul. 2020.

J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, “Reliable federated learning for mobile networks,” IEEE Wireless Commun., vol. 27, no. 2, pp. 72–80, Apr. 2020.

Y. Liu, Y. Xu, Z. Xiong, J. Kang, X. Wang, and D. Niyato, “Federated learning for 6G communications: Challenges, methods, and future directions,” China Commun., vol. 17, no. 9, pp. 105–118, Sep. 2020.

J. S. Ng et al., “A comprehensive survey on coded distributed computing: Fundamentals, challenges, and networking applications,” IEEE Commun. Surv. Tuts., vol. 23, no. 3, pp. 1800–1837, Jul.–Sep. 2021.

J. W. Park, D. J. Han, M. Choi, and J. Moon, “Sageflow: Robust federated learning against both stragglers and adversaries,” Adv. Neural Inf. Process. Syst., vol. 34, pp. 1–12, Dec. 2021.

N. J. Hernandez Marcano, M. Moller, S. Hansen, and R. H. Jacobsen, “On fully homomorphic encryption for privacy-preserving deep learning,” in Proc. IEEE Globecom Workshops, 2019, pp. 1–6.

Z. Yue et al., “Privacy-preserving time-series medical images analysis using a hybrid deep learning framework,” ACM Trans. Intell. Syst. Technol., vol. 11, no. 3, pp. 1–29, Jun. 2020.

L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, “Privacy-preserving deep learning via additively homomorphic encryption,” IEEE Trans. Inf. Forensics Secur., vol. 13, no. 5, pp. 1333–1345, May 2018.

H. Fang and Q. Qian, “Privacy preserving machine learning with homomorphic encryption and federated learning,” Future Internet, vol. 11, no. 4, pp. 1–15, Apr. 2019.

P. Kairouz, Z. Liu, and T. Steinke, “The distributed discrete gaussian mechanism for federated learning with secure aggregation,” in Proc. Int. Conf. Mach. Learn. PMLR, 2021, pp. 5201–5212.

W. N. Chen, C. A. Choquette-Choo, and P. Kairouz, “Communication efficient federated learning with secure aggregation and differential privacy,” in Proc. NeurIPS Workshop Privacy Machine Learning, 2019.

K. Bonawitz, P. Kairouz, B. McMahan, and D. Ramage, “Federated learning and privacy,” ACM Queue, vol. 19, no. 5, pp. 1–28, Nov. 2021.

J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, “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.

D. Yang, G. Xue, X. Fang, and J. Tang, “Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones,” IEEE/ACM Trans. Netw., vol. 24, no. 3, pp. 1732–1744, Jun. 2016.

S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y.-C. Liang, “Joint service pricing and cooperative relay communication for federated learning,” in Proc. IEEE IThings, GreenCom, CPSCom, SmartData, 2018, pp. 815–819.

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.

D. Ding et al., “Incentive mechanism design for distributed coded machine learning,” in Proc. IEEE INFOCOM, 2021, pp. 1–10.
Yuris Mulya Saputra (Member, IEEE) received the BS degree in telecommunication engineering from the Institut Teknologi Bandung, Indonesia, in 2010, the MS degree in electrical and information engineering from the Seoul National University of Science and Technology (SeoulTech), South Korea, in 2014, and the PhD degree from the University of Technology Sydney, Australia, in 2022. He is currently a full-time lecturer and the vice head with the Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada, Indonesia. He was also a researcher on the ubiquitous communications for 5G networks project funded by Intel Corporation, the USA in 2019, a researcher on the IEEE802.11ax HEW project funded by LG Corporation, South Korea between 2014 and 2015, and an application software developer in Digital Appliance Division, Samsung Electronics, Indonesia from 2010 to 2012. His research interests include mobile computing, energy and economic efficiency, machine learning, and optimization problems for wireless communication networks. He was an exemplary reviewer of *IEEE Wireless Communications Letters* in 2020. He was the recipient of PhD Post-Thesis Award, Faculty of Engineering and Information Technology, UTS, in 2021. He is currently an active reviewer of various Q1 journals, including IEEE TMC, IEEE JSAC, IEEE WCM, IEEE TWC, IEEE IoT Journal, IEEE WCL, and Elsevier JCLEPRO.

Dinh Thai Hoang (Senior Member, IEEE) received the PhD degree in computer science and engineering from the Nanyang Technological University, Singapore, in 2016. He is currently a faculty member with the School of Electrical and Data Engineering, University of Technology Sydney, Australia. His research interests include emerging topics in wireless communications and networking, such as machine learning, edge intelligence, cybersecurity, IoT, and metaverse. He was the recipient of several awards, including Australian Research Council and IEEE TCSC Award for Excellence in Scalable Computing (Early Career Researcher). He is currently the editor of the *IEEE Transactions on Wireless Communications*, *IEEE Transactions on Cognitive Communications and Networking*, *IEEE Transactions on Vehicular Technology*, and an associate editor for *IEEE Communications Surveys and Tutorials*.

Quoc-Viet Pham (Member, IEEE) received the BS degree in electronics and telecommunications from the Hanoi University of Science and Technology, Vietnam, in 2013, and the PhD degree in telecommunications engineering from Inje University, South Korea, in 2017. Since February 2020, he has been a research professor with Pusan National University, South Korea. He is specialized in applying convex optimization, game theory, and machine learning to analyze and optimize edge computing and future wireless communications. He has been granted the Korea NRF Funding for outstanding young researchers for the term 2019–2024. He is the editor of the *Journal of Network and Computer Applications* (Elsevier), *Scientific Reports* (Nature), and *Frontiers in Communications and Networks*, and the lead guest editor of the *IEEE Internet of Things Journal*. He was the recipient of the Best PhD Dissertation Award from Inje University in 2017, Top Reviewer Award from the IEEE Transactions on Vehicular Technology in 2020, and Golden Globe Award 2021 from the Ministry of Science and Technology (Vietnam).

Diep N. Nguyen (Senior Member, IEEE) received the ME degree in electrical and computer engineering from the University of California at San Diego (UCSD), and the PhD degree in electrical and computer engineering from The University of Arizona. He is currently a faculty member with the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS). Before joining UTS, he was a DECRA research fellow with Macquarie University and a member of Technical Staff with Broadcom Corporation, Irvine, CA, USA, and ARCON Corporation, Boston, MA, USA, and consulting the Federal Administration of Aviation on turning detection of UAVs and aircraft, and the U.S. Air Force Research Laboratory on anti-jamming. His research interests include computer networking, wireless communications, and machine learning application, with emphasis on systems’ performance and security/privacy. He was the recipient of several awards from LG Electronics, UCSD, UA, the U.S. National Science Foundation, and Australian Research Council. He is currently the editor, an associate editor, the guest editor of the *IEEE Transactions on Mobile Computing*, *IEEE Access*, *Sensors Journal*, *IEEE Open Journal of the Communications Society*, and *Scientific Reports* (Nature’s).

Eryk Dutkiewicz (Senior Member, IEEE) received the BE degree in electrical and electronic engineering and the MSc degree in applied mathematics from the University of Adelaide, in 1988 and 1992, respectively, and the PhD degree in telecommunications from the University of Wollongong, in 1996. His industry experience include management of the Wireless Research Laboratory at Motorola in early 2000’s. He is currently the head of School of Electrical and Data Engineering with the University of Technology Sydney, Australia. He also holds a professorial appointment with Hokkaido University in Japan. His current research interests include 5G/6G and IoT networks.

Won-Joo Hwang (Senior Member, IEEE) received the BS and MS degrees in computer engineering from Pusan National University, Busan, South Korea, in 1998 and 2000, respectively, and the PhD degree in information systems engineering from Osaka University, Osaka, Japan, in 2002. From 2002 to 2004, he was a research professor with the Inje University, Gimhae, South Korea. He is currently a full professor with the Department of Biomedical Convergence Engineering, Pusan National University. His research interests include optimization theory, game theory, machine learning, and data science for wireless communications and networking.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.