Incentivize to Build: A Crowdsourcing Framework for Federated Learning

Shashi Raj Pandey[1], Nguyen H. Tran[2], Mehdi Bennis[1,3], Yan Kyaw Tun[1], Zhu Han[1,4], Choong Seon Hong[1]
[1]Department of Computer Science and Engineering, Kyung Hee University, 17104, Rep. of Korea.
[2]School of Computer Science, The University of Sydney, NSW 2006, Australia.
[3]Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland.
[4]Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004-4005 USA.
{shashiraj, ykyawtun7, cshong}@khu.ac.kr, nguyen.tran@sydney.edu.au, mehdi.bennis@oulu.fi, zhan2@uh.edu.

***Abstract*—Federated learning (FL) rests on the notion of training a global model in a decentralized manner. Under this setting, mobile devices perform computations on their local data before uploading the required updates to the central aggregator for improving the global model. However, a key challenge is to maintain communication efficiency (i.e., the number of communications per iteration) when participating clients implement uncoordinated computation strategy during aggregation of model parameters. We formulate a utility maximization problem to tackle this difficulty, and propose a novel crowdsourcing framework, involving a number of participating clients with local training data to leverage FL. We show the incentive-based interaction between the crowdsourcing platform and the participating client's independent strategies for training a global learning model, where each side maximizes its own benefit. We formulate a two-stage Stackelberg game to analyze such scenario and find the game's equilibria. Further, we illustrate the efficacy of our proposed framework with simulation results. Results show that the proposed mechanism outperforms the heuristic approach with up to 22% gain in the offered reward to attain a level of target accuracy.**

***Index Terms*—Decentralized machine learning, federated learning, mobile crowdsourcing, stackelberg game.**

I. Introduction

With dedicated chipsets for Machine Learning (ML) applications and in-built sensors [1]–[2], smartphone makers will achieve a significant market gain by offering cutting edge applications such as an ability to understand user behaviors, more secured facial recognition system and predictive future. This means *on-device intelligence* will be ubiquitous! However, the growing possibilities of on-device intelligence is challenged by data privacy concerns raised in a White House report published on the principle of data minimization [3]. The direct application of this report led to a ML technique in which the training data remain distributed on the mobile devices, called *Federated Learning* [2]–[4]. Unlike the conventional approaches on distributed optimization [5]–[6], this technique eliminates the mobility of distributed training data on local users to build a learning model, and thereby protecting user data privacy. Thus, there exists a huge market potential of untapped private data, and FL is a promising tool to explore more personalized service oriented applications.

The mobile users (participating clients) perform computation for the updates on their local training data with the current global model parameters, which are then aggregated and broadcast-back by the centralized coordinating server. In this way, FL decouples the training process to build a learning model, and this iterative process undergoes until an accuracy level of the learning model is maintained.

The processes of local computations at the devices and their communication with the centralized coordinating server to build a global learning model is interleaved in a complex manner. Therefore, there exist several challenges for having a communication efficient FL framework [7], [8]. Most of the existing works have focused on overcoming the challenges related to convergence properties in training the model [7], [9], and architecture design [8]–[10]. However, the difficulty to ensure cooperation of a number of participating clients in building the learning model, i.e., *how can we motivate a number of participating clients to enable FL?*, is still unaddressed. To tackle these overlooked issues, we design a value-based compensation mechanism for the participating clients, such as a bounty (e.g., data discount package) in the crowdsourcing framework. We characterize the level of participation in the framework by a local accuracy level, i.e., *prediction accuracy* that can protect the learning model against imperfect updates by limiting compromising clients (for instance, clients with the skewed data).

The goal of this paper is two-fold: First, we address the challenge of maintaining communication efficiency while exchanging the model parameters with a number of participating clients during aggregation; we develop a participatory method to enable computation and communication cost effective FL, and its criticality with a crowdsourcing structure. Second, we design an incentive mechanism to *reveal the iteration strategy* of the mobile clients to perform FL for improving the global model.

The remainder of this paper is organized as follows. We present the system model of our proposed crowdsourcing framework in Section II. In Section III, we formulate an incentive mechanism for the participating clients in the decentral-

The research is partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing), US MURI AFOSR MURI 18RT0073, NSF CNS-1717454, CNS- 1731424, CNS-1702850, CNS-1646607. Dr. CS Hong is the corresponding author.

ized learning framework with a two-stage Stackelberg game, and investigate the Nash equilibrium of the game. Numerical results are presented in Section IV. Finally, conclusions are drawn in Section V.

II. System Model

We consider a set of participating clients $\mathcal{K} = \{1, 2, \dots, K\}$, associated with a base station with a coordinating server in the crowdsourcing framework. The crowdsourcer (referred to as the Multi-access Edge Computing (MEC) server hereafter) can interact with mobile clients via an application interface, and aims at leveraging FL to build a global ML model with distributed local training data. The interaction is instantiated by an incentive mechanism to train a global model with an accuracy level.

A. Federated Learning Background

In a typical FL setting [4], within each global iteration, each participating client will iterate over its data samples to solve the local subproblem and produces the *model parameter vector*, commonly known as *weights* in ML. These local parameters are aggregated at the MEC and the resulting global parameter is broadcast-back to all the participating clients for the next global iteration. This iterative process continues until a global accuracy ϵ is obtained. For the sake of brevity, in this work, we will exclude the common details of FL algorithm (also briefly defined in our earlier work [10]), and rather focus on the constructed model with the perspective of economics and incentive mechanism design for FL participating clients.

Each participating client can use arbitrary optimization algorithm such as Stochastic Gradient Descent (SGD) or Stochastic Variance Reduced Gradient (SVRG) [9], and take multiple *local iterations* to attain a common threshold relative accuracy θ on local subproblem when solving the global learning problem for ϵ accuracy. Then, according to [5], a general upper-bound on the number of global iterations for strongly convex learning objectives is defined as

$$I^{\mathrm{g}}(\epsilon, \theta) = \frac{\zeta \cdot \log\left(\frac{1}{\epsilon}\right)}{1 - \theta}. \tag{1}$$

Here, we have replaced $\mathcal{O}(\log\left(\frac{1}{\epsilon}\right))$ in the numerator with $\zeta \cdot \log\left(\frac{1}{\epsilon}\right)$ for a constant $\zeta > 0$. For the fixed iterations I^{g}, we observe in (1) that a very high local accuracy (small θ) can significantly improve the accuracy ϵ. However, each client k has to spend excessive resources in terms of local iterations, I_k^{l} to attain a small θ_k accuracy, which is upper bounded for a wide range of iterative algorithms [5] as

$$I_k^{\mathrm{l}}(\theta_k) = \gamma_k \log\left(\frac{1}{\theta_k}\right), \tag{2}$$

where $\gamma_k > 0$ is a constant that depends on the parametric choice of client k based upon the data size and condition number of the local subproblem [11]. Further, for a given ϵ, larger θ_k exerts more global iterations. Therefore, to strike this discomfort, in each global iteration the MEC server can incentivize the participating clients to increase the number of local iterations for small θ_k accuracy (more accurate). In this scenario, the corresponding performance bound (1) for the heterogeneous responses θ_k can be modified by considering the worst case response of the participating client as

$$I^{\mathrm{g}}(\epsilon, \theta) = \frac{\zeta \cdot \log\left(\frac{1}{\epsilon}\right)}{1 - \max_k \theta_k}, \forall k \in \mathcal{K}. \tag{3}$$

In the following section, we will further discuss in details about the proposed incentive mechanism, and present the interaction between MEC server and participating clients as a two-stage Stackelberg game.

B. Cost Model

Training on local data for a defined accuracy level incurs two typical costs for the participating clients: the computing cost and the communication cost.

Computing cost: In a single round of communication with the MEC, each participating client strategically[1] iterates over its local data to a relative accuracy θ_k. With (2), we define the computing cost for client k.

Communication cost: This cost is incurred when client k interacts with MEC server for parameter updates to maintain θ_k accuracy. Intuitively, larger θ_k will induce more rounds of interaction between with the server until global convergence, as formalized in (1).

Thus, with the inverse relation of global iteration upon local relative accuracy in (1), we can characterize the total communication cost as

$$T(\theta_k) = \frac{T_k}{(1 - \theta_k)}, \tag{4}$$

where T_k as the time required for the client k to communicate with MEC server in each round of the model's parameter exchanges. Using first-order Taylor's approximation, we can approximate the total communication cost as $T(\theta_k) = T_k \cdot (1 + \theta_k)$. We assume that clients are allocated orthogonal sub-channels so that there is no interference between them[2]. Thus, having evaluated T_k, the increase in value of θ_k (poor local accuracy) will contribute for a larger communication cost.

Therefore, the participating client k's cost for the relative accuracy level θ_k on the local subproblem is

$$C_k(\theta_k) = (1 + \theta_k) \cdot \left(\nu_k \cdot T_k + (1 - \nu_k) \cdot \gamma_k \log\left(\frac{1}{\theta_k}\right)\right), \tag{5}$$

where $0 \leq \nu_k \leq 1$ is the normalized monetary weight for communication and computing costs (i.e., \$/ rounds of iteration). A **smaller value of relative accuracy θ_k indicates a higher local accuracy**. Thus, there exists a trade-off between the communication and the computing cost (5). A participating client can adjust its preference on each of these costs with weight metric ν_k.

[1] Fewer iterations might not be sufficient to have an optimal solution [12].

[2] Note that the scenario of possible delay introduced can be mitigated by adjusting maximum waiting time as in [13] at MEC.

III. Incentive Mechanism for the Client's Participation in the Decentralized Learning Framework

In this section, firstly, we present our motivation to realize the concept of FL by employing a crowdsourcing framework. We next advocate for an effective incentive mechanism required to realize this setting of the decentralized learning model with our proposed solution approach.

A. *Incentive Mechanism: A Two-Stage Stackelberg Game Approach*

The interaction scenario between MEC server and the participating clients can be realized with a Stackelberg game approach. We formulate our problem as a two-stage Stackelberg game between the MEC server (leader) and participating clients (followers).

Clients (Stage II): The MEC server is a leader with the first-move advantage in influencing the followers to participate for the local consensus accuracy. At first, it will announce a uniform reward rate[3] (e.g., a fair data package discount as $/accuracy level) $r > 0$ for the participating clients. Given r, at Stage II, a rational client k will try to improve the local model's accuracy for maximizing its net utility by training over the local data with global parameters. Note that the proposed utility model incorporates the upper-bound cost for the participating clients.

Client Utility Model: We use a valuation function $v_k(\theta_k)$ that characterizes the participating clients' economic return on response θ_k in the crowdsourcing model for FL.

Assumption 1. The valuation function $v_k(\theta_k)$ is a linear, decreasing function with $\theta_k > 0$, i.e., $v_k(\theta_k) = (1 - \theta_k)$.

Given $r > 0$, each participating client k's strategy is to maximize its own utility as follows:

$$\max_{0\leq\theta_k\leq 1} \qquad u_k(r,\theta_k) = r(1-\theta_k) - C_k(\theta_k), \forall k \in \mathcal{K}. \quad (6)$$

Also, we have $C_k^{''}(\theta_k) > 0$, which means $C_k(\theta_k)$ is a strictly convex function. Thus, there exists a unique solution $\theta_k^*(r), \forall k$.

MEC Server(Stage I): Knowing the response (strategy) of the participating clients, the MEC can evaluate an optimal reward rate r^* to maximize its utility. The utility $U(\cdot)$ of MEC server can be defined in relation to the satisfaction measure attained with respect to accuracy.

MEC Server Utility Model: We define $x(\epsilon)$ as the number of iterations required for an arbitrary algorithm to converge to some ϵ accuracy. We similarly define $I^{\text{g}}(\epsilon,\theta)$ as global iterations of the framework to reach a relative θ accuracy on the local subproblems.

From this perspective, we require an appropriate utility function $U(\cdot)$ as the satisfaction measure of the framework in reference to ϵ accuracy. We use the definition of the number of iterations for ϵ accuracy in a device as

$$x(\epsilon) = \zeta \cdot \log\left(\frac{1}{\epsilon}\right).$$

Due to large values of iterations, we approximate $x(\epsilon)$ as a continuous value, and with aforementioned relation, we choose $U(\cdot)$ as a strictly concave function of $x(\epsilon)$ for $\epsilon \in [0,1]$ i.e., with the increase in $x(\epsilon)$, $U(\cdot)$ also increases. Thus, we propose $U(x(\epsilon))$ as the normalized utility function bounded within $[0,1]$ as

$$U(x(\epsilon)) = 1 - 10^{-(ax(\epsilon)+b)}, \quad a \geq 0, b \leq 0, \quad (7)$$

which is strictly increasing concave function of $x(\epsilon)$ with design parameters a and b.

As for the global model, there exists an acceptable value of threshold accuracy correspondingly reflected by $x_{\min}(\epsilon)$. This suggests the possibility of near-zero utility for MEC server for failing to attain such value.

Furthermore, in our setting, $I^{\text{g}}(\epsilon,\theta)$ can be elaborated with a upper bound (maximum global iterations, δ) as

$$I^{\text{g}}(\epsilon,\theta) = \frac{x(\epsilon)}{1-\theta} \leq \delta. \quad (8)$$

(8) explains the efficiency paradigm of the proposed framework in terms of time required for the convergence to some accuracy ϵ. If $\tau^{\text{l}}(\theta)$ is the time per iteration to reach a relative θ accuracy at a local subproblem and $T(\theta)$ is the communication time required during a single iteration for any arbitrary algorithm, then we can analyze the result in (8) with the efficiency of the global model as

$$I^{\text{g}}(\epsilon,\theta) \cdot (T(\theta) + \tau^{\text{l}}(\theta)). \quad (9)$$

Because the cost of communication is proportional to the speed and energy consumption in a distributed scenario [6], the bound defined in (8) explains the efficiency in terms of MEC server's resource restriction for attaining ϵ accuracy.

Therefore, the utility of the MEC server can be defined for the set of measured best responses $\boldsymbol{\theta}^*$ as

$$\mathcal{U}(x(\epsilon), r|\boldsymbol{\theta}^*) = \beta\left(1 - 10^{-(ax(\epsilon)+b)}\right) - r\sum_{k\in\mathcal{K}}(1-\theta_k^*(r)),$$

where $\beta > 0$ is the system parameter [4], and $r\sum_{k\in\mathcal{K}}(1-\theta_k^*(r))$ is the cost spent for incentivizing participating clients in the crowdsourcing framework for FL. Thus, for the measured $\boldsymbol{\theta}^*$, the utility maximization problem can be formulated as follows:

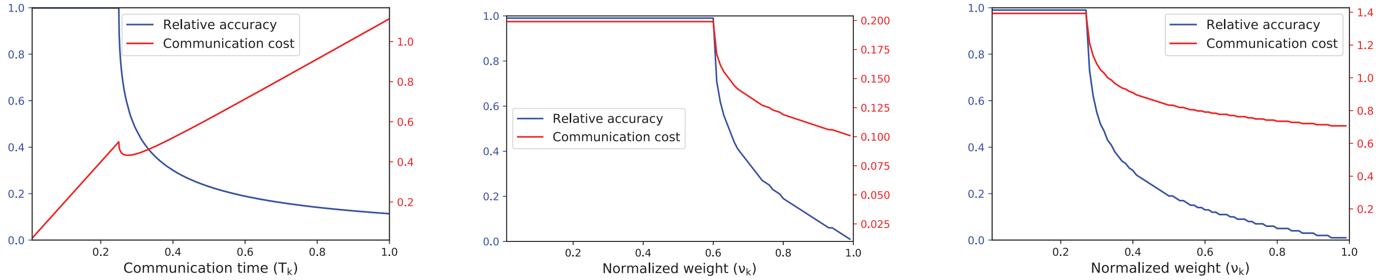
$$\max_{r\geq 0, x(\epsilon)} \qquad \mathcal{U}(x(\epsilon), r|\boldsymbol{\theta}^*), \quad (10)$$

$$\text{s.t.} \qquad \frac{x(\epsilon)}{1-\max_k \theta_k^*(r)} \leq \delta. \quad (11)$$

In constraint (11), $\max_k \theta_k^*(r)$ characterizes the worst case response for the server side utility maximization problem. Here, we can observe that with more accurate local solutions, the MEC server can attain better utility due to corresponding increment in value of $x(\epsilon)$.

[3] In this work, we have considered a uniform pricing scheme, which is meaningful in terms of fairness. Whereas, a differentiated pricing scheme is more complicated to design, and further requires more personal information of the clients.

[4] Note that $\beta > 0$ characterizes a linear scaling metric to the utility function which can be set arbitrarily and will not alter our evaluation.



(a)

(b)

(c)

Fig. 1: Solution Analysis (15) (Left Y-axis: Relative accuracy, Right Y-axis: Communication cost): (a) impact of communication adversity on local relative accuracy for a constant reward (b) normalized weight versus relative accuracy for a fair data rate (quality communication channel), and (c) normalized weight versus relative accuracy for an expensive data rate (poor communication channel).

Lemma 1. *The optimal solution $x^*(\epsilon)$ for (10) can be derived as $\delta(1-\max_k\ \theta_k^*(r))$.*

Proof: (by analyzing the KKT condition) Omitted due to limited space. ■

Therefore, for a given response $\theta^*(r)$, we can formalize (10) as

$$\max_{r\geq 0} \qquad \beta\left(1-10^{-(ax^*(\epsilon)+b)}\right)-r\sum_{k\in\mathcal{K}}(1-\theta_k^*(r)). \tag{12}$$

Stackelberg Equilibrium. With a solution to MEC server's utility maximization problem r^*, we have the following definition.

Definition 1. For any values of r, and $\boldsymbol{\theta}$, $(r^*,\boldsymbol{\theta}^*)$ is a Stackelberg equilibrium if the following conditions are satisfied:

$$\mathcal{U}(r^*,\boldsymbol{\theta}^*)\geq\mathcal{U}(r,\boldsymbol{\theta}^*), \tag{13}$$

$$u_k(\theta_k^*,r^*)\geq u_k(\theta_k,r^*),\ \ \forall k. \tag{14}$$

Next, we analyze the Stackelberg equilibria with the backward-induction method: the Stage-II problem is solved at first to obtain $\boldsymbol{\theta}^*$, which is then used for solving the Stage-I problem to obtain r^*.

B. Stackelberg Equilibrium: Algorithm and Solution Approach

With measured responses $\boldsymbol{\theta}$ from the participating clients, the server can design a proper incentive plan to improve the global model while maintaining the worst case relative accuracy $\max_k\ \theta_k^*$ as θ_{th}, which is the consensus on the maximum local accuracy. Note that the threshold accuracy θ_{th} can be adjusted by the MEC server as a client selection criteria [see Remark 1.] to improve the overall performance of the system.

For a scenario, where the offered reward r for the client k is insufficient to motivate it for participation with improved local relative accuracy, we might have $\max_k\ \theta_k^*(r)=1$ i.e., $\theta_{\text{th}}=1$, no participation.

Lemma 2. *For a given reward rate r, and T_k which is determined based upon the channel conditions, we have the unique solution $\theta_k^*(r)$ for the participating client k satisfying following relation:*

$$g_k(r)=\log\left(e^{1/\theta_k^*(r)}\theta_k^*(r)\right),\forall k\in\mathcal{K}, \tag{15}$$

for $g_k(r)\geq 1$, where

$$g_k(r)=\left[\frac{r+\nu_k T_k}{(1-\nu_k)\gamma_k}-1\right].$$

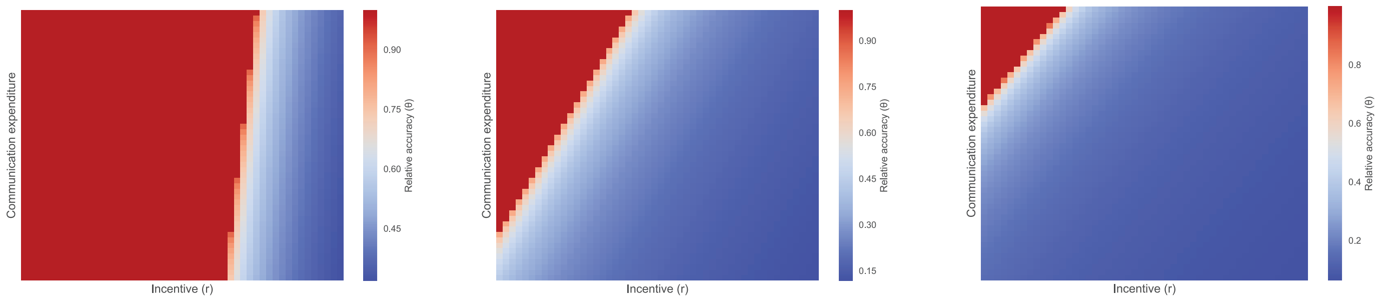
Proof: (by first-order condition on (6)) Omitted due to limited space. ■

We can therefore characterize the participating client k's best response under the proposed framework as

$$\theta_k^*(r)=\min\left\{\hat{\theta}_k(r)\mid_{g_k(r)=\log\left(e^{1/\hat{\theta}_k(r)}\hat{\theta}_k(r)\right)},\theta_{\text{th}}\right\},\forall k. \tag{16}$$

(16) represents the best response for the participating client k under our proposed framework.

In Fig. 1, we briefly present the solution analysis to (15) with the impact of channel condition (we define it as communication adversity) on the local relative accuracy for constant reward. For this, in Fig. 1a, we consider a participating client with the fixed offered reward setting r from uniformly distributed values of 0.1 to 5. We use the normalized T_k parameter for a client k to illustrate the response analysis scenario. In Fig. 1b and Fig. 1c, T_k is uniformly distributed on [0.1, 1], and ν_k is set at 0.6. Intuitively, as in Fig. 1a, the increase in communication time T_k for a fixed reward r will influence participating clients to iterate more locally for improving local accuracy than to rely upon the global model, which will minimize their total cost. However, the trend is significantly affected by normalized weights ν_k, as observed in Fig. 1b and Fig. 1c. For a larger value of T_k (poor channel condition) as in the case of Fig. 1c, increasing the value of ν_k, i.e., clients with more preference on the communication cost in the total cost model results to higher local iterations for solving local subproblems, as reflected by the better local accuracy, unlike in Fig. 1b. In both cases we observe the decrease in communication cost upon participation. However, in Fig.1c the communication cost is higher because of an expensive data



(a) (b) (c)

Fig. 2: Case Study: impact of communication cost and offered reward rate r for different values of normalized weight (preferences) ν_k that defines the client's categories as (a) Reluctant, $\nu_k = 0.1$ (b) Rational, $\nu_k = 0.5$ and (c) Sensitive, $\nu_k = 0.7$. X-axis shows the increase in incentive (r) value from left-to-right, and the y-axis defines the increase in the value of communication expenditure (top-to-bottom).

rate. Therefore, for a given r, client k can adjust its weight metrics accordingly to improve the response θ_k.

To solve (12) efficiently, we introduce a new variable z_k in relation with the consensus on local relative accuracy θ_{th} as

$$z_k = \begin{cases} 1, & \text{if} \quad r > \ \hat{r}_k, \\ 0, & \text{otherwise}, \end{cases} \tag{17}$$

where

$$\hat{r}_k = \left[g_k^{-1}\left(\log\left(e^{1/\theta_{\text{th}}} \theta_{\text{th}} \right) \right) \right]$$

is the minimum incentive value required obtained from (16) to attain the local consensus accuracy θ_{th} at client k for the defined parameters ν_k and T_k.

This means, $\theta_k(r) < \theta_{\text{th}}$ when $z_k = 1$, and $\theta_{\text{th}} \leq \theta_k(r) < 1$ when $z_k = 0$. MEC server can use this setting to drop the participants with poor accuracy. As discussed before, for the worst case scenario we consider $\theta_{\text{th}} = 1$.

Therefore, the utility maximization problem can be equivalently written as

$$\max_{r, \{z_k\}_{k \in \mathcal{K}}} \quad \beta \left(1 - 10^{-(ax^*(\epsilon)+b)} \right) - r \sum_{k \in K} z_k \cdot (1 - \theta_k^*(r)), \tag{18}$$

$$\text{s.t.} \quad r \geq 0, \tag{19}$$

$$z_k \in \{0, 1\}, \forall k. \tag{20}$$

The problem (18) is a mixed-boolean programming that may require exponential-complexity effort (i.e., 2^K configuration of $\{z_k\}_{k \in \mathcal{K}}$) to solve by the exhaustive search. To solve this problem with linear complexity, we refer to the solution approach as in Algorithm 1.

Proposition 1. *Algorithm 1 can solve the Stage-I equivalent problem (12) with linear complexity.*

Proof: As the clients are sorted in the order of increasing $\hat{r}_k$ (line 1), for sufficient condition $r > \hat{r}_k$ resulting $z_k = 1$, the MEC's utility maximization problem reduces to a single-variable problem that can be solved using popular numerical methods.

Algorithm 1 MEC Server's Utility Maximization

```
1: Sort clients as with $\hat{r}_1 < \hat{r}_2 < \ldots < \hat{r}_K$
2: $\mathcal{R} = \{\}, \mathcal{A} = \mathcal{K}, j = K$
3: while $j > 0$ do
4:     Obtain the solutions $r_j$ to the problem
           $\max_{r \geq \hat{r}_1} \beta \left( 1 - 10^{-(ax^*(\epsilon)+b)} \right) - r \sum_{k \in \mathcal{A}} (1 - \theta_k^*(r))$.
5:     if $r_j > \hat{r}_j$, then $\mathcal{R} = \mathcal{R} \cup \{r_j\}$;
6:     end if
7:     $\mathcal{A} = \mathcal{A} \backslash j$;
8:     $j = j - 1$;
9: end while
10: Return $r_j \in \mathcal{R}$ with the highest optimal values in the
    problem (4)
```

Remark 1. *Algorithm 1 can maintain consensus accuracy by formalizing the clients selection criteria. This is because from (17), $z_k = 1$ for $\theta_k(r) < \theta_{th}$, and $z_k = 0$ for $\theta_{th} \leq \theta_k(r) < 1$. Thus, MEC server uses this setting to drop the participants with $\theta_k(r) > \theta_k^*(r) = \theta_{th}$.*

Theorem 1. *The Stackelberg equilibria of the crowdsourcing framework are the set of pairs $\{r^*, \boldsymbol{\theta}^*\}$.*

Proof: For any given $\boldsymbol{\theta}$, it is obvious that $\mathcal{U}(r^*, \boldsymbol{\theta}) \geq \mathcal{U}(r, \boldsymbol{\theta}), \forall r$ since r^* is the solution to the Stage-I problem. Thus, we have $\mathcal{U}(r^*, \boldsymbol{\theta}^*) \geq \mathcal{U}(r, \boldsymbol{\theta}^*)$. In the similar way, for any given value of r and $\forall k$, we have $u_k(r, \theta_k^*) \geq u_k(r, \theta_k), \forall \theta_k$. Hence, $u_k(r^*, \theta_k^*) \geq u_k(r^*, \theta_k)$. Combining these facts, we conclude the proof being based upon the definitions of (13) and (14).

IV. Numerical Results

In this section, we analyze numerical simulations to illustrate our results. First, we show the optimal solution of Algorithm 1 (ALG. 1) and conduct a comparison of its performance with two baselines. The first one, named OPT, is the optimal solution of the problem (12) with exhaustive search for the optimal response $\boldsymbol{\theta}^*$. The second, called baseline is an offered price that considers the worst response amongst the participating clients to attain local consensus θ_{th} accuracy,

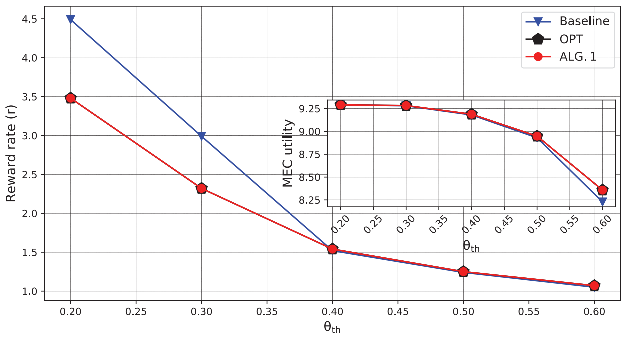


Fig. 3: Comparison of (a) Reward rate and (b) MEC utility, under three schemes for different values of threshold θ_{th} accuracy.

which is inefficient scheme but still enables us attain feasible solutions. Finally, we analyze the system performance by varying different parameters, and conduct a comparison on the incentive mechanism with the baseline and their corresponding utilities. In our analysis, the smaller values of local consensus are of specific interest as it reflects the competence of FL.

1) Settings: For a close understanding and easy interpretation, we fix the number of participating clients as 4, and parameters $\beta = 10$, $\delta = 10$. The MEC's utility $U(\cdot)$ model is defined with parameters $a = 0.3$, and $b = 0$. For each client k, we consider ν_k is uniformly distributed on [0.1, 0.5], which can provide an interesting insight of the system's efficacy as presented in the case study Fig. 2. We also use the normalized value of T_k for all participating clients.

2) Reward Rate: In Fig. 3 we observe a significant increase in reward rate when the value of accuracy level is improved (from 0.4 to 0.2). These results are consistent with the analysis in Section III-B. The cost for attaining higher local accuracy level requires more local iterations, and thus, participating clients exerts more incentive to compensate their cost.

We also examine that the reward variation is prominent for lower values of θ_{th}, and the scheme ALG. 1 and OPT achieves the same performance, while Baseline is not efficient as others. Here, we can observe up to 22% of gain in the offered reward against the Baseline by other two schemes. In Fig.3b, we see the corresponding MEC utilities for the offered reward that complements the competence of proposed ALG. 1. We observe that the trend of utility against the offered reward goes along with our analysis.

3) Client's Response: A Case Study: In Fig. 2, we dig for more details to explore the behaviors of the participating client through the heatmap plot. To explain better, we define three categories of participating clients based upon the value of normalized weight $\nu_k, \forall k$, which is their individual preferences upon the computation cost and the communication expenditure for the convergence of the learning framework. (i) *Reluctant* clients with a lower $\nu_k, \forall k$ consumes more reward to improve local accuracy, even though the value of T_k is larger (expensive), as observed in Fig. 2a. (ii) *Sensitive* clients are more inflicted by the channel quality for larger $\nu_k, \forall k$, and iterates more locally within a round of communication to the MEC server for improving local accuracy, as observed in Fig. 2c. (iii) *Rational* clients, as referred in Fig. 2b tend to balance these extreme preferences (say $\nu_k = 0.5$ for client k), which in fact would be unrealistic to expect all the time due to heterogeneity in participating client's resources.

V. Conclusions

In this paper, we developed and analyzed a novel crowdsourcing framework to enable computation and communication cost efficient FL. An incentive mechanism has been designed to enable the participation of several devices in FL. In particular, we have adopted a two-stage Stackelberg game model to jointly study the utility maximization of the participating clients and MEC server interacting via an application platform for building a high-quality learning model. Further, we derived the best response solution and proved the existence of Stackelberg equilibrium. We have examined characteristics of participating clients for different parametric configurations. Additionally, we have conducted numerical simulations and presented several case studies to evaluate the framework efficacy. For future work, we will consider the scenario without a central coordinating server to enable self-organizing FL amongst participating clients.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, Nov 2011.

[2] "We are making on-device AI ubiquitous," Tech. Rep. [Online]. Available: https://www.qualcomm.com/news/onq/2017/08/16/we-are-making-device-ai-ubiquitous

[3] W. House, "Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy," *White House, Washington, DC*, pp. 1–62, Feb 2012.

[4] B. McMahan and D. Ramage, "Federated learning: Collaborative machine learning without centralized training data," *Google Research Blog*, April 2017.

[5] C. Ma, J. Konečnỳ, M. Jaggi, V. Smith, M. I. Jordan, P. Richtárik, and M. Takáč, "Distributed optimization with arbitrary local solvers," *Optimization Methods and Software*, vol. 32, no. 4, pp. 813–848, Aug 2017.

[6] Y. Bao, Y. Peng, C. Wu, and Z. Li, "Online job scheduling in distributed machine learning clusters," *arXiv preprint arXiv:1801.00936*, Jun 2018.

[7] J. Konecnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, Oct 2016.

[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, Feb 2016.

[9] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, Oct 2016.

[10] N. H. Tran, W. Bao, A. Zomaya, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. of IEEE Conference on Computer Communications (INFOCOM), Paris, France*. IEEE, 2019, pp. 1387–1395.

[11] J. Konečnỳ, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," *Optimization Methods and Software*, vol. 32, no. 5, pp. 993–1005, Feb 2017.

[12] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," *arXiv preprint arXiv:1804.05271*, Aug 2018.

[13] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "On-device federated learning via blockchain and its latency analysis," *arXiv preprint arXiv:1808.03949*, Aug 2018.