Incentive Mechanism for AI-Based Mobile Applications with Coded Federated Learning

Yuris Mulya Saputra, Diep N. Nguyen, Dinh Thai Hoang, and Eryk Dutkiewicz
School of Electrical and Data Engineering, University of Technology Sydney, Australia

***Abstract*—Federated learning (FL) has emerged as a highly-effective distributed learning framework for various AI-based mobile applications. However, in conventional FL, participating mobile users (MUs) may have limited computing resources to train their local data, which leads to degradation of learning quality for the whole FL process. To address this problem, coded FL (codFL) has been recently introduced, allowing MUs to upload part of their coded data to a mobile application provider (MAP) before the learning process. As a result, codFL can not only deal with the MUs' limited computing resources, but also provide more benefits for the MUs to participate in the learning process. Nonetheless, in practice, the MAP and MUs often belong to different parties who unilaterally aim to maximize their individual utility functions. Thus, in this paper, we propose an effective mechanism for the codFL process to incentivize all the participating MUs while improving the learning quality of the MAP. Specifically, we first design a codFL contract optimization problem leveraging a multi-principal one-agent (MPOA) approach in contract theory, under limited computing resources at the MAP and MUs as well as information asymmetry between them. To find the optimal contracts for MUs, we develop an iterative contract algorithm which can produce maximum utilities for all MUs while satisfying all the constraints of the MAP. Numerical results show that our framework can enhance the utilities of MUs up to 113% and system performance in terms of social welfare up to 42% compared with the baseline method.**

Keywords- Federated learning, coded computing, multi-principal one-agent, contract theory, mobile application.

I. Introduction

Federated learning (FL) has been considered as a highly-effective learning method to deal with a massive demand of big data trading especially for emerging artificial intelligence (AI)-based mobile application services (e.g., healthcare, crowdsensing, and mobile social networks). Using FL, mobile users (MUs) can help the mobile application provider (MAP) build highly-accurate mobile applications without requiring MUs to share their raw data [1], [2]. Specifically, at each learning round, the participating MUs can first train their local data to generate local trained models individually. Then, they can upload their local models to the MAP for a global model update, aiming at improving the global model accuracy. However, in practice, participating MUs usually do not have sufficient computing resources to train their local datasets and may suffer from unrealiable wireless communication links when uploading the local trained models to the MAP. As such, the quality of the FL process can be significantly degraded due to low prediction model accuracy as well as unstable local trained model updates from straggling devices.

To address the aforementioned issues, the local data from unreliable MUs can be uploaded to an edge or the cloud server as investigated in recent works [3]–[5]. Although this data-sharing approach can relieve the burden of the MUs and maintain the FL performance, sharing local data to the edge/cloud server may violate the data protection and privacy preservation, the key advantages of FL. For that, utilizing an additional coded computing for the FL process (referred to as *coded* FL) has emerged as one of the most effective solutions to simultaneously maintain the FL performance and mitigate the privacy concern [6]–[8]. In particular, parts of local data from all participating MUs can be first encoded to protect their data privacy. Then, the coded data can be offloaded to the MAP for the additional training process (in addition to the local training process using the rest of local data at all the MUs). In this way, the coded FL can not only compensate the MUs' limited computing resource problem, but also bring more benefits for the MUs to join in the learning process. However, the above coded FL works do not consider the conflicts-of-interests between the MUs and MAP as well as the competition among the MUs to maximize their own benefits. For that, an effective incentive mechanism for both the MAP and the MUs is required to guarantee that the MUs and the MAP are willing to join the learning processes (including local/coded data training and local/global model exchanges). Alternatively, it is critical to find a mutual incentive mechanism that can maximize the utilities for the MUs and at the same time improve the learning quality of the MAP under their constraints, e.g., limited computing resources.

To achieve the above goal, it is essential to consider the self-interest interaction between the MAP and participating MUs. For example, the authors in [10]–[12] discuss Stackelberg game models where the MAP and MUs can act as leaders or followers. Nonetheless, these models are only applicable when both the MAP and the MUs have the complete knowledge from each other (referred to as *information symmetry*), e.g., the MUs know the MAP's available computing resource. In practice, the MAP may keep its computing resource information as a private information (referred to as *information asymmetry*). Additionally, the participating MUs may not fully follow the controls from the MAP due to the conflict of economic interests between them. Consequently, all the aforementioned works may not be effective to implement in our problem.

Given the above, the contract-based game approach [13] can be used to cope with the information asymmetry issue under the competition among participating MUs while considering the economic interests between the MAP and the MUs. For that, in this paper, we propose an effective incentive mechanism for coded FL (codFL) leveraging a multi-principal one-agent (MPOA)-based contract approach [14], aiming at maximizing the utilities for both the MAP and participating MUs in the codFL process. In particular, the participating MUs act as the principals which offer contracts containing the sizes of local and coded datas as well as their offered payments to the MAP individually. Meanwhile, the MAP operates as the agent which can optimize the offered contracts on behalf of the participating MUs. To this end, we formulate the contract model as a codFL contract optimization problem under the limited computing resources at the MAP and MUs, information asymmetry between them, and the MAP's common constraints. These common constraints guarantee that the MAP will always obtain a non-negative and maximum utility during the codFL process. To address the problem, we develop an iterative algorithm at the MAP to not only obtain the optimal contracts but also find the equilibrium solution for all participating MUs. This solution can achieve the performance gap within 0.57% compared with that obtained by the information-symmetry FL contract, i.e., when each MU completely knows other MUs' available data sizes and the MAP's current available computing resource. Through numerical results, we demonstrate that our framework can improve the utilities of MUs up to 113% and social welfare up to 42% compared with the baseline method.

II. System Model

The system model of the considered codFL process is shown in Fig. 1. We consider an MAP as a parameter server and multiple participating MUs as the workers who wish to contribute to the FL process. Let $\mathcal{I} = \{1, \dots, i, \dots, I\}$ denote the set of participating MUs in the codFL process. Prior to the FL process, each participating MU-i can capture sensing information via its embedded sensor devices, e.g., smartphones and smartwatches, during a particular period. This sensing information can be stored in a log file at the MU's internal storage and then can be used as the available dataset with size δ_i for the learning process. Nonetheless, due to the limited available computing resources and unstable communication links of the participating MUs when implementing the learning process [2], [8], each MU-i may want to offload a part of its local data to the MAP before the FL process. The coded dataset with size δ_i^c is first generated using a privacy-preserving encoding method [8]. Then, the generated coded dataset is uploaded to the MAP once and trained during the coded computing process iteratively. Meanwhile, the remaining local datasets with sizes $\delta_i^l, \forall i \in \mathcal{I}$, are trained locally at MUs. The local and coded trained models which are produced from the codFL process at the participating MUs and the MAP, respectively, can be then aggregated to update the global model at each learning round until the codFL process converges. However, due to limited computing resource at the MAP to train the coded datasets uploaded from all the participating

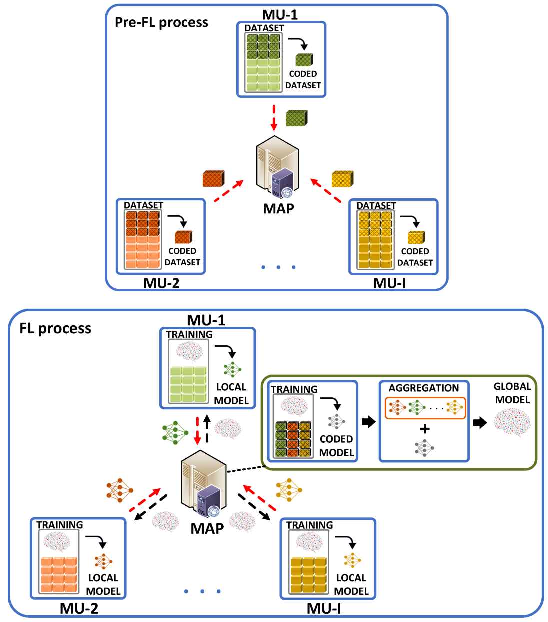


Fig. 1: The codFL process in a mobile application service.

MUs during the codFL process, the total coded dataset that can be collected by the MAP is constrained by the MAP's total computing resource, which can be expressed by the maximum size of trainable coded dataset δ_{max}^c of the MAP. Additionally, due to limited computing resource at each MU-i, the size of local dataset must be less than or equal to the maximum size of dataset that can be processed at the MU-i, i.e., $\hat{\delta}_i^l$, for the entire FL process, which is $\delta_i^l \leq \hat{\delta}_i^l$ [2].

In our considered problem, the MAP keeps its willingness to train the total coded dataset as an information asymmetry for the MUs due to its economic benefit. This willingness is defined as the type of the MAP [13] and influenced by the MAP's current available computing resource. Specifically, a higher type indicates the willingness to train more coded datasets from the participating MUs due to its higher available computing resource. In other words, the willingness to train more coded datasets can compensate the low computing resource problem more, and thus bring more benefits for the MUs and better learning quality for the MAP. In this way, a finite set of the MAP's types can be defined as $\mathcal{Z} = \{\pi_1, \dots, \pi_k, \dots, \pi_K\}$, and $\pi_1 < \pi_2 < \dots < \pi_k < \dots < \pi_{K-1} < \pi_K$, where $k \in \mathcal{K}$ with $\mathcal{K} = \{1, \dots, k, \dots, K\}$ specifies the type index. Although the MAP's type is unknown, the participating MUs can still observe the distribution of MAP's types, i.e, ρ_k, where $\sum_{k=1}^{K} \rho_k = 1, \forall k \in \mathcal{K}$ [16], e.g., by monitoring public workloads of the MAP in the previous learning processes. For the MAP with type π_k, it has the maximum size of trainable coded dataset $\hat{\delta}_k^c$, where $\hat{\delta}_k^c = \frac{\pi_k}{\pi_K} \delta_{max}^c, \forall k \in \mathcal{K}$. Based on $\hat{\delta}_k^c$, the MAP can determine the coded dataset proportion for each MU-i accordingly. For the MAP with type π_k, the coded dataset proportion vector of all participating MUs can be denoted by $\boldsymbol{\phi} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_k, \dots, \boldsymbol{\phi}_K]$, where $\boldsymbol{\phi}_k = [\phi_k^1, \dots, \phi_k^i, \dots, \phi_k^I]$, and $0 \leq \phi_k^i \leq 1, \forall k \in \mathcal{K}, \forall i \in \mathcal{I}$. We denote the size of coded dataset and corresponding payment vectors of all participating

MUs for all MAP's types in the contract implementation as $\boldsymbol{\delta}^c = [\boldsymbol{\delta}_1^c, \dots, \boldsymbol{\delta}_k^c, \dots, \boldsymbol{\delta}_K^c]$ and $\boldsymbol{\varrho}^c = [\boldsymbol{\varrho}_1^c, \dots, \boldsymbol{\varrho}_k^c, \dots, \boldsymbol{\varrho}_K^c]$, respectively, where $\boldsymbol{\delta}_k^c = [\delta_{k,1}^c, \dots, \delta_{k,i}^c, \dots, \delta_{k,I}^c]$ and $\boldsymbol{\varrho}_k^c = [\varrho_{k,1}^c, \dots, \varrho_{k,i}^c, \dots, \varrho_{k,I}^c]$. As such, each MU-$i$ can offer higher size of coded dataset to the MAP when the MAP's type gets higher since the MU-i can obtain higher payment from the MAP. Moreover, we denote $\boldsymbol{\delta}^l = [\delta_1^l, \dots, \delta_i^l, \dots, \delta_I^l]$ and $\boldsymbol{\varrho}^l = [\varrho_1^l, \dots, \varrho_i^l, \dots, \varrho_I^l]$ to be the size of local dataset and corresponding payment vectors of all the MUs, respectively.

III. MPOA-Based CodFL Contract Problem

In this section, we formulate and exploit the codFL contract optimization problem using the MPOA approach, i.e., the MAP and participating MUs act as the agent and principals, respectively, to maximize the utilities of the MAP and participating MUs in the FL process. Specifically, the participating MUs can first send initial contracts containing local dataset sizes $\delta_i^l, \forall i \in \mathcal{I}$ (where $0 \leq \delta_i^l \leq \delta_i$), and payments $\varrho_i^l = \gamma_l \delta_i^l, \forall i \in \mathcal{I}$, as well as the expected coded dataset sizes $\delta_{k,i}^c, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}$ (where $0 \leq \delta_{k,i}^c \leq \delta_i$) and payments $\varrho_{k,i}^c = \gamma_c \delta_{k,i}^c, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}$. The constants γ_l and γ_c indicate unit fees of utilizing a local dataset sample for local training at an MU and a coded dataset sample for coded training at the MAP, respectively. Then, the utility optimizations for both the MAP and participating MUs can be described below.

A. The MAP's Utility Maximization

Given the coded dataset proportion vector for the MAP with type π_k, i.e., $\boldsymbol{\phi}_k$, the local dataset size vector, i.e., $\boldsymbol{\delta}^l$, and its payment vector, i.e., $\boldsymbol{\varrho}^l$, the coded dataset size vector for the MAP with type π_k, i.e., $\boldsymbol{\delta}_k^c$, and its payment vector for the MAP with type π_k, i.e., $\boldsymbol{\varrho}_k^c$, the MAP's utility for type π_k in the codFL process with the MU-$i, \forall i \in \mathcal{I}$, can be derived by

$$\psi_{\text{MAP}}^k = \underbrace{\pi_k S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c) - C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c)}_{\textit{Utility of coded training}} + \underbrace{S_l(\boldsymbol{\delta}^l) - C_l(\boldsymbol{\varrho}^l)}_{\textit{Utility of local training}}. \tag{1}$$

In this case, the utility of MAP with type π_k can be divided into two utility functions. For the former one, it has the gain function $S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c)$ and cost function $C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c)$ for collecting and training the coded dataset at the MAP. The type π_k specifies the weight of $S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c)$ for the MAP with type index k. As such, the MAP with a higher type obtains a higher weight due to the MAP's willingness to train more coded dataset in the codFL process. Meanwhile, for the later utility, it contains the gain function $S_l(\boldsymbol{\delta}^l)$ and cost function $C_l(\boldsymbol{\varrho}^l)$ for local training at the MUs.

For both gain functions, we utilize a squared-root function in a similar way as that of [16]. In this case, the gain functions for training coded and local datasets can be respectively formulated by

$$S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c) = \alpha_c \sqrt{\sum_{i \in \mathcal{I}} \phi_k^i \delta_{k,i}^c}, \text{ and } S_l(\boldsymbol{\delta}^l) = \alpha_l \sqrt{\sum_{i \in \mathcal{I}} \delta_i^l}, \tag{2}$$

where $\alpha_c > 0$ and $\alpha_l > 0$ are the conversion parameters representing the monetary unit of using the coded and local datasets, respectively, which are determined by the current data trading market [16]. For the MAP's cost function $C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c)$, we define it as the sum of total payments to the participating MUs (regarding their coded datasets) and energy consumption cost for training all coded datasets in the codFL process, i.e.,

$$C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c) = \sum_{i \in \mathcal{I}} \phi_k^i \varrho_{k,i}^c + \xi \beta f^2 \sum_{i \in \mathcal{I}} \phi_k^i \delta_{k,i}^c, \tag{3}$$

where ξ, β, and f are respectively the capacitance parameter of computing chipset [9], the number of CPU cycles to train one coded dataset sample, and used computing resource for the MAP. For the MAP's cost function $C_l(\boldsymbol{\varrho}^l)$, we can simply formulate it as the total payments to the participating MUs with respect to their local training processes, i.e.,

$$C_l(\boldsymbol{\varrho}^l) = \sum_{i \in \mathcal{I}} \varrho_i^l. \tag{4}$$

From (1)-(4), we can formulate the optimization problem of the MAP with type π_k whose optimal proportion vector $\hat{\boldsymbol{\phi}}_k$ can maximize its utility as follows:

$$(\mathbf{P}_1) \quad \max_{\boldsymbol{\phi}_k} \ \psi_{\text{MAP}}^k, \tag{5}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}} \phi_k^i \delta_{k,i}^c \leq \hat{\delta}_k^c, \tag{6}$$

$$0 \leq \phi_k^i \leq 1, \forall i \in \mathcal{I}, \tag{7}$$

where the constraint (6) implies that the total coded dataset trained at the MAP cannot exceed the maximum trainable coded dataset at the MAP with type π_k, i.e., $\hat{\delta}_k^c$. From $(\mathbf{P}_1)$, we can obtain the optimal $\hat{\boldsymbol{\phi}}_k = [\hat{\phi}_k^1, \dots, \hat{\phi}_k^i, \dots, \hat{\phi}_k^I], \forall k \in \mathcal{K}$.

B. Participating MUs' Utility Maximization

Next, we derive the utility optimization for the participating MUs to maximize expected utility of each MU-i independently. Particularly, given $\hat{\boldsymbol{\phi}}_k, \forall k \in \mathcal{K}$, each MU-$i$ can obtain its expected utility by considering all possible types of the MAP as expressed in (8). Specifically, the first term in (8) indicates the utility of coded dataset training implemented by the MAP. As such, the MU-i can receive the payment $\hat{\phi}_k^i \varrho_{k,i}^c$ for sharing the coded dataset under privacy protection and transmission costs of the coded dataset. In this case, $\frac{\sigma}{2} \log_2 \left(1 + \frac{\varepsilon_i \hat{\phi}_k^i \delta_{k,i}^c}{A_i^2}\right)$ [15] specifies the privacy cost for the coded dataset generation of MU-i when the MAP has type π_k, where ε_i ($\varepsilon_i \leq 1$) [16] is the privacy protection level, σ is the unit cost of the privacy, and A_i is the amount of additional noise to the coded dataset of MU-i. Moreover, $\upsilon \hat{\phi}_k^i \delta_{k,i}^c$ indicates the transmission cost of coded dataset, where υ is the unit cost of sending a coded dataset sample to the MAP. Meanwhile, the second term in (8) represents the utility of local dataset training at the MU-i. Particularly, the MU-i receives the payment for training the local dataset with the cost of energy consumption $\xi_i \beta_i f_i^2 \delta_i^l$, where ξ_i, β_i, and f_i are the capacitance parameter of computing chipset, the number of CPU cycles to execute a local dataset sample, and used computing resource for the MU-i, respectively.

To maximize the expected utility of all participating MUs, we need to satisfy the IR and IC constraints from the MAP as stated in Definition 1 and 2.

$$\psi_i(\boldsymbol{\delta}^c, \boldsymbol{\varrho}^c, \boldsymbol{\delta}^l, \boldsymbol{\varrho}^l) = \sum_{k=1}^{K} \Bigg(\underbrace{\hat{\phi}_k^i \varrho_{k,i}^c - \frac{\sigma}{2}\log_2\left(1 + \frac{\varepsilon_i \hat{\phi}_k^i \delta_{k,i}^c}{A_i^2}\right) - \upsilon \hat{\phi}_k^i \delta_{k,i}^c}_{\textit{Utility of coded training}} + \underbrace{\varrho_i^l - \xi_i \beta_i f_i^2 \delta_i^l}_{\textit{Utility of local training}} \Bigg) \rho_k. \tag{8}$$

Definition 1. *IR constraint: The MAP with type $\pi_k, k \in \mathcal{K}$, must obtain a non-negative utility to join in the codFL contract optimization, i.e.,*

$$\pi_k S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c) - C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c) + S_l(\boldsymbol{\delta}^l) - C_l(\boldsymbol{\varrho}^l) \geq 0, \quad \forall k \in \mathcal{K}. \tag{9}$$

Definition 2. *IC constraint: The MAP with true type $\pi_k, k \in \mathcal{K}$, will choose a contract designed for its current type π_k rather than with another type π_{k^*} to maximize its utility, i.e.,*

$$\pi_k S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c) - C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c) \geq \pi_k S_c(\boldsymbol{\phi}_{k^*}, \boldsymbol{\delta}_{k^*}^c) - C_c(\boldsymbol{\phi}_{k^*}, \boldsymbol{\varrho}_{k^*}^c, \boldsymbol{\delta}_{k^*}^c), k \neq k^*, \forall k, k^* \in \mathcal{K}. \tag{10}$$

Based on the above IR and IC constraint definitions, the codFL contract optimization problem $(\mathbf{P}_2)$ which can maximize the expected individual utility for MU-i via the MAP (under the MAP's constraints and the MUs' limited computing resources) can be expressed by

$$(\mathbf{P}_2) \quad \max_{\boldsymbol{\delta}^c, \boldsymbol{\varrho}^c, \boldsymbol{\delta}^l, \boldsymbol{\varrho}^l} \psi_i(\boldsymbol{\delta}^c, \boldsymbol{\varrho}^c, \boldsymbol{\delta}^l, \boldsymbol{\varrho}^l), \forall i \in \mathcal{I}, \tag{11}$$

$$\text{s.t.} \quad \sum_{i \in \mathcal{I}} \hat{\phi}_k^i \delta_{k,i}^c \leq \hat{\delta}_k^c, \forall k \in \mathcal{K}, \tag{12}$$

$$\delta_i^l \leq \hat{\delta}_i^l, \forall i \in \mathcal{I}, \tag{13}$$

$$\hat{\phi}_k^i \delta_{k,i}^c + \delta_i^l \leq \delta_i, \forall k \in \mathcal{K}, \forall i \in \mathcal{I}, \tag{14}$$

$$\pi_k S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c) - C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c) + S_l(\boldsymbol{\delta}^l) - C_l(\boldsymbol{\varrho}^l) \geq 0, \forall k \in \mathcal{K}, \tag{15}$$

$$\pi_k S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c) - C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c) \geq \pi_k S_c(\boldsymbol{\phi}_{k^*}, \boldsymbol{\delta}_{k^*}^c) - C_c(\boldsymbol{\phi}_{k^*}, \boldsymbol{\varrho}_{k^*}^c, \boldsymbol{\delta}_{k^*}^c), k \neq k^*, \forall k, k^* \in \mathcal{K}. \tag{16}$$

IV. MPOA-Based CodFL Contract Solution

Based on $(\mathbf{P}_2)$, it is intractable to solve the problem directly due to complicated IR and IC constraints, especially when the number of MAP's possible types is high. To solve this problem, we can modify the problem into an equivalent problem using a transformation method, i.e., IR and IC constraint reduction. Specifically, we first can show that the MAP's utility is monotonically increasing in its type in Lemma 1.

Lemma 1. *For any feasible contract $(\boldsymbol{\delta}^c, \boldsymbol{\varrho}^c, \boldsymbol{\delta}^l, \boldsymbol{\varrho}^l)$, the MAP's utility must hold the following condition*

$$\pi_k S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c) - C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c) \geq \pi_{k^*} S_c(\boldsymbol{\phi}_{k^*}, \boldsymbol{\delta}_{k^*}^c) - C_c(\boldsymbol{\phi}_{k^*}, \boldsymbol{\varrho}_{k^*}^c, \boldsymbol{\delta}_{k^*}^c), k \neq k^*, \tag{17}$$

where $\pi_k \geq \pi_{k^*}$, $k \neq k^*, k, k^* \in \mathcal{K}$.

Proof. Due to limited space, we briefly prove Lemma 1 as follows. We first prove that if $\pi_k \geq \pi_{k^*}$, then $\boldsymbol{\delta}_k^c \geq \boldsymbol{\delta}_{k^*}^c$, where $k \neq k^*, k, k^* \in \mathcal{K}$. Moreover, if $\boldsymbol{\delta}_k^c \geq \boldsymbol{\delta}_{k^*}^c$, then $\boldsymbol{\varrho}_k^c \geq \boldsymbol{\varrho}_{k^*}^c$. Hence, if $\pi_k \geq \pi_{k^*}$, then the condition (17) hold. □

Using Lemma 1, we then can decrease the number of IR constraints by using the MAP's minimum type, i.e., π_1, such that $\pi_k S_c(\hat{\boldsymbol{\phi}}_k, \boldsymbol{\delta}_k^c) - C_c(\hat{\boldsymbol{\phi}}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c) + S_l(\boldsymbol{\delta}^l) - C_l(\boldsymbol{\varrho}^l) \geq \pi_k S_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\delta}_1^c) - C_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\varrho}_1^c, \boldsymbol{\delta}_1^c) + S_l(\boldsymbol{\delta}^l) - C_l(\boldsymbol{\varrho}^l) \geq \pi_1 S_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\delta}_1^c) - C_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\varrho}_1^c, \boldsymbol{\delta}_1^c) + S_l(\boldsymbol{\delta}^l) - C_l(\boldsymbol{\varrho}^l) \geq 0$. In this case, the IR constraints for other types π_k, where $k > 1$, are satisfied if and only if we can hold the IR constraint for π_1. As a result, we can transform the IR constraints in (15) into

$$\pi_1 S_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\delta}_1^c) - C_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\varrho}_1^c, \boldsymbol{\delta}_1^c) + S_l(\boldsymbol{\delta}^l) - C_l(\boldsymbol{\varrho}^l) \geq 0. \tag{18}$$

Additionally, we can reduce the number of IC constraints in (16) using the following transformation in Lemma 2.

Lemma 2. *The IC constraints in (16) of $(\mathbf{P}_2)$ can be modified into the local downward incentive constraints (LDIC) by*

$$\pi_k S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c) - C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c) \geq \pi_k S_c(\boldsymbol{\phi}_{k-1}, \boldsymbol{\delta}_{k-1}^c) - C_c(\boldsymbol{\phi}_{k-1}, \boldsymbol{\varrho}_{k-1}^c, \boldsymbol{\delta}_{k-1}^c), \forall k \in \{2, \ldots, K\}, \tag{19}$$

where $\boldsymbol{\delta}_k^c \geq \boldsymbol{\delta}_{k-1}^c, \forall k \in \{2, \ldots, K\}$.

Proof. Due to limited space, we briefly prove Lemma 2 as follows. Specifically, we first specify local downward ICs (LDICs) and local upward ICs (LUICs). Then, using some derivations, we can obtain that $\pi_{k+1} S_c(\hat{\boldsymbol{\phi}}_{k+1}, \boldsymbol{\delta}_{k+1}^c) - C_c(\hat{\boldsymbol{\phi}}_{k+1}, \boldsymbol{\varrho}_{k+1}^c, \boldsymbol{\delta}_{k+1}^c) \geq \pi_{k+1} S_c(\hat{\boldsymbol{\phi}}_{k-1}, \boldsymbol{\delta}_{k-1}^c) - C_c(\hat{\boldsymbol{\phi}}_{k-1}, \boldsymbol{\varrho}_{k-1}^c, \boldsymbol{\delta}_{k-1}^c) \geq \ldots \geq \pi_{k+1} S_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\delta}_1^c) - C_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\varrho}_1^c, \boldsymbol{\delta}_1^c), \forall k \in \{1, \ldots, K-1\}$. □

Equations (19) specifies that if the IC constraint for type π_{k-1} holds, then all other IC constraints are also satisfied as long as the conditions $\boldsymbol{\delta}_k^c \geq \boldsymbol{\delta}_{k-1}^c, \forall k \in \{2, \ldots, K\}$, hold. To this end, using (18) and (19), we can transform the optimization problem $(\mathbf{P}_2)$ into the problem $(\mathbf{P}_3)$ as follows:

$$(\mathbf{P}_3) \quad \max_{\boldsymbol{\delta}^c, \boldsymbol{\varrho}^c, \boldsymbol{\delta}^l, \boldsymbol{\varrho}^l} \psi_i(\boldsymbol{\delta}^c, \boldsymbol{\varrho}^c, \boldsymbol{\delta}^l, \boldsymbol{\varrho}^l), \forall i \in \mathcal{I}, \tag{20}$$

s.t. (12)-(14), and,

$$\pi_1 S_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\delta}_1^c) - C_c(\hat{\boldsymbol{\phi}}_1, \boldsymbol{\varrho}_1^c, \boldsymbol{\delta}_1^c) + S_l(\boldsymbol{\delta}^l) - C_l(\boldsymbol{\varrho}^l) \geq 0, \tag{21}$$

$$\pi_k S_c(\boldsymbol{\phi}_k, \boldsymbol{\delta}_k^c) - C_c(\boldsymbol{\phi}_k, \boldsymbol{\varrho}_k^c, \boldsymbol{\delta}_k^c) \geq \pi_k S_c(\boldsymbol{\phi}_{k-1}, \boldsymbol{\delta}_{k-1}^c) - C_c(\boldsymbol{\phi}_{k-1}, \boldsymbol{\varrho}_{k-1}^c, \boldsymbol{\delta}_{k-1}^c), \forall k \in \{2, \ldots, K\}, \tag{22}$$

$$\boldsymbol{\delta}_k^c \geq \boldsymbol{\delta}_{k-1}^c, \forall k \in \{2, \ldots, K\}. \tag{23}$$

To obtain optimal contracts $\left(\hat{\boldsymbol{\delta}}^c, \hat{\boldsymbol{\varrho}}^c, \hat{\boldsymbol{\delta}}^l, \hat{\boldsymbol{\varrho}}^l\right)$ from problem $(\mathbf{P}_3)$, we implement an iterative process as shown in Algorithm 1. Particularly, the optimal values of $\hat{\boldsymbol{\phi}}$ which maximize $(\mathbf{P}_3)$ are required to be found first. Using the iterative algorithm, the MAP can then update contract for each MU-i iteratively considering that other MUs' current contracts remain fixed [17], aiming at maximizing the objective function in $(\mathbf{P}_3)$ at each iteration. The above process terminates when the gaps between the expected utilities of MU-i, $\forall i \in \mathcal{I}$, at previous and current iterations are equal or less than the optimality tolerance κ. In this way, the algorithm converges and the equilibrium contract solution can be found. Alternatively, considering the equilibrium contract solution of other MUs $\left(\hat{\boldsymbol{\delta}}^c_{-i}, \hat{\boldsymbol{\varrho}}^c_{-i}, \hat{\boldsymbol{\delta}}^l_{-i}, \hat{\boldsymbol{\varrho}}^l_{-i}\right)$, the expected utility of the MU-i at the equilibrium contract solution $\left(\hat{\boldsymbol{\delta}}^c_{i}, \hat{\boldsymbol{\varrho}}^c_{i}, \hat{\boldsymbol{\delta}}^l_{i}, \hat{\boldsymbol{\varrho}}^l_{i}\right)$ will produce the highest value compared with the ones using all other contract solutions. This condition is formally described in Definition 3.

Definition 3. *Equilibrium contract solution for* $(\mathbf{P}_3)$*: The optimal contracts* $\left(\hat{\boldsymbol{\delta}}^c, \hat{\boldsymbol{\varrho}}^c, \hat{\boldsymbol{\delta}}^l, \hat{\boldsymbol{\varrho}}^l\right)$ *are the equilibrium solution of the* $(\mathbf{P}_3)$ *if and only if the conditions*

$$\psi_i\left(\hat{\boldsymbol{\delta}}_i, \hat{\boldsymbol{\varrho}}_i, \hat{\boldsymbol{\delta}}_{-i}, \hat{\boldsymbol{\varrho}}_{-i}\right) \geq \psi_i\left(\boldsymbol{\delta}_i, \boldsymbol{\varrho}_i, \hat{\boldsymbol{\delta}}_{-i}, \hat{\boldsymbol{\varrho}}_{-i}\right), \tag{24}$$

$\forall i \in \mathcal{I}$*, are satisfied and the optimal contracts* $\left(\hat{\boldsymbol{\delta}}^c, \hat{\boldsymbol{\varrho}}^c, \hat{\boldsymbol{\delta}}^l, \hat{\boldsymbol{\varrho}}^l\right)$ *still hold the constraints (21)-(23), where* $\hat{\boldsymbol{\delta}}_i = \left(\hat{\boldsymbol{\delta}}^c_i, \hat{\boldsymbol{\delta}}^l_i\right)$, $\boldsymbol{\delta}_i = \left(\boldsymbol{\delta}^c_i, \boldsymbol{\delta}^l_i\right)$, $\hat{\boldsymbol{\varrho}}_i = \left(\hat{\boldsymbol{\varrho}}^c_i, \hat{\boldsymbol{\varrho}}^l_i\right)$, $\boldsymbol{\varrho}_i = \left(\boldsymbol{\varrho}^c_i, \boldsymbol{\varrho}^l_i\right)$, $\hat{\boldsymbol{\delta}}_{-i} = \left(\hat{\boldsymbol{\delta}}^c_{-i}, \hat{\boldsymbol{\delta}}^l_{-i}\right)$, *and* $\hat{\boldsymbol{\varrho}}_{-i} = \left(\hat{\boldsymbol{\varrho}}^c_{-i}, \hat{\boldsymbol{\varrho}}^l_{-i}\right)$.

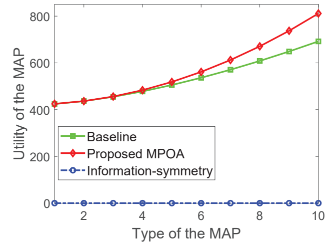
Algorithm 1 CodFL Contract Algorithm

Initialize iteration $\tau = 0$ and κ
Set $\boldsymbol{\delta}^{(\tau)}_i = \left(\boldsymbol{\delta}^{c,(\tau)}_i, \boldsymbol{\delta}^{l,(\tau)}_i\right)$, $\boldsymbol{\varrho}^{(\tau)}_i = \left(\boldsymbol{\varrho}^{c,(\tau)}_i, \boldsymbol{\varrho}^{l,(\tau)}_i\right)$, $\boldsymbol{\delta}^{(\tau)}_{-i} = \left(\boldsymbol{\delta}^{c,(\tau)}_{-i}, \boldsymbol{\delta}^{l,(\tau)}_{-i}\right)$, and $\boldsymbol{\varrho}^{(\tau)}_{-i} = \left(\boldsymbol{\varrho}^{c,(\tau)}_{-i}, \boldsymbol{\varrho}^{l,(\tau)}_{-i}\right)$
All MUs in $\mathcal{I}$ send initial contracts $\left(\boldsymbol{\delta}^{(\tau)}_i, \boldsymbol{\varrho}^{(\tau)}_i\right)$ to the MAP
repeat
  Find $\hat{\boldsymbol{\phi}}^{(\tau)}$ values which maximize $(\mathbf{P}_1)$ using $\left(\boldsymbol{\delta}^{(\tau)}_i, \boldsymbol{\varrho}^{(\tau)}_i\right), \forall i \in \mathcal{I}$
  for $\forall i \in \mathcal{I}$ **do**
    Find the new contract $\left(\boldsymbol{\delta}^{\text{new}}_i, \boldsymbol{\varrho}^{\text{new}}_i\right)$, which maximizes $(\mathbf{P}_3)$ given $\hat{\boldsymbol{\phi}}^{(\tau)}$ and $\left(\boldsymbol{\delta}^{(\tau)}_{-i}, \boldsymbol{\varrho}^{(\tau)}_{-i}\right)$
    if $\left[\psi_i\left(\boldsymbol{\delta}^{\text{new}}_i, \boldsymbol{\varrho}^{\text{new}}_i, \boldsymbol{\delta}^{(\tau)}_{-i}, \boldsymbol{\varrho}^{(\tau)}_{-i}\right) - \psi_i\left(\boldsymbol{\delta}^{(\tau)}_i, \boldsymbol{\varrho}^{(\tau)}_i, \boldsymbol{\delta}^{(\tau)}_{-i}, \boldsymbol{\varrho}^{(\tau)}_{-i}\right)\right] > \kappa$ **then**
      Set $\left(\boldsymbol{\delta}^{(\tau+1)}_i, \boldsymbol{\varrho}^{(\tau+1)}_i\right) = \left(\boldsymbol{\delta}^{\text{new}}_i, \boldsymbol{\varrho}^{\text{new}}_i\right)$
    else
      Set $\left(\boldsymbol{\delta}^{(\tau+1)}_i, \boldsymbol{\varrho}^{(\tau+1)}_i\right) = \left(\boldsymbol{\delta}^{(\tau)}_i, \boldsymbol{\varrho}^{(\tau)}_i\right)$
    end if
  end for
  $\tau = \tau + 1$
until $\psi_i\left(\boldsymbol{\delta}^{(\tau)}_i, \boldsymbol{\varrho}^{(\tau)}_i, \boldsymbol{\delta}^{(\tau)}_{-i}, \boldsymbol{\varrho}^{(\tau)}_{-i}\right), \forall i \in \mathcal{I}$, remain unchanged
Obtain optimal contracts $\left(\hat{\boldsymbol{\delta}}^c, \hat{\boldsymbol{\varrho}}^c, \hat{\boldsymbol{\delta}}^l, \hat{\boldsymbol{\varrho}}^l\right)$

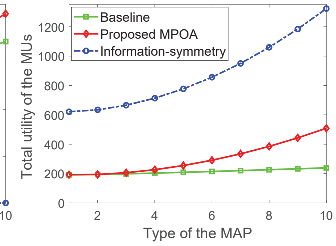
V. Performance Evaluation

A. Simulation Setup

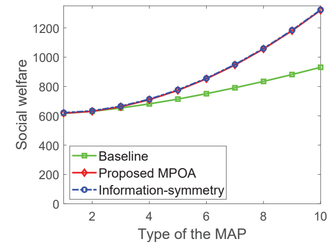
For performance evaluation, we use 10 types of the MAP with uniform distribution of the types. We set $\hat{\delta}^c_k = 5 \times 10^5$ samples. We also define α_c and α_l at 0.125 and 3, respectively, and γ_c and γ_l at 0.001 and 0.005, respectively, to show that the unit payment for local training is higher than that for coded training at the MAP (since the MUs help the MAP to complete its FL tasks). We denote $\sigma = 1$ and $\upsilon = 0.0001$. We use $\xi_i = 0.5 \times 10^{-26}$ [11], and $f_i = 2$GHz, $\forall i \in \mathcal{I}$. We then compare our proposed method, i.e., MPOA, with baseline and information-symmetry methods. For the baseline method,



(a) Utility of the MAP



(b) Total utility of the MUs



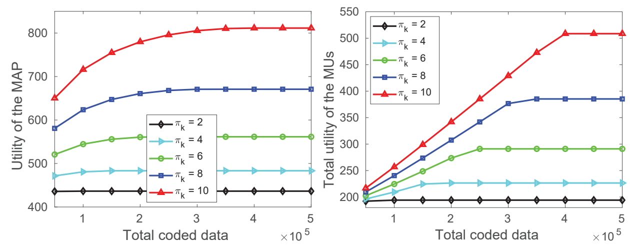
(c) Social welfare

Fig. 2: The codFL contract performance for various methods.

each MU can offer the proportional amount of coded dataset to the MAP (with respect to the MAP's available computing resource) without using contract policy. Meanwhile, for the information-symmetry method, each MU is assumed to completely know the MAP's true type and other MUs' contracts to obtain the optimal contract policy (which is considered as the upper bound solution).

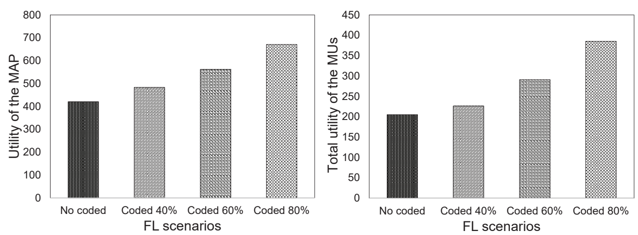
B. Simulation Results

We first compare the MAP's utility of the proposed mechanism with those of the baseline and information-symmetry methods. As shown in Fig. 2(a), the proposed method can achieve the highest utility for all types of the MAP. In particular, the proposed method can produce higher utility (up to 17%) than that of the baseline method. The reason is that the baseline method cannot optimize the coded dataset proportion from the MAP since the MAP can only collect the proportional amount of coded datasets from the participating MUs under the MAP's current available computing resource. For the information-symmetry method, it suffers from zero utility for all types of the MAP since all participating MUs can collaborate together to obtain maximum utility by fully perceiving the current true type, i.e., the available computing resource, of the MAP. As a result, the information-symmetry method can maximize the total utility of participating MUs as shown in Fig. 2(b). To this end, the proposed method can still obtain higher total utility of participating MUs up to 113% compared with that of the baseline method. Although the information-symmetry can obtain the total utility of the MUs up to 2.6 times higher than that of the proposed method, our proposed method can achieve the social welfare within 0.57% as close as that obtained by the information-symmetry method (referred to as the upper-bound solution) as observed in Fig. 2(c). Moreover, the proposed method can obtain the social welfare up to 42% higher than that of the baseline method. To this end, we can conclude that our proposed method is highly economic-effective to implement in the codFL process by balancing the utility performance of the MAP and participating MUs.



(a) Utility of the MAP (b) Total utility of the MUs

Fig. 3: The utility performance when the total offloaded coded dataset increases for various types of the MAP.



(a) Utility of the MAP (b) Total utility of the MUs

Fig. 4: The utility performance for various FL scenarios.

We then observe the utility performances of proposed method in Fig. 3 when the total offloaded coded dataset in the FL process increases for various types of the MAP. In particular, when the MAP has low types, i.e., type 2 and 4, the utilities of the MAP and all the participating MUs will converge to low utility values for most of the total coded dataset scenarios due to small computing resources of the MAP to train the coded dataset. Meanwhile, when the MAP has high types, i.e., type 8 and 10, both the MAP and the participating MUs can improve their utilities and reach much higher convergence values up to 86% and 126%, respectively. Moreover, we present Fig. 4 to further show that more offloaded coded dataset can improve utilities for both the MAP and MUs under sufficient computing resource of the MAP with a certain type, e.g., type 10 in this case. Specifically, the codFL scenarios when 40%, 60%, and 80% local data are coded and trained at the MAP, i.e., coded 40%, 60%, and 80% scenarios, can achieve higher utilities of the MAP and the MUs up to 60% and 88%, respectively, compared with that of the FL scenario without any coded data and coded computing, i.e., no coded scenario. The reason is that the MUs can reduce significant training tasks by uploading the coded data to the MAP for the coded computing, which then leads to better learning quality, especially when some MUs suffer from low computing resources. The above results imply that the MUs can obtain more benefits by contributing more coded data to the MAP and at the same time the MAP can improve its system performance, e.g., highly-accurate global model, while compensating the straggling problem, i.e., long training time.

VI. Conclusion

In this paper, we have proposed an effective codFL-based incentive mechanism using the MPOA contract scheme to maximize the utilities for participating MUs while maintaining the learning quality of the MAP. Particularly, we have first formulated the codFL contract optimization problem and then developed the iterative codFL contract algorithm to find optimal contracts that can achieve the equilibrium solution for all participating MUs. Through numerical results, we have demonstrated that our incentive mechanism can obtain the close performance with the upper bound solution, and enhance the utilities of MUs as well as social welfare significantly compared with the baseline method. Furthermore, the proposed method can stabilize the utility optimizations of all participating MUs and the MAP based on their optimal contracts and optimal coded dataset proportions, respectively.

References

[1] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning: synthesis lectures on artificial intelligence and machine learning," vol. 13, no. 3, pp. 1-207, Dec. 2019.

[2] 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.

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

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

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

[6] S. Dhakal, *et al.*, "Coded federated learning," in *IEEE Globecom Workshops*, Dec. 2019, pp. 1-6.

[7] J. S. Ng, *et al.*, "A survey of coded distributed computing," arXiv:2008.09048 [cs.DC], Aug. 2020.

[8] 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.

[9] 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.

[10] 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.

[11] N. Kim, *et al.*, "Incentive-based coded distributed computing management for latency reduction in IoT services - a game theoretic approach," *IEEE Internet Things J.*, early access, Dec. 2020.

[12] N. Ding, *et al.*, "Incentive mechanism design for distributed coded machine learning", arXiv:2012.08715 [cs.GT], Dec. 2020.

[13] P. Bolton and M. Dewatripont, *Contract Theory*. Cambridge, MA, USA: MIT Press, 2005.

[14] B. D. Bernheim and M. .D Whinston, "Common agency," *Econometrica*, vol. 54, no. 4, pp. 923-942, Jul. 1986.

[15] M. Showkatbakhsh, C. Karakus, and S. Diggavi, "Privacy-utility trade-off of linear regression under random projections and additive noise," in *IEEE ISIT*, Jun. 2018, pp. 186-190.

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

[17] 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.