Double Insurance: Incentivized Federated Learning with Differential Privacy in Mobile Crowdsensing

Chenhao Ying[†§], Haiming Jin[‡], Xudong Wang[†§], and Yuan Luo[†§]
[†]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
[§]SJTU-China Unicom-Unioncast Big Data Joint Lab, Shanghai Jiao Tong University, Shanghai, 200240, China
[‡]John Hopcroft Center for Computer Science, Shanghai Jiao Tong University, Shanghai, 200240, China
E-mail: {yingchh1565, jinhaiming, dongte, yuanluo@sjtu.edu.cn}

***Abstract*—Exploiting the computing capability of mobile devices with specialized engines (*e.g.,* Neural Engine in iPhone), an attractive paradigm of federated learning that combines the mobile crowdsensing (MCS) has been deeply investigated recently (*e.g.,* Google AI and Nvidia), where the training task is offloaded to the mobile crowd. However, this new paradigm still has numerous problems. Since executing the training task is costly for individual workers, the first problem is how to attract more participants. Following the incentive requirement, the second is how to preserve the workers' bid privacy since the reported costs are usually sensitive. Finally, the third problem is to guarantee the privacy protection on locally training models in the federated learning which involve the private information of local data.**

In this paper, we propose an incentivized federated learning with differential privacy in MCS system, namely, SHIELD, to solve the three significant problems. In fact, SHIELD satisfies the truthfulness and individual rationality while preserving the differential privacy of workers' bids and locally training models. Furthermore, for accuracy, the excess empirical risk of SHIELD is proved to be upper bounded by $\mathcal{O}(\frac{(\ln(Kn_{min}))^{\frac{1}{2}}}{Kn_{min}} + \frac{\ln(Kn_{min})}{K^2 n_{min}^2})$, where a special case for totally distributed scenario leads to a much sharper bound $\mathcal{O}(\frac{\log(n)}{n^2})$ than the latest result $\mathcal{O}(\frac{\ln(mn_{min})}{m^2 n_{min}^2})$. Finally, comparing with the state-of-art approaches, SHIELD illustrates superior performance by numerous experiments in classification and regression tasks.

I. Introduction

With the dramatic development of distributed machine learning, an attractive framework, namely, federated learning, has been proposed [1, 2]. In fact, the remote cloud-based platform selects some distributed servers, and then, each selected server trains a local model by utilizing the local data, respectively, which is then transmitted to the platform for global aggregation. However, the traditional federated learning is implemented by renting the servers from some companies (*e.g.,* Google), which needs expensive cost and is not convenient.

Furthermore, the recent ubiquitous mobile devices which are embedded with plentiful sensors (*e.g.,* GPS, camera) have impelled a newly emerged sensing paradigm, namely, mobile crowd sensing (MCS), where the sensory tasks are executed by the crowd who carry the mobile devices. However, the traditional MCS system, which is only applied to collect the massive sensory data, does not exploit the computing capability of mobile devices brought by their embedded specialized computing engines (*e.g.,* Neural Engine in iPhone).

Therefore, to reduce the implementation cost and improve the efficiency, some new frameworks of federated learning in MCS have been proposed [3] by exploiting the computing power of mobile devices. As illustrated in Fig. 1, in such framework, instead of the servers, the cloud-based platform recruits some mobile workers to execute the training task. However, this new framework in MCS still has numerous problems that need to be solved. On one hand, although instead of raw data, transmitting the updates of local model can avoid the direct privacy leakage, it may reveal the sensitive information in undirect ways [22]. On the other hand, since participating in such MCS system to train the learning model is usually costly for individual workers (*e.g.,* the consumption of battery), it is paramountly important to attract the maximum participation [4–6]. Therefore, in this paper, *we aim to design a new framework of federated learning in MCS system that can attract more participants while preserving their privacy.*

However, due to the multiple requirements, designing an appropriate framework in MCS system needs to address many new challenges and we elaborate upon some main challenges.

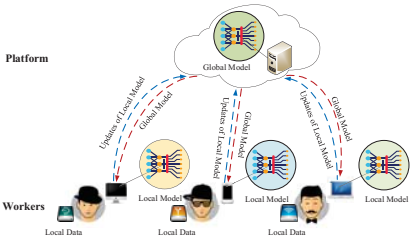


Fig. 1: Illustration of federated learning in MCS system.

On one hand, attracting the participation is challenging since it needs to guarantee two fundamental characteristics, the truthfulness (*i.e.,* each worker reports the cost truthfully) and the individual rationality (*i.e.,* each worker obtains a non-negative utility). Furthermore, since worker's cost is also the sensitive information whose privacy should be preserved, truthfully reporting the cost provides an opportunity for other workers to infer this private information. Therefore, another challenge is to guarantee the differential privacy [14] of workers' truthfully reported costs.

On the other hand, eavesdropping the transmitted updates of local models can also obtain the private information of workers' data [25]. Therefore, apart from the cost, it also

2575-8462/20/$31.00 ©2020 IEEE
DOI 10.1109/SRDS51746.2020.00016

requires us to preserve the differential privacy [14] of workers' transmitted models by adding the artificial noise. However, unlike the existing frameworks of federated learning [25, 29], where the workers are selected uniformly at random, we need to meticulously design the probability for worker selection such that more workers can be attracted with satisfying the truthfulness and individual rationality, and the privacy of their costs can also be protected. Therefore, another challenge is that rather than simply following the existing works, we need to redetermine the type of the artificial noise (*e.g.,* adding Gaussian noise or Laplace noise) and the corresponding parameters (*e.g.,* the variance of Gaussian noise). Furthermore, it is challenging to guarantee the accuracy of federated learning in this paper due to the joint requirements of incentive and privacy preserving.

Therefore, to overcome the above challenges, we propose a novel framework, namely, SHIELD[1], by jointly considering the problems of incentive, privacy and accuracy. We assume that the platform is trusty throughout this paper, which means that the communication between the platform and workers is secure. Thus, we aim to achieve the differential privacy on both workers' costs and local models at the platform end such that eavesdropping on the platform can not obtain the sensitive information. Furthermore, similar to those in the existing works [16], we also apply the gradient perturbation to preserve the privacy of workers' local models, *i.e.,* adding noise to the gradient by the platform or workers. In fact, to further reduce the computation operations of platform and exploit the computing capability of mobile devices, in SHIELD, the artificial noise is individually added to the local gradient by each selected worker rather than the platform. The main contributions of this paper are as follows.

- *Framework:* Different from the exiting works, we propose a novel framework of federated learning in MCS system, namely, SHIELD, by jointly considering problems of incentive, privacy and accuracy.
- *Incentive Requirements:* By meticulously designing the selection probability, SHIELD is able to stimulate the participation of workers, and then bears the properties of truthfulness and individual rationality.
- *Privacy Preserving:* Apart from stimulating the participation, the designed probability also allows SHIELD to achieve (ϵ_E, δ_E)-differential privacy on workers' reported costs. Furthermore, SHIELD achieves (ϵ_G, δ_G)-differential privacy on workers' local models by adding noise.
- *Bounds on Accuracy:* The excess empirical risk of SHIELD is proved to be upper bounded by $\mathcal{O}(\frac{(\ln(Kn_{min}/\delta_G))^{\frac{1}{2}}}{\epsilon_G K n_{min}} + \frac{\ln(Kn_{min}/\delta_G)}{\epsilon_G^2 K^2 n_{min}^2})$, where a special case for totally distributed scenario leads to a much sharper bound $\mathcal{O}(\frac{\log(n/\delta_G)}{\epsilon_G^2 n^2})$ than the latest result $\mathcal{O}(\frac{\ln(mn_{min}/\delta_G)}{\epsilon_G^2 m^2 n_{min}^2})$ in [18], where K is the number of workers selected among m total workers, n is the total number of all data, and n_{min} is the minimum size of workers' data sets.
- *Evaluations:* Finally, numerous experiments are implemented for both classification and regression tasks over two real-world data sets with varying parameters. Apart from the theoretical analysis, the experimental results also validate that SHIELD has superior performance on training accuracy comparing with the state-of-the-art approaches.

[1]The name SHIELD comes from double inSurance: incentivized federated learning witH dIfferEntial privacy in mobiLe crowDsensing.

II. Related Works

Incentive Mechanism in MCS Systems: Due to the paramount significance of attracting more participation, various incentive mechanisms [4–13] for MCS system have been developed recently. Jin *et al.* in [6] investigated the problem of task pricing in MCS system by utilizing Markov game. Lin *et al.* [7] proposed an incentive mechanism to minimize social cost while preserving workers' private information.

Distributed Machine Learning with Differential Privacy: Since the differential privacy was proposed, it has been widely applied to distributed machine learning [18–29]. Combining the differential privacy and secure multi-party computation, Jayaraman *et al.* in [18] introduced a distributed learning method. Zhang *et al.* in [21] decentralized the learning algorithm using the ADMM, and proposed some variable perturbations to provide dynamic differential privacy. Li *et al.* in [22] proposed a new differentially private algorithm in meta learning for gradient-based parameter transfer that not only satisfies the privacy requirement but also guarantees the provable transfer learning in convex settings. Furthermore, following the federated learning proposed by McMahan *et al.* in [1], Triastcyn *et al.* in [25] adopted the Bayesian differential privacy to the federated setting.

III. Preliminaries

In this section, we introduce the system overview, incentive model, federated learning and differential privacy.

A. System Overview

We consider an MCS system consisting of a cloud-based platform and a set of m mobile workers, denoted as $\mathcal{W} = \{w_1, \ldots, w_m\}$, to locally perform training task. For $1 \leq i \leq m$ and $1 \leq j \leq n_i$, each worker w_i has a set of data denoted as $\mathcal{D}_i = \{(\mathbf{x}_1^{(i)}, y_1^{(i)}), \ldots, (\mathbf{x}_{n_i}^{(i)}, y_{n_i}^{(i)})\}$ with size n_i, where $\mathbf{x}_j^{(i)} \in \mathbb{R}^d$ is an input d-dimensional sample vector (*e.g.,* the pixels of an image) and $y_j^{(i)}$ is the output (*e.g.,* the labeled output of the image). Let $\mathcal{D} = \cup_{w_i \in \mathcal{W}} \mathcal{D}_i = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be the total data set with the size n, where the superscript of each data is omitted for notational convenience. At the t-th training round, the workflow of SHIELD is shown in Fig. 2 and described as follows.

- **Incentive Procedure:** The platform announces the training task to workers (Step ①). After receiving the training task, each worker w_i submits a bid b_i to the platform (Step ②), which is her bidding price for executing the task in this training round. Based on the received bids,

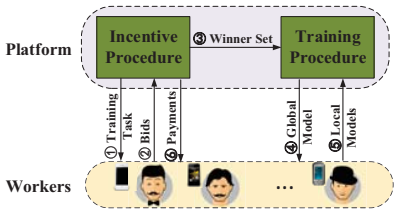


Fig. 2: System overview, where the circled numbers represent the event orders.

the platform determines the set $\mathcal{S}_t$ of winning workers for the t-th training round (Step ③), and the corresponding payment p_i paid to every winning worker $w_i \in \mathcal{S}_t$.

- **Training Procedure:** Then, the platform sends the current global model to workers in $\mathcal{S}_t$ (Step ④). Based on the received global model, the winning workers do the corresponding training locally, and then, submit the local models to the platform (Step ⑤). According to the local models, the platform updates the aggregation result, and pays p_i to each winning worker w_i (Step ⑥).

For convenience, some essential symbols are listed in Table I.

TABLE I: Some Important Notations.

Notation	Definition
$\mathcal{O}$	Big O notation
$\widetilde{\mathcal{O}}$	Some logarithm multiplier terms are hiding in $\widetilde{\mathcal{O}}$ notation
$\mathcal{D}_i$	Data set of worker w_i with size n_i
$\mathcal{S}_t$	Set of winning workers in the t-th training round
$\boldsymbol{\theta}^t$	Global model parameter in the t-th training round
z_i^t	Gaussian noise added by worker w_i in the t-th round
b_i	Bid of worker w_i for executing training task
p_i	Payment to worker w_i
$\ell(\boldsymbol{\theta}, \cdot, \cdot)$	Loss function
$L(\boldsymbol{\theta}, \mathcal{D})$	Function of empirical risk

B. Incentive Model

In the Incentive Procedure of SHIELD shown in Fig. 2, to incentivize the workers who are strategic and selfish to maximize their own utilities, we employ a randomized incentive model defined as follows.

Definition 1 (Randomized Incentive Model)**.** In a randomized incentive model for SHIELD, after receiving the training task, each worker w_i submits a bid b_i to the platform, which is her bidding price for executing the learning task. Her actual cost for executing the task is denoted as c_i, which is unknown to the platform. According to the received bids, the platform meticulously designs a probability distribution, which is utilized to select a set $\mathcal{S}_t \subseteq \mathcal{W}$ of K winning workers, where the selection probability of worker w_i is denoted as $\mathbf{Pr}(b_i)$.

According to the actual cost c_i of worker w_i, we define her utility in the t-th training round as

$$u_i = \begin{cases} p_i - c_i & \text{if } w_i \in \mathcal{S}_t \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

Furthermore, in this paper, we assume that the actual cost c_i of each worker w_i is proportional to her data quantity, *i.e.,* $c_i = \lambda n_i$, where λ is a coefficient known to the platform. Actually, this assumption is practical since more data leads to more consumption of training resources (*e.g.,* time and energy). Furthermore, since the number of data owned by each worker is finite, we assume that the bid b_i of each worker w_i is bounded by $[b_{min}, b_{max}]$, where $0 < b_{min} < b_{max} < \infty$.

The incentive model has two well-known characteristics called the truthfulness and the individual rationality.

Definition 2 (Truthfulness)**.** An incentive model is truthful if for any worker $w_i \in \mathcal{W}$, her utility is maximized when bidding her actual cost c_i.

Definition 3 (Individual Rationality)**.** An incentive model is individual rational if for any worker $w_i \in \mathcal{W}$, her utility u_i satisfies $u_i \geq 0$.

C. Federated Learning

As illustrated in Fig. 2, we develop a federated learning in the Training Procedure of SHIELD. In particular, the federated learning problem, for the union data set $\mathcal{D} = \cup_{w_i \in \mathcal{W}} \mathcal{D}_i$ where $\mathcal{D}_i = \{(\mathbf{x}_1^{(i)}, y_1^{(i)}), \ldots, (\mathbf{x}_{n_i}^{(i)}, y_{n_i}^{(i)})\}$, is to find the model parameter vector $\boldsymbol{\theta} \in \mathcal{C}$ that minimizes the following function of empirical risk:

$$\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D}) \tag{2}$$

where

$$L(\boldsymbol{\theta}, \mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(\boldsymbol{\theta}, \mathbf{x}_j^{(i)}, y_j^{(i)}) \tag{3}$$

and $\ell(\boldsymbol{\theta}, \cdot, \cdot)$ is G-Lipschitz and L-smooth over $\boldsymbol{\theta} \in \mathcal{C}$.

Note that in this paper, we consider both convexity and non-convexity of loss function $\ell(\boldsymbol{\theta}, \cdot, \cdot)$. In the convexity case, similar to [17], we assume that the loss function $\ell(\boldsymbol{\theta}, \cdot, \cdot)$ is convex and the domain $\mathcal{C}$ of $\boldsymbol{\theta}$ is a closed convex set. While in the non-convexity case, we assume that the empirical risk function $L(\boldsymbol{\theta}, \mathcal{D})$ satisfies the Polyak-Lojasiewicz condition.

Algorithm 1: FedAvg [1]

Input: worker set $\mathcal{W}$, learning rate η, number T of training rounds.
Output: model parameter $\boldsymbol{\theta}^T$.

```
Platform initializes the model parameter θ^0;
for t=0,...,T-1 do
    Platform sends θ^t to workers and selects a subset W_t ⊆ W of
      K workers uniformly at random;
    for each selected worker w_i ∈ W_t do
        Calculate θ_i^{t+1} by Eq. (4), and send it to the platform;
    Platform updates the model parameter θ^{t+1} by Eq. (6);
Return θ^T;
```

By Eq. (3), a typical method for federated learning called federated averaging (FedAvg) is proposed in [1] whose procedure is shown in Algorithm 1. In particular, at the t-th training round in FedAvg, the platform sends the current model parameter $\boldsymbol{\theta}^t$ to the workers and selects a subset $\mathcal{W}_t$ of K workers uniformly at random to execute the training task. After receiving $\boldsymbol{\theta}^t$, with the learning rate η, each worker $w_i \in \mathcal{W}_t$ locally trains the model parameter $\boldsymbol{\theta}_i^{t+1}$ according to

$$\boldsymbol{\theta}_i^{t+1} = \boldsymbol{\theta}^t - \eta \nabla L_i(\boldsymbol{\theta}^t, \mathcal{D}_i) \tag{4}$$

and transmits it to the platform, where

$$L_i(\boldsymbol{\theta}^t, \mathcal{D}_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(\boldsymbol{\theta}^t, \mathbf{x}_j^{(i)}, y_j^{(i)}) \tag{5}$$

is the function of empirical risk for worker w_i and ∇ is the differential operator. Then, the platform updates the model parameter $\boldsymbol{\theta}^{t+1}$ as

$$\boldsymbol{\theta}^{t+1} = \frac{1}{K} \sum_{i:w_i \in \mathcal{W}_t} \boldsymbol{\theta}_i^{t+1}. \tag{6}$$

By denoting the ℓ_2-norm as $||\cdot||$ and the inner product as $<\cdot,\cdot>$, we have the following definitions.

Definition 4 (G-Lipschitz)**.** A loss function $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ is G-Lipschitz over $\boldsymbol{\theta}$, if for any j-th data $(\mathbf{x}_j, y_j) \in \mathcal{D}$ and model parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{C}$, where $0 < j \leq n$, it has $|\ell(\boldsymbol{\theta}_1, \mathbf{x}_j, y_j) - \ell(\boldsymbol{\theta}_2, \mathbf{x}_j, y_j)| \leq G||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||$.

Definition 5 (L-Smooth)**.** A differentiable loss function $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ is L-smooth over $\boldsymbol{\theta}$, if for any j-th data $(\mathbf{x}_j, y_j) \in \mathcal{D}$ and model parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{C}$, where $0 < j \leq n$, it has $\ell(\boldsymbol{\theta}_1, \mathbf{x}_j, y_j) \leq \ell(\boldsymbol{\theta}_2, \mathbf{x}_j, y_j) + < \ell(\boldsymbol{\theta}_2, \mathbf{x}_j, y_j), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 > + \frac{L}{2}||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||^2$.

Definition 6 (φ-Strongly Convex)**.** A differentiable empirical risk function $L : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ is φ-strongly convex if for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{C}$, it has

$$L(\boldsymbol{\theta}_2, \mathcal{D}) \geq L(\boldsymbol{\theta}_1, \mathcal{D}) + < \nabla L(\boldsymbol{\theta}_1, \mathcal{D}), \boldsymbol{\theta}_2 - \boldsymbol{\theta}_1 > + \frac{\varphi}{2}||\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1||^2. \tag{7}$$

Definition 7 (Polyak-Lojasiewicz Condition [15])**.** For an empirical risk function $L : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$, assume that it has a nonempty set of global minimizers denoted as Θ^*. If there exists φ such that for any $\boldsymbol{\theta} \in \mathcal{C}$, it has

$$||\nabla L(\boldsymbol{\theta}, \mathcal{D})|| \geq 2\varphi(L(\boldsymbol{\theta}, \mathcal{D}) - L^*), \tag{8}$$

where $L^* = L(\boldsymbol{\theta}^*, \mathcal{D})$ for $\boldsymbol{\theta}^* \in \Theta^*$, then, the function $L(\boldsymbol{\theta}, \mathcal{D})$ satisfies the Polyak-Lojasiewicz condition.

D. Differential Privacy

We introduce the differential privacy in this subsection, which will be applied to SHIELD to preserve the privacy on workers' costs and local models. It is defined in terms of the specific concept of adjacent sets. Informally, two sets $\mathcal{B}$ and $\mathcal{B}'$ are adjacent if they differ by only one entry. Based on the adjacent sets, (ϵ, δ)-differential privacy is defined as follows.

Definition 8 ((ϵ, δ)-Differential Privacy [14])**.** A randomized mechanism $\mathcal{M} : \mathcal{A} \to \mathcal{R}$ is (ϵ, δ)-differential private if for any two adjacent sets $\mathcal{B}, \mathcal{B}' \subseteq \mathcal{A}$, it has

$$\mathbf{Pr}[\mathcal{M}(\mathcal{B}) \in \mathcal{V}] \leq e^{\epsilon} \mathbf{Pr}[\mathcal{M}(\mathcal{B}') \in \mathcal{V}] + \delta \tag{9}$$

where $\mathcal{A}$ and $\mathcal{R}$ are the domain and the output range of mechanism $\mathcal{M}$, respectively, and $\mathcal{V} \subseteq \mathcal{R}$ is a set of outputs of mechanism $\mathcal{M}$.

Next, we introduce two mechanisms that are widely employed to some practical applications. The first is called the exponential mechanism that is defined with respect to a score function $\mu : \mathcal{A} \times \mathcal{R} \to \mathbb{R}$. For any two adjacent sets $\mathcal{B} \subseteq \mathcal{A}$ and $\mathcal{B}' \subseteq \mathcal{A}$ and all possible outputs $r \in \mathcal{R}$, let $\Delta_E = \sup_r \sup_{\mathcal{B},\mathcal{B}'} |\mu(\mathcal{B}, r) - \mu(\mathcal{B}', r)|$. The exponential mechanism is defined as follows.

Definition 9 (Exponential Mechanism [14])**.** Given a score function $\mu : \mathcal{A} \times \mathcal{R} \to \mathbb{R}$ and an input set $\mathcal{B} \subseteq \mathcal{A}$, $\mathcal{M}_E : \mathcal{A} \to \mathcal{R}$ is an exponential mechanism if an output $r \in \mathcal{R}$ is selected by the probability that is proportional to $\exp(\epsilon_E \mu(\mathcal{B}, r))$, *i.e.,*

$$\mathbf{Pr}[\mathcal{M}(\mathcal{B}) = r] \propto \exp(\epsilon_E \mu(\mathcal{B}, r)). \tag{10}$$

In fact, the exponential mechanism will be applied to SHIELD to protect the privacy of workers' reported costs.

The second is the Gaussian mechanism, which will be employed to preserve the privacy of workers' local models. For any function $f : \mathcal{A} \to \mathbb{R}^d$, and any two adjacent sets $\mathcal{B} \subseteq \mathcal{A}$ and $\mathcal{B}' \subseteq \mathcal{A}$, let $\Delta_G = \sup_{\mathcal{B},\mathcal{B}'} ||f(\mathcal{B}) - f(\mathcal{B}')||$. Then the Gaussian mechanism can be defined as follows.

Definition 10 (Gaussian Mechanism [14])**.** Given any function $f : \mathcal{A} \to \mathbb{R}^d$, the Gaussian mechanism is defined as

$$\mathcal{M}_G(\mathcal{B}) = f(\mathcal{B}) + z \tag{11}$$

for any $\mathcal{B} \subseteq \mathcal{A}$, where z is the additive noise that is sampled from the Gaussian distribution $\mathcal{N}(0, \sigma^2 I_d)$ with $\sigma \geq \frac{\sqrt{2\ln(1.25/\delta_G)}\Delta_G}{\epsilon_G}$ and $\Delta_G = \sup_{\mathcal{B},\mathcal{B}'} ||f(\mathcal{B}) - f(\mathcal{B}')||$.

Note that in both Incentive and Training Procedures, the platform is assumed to be trusty, which means that the communication between the platform and workers is secure. Therefore, we aim to guarantee the differential privacy on workers' costs and local models at the end of platform such that eavesdropping on the platform can not obtain any sensitive information. Furthermore, as shown in [18], we can utilize two frameworks to preserve such differential privacy on workers' local models, *i.e.,* adding noise to the global parameter by the platform and adding noise to the local models by the workers. In fact, SHIELD applies the later framework in order to reduce the computation operations of platform and exploit the computing capability of mobile devices. Therefore, adding noise by the workers in SHIELD is only for the above consideration and does not mean the platform is unreliable.

IV. Framework Illustration

In this section, we present the design details of SHIELD and then show the outlines of algorithm (Algorithm 2) which can attract more participants by rewarding them a non-negative utility and preserving the privacy of their costs and local models. Finally, Example 1 illustrates the workflow of SHIELD.

A. Design Rationale

To design an appropriate framework of federated learning in MCS system, we consider the following three requirements.

- *Incentive Requirement:* Since the federated learning in MCS system needs to recruit mobile workers to train the learning model, which is usually costly for individual workers (*e.g.,* the consumption of battery), it is important to design an efficient incentive procedure to attract the maximum participation. *Therefore, we propose a randomized incentive procedure to deal with the incentive issues.*
- *Privacy Preserving on Workers' Costs:* Furthermore, since our incentive procedure is able to stimulate all

participating workers to bid their actual costs, which are actually workers' private information. Therefore, when the actual cost is submitted to the platform, other workers might infer this private information. It urges us to protect the privacy of workers' costs when designing the incentive procedure. *To achieve this goal, we utilize an exponential mechanism to guarantee the differential privacy on their reported costs.*

- *Privacy Preserving on Local Models:* Finally, although the federated learning allows workers to locally train the model without transmitting the local data to the platform, the private information may still be divulged by analysing the submitted local models. *Therefore, we design a new training procedure by adding the Gaussian noise to the local models to prevent the information leakage.*

Therefore, motivated by these paramountly important requirements, we propose our framework, namely, SHIELD.

B. Design Details

In this subsection, we present the design details of SHIELD and then show the outlines of algorithm. Finally, we give an example to illustrate the workflow of SHIELD.

As shown in Algorithm 2, SHIELD consists of two procedures, *i.e.,* the Incentive Procedure and the Training Procedure.

After initializing the global parameter $\boldsymbol{\theta}^0$, the platform carries out the Incentive Procedure.

Algorithm 2: SHIELD

Input: worker set $\mathcal{W}$, learning rate η, bid b_i of each worker w_i, number T of training rounds, number K of recruited workers.
Output: model parameter $\boldsymbol{\theta}^T$.

Platform initializes the model parameter $\boldsymbol{\theta}^0$;
for *t=0,…,T-1* **do**
  Define a copy set $\widehat{\mathcal{W}} \leftarrow \mathcal{W}$;
  $\mathcal{S}_t \leftarrow \emptyset$;
  // **Incentive Procedure:**
  while $|\mathcal{S}_t| < K$ **do**
    for *each worker* $w_i \in \widehat{\mathcal{W}}$ **do**
      // *Winner Selection:*
      Calculate the selection probability $\mathbf{Pr}(b_i)$ of each worker w_i by Eq. (12) and select her with this probability;
      // *Payment Determination:*
      if *worker* w_i *is selected* **then**
        Calculate the payment to worker w_i by Eq. (13);
        $\mathcal{S}_t \leftarrow \mathcal{S}_t \cup \{w_i\}$, $\widehat{\mathcal{W}} \leftarrow \widehat{\mathcal{W}} \backslash \{w_i\}$;
  // **Training Procedure:**
  // *Local Training:*
  Platform sends $\boldsymbol{\theta}_t$ to workers in $\mathcal{S}_t$;
  for *each worker* $w_i \in \mathcal{S}_t$ **do**
    Calculate $G_i(\boldsymbol{\theta}^t)$ by Eq. (14), and send it to the platform;
  // *Global Aggregation:*
  Platform updates the model parameter $\boldsymbol{\theta}^{t+1}$ by Eq. (15);
Return $\boldsymbol{\theta}^T$;

Incentive Procedure: In particular, this procedure further contains two sub-phases, namely, Winner Selection and Payment Determination, which work as follows.

- *Winner Selection:* As mentioned previously, since other workers may infer the reported cost, we apply an exponential mechanism to guarantee the privacy preserving. To employ this mechanism, we set the score function to $\mu(\widehat{\mathcal{W}}, w_i) = 1 - \frac{b_i}{b_{max}}$, where $\widehat{\mathcal{W}}$ is a copy set of worker set $\mathcal{W}$. The score function means that for the input $\widehat{\mathcal{W}}$, once worker w_i is selected to execute the training task, the score is $1 - \frac{b_i}{b_{max}}$. Then, according to this score function, the selection probability of worker w_i is

$$\mathbf{Pr}[\mathcal{M}_E(\widehat{\mathcal{W}})=w_i]=\begin{cases}\frac{\exp(\epsilon'(1-\frac{b_i}{b_{max}}))}{\sum_{j:w_j\in\widehat{\mathcal{W}}}\exp(\epsilon'(1-\frac{b_j}{b_{max}}))} & \text{if } w_i\in\widehat{\mathcal{W}}\\ 0 & \text{otherwise}\end{cases} \tag{12}$$

  where $\epsilon' = \epsilon_E/(e\ln(e/\delta_E))$ for the given privacy budget ϵ_E and failure probability δ_E, and b_i is worker w_i's bid. It can be obtained that $\Delta_E = |\frac{1-b_{min}}{b_{max}}| \leq 1$. For convenient explanation, we denote $\mathbf{Pr}[\mathcal{M}_E(\widehat{\mathcal{W}}) = w_i]$ as $\mathbf{Pr}(b_i)$.
- *Payment Determination:* Since our mechanism is based on the incentive model, which needs to guarantee the truthfulness and individual rationality, we determine the payment p_i to each wining worker $w_i \in \mathcal{S}_t$ as

$$p_i = b_i + \frac{\int_{b_i}^{b_{max}} \mathbf{Pr}(b)db}{\mathbf{Pr}(b_i)}. \tag{13}$$

  It will be observed in the following parts that this payment make our procedure truthful and individual rational.

Training Procedure: Similarly, this procedure also contains two sub-phases, namely, Local Training phase and Global Aggregation phase, which work as follows.

- *Local Training:* At the t-th round, each worker $w_i \in \mathcal{S}_t$ calculates the noisy gradient $G_i(\boldsymbol{\theta}^t)$ of local objective function $L_i(\boldsymbol{\theta}^t, \mathcal{D}_i)$ according to

$$G_i(\boldsymbol{\theta}^t) = \nabla L_i(\boldsymbol{\theta}^t, \mathcal{D}_i) + z_i^t \tag{14}$$

  where $L_i(\boldsymbol{\theta}^t, \mathcal{D}_i)$ is given by Eq. (5) and $z_i^t \sim \mathcal{N}(0, \sigma^2 I_d)$ is the Gaussian noise added to worker w_i's transmitted parameter to guarantee differential privacy.
- *Global Aggregation:* Receiving the updates of local models from workers in $\mathcal{S}_t$, the platform updates the global model $\boldsymbol{\theta}^{t+1}$ according to

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \cdot \sum_{i:w_i\in\mathcal{S}_t} \frac{n_i}{\sum_{j:w_j\in\mathcal{S}_t} n_j} G_i(\boldsymbol{\theta}^t), \tag{15}$$

  where η is the learning rate and n_i is the size of worker w_i's data set.

Example 1. In this example, there are three workers denoted as $\mathcal{W} = \{w_1, w_2, w_3\}$ with data quantities $n_1 = 10, n_2 = 25, n_3 = 30$ and coefficient $\lambda = 2$, which means that their costs are $c_1 = 20, c_2 = 50, c_3 = 60$. Since in the **Incentive Procedure**, each worker truthfully reports the cost, her bid and cost are equal, *i.e.,* $b_1 = c_1, b_2 = c_2$ and $b_3 = c_3$. We assume that the platform collects the local models from two workers, *i.e.,* $K = 2$. At the t-th training round, the platform will define a copy set $\widehat{\mathcal{W}} = \mathcal{W} = \{w_1, w_2, w_3\}$ and a winning worker set $\mathcal{S}_t = \emptyset$. Then, it carries out the *Winner Selection* in **Incentive Procedure**. By assuming $\epsilon_E = 0.1$ and $\delta_E = 0.1$, we can obtain that $\epsilon' = 0.1/(e\ln(e/0.1)) = 0.0111$. According to Eq.

(12), we need to calculate the selection probability of workers in $\widehat{\mathcal{W}} = \{w_1, w_2, w_3\}$. In fact, the selection probability of worker w_1 is $\mathbf{Pr}(b_1) = \exp(0.0111 \times (1 - \frac{20}{60}))/(\exp(0.0111 \times (1 - \frac{20}{60})) + \exp(0.0111 \times (1 - \frac{50}{60})) + \exp(0.0111 \times (1 - \frac{60}{60}))) = 0.3348$. Similarly, the corresponding probabilities of worker w_2 and worker w_3 are $\mathbf{Pr}(b_2) = 0.3329$ and $\mathbf{Pr}(b_3) = 0.3323$, respectively. We assume that worker w_1 is selected, after which, the *Payment Determination* is carried out. According to Eq. (13), the payment to worker w_1 is $p_1 = 20 + \frac{\int_{20}^{60} \mathbf{Pr}(r)dr}{0.3348} = 59.90$, where $\mathbf{Pr}(r) = \exp(0.0111 \times (1 - \frac{r}{60}))/(\exp(0.0111 \times (1 - \frac{r}{60})) + \exp(0.0111 \times (1 - \frac{50}{60})) + \exp(0.0111 \times (1 - \frac{60}{60})))$. Then, we have $\mathcal{S}_t = \{w_1\}$ and $\widehat{\mathcal{W}} = \{w_2, w_3\}$. Since $K = 2$, the platform will further go on the next iteration of **Incentive Procedure**. In the corresponding *Winner Selection*, we only need to calculate the selection probability of worker w_2 and worker w_3 since $\widehat{W} = \{w_2, w_3\}$ after the previous iteration. In fact, the probability of worker w_2 is $\mathbf{Pr}(b_2) = \exp(0.0111 \times (1 - \frac{50}{60}))/(\exp(0.0111 \times (1 - \frac{50}{60})) + \exp(0.0111 \times (1 - \frac{60}{60}))) = 0.5005$. Similarly, the probability of worker w_3 is $\mathbf{Pr}(b_3) = 0.4995$. We assume that worker w_2 is selected in this iteration, after which, the platform carries out the *Payment Determination* again and determines the payment to worker w_2 as $p_2 = 50 + \frac{\int_{50}^{60} \mathbf{Pr}(r)dr}{0.5005} = 60.00$, where $\mathbf{Pr}(r) = \exp(0.0111 \times (1 - \frac{r}{60}))/(\exp(0.0111 \times (1 - \frac{r}{60})) + \exp(0.0111 \times (1 - \frac{60}{60})))$. Then, we have $\mathcal{S}_t = \{w_1, w_2\}$ and $\widehat{\mathcal{W}} = \{w_3\}$. Since two workers have been selected, the platform will stop the **Incentive Procedure** and carry out the **Training Procedure** by receiving the local models from workers in $\mathcal{S}_t = \{w_1, w_2\}$, which is omitted in this example and will be shown in our performance evaluation.

V. Theoretical Analysis

We will show that SHIELD satisfies the properties introduced in Section III. In brief, it preserves the differential privacy on workers' reported costs (Theorem 1) and transmitted local models (Theorem 2) while maintaining the truthfulness (Lemma 1) and the individual rationality (Lemma 2). We further derive the tight bounds on the excess empirical risk in both convex and non-convex scenarios (Theorems 3 and 4). Their proofs are available in our technical report [39].

A. Incentive Requirements

Since SHIELD is based on a randomized incentive model, it needs to satisfy the truthfulness and the individual rationality.

Lemma 1. *In each iteration of the Winner Selection, SHIELD is truthful.*

Lemma 2. *In each iteration of the Payment Determination, SHIELD is individual rational.*

Remark 1. Since the truthfully reported cost c_i of each worker w_i is proportional to her data quantity n_i, *i.e.,* $c_i = \lambda n_i$, once the platform knows the coefficient λ, it can carry out the federated learning directly without the assumption applied to other existing workers that the data quantity of each worker is known to the platform a priori.

B. Privacy Preserving

Since each worker truthfully submits her actual bid and local model, the adversaries may have chance to infer her private information. Therefore, it is essential to preserve the differential privacy on each worker's cost and local model, which urges us to obtain the following two theorems.

Theorem 1 (Differential Privacy on Costs)**.** *In each training round of SHIELD in Algorithm 2, for any privacy budget $\epsilon_E > 0$ and failure probability $\delta_E \in (0, \frac{1}{2}]$, if the platform selects K ($\leq m$) workers by Winner Selection, SHIELD will achieve $(\epsilon_E(e-1)/e, \delta_E)$-differential privacy on workers' costs, where e is the base of the natural logarithm.*

Remark 2. Combining Lemma 2 and Theorem 1, the workers recruited by the Incentive Procedure always get a non-negative utility without the privacy leakage of reported cost. Therefore, compared with the existing works on the federated learning, our approach can significantly attract more workers to participate in the model training.

By adding the artificial noises to the local models, in the following parts, we will show that the Training Procedure in SHIELD is also (ϵ_G, δ_G)-differential private.

Theorem 2 (Differential Privacy on Local Model)**.** *For any privacy budget $\epsilon_G > 0$ and failure probability $\delta_G > 0$, if the loss function $\ell(\boldsymbol{\theta}, \cdot, \cdot)$ is G-Lipschitz over $\boldsymbol{\theta}$, we have that SHIELD is (ϵ_G, δ_G)-differential private on workers' locally training models by setting*

$$\sigma^2 = c \cdot \frac{\ln(1/\delta_G)}{\epsilon_G^2} \cdot \frac{G^2 T}{K^2 n_{min}^2}, \tag{16}$$

where c is a constant, T is the number of training rounds, n_{min} is the size of the minimum data set and K is the number of workers selected by the Incentive Procedure in each training round.

Remark 3. Theorem 2 holds in both convexity and non-convexity since the constraint of the loss function $\ell(\boldsymbol{\theta}, \cdot, \cdot)$ is only G-Lipschitz.

C. Bounds on Accuracy

Although applying the exponential mechanism and Gaussian mechanism to SHIELD allows the participating workers to preserve their private information, they will influence the accuracy of SHIELD, especially the Gaussian mechanism, which adds the artificial noise to workers' local models. Therefore, in this part, we will investigate the accuracy of SHIELD by considering both convexity and non-convexity of the loss function and showing them in Theorems 3 and 4.

For convenience, by denoting $z_{[t]} = \{z_0, \ldots, z_t\}$ and $\mathcal{S}_{[t]} = \{\mathcal{S}_0, \ldots, \mathcal{S}_t\}$, we obtain the following theorems.

Theorem 3 (Accuracy in Convex Scenario)**.** *Without loss of generality, we assume that the loss function $\ell(\boldsymbol{\theta}, \cdot, \cdot)$ is G-Lipschitz and L-smooth over $\boldsymbol{\theta}$. Further assume that $L(\boldsymbol{\theta}, \mathcal{D})$ is φ-strongly convex and differentiable. When the platform*

selects $K \approx m$ workers, by setting σ according to Eq. (16) and learning rate $\eta = \frac{1}{L}$ in the SHIELD, we have

$$\mathbb{E}_{\mathcal{S}_{[T-1]}, z_{[T-1]}}[L(\boldsymbol{\theta}^T, \mathcal{D})] - L^* \leq \mathcal{O}\left(\frac{(\ln(Kn_{min}/\delta_G))^{\frac{1}{2}}}{\epsilon_G K n_{min}} + \frac{\ln(Kn_{min}/\delta_G)}{\epsilon_G^2 K^2 n_{min}^2}\right) \tag{17}$$

where $T = \widetilde{\mathcal{O}}\left(\ln\left(\frac{\epsilon_G^2 K^2 n_{min}^2}{dG^2 \ln(1/\delta_G)}\right)\right)$ is the number of training rounds, $L^* = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D})$, d is the dimension of $\boldsymbol{\theta}$ and n_{min} is the size of minimum data set.

Utilizing the similar method in Theorem 3, we can obtain the following corollary for the totally distributed scenario where all workers are selected to execute the training task.

Corollary 1 (Accuracy in Totally Distributed Scenario)**.** *With the same assumptions in Theorem 3, when the platform selects all m workers, by setting σ according to Eq. (16) and learning rate $\eta = \frac{1}{L}$ in the SHIELD, we have*

$$\mathbb{E}_{\mathcal{S}_{[T-1]}, z_{[T-1]}}[L(\boldsymbol{\theta}^T, \mathcal{D})] - L^* \leq \mathcal{O}\left(\frac{\log(n/\delta_G)}{\epsilon_G^2 n^2}\right). \tag{18}$$

where $T = \widetilde{\mathcal{O}}\left(\ln\left(\frac{\epsilon_G^2 n^2}{dG^2 \ln(1/\delta_G)}\right)\right)$.

Remark 4. Corollary 1 is a result for the totally distributed learning obtained from Theorem 3, where all workers are selected to participate in the training task. Furthermore, it can be seen that the upper bound provided by Corollary 1 is tighter than that derived in [18], which is $\mathcal{O}(\frac{\ln(mn_{min}/\delta_G)}{\epsilon_G^2 m^2 n_{min}^2})$.

We will further investigate the corresponding property in the non-convex scenario. To derive the result, we need that $L(\boldsymbol{\theta}, \mathcal{D})$ satisfies the Polyak-Lojasiewicz condition.

Theorem 4 (Accuracy in Non-convex Scenario)**.** *Without loss of generality, we assume that the loss function $\ell(\boldsymbol{\theta}, \cdot, \cdot)$ is G-Lipschitz and L-smooth over $\boldsymbol{\theta}$. Further assume $L(\boldsymbol{\theta}, \mathcal{D})$ is differentiable and satisfies the Polyak-Lojasiewicz condition. When the platform selects $K \approx m$ workers, by setting σ according to Eq. (16) and learning rate $\eta = \frac{1}{L}$ in the SHIELD, we have*

$$\mathbb{E}_{\mathcal{S}_{[T-1]}, z_{[T-1]}}[L(\boldsymbol{\theta}^T, \mathcal{D})] - L^* \leq \mathcal{O}\left(\frac{(\ln(Kn_{min}/\delta_G))^{\frac{1}{2}}}{\epsilon_G K n_{min}} + \frac{\ln(Kn_{min}/\delta_G)}{\epsilon_G^2 K^2 n_{min}^2}\right) \tag{19}$$

where $T = \widetilde{\mathcal{O}}\left(\ln\left(\frac{\epsilon_G^2 K^2 n_{min}^2}{dG^2 \ln(1/\delta_G)}\right)\right)$ is the number of training rounds, $L^* = \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathcal{D})$, d is the dimension of $\boldsymbol{\theta}$ and n_{min} is the size of minimum data set.

D. Discussion

As illustrated in Subsection III-D, apart from the workers, adding noise by the platform can also guarantee the differential privacy on workers' local models. Therefore, in this subsection, we briefly describe how to realize this framework.

In the Training Procedure, each worker calculates and uploads Eq. (14), where the Gaussian noise term needs to be deleted. After that, the platform updates the model parameter by Eq. (15), where a Gaussian noise term needs to be added.

Next, it can be verified that under the same conditions in Theorem 2, the noise added by the platform is similar to that added by the workers since the probability distributions of adjacent databases after adding noise by the platform are also the mixture Gaussian distribution. Furthermore, we can also obtain the similar upper bounds on the excess empirical risk.

From the discussion, we observe that both frameworks can guarantee the differential privacy on workers' local models with the similar accuracy by adding the similar Gaussian noise. The only difference between them is the time to add noise. In fact, the platform adds the noise to the aggregated parameter after receiving all local models, while the workers add the noise to the local models before the transmission. In SHIELD, we apply the later framework. It should be pointed out again that adding noise by the workers in SHIELD does not mean that the platform is unreliable. In contrast, the platform is assumed to be trusty throughout this paper.

VI. Evaluation

In this section, we introduce the baseline methods, the experimental data sets, as well as the experimental results of the performance evaluation about SHIELD.

A. Baseline Methods

As introduced previously, SHIELD consists of an incentive procedure and a training procedure. Therefore, we will evaluate its performance from these two aspects.

1) Incentive Procedure: Since no any existing works can be applied to the federated learning in MCS system considered in this paper, we consider two baseline methods.

The first method is modified from the incentive mechanism proposed in [35], namely, Enabling Privacy-Preserving Incentive Mechanism (EPPIM), which is originally designed for collecting the sensory data with high quality. In fact, we modify it as follows. When recruiting a fixed number of workers to execute the training task (*e.g.,* 80 workers), the platform randomly selects a worker set from some possible worker sets, where the size of each possible set is no less than the above fixed number. Once a worker set is selected, each contained worker executes the task by spending a cost. Then, the platform pays all of them some payments but only chooses the fixed number of workers for global aggregation.

The second method is modified from the Incentive Procedure of SHIELD, namely, Modified Randomized Incentive Mechanism (MRIM). In contrast to the original Winner Selection, the selection probability of each worker w_i in MRIM is proportional to her bid (*i.e.,* $\frac{b_i}{b_{max}}$). The Payment Determination of MRIM is the same as that of SHIELD.

2) Training Procedure: To evaluate the performance of the Training Procedure of SHIELD, we compare it with some existing works on distributed learning with differential privacy.

The first distributed learning is proposed by Rajkumar and Agarwal in [36], namely, Rajkumar and Agarwal's Differential Private Distributed Learning (RA-DPDL). In particular, RA-DPDL adds both Laplace noise and Gaussian noise to guarantee the differential privacy.

The second is proposed by Jayaraman in [18], namely, Output Perturbation based Differential Private Distributed Learning (OP-DPDL), where the Laplace noise is added.

The third is also proposed by Jayaraman in [18] by adding the Gaussian noise, namely, Gradient Perturbation based Differential Private Distributed Learning (GP-DPDL).

Note that all of the above baselines are totally distributed learning, where all workers are recruited to train the learning model. Therefore, to compare with SHIELD, we modify them as follows. In each training round, the platform only chooses some workers to execute the training task uniformly at random. Obviously, all of them still guarantee the differential privacy.

B. Experimental Data Sets

1) Incentive Procedure: As introduced previously, we assume that the cost of each worker is proportional to her data quantity with the coefficient λ. Therefore, in the evaluation of the Incentive Procedure in SHIELD, we set $\lambda = 0.01$, the number of training rounds $T = 1500$, the number of recruited workers $K = 80$, the privacy budget $\epsilon_E = 0.1$ and the corresponding failure probability $\delta_E = 0.25$. Then, we investigate the social cost and the total payment achieved by the Incentive Procedure in one training round by varying the number of recruited workers from 10 to 90. We further study them in different number of training rounds by increasing the number from 100 to 1500. Finally, the privacy leakage is further investigated by varying the privacy budget in the Incentive Procedure from 0.1 to 1.

2) Training Procedure: To evaluate the performance of the Training Procedure in SHIELD, we consider both classification and regression tasks by utilizing two popular data sets.

- *Classification Tasks:* For the classification, we apply a regularized logistic regression model over the KDDCup99 [37] data set. In particular, there are about $5,000,000$ network instances in KDDCup99 data set. The task is to predict whether a network connection is a denial-of-service attack or not.
- *Regression Tasks:* For the regression, we train a ridge regression model over the KDDCup98 [38], which contains the demographic and other related information of approximately $200,000$ American veterans. The task is to predict the individuals' donation amounts.

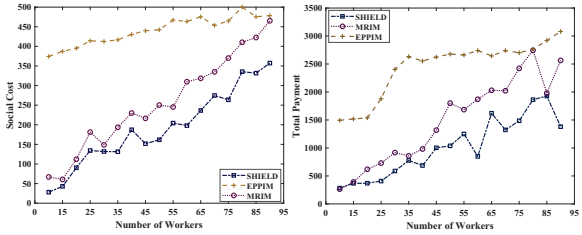
In both data sets, we randomly choose $70,000$ samples and divide them into two sets, *i.e.,* a training set with $50,000$ samples and a test set with $20,000$ samples, respectively.

In our experiments, we divide the training set into 100 subsets randomly, where each worker selects one subset among them, *i.e.,* there are total 100 workers. Furthermore, we set the Lipschitz constant $G = 1$, the learning rate $\eta = 1$, the privacy budget $\epsilon_G = 0.5$, the corresponding failure probability $\delta_G = 0.001$ and the total number of training rounds $T = 1500$. When comparing the performance for different number T of training rounds and different values of privacy budget ϵ_G, we increase them from 100 to 1500 and from 0.01 to 0.5, respectively. While for different number K of recruited workers, we increase it from 10 to 90. Note that when we vary the number of training rounds and the privacy budget, the number of recruited workers is fixed to 80. Furthermore, the privacy budget ϵ_E and the failure probability δ_E in the Incentive Procedure are fixed to 0.1 and 0.25, respectively.

C. Experimental Results

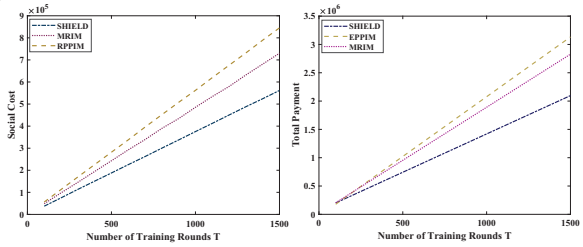
We will show the experimental results of SHIELD.

1) Incentive Procedure: In this subsection, we show the evaluation results for the Incentive Procedure in SHIELD.



(a) (b)

Fig. 3: (a) Social cost versus different numbers of recruited workers. (b) Total payment versus different numbers of recruited workers



(a) (b)

Fig. 4: (a) Social cost versus different number of training rounds. (b) Total payment versus different number of training rounds

Fig. 3 shows the social cost and the total payment in one training round for different number of workers. The increase in number of recruited workers leads to more total payment and social cost since more recruited workers will spend the costs and obtain the payments. Furthermore, Fig. 4, shows them for different number of training rounds. With the increasing number of training rounds, both social cost and total payment increase since more workers need to be recruited.

As shown in Fig. 3 and Fig. 4, the social cost and total payment of the Incentive Procedure in SHIELD is less than those of MRIM and EPPIM due to the following reasons.

- As introduced in Subsection VI-A, in contrast to that in SHIELD, the selection probability in MRIM is proportional to workers' bids, which means that the worker with higher cost has a higher probability to be selected. Therefore, it needs more social cost and total payment.
- Furthermore, as illustrated in Subsection VI-A, unlike that in SHILED, the platform in EPPIM selects a worker set to execute the training task, where the set size is larger than the number of required workers. Then, all workers in the selected set need to spend costs and obtain payments. Therefore, it increases the social cost and total payment.

We further plot the social cost and the privacy leakage in Fig. 5. For any fixed ϵ_E, the privacy leakage is the Kullback-Leibler divergence between the distributions of outputs of

exponential mechanism caused by one worker's different bids [35]. As shown in Fig. 5, the increasing value of ϵ_E leads to the increasing privacy leakage but the decreasing social cost.

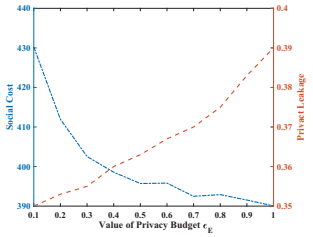
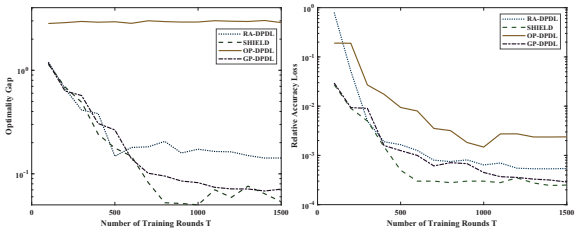


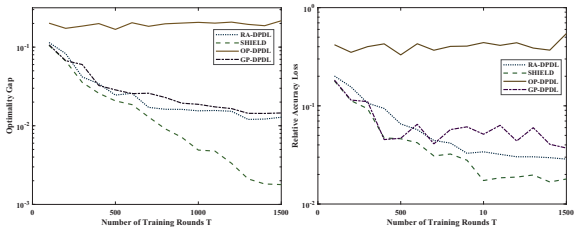
Fig. 5: Privacy leakage versus different values of privacy budget ϵ_E in the Incentive Procedure, where the platform selects 95 workers from the worker set in one training round by still setting the failure probability $\delta_E = 0.25$.

2) Training Procedure: In this subsection, we plot the evaluation results of the Training Procedure in SHIELD compared with the baseline methods. As illustrated previously, in SHIELD, workers are selected by the Incentive Procedure, while they are selected uniformly at random in the baselines.



(a) (b)

Fig. 6: Optimality gap and relative accuracy loss on KDDCup99 data set versus different number of training rounds, where the probability of selecting each worker is 0.4 in the baselines.

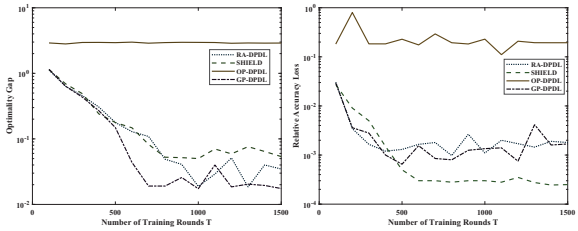


(a) (b)

Fig. 7: Optimality gap and relative accuracy loss on KDDCup98 data set versus different number of training rounds, where the probability of selecting each worker is 0.4 in the baselines.

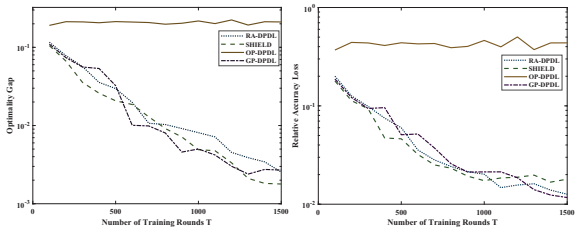
Fig. 6, Fig. 7, Fig. 8 and Fig. 9 show the optimality gap and the relative accuracy loss of SHIELD and other baselines for different number of training rounds over KDDCup99 and KDDCup98 data sets, respectively. Furthermore, in Fig. 6 and Fig. 7, the selection probability in baselines is 0.4. In contrast, in Fig. 8 and Fig. 9, the selection probability is 0.8. It can be seen that in Fig. 6 and Fig. 7, the optimality gap and the accuracy loss of SHIELD are both better than those of baselines. However, in Fig. 8 and Fig. 9, both optimality gap and accuracy loss of baselines are better than those of SHIELD sometimes, especially for the optimality gap.

In Fig. 10 and Fig. 11, we show the optimality gap and the relative accuracy loss with different privacy budgets over KDDCup99 and KDDCup98 data sets, respectively, where the



(a) (b)

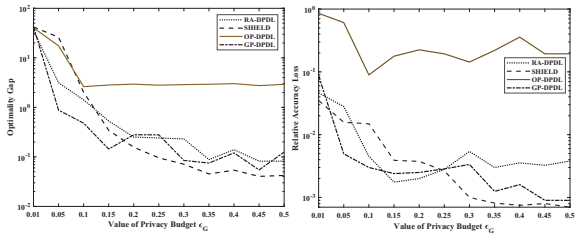
Fig. 8: Optimality gap and relative accuracy loss on KDDCup99 data set versus different number of training rounds, where the probability of selecting each worker is 0.8 in the baselines.



(a) (b)

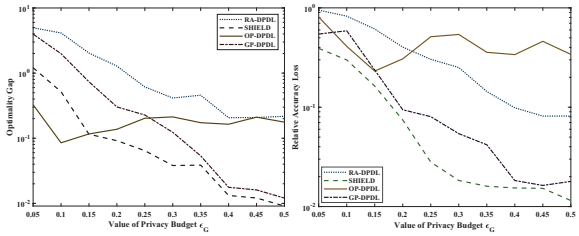
Fig. 9: Optimality gap and relative accuracy loss on KDDCup98 data set versus different number of training rounds, where the probability of selecting each worker is 0.8 in the baselines.

selection probability is 0.8 in Fig. 10, while it is 0.4 in Fig. 11. With the increasing privacy budget, both optimality gap and relative accuracy loss decrease since the smaller privacy budget means that the noise added to each local model is also smaller. Furthermore, similar to the results shown previously, compared with the baselines, SHIELD has better performance for the low selection probability, while it has a little worse performance for the high probability sometimes.



(a) (b)

Fig. 10: Optimality gap and relative accuracy loss on KDDCup99 data set versus different values of privacy budget ϵ_G in Learning Procedure, where the probability of selecting each worker is 0.8 in the baselines.



(a) (b)

Fig. 11: Optimality gap and relative accuracy loss on KDDCup98 data set versus different values of privacy budget ϵ_G in Learning Procedure, where the probability of selecting each worker is 0.4 in the baselines.

It can be seen from the experimental results that the Training Procedure in SHIELD achieves better performance when the

selection probability in other baseline methods is 0.4 due to the following reasons.

- Unlike the baselines, where the platform updates the model parameter by simply averaging the local models, in SHIELD, it calculates the weighted average since the data scales are usually uneven on different workers.
- Due to the Incentive Procedure in SHIELD, and low selection probability in baselines, the platform in SHIELD has more workers to execute the training task such that more data can be applied to improve the accuracy.

In contrast, when the selection probability is 0.8, the performances of some baselines are better than that of SHIELD. This is because that with the high probability, the platform in the baselines can select more workers to participate in the training task. However, they can not achieve such high probability in practice due to the lack of incentive procedure.

VII. Conclusion

In this paper, we have proposed a novel incentivized federated learning in MCS system, namely, SHIELD, which is able to attract more mobile workers to participate in the training task. Furthermore, it has been proved that SHIELD guarantee the differential privacy on workers' reported costs and local models. Additionally, the theoretical analysis has shown that SHIELD has tight bounds on the excess empirical risk. Finally, the experimental results have shown that SHIELD achieves superior performance.

VIII. Acknowledgement

This work was partially supported by National Natural Science Foundation of China under Grants 61871264 and 61902244, as well as SJTU-CUHK Joint Research Collaboration Fund and Shanghai Municipal Science and Technology Commission under Grant 19YF1424600.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS,* 2017.

[2] C. Dinh, N. H. Tran, M. N. Nguyen, C. S. Hong, W. Bao, A. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *arXiv:1910.13067,* 2019.

[3] S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor, and C. S. Hong, "A crowdsourcing framework for on-device federated learning," *IEEE Trans. Wireless. Commun.,* 2020.

[4] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *INFOCOM,* 2019.

[5] R. Zhou, Z. Li, and C. Wu, "A truthful online mechanism for location-aware tasks in mobile crowd sensing," in *IEEE Trans. Mob. Comput.,* 2018.

[6] H. Jin, H. Guo, L. Su, K. Nahrstedt, and X. Wang, "Dynamic task pricing in multi-requester mobile crowd sensing with markov correlated equilibrium," in *INFOCOM,* 2019.

[7] J. Lin, D. Yang, M. Li, J. Xu, and G. Xue, "Frameworks for privacypreserving mobile crowdsensing incentive mechanisms," *IEEE Trans. Mobile Comput.,* 2018.

[8] M. Karaliopoulos, I. Koutsopoulos, L. Spiliopoulos, "Optimal user choice engineering in mobile crowdsensing with bounded rational users," in *INFOCOM,* 2019.

[9] Y. Zhang, Y. Gu, M. Pan, N. H. Tran, Z. Dawy, and Z. Han, "Multi-dimensional incentive mechanism in mobile crowdsourcing with moral hazard," in *IEEE Trans. Mob. Comput.,* 2018.

[10] J. Wang, Y. Wang, D. Zhang, W. Feng, H. Xiong, C. Chao, L. Qin, and Z. Qiu, "Multi-task allocation in mobile crowd sensing with individual task quality assurance," *IEEE Trans. Mob. Comput.,* 2018.

[11] Y. Zhao, and X. Gong, "Truthful quality-aware data crowdsensing for machine learning", in *SECON,* 2019.

[12] M. Tang, H. Pang, S. Wang, L. Gao, J. Huang, and L. Sun, "Multidimensional auction mechanisms for crowdsourced mobile video streaming," in *IEEE/ACM Trans. Netw.,* 2018.

[13] F. Restuccia, P. Ferraro, S. Silvestri, S. K. Das and G. L. Re, "IncentMe: Effective mechanism design to stimulate crowdsensing participants with uncertain mobility," in *IEEE Trans. Mob. Comput.,* 2019.

[14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.,* 2014.

[15] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-lojasiewicz condition," in *ECML PKDD,* 2016.

[16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *CCS,* 2016.

[17] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *NeurIPS,* Dec. 2017.

[18] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Distributed learning without distress: Privacy-preserving Empirical Risk Minimization," *NeurIPS,* 2018.

[19] Z. Huang, R. Hu, Y. Gong, and E. Chan-Tin, "Dp-admm: Admmbased distributed learning with differential privacy," *IEEE Trans. Inf. Forensics Security,* 2019.

[20] M. Wang, C. Xu, X. Chen, H. Hao, L. Zhong, and S. Yu, "Differential privacy oriented distributed online learning for mobile social video prefetching," *IEEE Trans. Multimedia,* 2019.

[21] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forensics Security,* 2017.

[22] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially private meta-learning," in *ICLR,* 2020.

[23] H. Ono, and T. Takahash, "Locally private distributed reinforcement learning," *arXiv preprint arXiv:2001.11718,* 2020.

[24] C. Li, P. Zhou, G. Chen, and Y. Jiang, "Differentially private distributed online learning," *IEEE Trans. Knowl. Data Eng.,* 2018.

[25] A. Triastcyn and B. Faltings, "Federated learning with bayesian differential privacy," in *BigData,* 2019.

[26] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Trans. Ind. Inform.,* 2020.

[27] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *ICC,* 2019.

[28] J. Ding, Y. Gong, M. Pan, and Z. Han, "Optimal differentially private ADMM for distributed machine learning," *arXiv:1901.02094,* 2019

[29] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," *arXiv preprint arXiv:1812.00984,* 2018.

[30] A. Archer and E. Tardos, "Truthful mechanisms for one-parameter agents," in *FOCS,* 2001.

[31] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar, "Differentially private combinatorial optimization," in *SODA,* 2010.

[32] T. van Erven and P. Harremoes, "Rényi divergence and Kullback-Leibler divergence," *IEEE Trans. Inf. Theory,* 2014.

[33] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *IACR,* 2016.

[34] D. Csiba and P. Richtárik, "Global convergence of arbitrary-block gradient methods for generalized polyak-Lojasiewicz functions," *arXiv preprint arXiv:1709.03014,* 2017.

[35] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy preserving incentives for mobile crowd sensing systems," in *ICDCS,* 2016.

[36] A. Rajkumar and S. Agarwal, "A differentially private stochastic gradient descent algorithm for multiparty classification," in *AISTATS,* 2012.

[37] S. Hettich and S. D Bay. UCI machine learning repository, 1999.

[38] I. Parsa and K. Howes. UCI machine learning repository, 1998.

[39] Technical Report:"Double insurance: Incentivized federated learning with differential privacy in mobile crowdsensing," https://www.dropbox.com/s/372mwtskxsgwedt/Federated-Learning-Crowdsensing-TR.pdf?dl=0.