The Value of Collaboration in Convex Machine Learning with Differential Privacy

Nan Wu[‡], Farhad Farokhi[*,†], David Smith[*,§], and Mohamed Ali Kaafar[*,‡]
[‡]Macquarie University [*]CSIRO's Data61 [†]The University of Melbourne [§]Australian National University

***Abstract*—In this paper, we apply machine learning to distributed private data owned by multiple data owners, entities with access to non-overlapping training datasets. We use noisy, differentially-private gradients to minimize the fitness cost of the machine learning model using stochastic gradient descent. We quantify the quality of the trained model, using the fitness cost, as a function of privacy budget and size of the distributed datasets to capture the trade-off between privacy and utility in machine learning. This way, we can predict the outcome of collaboration among privacy-aware data owners prior to executing potentially computationally-expensive machine learning algorithms. Particularly, we show that the difference between the fitness of the trained machine learning model using differentially-private gradient queries and the fitness of the trained machine model in the absence of any privacy concerns is inversely proportional to the size of the training datasets squared and the privacy budget squared. We successfully validate the performance prediction with the actual performance of the proposed privacy-aware learning algorithms, applied to: financial datasets for determining interest rates of loans using regression; and detecting credit card frauds using support vector machines.**

***Index Terms*—Machine learning; Differential privacy; Stochastic gradient algorithm.**

I. Introduction

A. Motivation and Contributions

Data analysis methods using machine learning (ML) can unlock valuable insights for improving revenue or quality-of-service from, potentially proprietary, private datasets. Having large high-quality datasets improves the quality of the trained ML models in terms of the accuracy of predictions on new, potentially untested data. The subsequent improvements in quality can motivate multiple data owners to share and merge their datasets in order to create larger training datasets. For instance, financial institutes may wish to merge their transaction or lending datasets to improve the quality of trained ML models for fraud detection or computing interest rates. However, government regulations (e.g., the roll-out of the General Data Protection Regulation in EU, the California Consumer Privacy Act or the development of the Data Sharing and Release Bill in Australia) increasingly prohibit sharing customer's data without consent [1]. Our work here is motivated by the need to conciliate the tension between quality improvement of trained ML models and the privacy concerns for data sharing.

We investigate a machine learning setup in which a learner wants to train a model based on multiple datasets from different data owners. For the purpose of preserving privacy for data contributors, the learner can only submit queries to data owners and they respond by providing differentially-private (DP) responses as illustrated in Figure 1. We specifically consider honest-but-curious threat models in which different private data owners do not trust each other (or the central learner) for sharing private training datasets, but trust the learner to train the model correctly. As an example, in financial services, a central learner, such as a central bank or government, can be trusted for facilitating computations among banks although they may not trust each other or the learner for accessing private data. Another example is for smart grid in which electricity retailers are private data owners and the electricity market operator can facilitate learning. In this paper, the learner submits a gradient query to each data owner. Upon receiving DP responses from data owners to the gradient queries, the learner adjusts the parameters of the ML model in the direction of the average of the DP gradients. Therefore, the quality of the DP responses (in terms of the magnitude of the additive DP noise) from the data owners to the gradient queries determines the performance of the ML training algorithm.

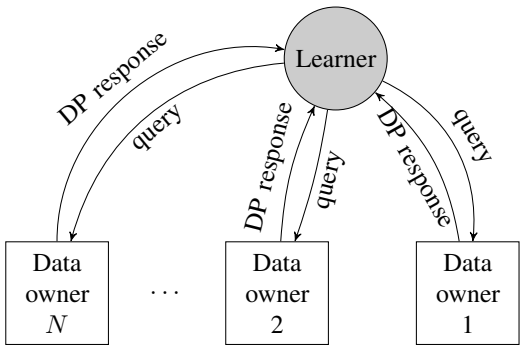

Fig. 1. The communication structure between the learner and the distributed data owners for submitting queries and providing differentially-private (DP) responses.

An important parameter in the ML training algorithm is the step size, the amount by which the model parameters are adjusted in each iteration. If the fitness cost of the ML meets the assumptions of smoothness, strong convexity, and Lipschitz-continuity of the gradient, we can prove that, by selecting the step sizes to be inversely proportional with the iteration number and inversely proportional with the maximum number of iterations squared (see Algorithm 1 in Section II), the difference between the fitness of the trained ML model using DP gradient queries and the fitness of the trained ML model in the absence of any privacy concerns becomes small. In fact, the magnitude of the difference becomes inversely

DOI 10.1109/SP40000.2020.00025

proportional to the size of the training datasets squared and the privacy budgets of the data owners squared; see Theorem 2 in Section III. Several ML models and fitness costs, such as linear and logistic regression, satisfy the above-mentioned assumptions. This enables us to predict the outcome of collaboration among privacy-aware data owners and the learner in terms of the fitness cost of the ML training model. However, if the fitness function does not meet these assumptions, we must select the step size to be inversely proportional to the square root of the iteration number. This way, the step size fades away much slower and the effect of the DP noise is more pronounced on the iterates of the learning algorithm. Therefore, we must add an averaging layer on top of the algorithm to reduce the negative impact of the DP noise; see Algorithm 2 in Section II. This is based on the developments of [2] with appropriate changes in the averaging step to suit the ML problem with DP gradient queries. In this case, we can prove that the difference between the fitness of the trained ML model using DP gradient queries and the fitness of the trained ML model in the absence of any privacy concerns is inversely proportional to the size of the training datasets (no longer squared) and the privacy budget (no longer squared); see Theorem 3 in Section III.

In this paper, we focus on the case where the datasets in possession of the private data owners in Figure 1 are mutually exclusive or non-overlapping, i.e., two identical records are not shared across the datasets. In many real-life applications within the financial and energy sectors, this is a realistic assumption, e.g, transactional records (e.g. for purchasing goods) are unique by the virtue of timestamps, amounts, and the uniqueness of purchases by an individual. This assumption is set in place to ensure differential privacy using independent additive noises. In the absence of such an assumption, there also needs to be a privacy-preserving mechanism for identifying those common entries without potential information leakage regarding non-common entries, which itself is a daunting task and open problem for research.

For experimental verification of the theoretical results, two financial datasets are used in this paper. First, we use a regression model on a dataset containing information on loans made on Lending Club, a peer-to-peer lending platform [3], to automate the process of setting interest rates of loans. Second, we train a support vector machine for detecting fraudulent transactions based on a dataset containing transactions made by European credit card-holders in September 2013 [4]. We use the experiments to validate theoretical predictions and to gain important insights into the outcome of collaborations among privacy-aware data owners. For instance, even if the learner has access to one large dataset with relaxed privacy constraints, the performance of the trained ML model can be very bad if small conservative datasets (i.e., datasets with very small privacy budgets) also contribute to the learning. Therefore, it is best to exclude smaller conservative datasets from collaboration. This is a *counter-intuitive observation as it clearly indicates that more data is not always good*, if it is obfuscated by conservative data owners. Larger, but conservative, datasets are sometimes worth including in the training as they do not degrade performance heavily with their conservative privacy budgets, yet improve the performance of the trained ML model because of their size. These observations can be alternatively interpreted as: collaboration in training a model with a dataset can only be useful if and only if it has enough information (i.e., enough data entries) to suppress the impact of random noise added for privacy guarantees.

In summary, this paper makes the following contributions:

- We develop DP gradient descent algorithms for training ML models on distributed private datasets owned by different entities; see Algorithms 1 and 2 in Section II.
- We prove that the quality of the trained ML model using DP gradient descent algorithm scales inversely with privacy budgets squared, and the size of the distributed datasets squared, which can establish a trade-off between privacy and utility in privacy-preserving ML;
- We develop a theory that enables to predict the outcome of a potential collaboration among privacy-aware data owners (or data custodians) in terms of the fitness cost of the ML training model prior to executing potentially computationally-expensive ML algorithms on distributed privately-owned datasets; see Theorems 2 and 3 in Section III. The bounds in these theorems are not necessarily optimal, i.e., there might exist better performance bounds for other privacy-preserving learning algorithms, but, if the data owners follow Algorithms 1 and 2, they can predict their success or failure.
- We validate our theoretical analysis by evaluating our differentially private ML algorithms using distributed non-overlapping financial datasets belonging to multiple institutes/banks for determining interest rates of loans using regression, and for detecting credit card fraud using support vector machine classifier; We further validate the predictions of the analysis with the actual performance of the proposed privacy-aware learning algorithms applied to the distributed financial datasets; see Section IV.
- Our experimental results indicate that, in the case of three banks collaborating to train a support vector machine classifier to detect credit card fraud, within only 100 iterations, the fitness of the trained model using DP gradient queries is in average within 90% of the fitness of the trained model in the absence of privacy concern if the privacy budget is equal to 1 and each bank has access to a dataset of 30,000 records of credit card transactions and their validity. We observe similar performance results for training a regression model over interest rates of loans with the privacy budget of 10 and datasets of 350,000 records each.

B. Related Work

ML using Secure Multi-Party Computation and Encryption. Secure multi-party computation provide avenues for securing the iterations of distributed ML algorithms across multiple data owners. In the past, secure multi-party computation has been used in various ML models, such as decision trees [5], regression [6], association rules [7], and clustering [8],

[9]. Training ML models using encrypted data was discussed in [10]–[14]. In [15], efficient conversion of models for use of encrypted input data was discussed. The use of secure multi-party computation reduces the computational efficiency of ML algorithms by adding a non-trivial computational and communication performance overhead.

ML with Differential Privacy. A natural way for alleviating privacy concerns is to deploy privacy-enabled ML using differential privacy (DP) [16]–[19]. In [18], a privacy-preserving regularized logistic regression algorithm is provided for learning from private databases by bounding the sensitivity of regularized logistic regression, and perturbing the learned classifier with noise proportional to the sensitivity. This technique is proved to be DP and simulations are used to investigate the trade-off between privacy and learning utility. In [17], a large class of optimization-based DP machine learning algorithms are developed by appropriately perturbing the objective function of the ML training algorithm. The mechanism is applied to linear and logistic regression models and shown to provide high accuracy. In the mentioned studies, privacy-preserving ML, however, often relies on an entire dataset, constructed by merging smaller datasets, being stored in one location. The ML model is then either trained on the aggregated dataset, and is systematically obfuscated using additive noise to guarantee differential privacy, or trained on an obfuscated centrally-located data. Such methods do not address the underlying problem that the smaller datasets are owned by multiple entities with restrictions on sharing sensitive data.

Distributed/Collaborative Privacy-Preserving ML. ML based on distributed private datasets has been recently investigated in, e.g., [20]–[24]. Note that this problem is intimately related to distributed optimization using differentially-private oracles, as such ML problems can be cast as distributed optimization problems in which distributed training datasets are represented within cost functions or constraints of the entities. Using stochastic gradient descent with additive Gaussian/Laplace noise to ensure DP is also common in the literature; (e.g., [25]–[28]). In [25], noisy gradients are used to train a deep neural network. The scale of the required additive noise for DP is reduced in [26] by employing the idea of moment accountant, instead of standard composition rules. Stochastic gradient descent is also utilized in [27] for recurrent neural network language models. Generalizations for obfuscating individual and group-level trends by DP additive noise are presented in [28]. Because iterative methods rely on multiple rounds of inquiries of private datasets, for instance, by submitting multiple gradient queries, the privacy budget must be inversely scaled by the total number of iterations to ensure that a reasonable privacy guarantee can be achieved (alternatively, privacy guarantees get weaker as the number of iterations grows because of the composition rule of differential privacy). Hence, if the parameters of the optimization algorithm are not carefully chosen, bounds on the performance of the ML training algorithm deteriorates with an increasing total number of iterations; e.g., see [29]. In [20], [21], the privacy budget was kept constant and therefore by communicating more, as the number of the iterations grows, the privacy guarantee weakens. However, in those studies, if the privacy budget had been scaled inversely proportional to the total number of iterations, privacy guarantees would be maintained over the entire horizon but performance would deteriorate with increasing total number of iterations, as in [29].

All these studies, however, do not address the issues of convergence of the learning algorithm, selection of appropriate step size in the stochastic gradient descent, and forecasting of the quality of the trained ML model based on the privacy budget prior to running extensive potentially computationally-expensive experiments. These missing steps are some of the important contributions of this paper.

C. Paper Organization

The rest of the paper is organized as follows. We introduce our system model and propose privacy-aware ML algorithms with distributed private datasets in Section II. We analyze and provide theoretical results for predicting the performance of the privacy-preserving training algorithms in Section III. We present the experimental results in Section IV. Finally, we conclude the paper in Section V.

II. ML Training Algorithm Based on Distributed Private Data with DP Gradient Queries

A. Setup

Consider a group of $N \in \mathbb{N}$ private agents or data owners $\mathcal{N} := \{1, \dots, N\}$ that are connected to a node responsible for training a ML model, identified as a learning agent, over an undirected communication graph as in Figure 1. Each agent has access to a set of private training data $\mathcal{D}_i := \{(x_i, y_i)\}_{i=1}^{n_i} \subseteq \mathbb{X} \times \mathbb{Y} \subseteq \mathbb{R}^{p_x} \times \mathbb{R}^{p_y}$, where x_i and y_i, respectively, denote inputs and outputs. Each data owner, for instance, could be a private bank/financial institution. In this case, the private datasets can represent information about loan applicants (such as salary, employment status, and credit rating[1]) as inputs and historically approved interest rates per annum by the bank (in percentage points) as outputs.

Assumption 1. *Private datasets are mutually exclusive, i.e., $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ for all $i, j \in \mathcal{N}$.*

Assumption 1 states that two identical records, equal in every possible aspect, cannot be in two or more datasets. This is a realistic assumption in many real-life applications, such as financial and energy data. For instance, across multiple banks and financial-service providers, transaction records (e.g. for purchasing goods) are unique by the virtue of timestamps, amounts, and the uniqueness of purchases for an individual. In energy systems, one household cannot transact (for purchasing power) with two or more energy retailers and thus its consumption pattern can only be stored by one retailer. The reasons behind this assumption are two-fold. First, to guarantee ϵ-differential privacy, we need to ensure that the records are not

[1]Categorical attributes, such as gender, can always be translated into numerical ones according to a rule.

repeated so that an adversary cannot reduce the noise levels by averaging the reports containing information about repeated entries and thus exceeding ϵ (due to the composition rule for differential privacy). If the datasets had common entries, there would need to be a privacy-preserving mechanism for identifying those common entries without potential information leakage with respect to non-common entries, which is a daunting task. The mutually exclusive or non-overlapping nature of the datasets also results in statistical independence of additive privacy-preserving noise. This independence is extremely useful in computing the magnitude of the additive noise for forecasting the performance of privacy-aware learning algorithms. If records can appear in at most $\kappa \in \{1, \ldots, N\}$ datasets and we do not exclude the overlapping entries during the learning, we must ensure that the gradient queries are DP with privacy budget $\epsilon_i/\kappa, \forall i \in \mathcal{N}$. This is to ensure that we can guarantee privacy budget ϵ_i for the repeated entries across the datasets by using the composition rule for differential privacy. This results in degradation of the fitness of the trained ML model with privacy-preserving algorithms. For instance, in Theorem 2, we show that the difference between the fitness of the trained ML model using DP gradient queries and the fitness of the trained ML model in the absence of any privacy concerns is inversely proportional to the size of the training datasets squared and the privacy budget squared. Therefore, when allowing repeated entries, the difference between the fitness of the private ML model and the fitness of the trained machine model without privacy concerns degrades by a factor of κ^2.

The learning agent is interested in extracting a meaningful relationship between the inputs and outputs using ML model $\mathfrak{M} : \mathbb{X} \times \mathbb{R}^{p_\theta} \to \mathbb{Y}$ and the available training datasets $\mathcal{D}_i$, $\forall i \in \mathcal{N}$, by solving the optimization problem in

$$\theta^* \in \arg\min_{\theta \in \Theta} \left[g_1(\theta) + \frac{1}{n} \sum_{j \in \mathcal{N}} \sum_{\{x,y\} \in \mathcal{D}_j} g_2(\mathfrak{M}(x;\theta), y) \right], \quad (1)$$

where $g_2(\mathfrak{M}(x;\theta), y)$ is a loss function capturing the "closeness" of the outcome of the trained ML model $\mathfrak{M}(x;\theta)$ to the actual output y, $g_1(\theta)$ is a regularizing term, $n := \sum_{\ell \in \mathcal{N}} n_\ell$, and $\Theta := \{\theta \in \mathbb{R}^{p_\theta} \mid \|\theta\|_\infty \leq \theta_{\max}\}$. Note that a large enough $\theta_{\max}$ can always be selected such that the search over Θ does not add any conservatism (in comparison to the unconstrained case), if desired. We use $f(\theta)$ to denote the cost function of (1) for the sake of the brevity of the presentation, i.e.,

$$f(\theta) := g_1(\theta) + \frac{1}{n} \sum_{\{x,y\} \in \bigcup_{j \in \mathcal{N}} \mathcal{D}_j} g_2(\mathfrak{M}(x;\theta), y). \quad (2)$$

Remark 1 (Generality of Optimization-Based ML)**.** *In an automated loan assessment example, a bank maybe interested in employing a linear regression model to estimate the interest rate of the loans based on attributes of customers (thus developing an "AI platform" for loan assessment and delivery). A linear regression model, as the name suggests, considers a linear relationship between input x and output y in the form of $y = \mathfrak{M}(x;\theta) := \theta^\top x$, where $\theta \in \mathbb{R}^{p_\theta}$ is the parameter of the ML model. We can train the regression model by solving the optimization problem* (1) *with $g_2(\mathfrak{M}(x;\theta), y) = \|y - \mathfrak{M}(x;\theta)\|_2^2$, and $g_1(\theta) = 0$. In addition to linear (or non-linear) regression discussed earlier, which clearly is of the form in* (1)*, several other ML algorithms follow this formulation. Another example is linear support vector machines (L-SVM). In this problem, it is desired to obtain a separating hyper plane of the form $\{x \in \mathbb{R}^{p_x} : \theta^\top [x^\top \ 1]^\top = 0\}$ with its corresponding classification rule $\text{sign}(\mathfrak{M}(x;\theta))$ with $\mathfrak{M}(x;\theta) := \theta^\top [x^\top \ 1]^\top$ to group the training data into two sets (corresponding to $y = +1$ and $y = -1$). This problem can be cast as* (1) *with $g_1(\theta) := (1/2)\theta^\top\theta$ and $g_2(\mathfrak{M}(x;\theta), y) := \max(0, 1 - \mathfrak{M}(x;\theta)y)$. We can easily see that the extension to non-linear SVM can also be cast as an optimization-based ML problem. Another example is artificial neural network (ANN). In this case, $\mathfrak{M}(x;\theta)$ describes the input-output behaviour of the ANN with θ capturing parameters, such as internal thresholds. This problem can be cast as* (1) *with $g_1(\theta) := 0$ and $g_2(\mathfrak{M}(x;\theta), y) := \|y - \mathfrak{M}(x;\theta))\|_2$.*

If the data owners could come to an agreement to share private data (and it was not illegal to disclose customers' private information without their consent), the learning agent could train the ML model by solving the optimization problem (1) directly. In practice, however, data owners may not be able to share their private data. In this case, the learning agent can submit queries $\mathfrak{Q}_i(\mathcal{D}_i; k) \in \mathcal{Q}$ to agent $i \in \mathcal{N}$ for $k \in \mathcal{T} := \{1, \ldots, T\}$, where T denotes the number of communication rounds (i.e., the number of queries) agreed upon by all the data owners prior to the exchange of information, index k identifies the current communication round, and $\mathcal{Q}$ denotes the output space of the query. Agent $i \in \mathcal{N}$ can then provide a differentially-private response $\overline{\mathfrak{Q}}_i(\mathcal{D}_i; k) \in \mathcal{Q}$ to the query $\mathfrak{Q}_i(\mathcal{D}_i; k) \in \mathcal{Q}$.

Definition 1 (Differential Privacy)**.** *The response policy of data owner $\ell \in \mathcal{N}$ is ϵ_ℓ-differentially private over the horizon T if*

$$\mathbb{P}\Big\{ (\overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell; k))_{k=1}^T \in \mathcal{Y} \Big\} \leq \exp(\epsilon_\ell) \mathbb{P}\Big\{ (\overline{\mathfrak{Q}}_\ell(\mathcal{D}'_\ell; k))_{k=1}^T \in \mathcal{Y} \Big\},$$

where $\mathcal{Y}$ is any Borel-measurable subset of $\mathcal{Q}^T$, and $\mathcal{D}_\ell$ and $\mathcal{D}'_\ell$ are two adjacent datasets differing at most in one entry, i.e., $|\mathcal{D}_\ell \setminus \mathcal{D}'_\ell| = |\mathcal{D}'_\ell \setminus \mathcal{D}_\ell| \leq 1$.

The learning agent then processes all the received responses to the queries in order to generate its ML model:

$$\hat{\theta} := \varsigma((\overline{\mathfrak{Q}}_j(\mathcal{D}_j; k))_{k \in \mathcal{T}, j \in \mathcal{N}}),$$

where $\varsigma : \prod_{k \in \mathcal{T}} \mathcal{Q}^T \to \mathbb{R}^{p_\theta}$ is a mapping used by the learning agent for fusing all the available information.

In the next subsection, we present an algorithm for generating queries, and then use the provided differentially-private responses for computing a trained ML model.

B. Algorithm

In the absence of privacy concerns, one strategy for training the ML model by the learning agent is to provide unfettered access to the original private data of the data owners in $\mathcal{N}$. In this case, the learning agent can follow the projected (sub)gradient descent iterations in

$$\theta[k+1] = \Pi_\Theta[\theta[k] - \rho_k \xi_f(\theta[k])], \tag{3}$$

where $\rho_k > 0$ is the step-size at iteration k, $\xi_f(\theta[k])$ is a sub-gradient, an element of sub-differentials $\partial_\theta f(\theta[k])$, of the cost function f with respect to the variable θ evaluated at $\theta[k]$ [30], and $\Pi_\Theta[\cdot]$ denotes projection operator into the set Θ defined as $\Pi_\Theta[a] := \arg\min_{b\in\Theta} \|a-b\|_2$. For continuously differentiable functions, the gradient is the only sub-gradient. The use of sub-gradients, instead of gradient in this paper, is motivated by the possible choice of non-differentiable loss functions in ML, e.g., the cost function of the L-SVM.

Assumption 2. *g_1 and g_2 are convex functions of θ.*

Assumption 2 implies that f is also a convex function of θ. The existence of sub-differentials is guaranteed for convex functions [30]. We define $\bar{g}_2^{x,y}(\theta) = g_2(\mathfrak{M}(x;\theta), y)$. The update law in (3) can be rewritten as

$$\begin{aligned}\theta[k+1] &= \Pi_\Theta\Bigg[\theta[k] - \rho_k \xi_{g_1}(\theta[k]) \\ &\qquad - \frac{\rho_k}{n} \sum_{\ell\in\mathcal{N}_j} \sum_{\{x,y\}\in\mathcal{D}_\ell} \xi_{\bar{g}_2^{x,y}}(\theta[k])\Bigg], \\ &= \Pi_\Theta\Bigg[\theta[k] - \rho_k \xi_{g_1}(\theta[k]) \\ &\qquad - \frac{\rho_k}{n} \sum_{\ell\in\mathcal{N}_j\setminus\{j\}} n_\ell \mathfrak{Q}_\ell(\mathcal{D}_\ell; k)\Bigg],\end{aligned} \tag{4}$$

where ξ_{g_1} is a sub-gradient of g_1, $\xi_{\bar{g}_2^{x,y}}$ is a sub-gradient of $\bar{g}_2^{x,y}$, and $\mathfrak{Q}_\ell(\mathcal{D}_\ell; k)$ is a query that can be submitted by the learning agent to data owner $\ell \in \mathcal{N}$ in order to provide the aggregate sub-gradient:

$$\mathfrak{Q}_\ell(\mathcal{D}_\ell; k) = \frac{1}{n_\ell} \sum_{\{x,y\}\in\mathcal{D}_\ell} \xi_{\bar{g}_2^{x,y}}(\theta[k]). \tag{5}$$

Responding to the query $\mathfrak{Q}_\ell(\mathcal{D}_\ell; k)$ clearly intrudes on the privacy of the individuals in dataset $\mathcal{D}_\ell$. Therefore, data owner ℓ only responds in a differentially-private manner by reporting the noisy aggregate:

$$\overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell; k) = \mathfrak{Q}_\ell(\mathcal{D}_\ell; k) + w_\ell[k], \tag{6}$$

where $w_\ell[k]$ is an additive noise to establish differential privacy with privacy budget ϵ_ℓ over the horizon T; see Definition 1. As stated before, here, the horizon T is the total number of iterations of the projected sub-gradient algorithm. Note that each neighbour responds to one query in each iteration.

Assumption 3. $\Xi := \max_{(x,y)\in\mathbb{X}\times\mathbb{Y}} \left\|\xi_{\bar{g}_2^{x,y}}(\theta[k])\right\|_1 < \infty$.

Algorithm 1 ML training algorithm with distributed private datasets using DP gradients for strongly-convex smooth fitness cost.

Require: T
Ensure: $(\theta[k])_{k=1}^T$
1: Initialize $\theta[1]$
2: **for** $k = 1, \ldots, T-1$ **do**
3: Learner submits query $\mathfrak{Q}_\ell(\mathcal{D}_\ell; k)$ to data owners in $\mathcal{N}$
4: Data owners return DP responses $\overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell; k)$
5: Learner follows the update rule

$$\theta[k+1] = \theta[k] - \frac{\rho}{T^2 k}\left(\xi_{g_1}(\theta[k]) + \sum_{\ell\in\mathcal{N}} \frac{n_\ell}{n} \overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell; k)\right),$$

6: **end for**

Assumption 3 implies the gradients or the sub-gradients of fitness function have a bounded magnitude. For strongly convexity loss functions with Lipschitz gradients, this assumption can be satisfied. This is because, for strongly convex functions, the decision variables, i.e., the ML model, remains within a compact set. However, for non-strongly convex functions, we need to restrict the ML models to the compact set Θ; see (1).

Theorem 1. *The policy of data owner ℓ in* (6) *for responding to the queries is ϵ_ℓ-differentially private over horizon $\{1, \ldots, T\}$ if $w_\ell[k]$ are i.i.d.*[2] *noises with the density function*

$$p(w) = \left(\frac{1}{2b}\right)^{p_\theta} \exp\left(-\frac{\|w\|_1}{b}\right)$$

with scale $b = 2\Xi T/(n_\ell \epsilon_\ell)$.

Proof. See Appendix A. □

Theorem 1 states that i.i.d. Laplace additive noise can ensure DP gradients. Each response in (6), for a given k, using the additive noise density in Theorem 1 is (ϵ_ℓ/T)-differentially private. Therefore, over the whole horizon $\{1, \ldots, T\}$, all the responses meet the definition of ϵ_ℓ-differential privacy. This follows from the composition of T differentially-private mechanisms [31]. In [20], [21], each response is constructed to ensure ϵ-differential privacy, which implies that the overall algorithm is ϵT-differentially private, thus reducing the privacy guarantee with increasing the number of the iterations.

In the presence of the additive noise, the iterates of the learner follow the stochastic map

$$\theta[k+1] = \Pi_\Theta[\theta[k] - \rho_k(\xi_f(\theta[k]) + w[k])], \tag{7}$$

where

$$w[k] := \frac{1}{n} \sum_{\ell\in\mathcal{N}} n_\ell w_\ell[k].$$

Algorithm 1 summarizes our proposed ML algorithm with distributed private datasets using DP gradients. Note that, in Algorithm 1, the step size, or the learning rate, decreases with the iteration number k. This is done to reduce the influence of

[2] independently and identically distributed

Algorithm 2 ML algorithm with distributed private datasets using DP sub-gradients.

Require: T, c_1
Ensure: $(\theta[k])_{k=1}^{T}$
1: Initialize $\theta[1]$ within Θ
2: **for** $k = 1, \ldots, T-1$ **do**
3: Learner submits query $\mathfrak{Q}_\ell(\mathcal{D}_\ell; k)$ to data owners in $\mathcal{N}$
4: Data owners return DP responses $\overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell; k)$
5: Learner follows the update rule

$$\theta[k+1] = \Pi_\Theta\left[\theta[k] - \frac{c_1}{\sqrt{k}}\left(\xi_{g_1}(\theta[k]) + \sum_{\ell\in\mathcal{N}} \frac{n_\ell}{n}\overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell; k)\right)\right],$$

6: Learner follows the averaging rule

$$\bar{\theta}[k+1] = \frac{k-1}{1/\sqrt{T}+k}\bar{\theta}[k] + \frac{1/\sqrt{T}+1}{1/\sqrt{T}+k}\theta[k].$$

7: **end for**

the privacy-preserving additive noise in the performance of the trained model. In the non-private training (i.e., when $\epsilon = +\infty$), we do not need to reduce the step size with iteration number k as there is no privacy-preserving noise. In fact, we can select a constant learning rate to extract the non-private model; see [32] for convergence analysis of optimization algorithms with constant steps sizes.

In Section III, we observe that the performance of Algorithm 1 can only be assessed under the assumptions of differentiability, smoothness, and strong convexity of the fitness cost. These assumptions are satisfied for several ML models and fitness costs, such as regression. To avoid these assumptions and to also reduce the effect of the additive noise, we can define the averaging variable

$$\begin{aligned}\bar{\theta}[k+1] &= \left(1 - \frac{1/\sqrt{T}+1}{1/\sqrt{T}+k}\right)\bar{\theta}[k] + \frac{1/\sqrt{T}+1}{1/\sqrt{T}+k}\theta[k] \\ &= \frac{k-1}{1/\sqrt{T}+k}\bar{\theta}[k] + \frac{1/\sqrt{T}+1}{1/\sqrt{T}+k}\theta[k]. \end{aligned} \tag{8}$$

Algorithm 2 summarizes the proposed ML algorithm with distributed private datasets using DP sub-gradients with the additional averaging step as per equation (8). Now, we are ready to analyze the performance our privacy-preserving ML training algorithms.

III. Predicting the Performance of ML on Distributed Private Data

For Algorithm 1, we can prove the following convergence result under the assumptions of differentiability, smoothness, and strong convexity of the ML fitness function.

Theorem 2. *Assume that f is a L-strongly convex continuously-differentiable function with λ-Lipschitz gradient and $\theta_{\max} = \infty$ (i.e., there is no constraint). For any $\varepsilon > 0$, there exists a large enough T such that the iterates of Algorithm 1 satisfy*

$$\min_{1\le k\le T} \mathbb{E}\{f(\theta[k])\} - f(\theta^*) \le \frac{8\Xi^2\rho}{Ln^2}\left(\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}\right) + \varepsilon, \tag{9}$$

and

$$\min_{1\le k\le T} \mathbb{E}\{\|\theta[k] - \theta^*\|_2^2\} \le \frac{32\Xi^2\rho}{L^2n^2}\left(\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}\right) + \frac{\varepsilon}{4L}. \tag{10}$$

Proof. See Appendix B. □

Theorem 2 establishes the convergence of Algorithm 1 for smooth strongly convex functions. This quantifies the *trade-off between privacy and utility* by capturing the closeness to the trained ML model with and without taking into account the privacy constraints of the data owners. In fact, the inequalities in (9) and (10) enable us to predict the outcome of a potential collaboration among privacy-aware data owners (or data custodians) in terms of the fitness cost of the ML training model prior to executing potentially computationally-expensive ML algorithms on distributed privately-owned datasets.

To relax the conditions required for convergence of the ML training, we can use Algorithm 2. In this case, we do not even need the fitness function to be differentiable because the algorithm uses sub-gradients, rather than gradients. For the noisy projected sub-gradient decent algorithm in Algorithm 2, the following result can be proved.

Theorem 3. *For any T, there exists large enough constants*[3] *$c_1, c_2 > 0$ such that the iterates of Algorithm 2 satisfy*

$$\mathbb{E}\{f(\bar{\theta}[T])\} - f(\theta^*) \le \frac{c_2\Xi}{n}\sqrt{\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}}, \tag{11}$$

Further, if g_1 is a L-strongly convex function,

$$\mathbb{E}\left\{\left\|\bar{\theta}[T] - \theta^*\right\|_2^2\right\} \le \frac{4c_2\Xi}{Ln}\sqrt{\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}}. \tag{12}$$

Proof. See Appendix C. □

The upper bounds on the performance of the training Algorithms 1 and 2 in Theorems 2 and 3 are increasing functions of $(1/n^2)\sum_{\ell\in\mathcal{N}} 1/(\epsilon_\ell)^2$ and $(1/n)[\sum_{\ell\in\mathcal{N}} 1/(\epsilon_\ell)^2]^{1/2}$, respectively. By increasing ϵ_ℓ, i.e., relaxing the privacy guarantees of data owners, the performance of the ML training algorithm improves, as expected because of having access to better quality gradient oracles.

Remark 2 (Comparison with Central Bounds)**.** *Under the assumption that all the data owners have equal privacy budgets $\epsilon_i = \epsilon$, $\forall i$, the bound in* (9) *scales as ϵ^{-2} and the bound in* (11) *scales as ϵ^{-1}. These bounds are in line with the lower and the upper bounds in [33] for strongly convex and general convex loss functions. The same outcome also*

[3]Note that the constants in the statement of the theorem can be functions of T and, therefore, the bounds in (11) and (12) are useful for comparing the variations in the performance of the sub-gradient descent algorithm for various privacy budgets and sizes of the datasets as long as T is fixed.

holds if $N = 1$ and $\epsilon_1 = \epsilon$, which is the case of centralized privacy-preserving learning.

Finally, we note that these results provide bounds on the distance between the non-private ML model and the privacy-preserving ML models learned in a distributed manner as a function of the privacy budgets and the size of the datasets. Issues, such as non-independent and non-identical datasets, influence the performance of the non-private model and thus also indirectly influence the performance of the privacy-preserving models. In the next section, although the datasets are not restricted be i.i.d. (e.g., the number of fraudulent transactions in the credit card fraud detection is low and arguably contains activities that have originated from same/similar fraudsters), the theoretical bounds tightly match the experimental results.

IV. Experimental Validation of the Performance of ML on Distributed Private Data

In this section, we examine the results of the paper, specifically the performance of Algorithm 2, on two financial datasets on lending and credit card fraud. Particularly, we use the relative fitness of the iterates in Algorithm 2 to illustrate its performance. The relative fitness of θ is given by

$$\psi(\theta) := \frac{f(\theta)}{f(\theta^*)} - 1. \tag{13}$$

This measure shows how good θ is in comparison to the optimal ML model θ^* in terms of the training cost in (1). We opt for studying the relative fitness, scaled by $f(\theta^*)$ as opposed as the absolute fitness $f(\theta) - f(\theta^*)$, because we consider datasets with different sizes for two distinct ML learning models and thus we want to factor out the effects of the variations of $f(\theta^*)$. Finally, note that, by construction, $\psi(\theta) \geq 0$. Further, the lower the value of $\psi(\theta)$, the better θ performs in comparison to θ^*. In what follows, we use Algorithm 1 with $\epsilon = +\infty$ for non-private learning of θ^*; this is equivalent to setting the magnitude of the additive privacy-preserving noise in the gradients to zero.

A. Lending Dataset

First, we use a lending dataset with a linear regression model to demonstrate the value of the methodology and to validate the theoretical results.

1) Dataset Description: The dataset contains information regarding nearly 890,000 loans made on a peer-to-peer lending platform, called the Lending Club, which is available on Kaggle [3]. The inputs contain loan attributes, such as total loan size, and borrower information, such as number of credit lines, state of residence, and age. The outputs are the interest rates of the loans per annum. We encode categorical attributes, such as state of residence and loan grade assigned by the Loan Club, with integer numbers. We also remove unique identifier attributes, such as id and member id, as well as irrelevant attributes, such as the uniform resource locator (URL) for the Loan Club page with listing data. Finally, we perform feature selection using the Principal Component Analysis (PCA) to select the top ten important features. This step massively improves the numerical stability of the algorithm. For the PCA, we only use the last ten-thousand entries of the dataset to ensure that the feature selection does not violate the distributed nature of the algorithm. Note that, if we were to use the entire dataset for the PCA, the data should have been available at one location for processing which is contradictory to the assumptions of the paper regarding the distributed nature of the dataset and the privacy requirements of the data owners. After performing the PCA, the eigenvectors corresponding to the most important features are communicated to the distributed datasets. The first n_1 entries of the Lending Club are assumed to be the private data of the first data owner. The entries between $n_1 + 1$ to $n_1 + n_2$ belong to the second data owner and the entries between $n_1 + n_2 + 1$ to $n_1 + n_2 + n_3$ are with the third data owner. Note that, by construct, these distributed datasets are non-overlapping, i.e., they do not share identical records. We may use any other approach for splitting the Lending Club dataset among the private data owners as long as the distributed datasets are not overlapping. The data owners then balance their datasets using the-said eigenvectors. The balancing refers to a transformation of the dataset using the eigenvectors to extract the most important independent features. The eigenvectors, here, serve as a common dictionary between the data owners for communication and training.

2) Experiment Setup: The experiments demonstrate the outcome of collaborations among $N = 3$ financial institutes, e.g., banks, for training a ML model to automate the process of assigning interest rates to loan applications based on the attributes of the borrower and the loan. Each institute has access to a private dataset of n_i historical loan applications and approved interest rates. The value of ϵ_i for each institute essentially determines eagerness for collaboration and openness to sharing private proprietary datasets. For a linear regression model, we consider a linear ML model relating the inputs and the outputs as in $y = \mathfrak{M}(x;\theta) := \theta^\top x$ with $\theta \in \mathbb{R}^{p_\theta}$ denoting the parameters of the ML model. We train the model by solving the optimization problem (1) with $g_2(\mathfrak{M}(x;\theta), y) = \|y - \mathfrak{M}(x;\theta)\|_2^2$, and $g_1(\theta) = 0$.

3) Results: First, we demonstrate the behaviour (e.g., convergence) of the iterates of the stochastic gradient descent procedure in Algorithm 2. Consider the case where $n_1 = n_2 = n_3 = 250,000$. Figure 2 shows the statistics of the relative fitness of the stochastic gradient method in Algorithm 2 for a ML model determining lending interest rates, $\psi(\bar{\theta}[k])$, versus the iteration number k for $T = 100$ for three choices of privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$ to illustrate the convergence of the learning algorithm as established in Theorem 3. The algorithm is stochastic because the data owners provide differentially-private responses to the gradient queries, obfuscated with Laplace noise in Theorem 1. Thus each run of the algorithm follows a different relative fitness trend. The boxes, i.e., the vertical lines at each iterations, illustrate the range of 25% to 75% percentiles of the relative fitness extracted from one-hundred runs of the algorithm. The black lines show the median relative fitness versus the iteration number. The effect of the privacy budgets on the quality of the iterates at the

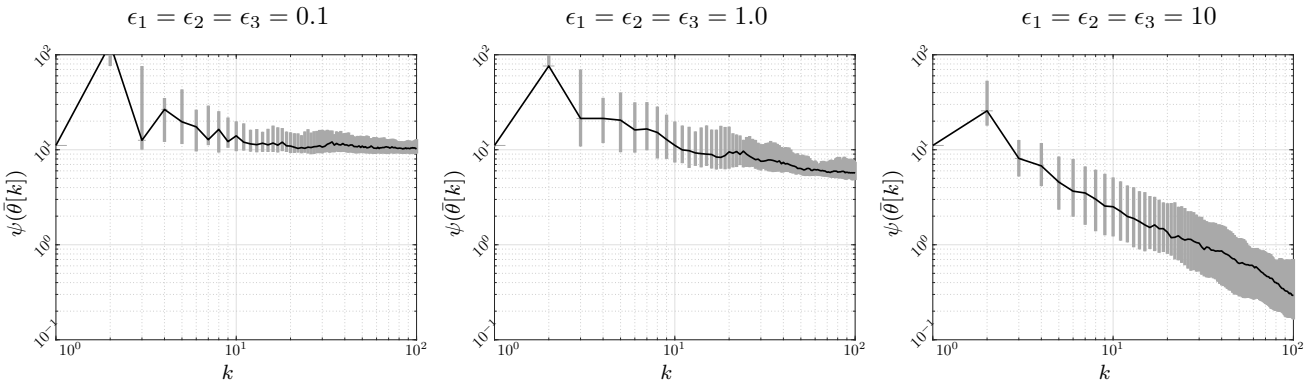

Fig. 2. Statistics of relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates versus the iteration number for $T = 100$ with various choices of privacy budgets. The boxes, i.e., the vertical lines at each iterations, illustrate the range of 25% to 75% percentiles for extracted from a hundred runs of the algorithm and the black lines show the median relative fitness.

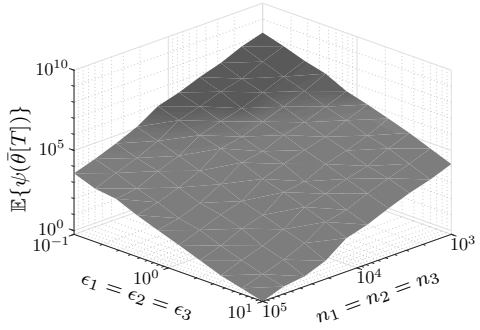

Fig. 3. Relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates after $T = 100$ iterations versus the size of the datasets and the privacy budgets.

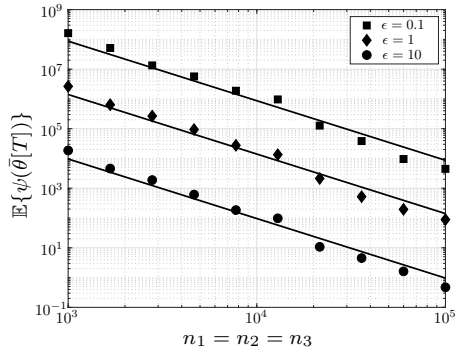

Fig. 5. Relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates after $T = 100$ iterations versus the size of the datasets. The solid line illustrate the bound in Theorem 2.

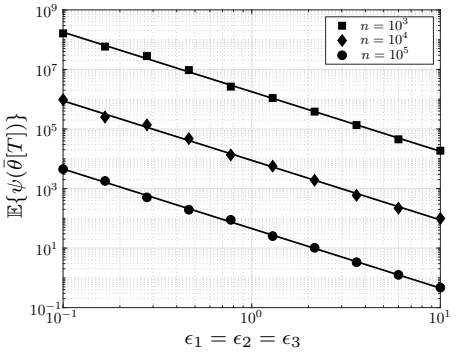

Fig. 4. Relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates after $T = 100$ iterations versus the privacy budgets. The solid line illustrate the bound in Theorem 2.

end of T iterations is evident, as expected from Theorem 3. As $\epsilon_1 = \epsilon_2 = \epsilon_3$ increases, i.e., the data owners become more willing to share data, the performance of the trained ML model improves. For instance, by increasing the privacy budget from $\epsilon_1 = \epsilon_2 = \epsilon_3 = 1$ to $\epsilon_1 = \epsilon_2 = \epsilon_3 = 10$, the relative fitness of the algorithm improves (i.e., decreases), on average, by approximately 100-fold.

After establishing the desired transient behaviour of the algorithm, we can investigate the effect of the size of the datasets and the privacy budgets on the performance of the trained ML model, i.e., the ML model after all the iterations have passed. Figure 3 shows the expectation (i.e., the statistical mean) of the relative fitness of the stochastic gradient method in Algorithm 2 for the trained ML model after $T = 100$ iterations versus the size of the datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. As predicted by Theorem 3, the fitness improves as the size of the datasets $n_1 = n_2 = n_3$ and/or the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$ increase. To quantify the tightness of the upper-bound in Theorem 3 for Algorithm 2, we isolate the effects of the size of the datasets and the privacy budgets on the relative fitness. Figure 4 illustrates the expectation of the relative fitness of the stochastic gradient method in Algorithm 2 after $T = 100$ iterations versus the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. In this figure, the markers (i.e., ■, ♦, and ●) are from the experiments and the solid lines are fitted to the experimental data. We can see that the slope of the linear lines in the log-log scale in Figure 4 is -2. This shows that $\psi(\bar{\theta}[k]) \propto \epsilon_i^{-2}$. Hence, our

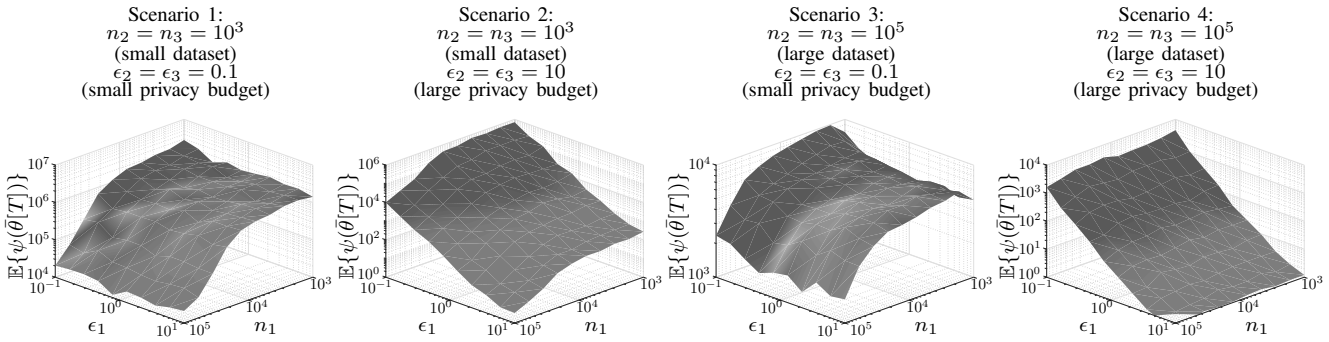

Fig. 6. Relative fitness of the stochastic gradient method in Algorithm 2 for learning lending interest rates after $T = 100$ iterations versus the size of the dataset and the privacy budget of the first data owner for four distinct scenarios of collaboration.

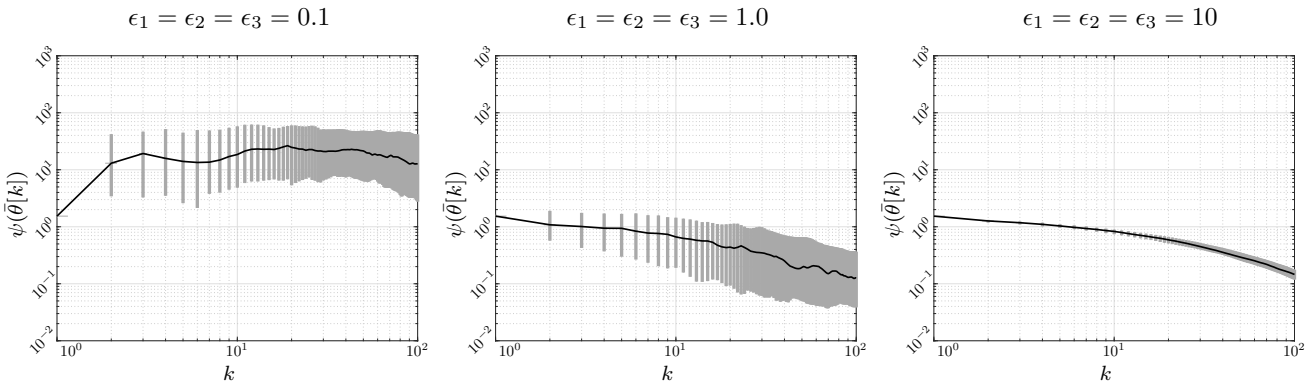

Fig. 7. Statistics of relative fitness of the stochastic gradient method in Algorithm 2 for fraud detection versus the iteration number for $T = 100$ with various choices of privacy budgets. The boxes, i.e., the vertical lines at each iterations, illustrate the range of 25% to 75% percentiles for extracted from a hundred runs of the algorithm and the black lines show the median relative fitness.

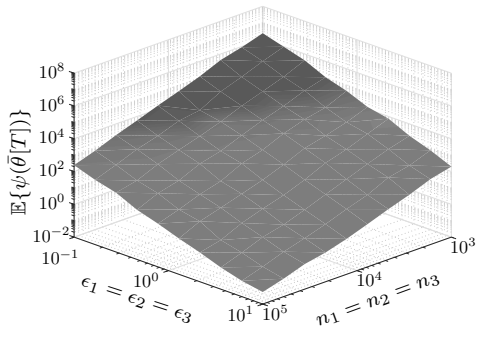

Fig. 8. Relative fitness of the stochastic gradient method in Algorithm 2 for fraud detection after $T = 100$ iterations versus the size of the datasets and the privacy budgets.

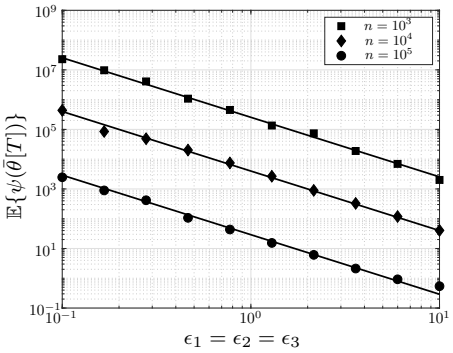

Fig. 9. Relative fitness of the stochastic gradient method in Algorithm 2 for fraud detection after $T = 100$ iterations versus the privacy budgets. The solid line illustrate the bound in Theorem 2.

bound in Theorem 3 is not tight as it states that $\psi(\bar{\theta}[k])$ is upper bounded by a function of the form ϵ_i^{-1}. This is because Theorem 3 does not use the fact that the cost function for the regression is strongly convex and has Lipschitz gradients. These assumptions are utilized in Theorem 2 and the bounds in this theorem are in fact tight, as Theorem 2 states that $\psi(\bar{\theta}[k])$ is upper bounded by a function of the form ϵ_i^{-2}. Figure 5 shows the expectation of the relative fitness of the stochastic gradient method in Algorithm 2 after $T = 100$ iterations versus the size of the datasets $n_1 = n_2 = n_3$. Similarly, the slop of the linear lines in the log-log scale in Figure 5 is -2 pointing to that $\psi(\bar{\theta}[k]) \propto n_i^{-2}$. This is again a perfect match for our theoretical bound in Theorem 2 (because $n = n_1 + n_2 + n_3 = 3n_i$).

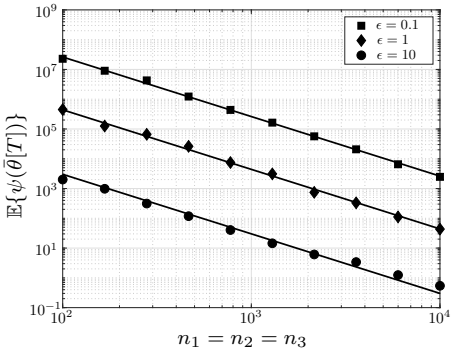

Fig. 10. Relative fitness of the stochastic gradient method in Algorithm 2 for fraud detection after $T = 100$ iterations versus the size of the datasets. The solid line illustrate the bound in Theorem 2.

Finally, we consider a few scenarios of collaboration for the data owners. Specifically, we evaluate the performance of the learning algorithm for four distinct scenarios in which the second and the third data owners have: (*i*) small datasets and small privacy budgets (i.e., reluctant to share due to privacy concerns); (*ii*) small datasets and large privacy budgets (i.e., eager to share); (*iii*) large datasets and small privacy budgets; (*iv*) large datasets and large privacy budgets. For each case, we vary the privacy budget and the size of the dataset of the first data owner. This allows us to investigate the potential benefit to data owners from collaboration in various scenarios. Figure 6 illustrates the expectation of the relative fitness of the stochastic gradient method in Algorithm 2, after $T = 100$ iterations, versus the size of the dataset n_1 and the privacy budget ϵ_1 for four distinct scenarios of collaboration. The first scenario in Figure 6 (the left most plot) shows that there is no point in collaboration with small data owners, even if the size of the dataset of the first data owner is large and it is eager to share its data; the relative fitness (capturing the distance between private ML model and the non-private model) is very large, it does not change significantly with ϵ_1, and it still remains large for relative large datasets $n_1 = 10^5$. We could foresee this from the bound in Theorem 3 without running Algorithm 2. This bound shows that $\psi(\bar{\theta}[k]) \propto 1/(2000+n_1)\sqrt{200 + 1/\epsilon_1^2}$; hence, no matter how large ϵ_1 gets (even if $\epsilon_1 = \infty$), the error's coefficient remains large due to small privacy budgets of the other two data owners and n_1 must become considerably large to compensate for it. In the second scenario (the second left most plot in Figure 6), the effect of ϵ_1 and n_1 are more pronounced. This is because, although the other two data owners are small, they do not hinder the learning process by adding large amounts of privacy-preserving noise because of their conservatively small privacy budgets. The third scenario is similar to the first one, albeit with better relative fitness as conservative data owners are relatively larger. The best scenario for collaboration, unsurprisingly, is the fourth scenario in which phenomenal performances can be achieved even without much consideration towards the size of the first dataset or its privacy budget as the other two datasets are large and eager to collaborate for learning.

B. Credit Card Fraud Detection

In this subsection, we use a credit card dataset with a L-SVM classifier to further demonstrate the value of the methodology and to validate the theoretical results.

1) Dataset Description: The datasets contains transactions made by European credit card holders in September 2013 available on Kaggle [4]. The inputs are vectors extracted by PCA (to avoid confidentiality issues) as well as the amount of the transaction. The output is a class, determining if the transactions was deemed fraudulent or not. The dataset is highly unbalanced, as the positive class (frauds) account for 0.172% of all transactions.

2) Experiment Setup: The experiments demonstrate the outcome of collaborations among $N = 3$ financial institutes for training a SVM classifier to detect fraudulent activities automatically and rapidly. Each institute has access to a private dataset of n_i historical credit card transactions and their authenticity. The value of ϵ_i for each institute determines eagerness for collaboration. In L-SVM, the model is $\mathfrak{M}(x;\theta) := \theta^\top [x^\top \ 1]^\top$, and $g_1(\theta) := (1/2)\theta^\top\theta$ and $g_2(\mathfrak{M}(x;\theta), y) := \max(0, 1 - \mathfrak{M}(x;\theta)y)$.

3) Results: First, we investigate the transient behaviour of the iterates of Algorithm 2. Assume that $n_1 = n_2 = n_3 = 30,000$. Figure 7 shows the statistics of the relative fitness of the iterates of Algorithm 2 for training a fraud detection SVM classifier, $\psi(\bar{\theta}[k])$, versus the iteration number k for $T = 100$ for three choices of privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The boxes, i.e., the vertical lines at each iterations, illustrate the range of 25% to 75% percentiles of relative fitness extracted from one-hundred runs of the algorithm and the black lines show the median relative fitness. As expected from Theorem 3, the performance of the trained SVM classifier gets closer to the SVM classifier trained with no privacy constraints θ^* as the privacy budgets increases.

Now, we can demonstrate the effect of the size of the datasets and the privacy budgets on the performance of the trained SVM classifier at the end of T training iterations. Figure 8 shows the expectation of the relative fitness of the stochastic gradient method in Algorithm 2 after $T = 100$ iterations versus the size of the datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. Similar to the theoretical results in Theorem 3, the fitness improves by increasing the size of the datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. We can also isolate the effects of the size of the datasets and the privacy budgets. Figure 9 illustrates the expectation of the relative fitness of the iterates of Algorithm 2 after $T = 100$ iterations versus the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. As all linear slopes in the log-log scale in Figure 9 are -2, the bound in Theorem 2 seems to be a perfect fit. Figure 5 shows the expectation of the relative fitness of the iterates of Algorithm 2 after $T = 100$ iterations versus the size of the datasets $n_1 = n_2 = n_3$ revealing the exact behaviour predicted in the bound in Theorem 2.

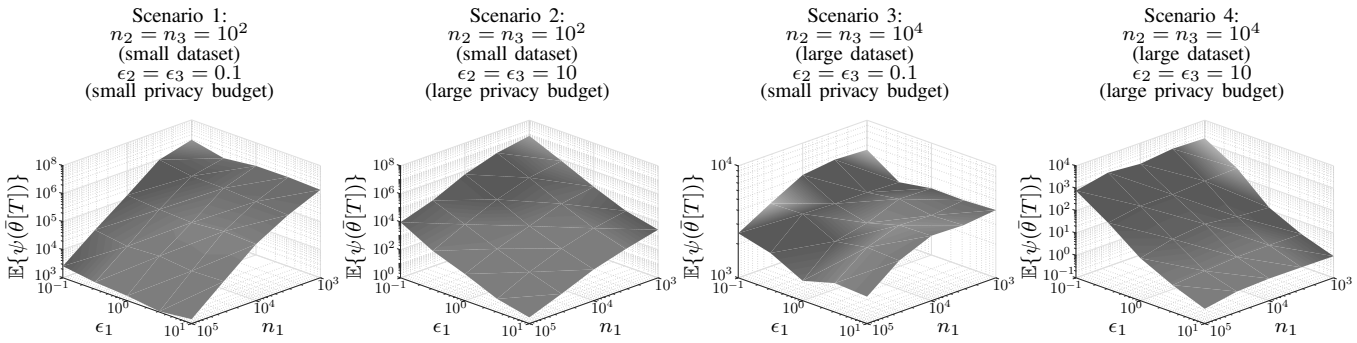

Fig. 11. Relative fitness of the stochastic gradient method in Algorithm 2 for a trained ML model determining lending interest rates after $T = 100$ iterations versus the size of the dataset and the privacy budget of the first data owner for four distinct scenarios of collaboration.

Finally, we evaluate the performance of the learning algorithm for four distinct scenarios, in which the second and the third data owners have: (*i*) small datasets and small privacy budgets; (*ii*) small datasets and large privacy budgets; (*iii*) large datasets and small privacy budgets; (*iv*) large datasets and large privacy budgets. Figure 11 illustrates the expectation of the relative fitness of Algorithm 2 after $T = 100$ iterations versus the size of the dataset n_1 and the privacy budget ϵ_1 for four distinct scenarios of collaboration. The first scenario in Figure 11 (the left most plot) illustrates that there is no point in collaboration with small data owners even if the size of the dataset of the first data owner is large and it is eager to share its data. In the second scenario (the second left most plot in Figure 11), the effect of ϵ_1 and n_1 are more pronounced because the privacy budgets of the second and the third data owners are large and thus they do not degrade the performance of the learning algorithm by injecting excessive privacy-preserving noise. The third scenario is again similar to the first one, albeit with better results as conservative data owners are relatively larger. The best scenario for collaboration, similar to the loan example, is the fourth scenario in which the training performances with and without privacy constraints are identical, so long as the dataset of the first subsystem is large, or its privacy budget is not too small.

V. Discussions, Conclusions, and Future Research

We considered privacy-aware optimization-based ML on distributed private datasets. We assumed that the data owners provide DP responses to gradient queries. The theoretical analysis of the proposed DP gradient descent algorithms provided a way for predicting the quality of ML models based on the privacy budgets and the size of the datasets. We proved that the difference between the training model with and without considering privacy constrains of the data owners is bounded by $(\sum_{\ell\in\mathcal{N}} n_\ell)^{-2} \sum_{\ell\in\mathcal{N}} \epsilon_\ell^{-2}$ in our proposed algorithms under smoothness and strong-convexity assumptions for the fitness cost. The empirical results with real-world financial datasets split between multiple institutes/banks while using regression and support vector machine models demonstrated that the relative fitness in fact follows ϵ_i^{-2} and n_i^{-2} for the proposed algorithm. This shows the tightness of the upper bounds on the difference between the trained ML models with and without privacy constraints from the theoretical analysis, which can be utilized for quantification of the privacy-utility trade-off in privacy-preserving ML.

Note that the data owners, themselves, can also play the role of the learner in Figure 1. In this case, the data owner who is interested in learning a model can query the other data owners to provide DP gradients to use for learning. Now, in this case, as the other data owners cannot access the trained model or the query responses, the data owner who is training the model can set its own privacy budget to infinity. Following this approach, by creating N copies of the algorithm discussed in this paper, we can remove the central learner and each data owner can learn its own ML model.

The results of this paper can be used or extended in multiple directions for future research:

- We can extend the framework to multiple learners aiming to train separate privacy-aware ML models with similar structures based on their own datasets and DP responses from other learners and private data owners. This is closer in nature to the distributed or federated ML framework over an arbitrary connected communication network. Note that, in this paper, the communication structure among the learner and the data owners is over a star graph with the learner at the center.
- The results of this paper can be used to understand the behaviour of data owners and learners in a data market for ML training. The utility-privacy trade-off in this paper, in terms of the quality of the trained ML models, can be used in conjunction with the cost of sharing private data of costumers with the learner (in terms of loss of reputation, legal costs, implementation of privacy-preserving mechanisms, and communication infrastructure) to setup a game-theoretic framework for modeling interactions across a data market. The learner can compensate the data owners for access to their private data, by essentially paying them for choosing larger privacy budgets. After negotiations between the data owners and the learners for setting the privacy budgets, the algorithm of this paper

can be used to then train ML models, while knowing in advance the expected quality of the trained model.

- Synchronous updates of the algorithm is indeed a bottleneck of the proposed algorithm. Future work can focus on extending the results of this paper to asynchronous gradient updates where, at each iteration, only a subset of the data owners update the ML model. To be able to ensure the convergence of the asynchronous algorithm, we need to ensure that all the data owners update the model as frequently as required.
- Another direction for future research is to extend the framework of this paper to adversarial learning scenarios that can admit more general adversaries (than the case of curious-but-honest adversaries in this paper).

ACKNOWLEDGEMENTS

We would like to thank Nicolas Papernot for shepherding our paper. His comments and suggestions greatly helped in improving the paper. The work has been funded, in part, by the "Data Privacy in AI Platforms (DPAIP): Risks Quantification and Defence Apparatus" project from the Next Generation Technologies Fund by the Defence Science and Technology (DST) in the Australian Department of Defence and the DataRing project funded by the NSW Cyber Security Network and Singtel Optus pty ltd through the Optus Macquarie University Cyber Security Hub.

REFERENCES

[1] C. J. Bennett and C. D. Raab, "Revisiting the governance of privacy: Contemporary policy instruments in global perspective," *Regulation & Governance*, 2018.

[2] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *International Conference on Machine Learning*, pp. 71–79, 2013.

[3] W. Kan, "Lending club loan data: Analyze lending club's issued loans." https://www.kaggle.com/wendykan/lending-club-loan-data, Date Accessed: 17 Oct 2018.

[4] Machine Learning Group–ULB, "Credit card fraud detection: Anonymized credit card transactions labeled as fraudulent or genuine." https://www.kaggle.com/mlg-ulb/creditcardfraud/home, Date Accessed: 27 Nov 2018.

[5] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in *Advances in Cryptology — CRYPTO 2000* (M. Bellare, ed.), (Berlin, Heidelberg), pp. 36–54, Springer Berlin Heidelberg, 2000.

[6] W. Du, Y. S. Han, and S. Chen, "Privacy-preserving multivariate statistical analysis: Linear regression and classification," in *Proceedings of the 2004 SIAM international conference on data mining*, pp. 222–233, SIAM, 2004.

[7] J. Vaidya and C. Clifton, "Privacy preserving association rule mining in vertically partitioned data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 639–644, ACM, 2002.

[8] J. Vaidya, M. Kantarcıoğlu, and C. Clifton, "Privacy-preserving naive bayes classification," *The VLDB Journal*, vol. 17, no. 4, pp. 879–898, 2008.

[9] G. Jagannathan and R. N. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 593–599, ACM, 2005.

[10] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, ACM, 2017.

[11] T. Graepel, K. Lauter, and M. Naehrig, "ML confidential: Machine learning on encrypted data," in *International Conference on Information Security and Cryptology*, pp. 1–21, Springer, 2012.

[12] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, "Chiron: Privacy-preserving machine learning as a service," *arXiv preprint arXiv:1803.05961*, 2018.

[13] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[14] Y. Aono, T. Hayashi, L. Wang, S. Moriai, *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.

[15] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International Conference on Machine Learning*, pp. 201–210, 2016.

[16] A. D. Sarwate and K. Chaudhuri, "Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data," *IEEE signal processing magazine*, vol. 30, no. 5, pp. 86–94, 2013.

[17] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: Regression analysis under differential privacy," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1364–1375, 2012.

[18] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Advances in Neural Information Processing Systems*, pp. 289–296, 2009.

[19] Z. Zhang, B. I. P. Rubinstein, and C. Dimitrakakis, "On the differential privacy of Bayesian inference," in *AAAI Conference on Artificial Intelligence*, pp. 2365–2371, 2016.

[20] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 172–187, 2017.

[21] Z. Huang, R. Hu, Y. Gong, and E. Chan-Tin, "DP-ADMM: ADMM-based distributed learning with differential privacy," *Preprint:* `arXiv preprint arXiv:1808.10101`, 2018.

[22] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, p. 4, 2015.

[23] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via functional perturbation," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 395–408, 2018.

[24] M. Hale and M. Egersted, "Differentially private cloud-based multi-agent optimization with constraints," in *Proceedings of the American Control Conference*, pp. 1235–1240, 2015.

[25] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321, ACM, 2015.

[26] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

[27] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[28] T. Zhang, Z. He, and R. B. Lee, "Privacy-preserving machine learning through data obfuscation," *arXiv preprint arXiv:1807.01860*, 2018.

[29] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 50–64, 2017.

[30] N. Z. Shor, *Minimization methods for non-differentiable functions*, vol. 3 of *Springer Series in Computational Mathematics*. Berlin, Heidelberg: Springer, 2012.

[31] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[32] B. Grimmer, "Convergence rates for deterministic and stochastic subgradient methods without lipschitz continuity," *SIAM Journal on Optimization*, vol. 29, no. 2, pp. 1350–1365, 2019.

[33] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473, IEEE, 2014.

[34] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization, Springer US, 2013.

APPENDIX A
PROOF OF THEOREM 1

First, because of (6), we have

$$\begin{aligned}\|\overline{\Omega}_\ell(\mathcal{D}_\ell;k)-\overline{\Omega}_\ell(\mathcal{D}'_\ell;k))\|_1 =&\frac{1}{n_\ell}\Bigg\|\sum_{\{x,y\}\in\mathcal{D}_\ell}\xi_{\bar{g}_2^{x,y}}(\theta[k])\\ &-\sum_{\{x,y\}\in\mathcal{D}'_\ell}\xi_{\bar{g}_2^{x,y}}(\theta[k])\Bigg\|_1\\ =&\frac{1}{n_\ell}\|\xi_{\bar{g}_2^{x,y}}(\theta[k])|_{\{x,y\}\in\mathcal{D}_\ell\subseteq\mathcal{D}'_\ell}\\ &-\xi_{\bar{g}_2^{x,y}}(\theta[k])|_{\{x,y\}\in\mathcal{D}'_\ell\subseteq\mathcal{D}_\ell}\|_1.\end{aligned}$$

This implies that $\|\overline{\Omega}_\ell(\mathcal{D}_\ell;k)-\overline{\Omega}_\ell(\mathcal{D}'_\ell;k))\|_1 \leq (2/n_\ell)\max_{\{x,y\}\in\mathcal{D}'_\ell\subseteq\mathcal{D}_\ell\cup\mathcal{D}_\ell\subseteq\mathcal{D}'_\ell}\|\xi_{\bar{g}_2^{x,y}}(\theta[k])\|_1 \leq 2\Xi/n_\ell$, where the last inequality follows from Assumption 3. Noting the exponential form of the Laplace random variable, we get

$$\begin{aligned}\frac{p((\overline{\Omega}_\ell(\mathcal{D}_\ell;k))_{k=1}^T)}{p((\overline{\Omega}_\ell(\mathcal{D}'_\ell;k))_{k=1}^T)}=&\prod_{k=1}^T\exp\left(\frac{\|\overline{\Omega}_\ell(\mathcal{D}'_\ell;k)\|_1}{b}-\frac{\|\overline{\Omega}_\ell(\mathcal{D}_\ell;k)\|_1}{b}\right)\\ \leq&\prod_{k=1}^T\exp(2\Xi/bn_\ell)\\ =&\exp(2\Xi T/bn_\ell),\end{aligned}$$

where, by some abuse of notation, $p(\cdot)$ denotes the probability density of the variable in its argument. Substituting $b=2\Xi T/(n_\ell\epsilon_\ell)$ in this inequality concludes the proof.

APPENDIX B
PROOF OF THEOREM 2

The magnitude of the DP noise is

$$\begin{aligned}\mathbb{E}\{\|w[k]\|_2^2\} =&\mathbb{E}\left\{\left\|\left(\frac{1}{\sum_{\ell\in\mathcal{N}}n_j}\right)\sum_{j\in\mathcal{N}}n_\ell w_\ell[k]\right\|_2^2\right\}\\ =&\left(\frac{1}{\sum_{\ell\in\mathcal{N}}n_j}\right)^2\sum_{\ell\in\mathcal{N}}n_\ell^2\mathbb{E}\{\|w_\ell[k]\|_2^2\}\\ =&\left(\frac{1}{\sum_{\ell\in\mathcal{N}}n_j}\right)^2\sum_{\ell\in\mathcal{N}}\frac{8\Xi^2T^2}{\epsilon_\ell^2}\\ =&\frac{8\Xi^2T^2}{n^2}\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}.\end{aligned}$$

Because ∇f is λ-Lipschitz, $f(z_1)\leq f(z_2)+\nabla f(z_2)^\top(z_1-z_2)+0.5\lambda\|z_2-z_1\|_2^2$ for all z_1,z_2 [34] and therefore

$$\begin{aligned}\mathbb{E}\{f(\theta[k+1])\} \leq&\mathbb{E}\{f(\theta[k])\}\\ &+\mathbb{E}\{\nabla f(\theta[k])^\top(\theta[k+1]-\theta[k])\}\\ &+\frac{\lambda}{2}\mathbb{E}\{\|\theta[k+1]-\theta[k]\|_2^2\}\\ \leq&\mathbb{E}\{f(\theta[k])\}\\ &+\rho_k\left(\frac{\lambda\rho_k}{2}-1\right)\mathbb{E}\{\|\nabla f(\theta[k])\|_2^2\}\\ &+\rho_k^2\frac{8\Xi^2T^2}{n^2}\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}.\end{aligned}$$

For all $\rho_k\leq 1/\lambda$, we have

$$\begin{aligned}\mathbb{E}\{f(\theta[k+1])\} \leq&\mathbb{E}\{f(\theta[k])\}-\frac{\rho_k}{2}\mathbb{E}\{\|\nabla f(\theta[k])\|_2^2\}\\ &+\rho_k^2\frac{8\Xi^2T^2}{n^2}\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}.\end{aligned}\tag{14}$$

For $\varepsilon>0$, we may define

$$k_0:=\inf_k\left\{k\,\middle|\,\mathbb{E}\{\|\nabla f(\theta[k])\|_2^2\}\leq\frac{16\Xi^2T^2\rho_k}{n^2}\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}+\varepsilon\right\}.$$

Here, k_0 is the iteration number at which the magnitude of the last term in the right hand side of (14) (a positive value) becomes larger than the magnitude of the second to the last term in the right hand side of (14) (a negative value). In essence, at k_0, the upper bound on the cost function does not reduce. If T is large enough, we can easily show that there exists $k_0<\infty$. This can be proved by contrapositive. Assume that this not the case. Therefore,

$$\begin{aligned}\lim_{k\to 0}\mathbb{E}\{f(\theta[k])\} =&\mathbb{E}\{f(\theta[1])\}\\ &+\sum_{t=2}^k(\mathbb{E}\{f(\theta[t])\}-\mathbb{E}\{f(\theta[t-1])\})\\ \leq&\mathbb{E}\{f(\theta[1])\}-\sum_{t=2}^k\varepsilon\rho_k\\ =&-\infty.\end{aligned}$$

This is however not possible. Since f is L-strongly convex, Polyak-Lojasiewicz inequality [34] implies that

$$\begin{aligned}\mathbb{E}\{f(\theta[k_0])\}-f(\theta^*)\leq&\frac{1}{2L}\mathbb{E}\{\|\nabla f(\theta[k])\|_2^2\}\\ \leq&\frac{8\Xi^2T^2\rho_k}{Ln^2}\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}+\frac{\varepsilon}{2L}.\end{aligned}$$

Now, because $k_0\leq T$, we get

$$\begin{aligned}\min_{1\leq k\leq T}\mathbb{E}\{f(\theta[k])\}-f(\theta^*)\leq&\mathbb{E}\{f(\theta[k_0])\}-f(\theta^*)\\ \leq&\frac{8\Xi^2T^2\rho_k}{Ln^2}\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}+\frac{\varepsilon}{2L}.\end{aligned}$$

Again, because f is L-strongly convex, we can see that

$$\begin{aligned}f(\theta^*)\leq&f(t\theta+(1-t)\theta^*)\\ \leq&tf(\theta)+(1-t)f(\theta^*)-\frac{L}{2}t(t-1)\|\theta-\theta^*\|_2^2,\end{aligned}$$

for all $t\in(0,1)$. Setting $t=1/2$ results in

$$\|\theta-\theta^*\|_2^2\leq 4(f(\theta)-f(\theta^*))/L.\tag{15}$$

Hence,

$$\begin{aligned}\min_{1\leq k\leq T}\|\theta[k]-\theta^*\|_2^2\leq&\frac{4}{L}\left(\min_{1\leq k\leq T}\mathbb{E}\{f(\theta[k])\}-f(\theta^*)\right)\\ \leq&\frac{32\Xi^2T^2\rho_k}{L^2n^2}\sum_{\ell\in\mathcal{N}}\frac{1}{\epsilon_\ell^2}+\frac{\varepsilon}{8L^2}.\end{aligned}$$

This concludes the proof.

APPENDIX C
PROOF OF THEOREM 3

The proof for this theorem follows from modification of the results of [2]. In fact, the inequality in (11) follows from the result of [2] using the optimal selection of c in [29]. The only difference with the proofs in [2] is to appreciate that

$$\zeta_k - \zeta_{k-1} \leq \frac{2}{\sqrt{T}T(T+1)},$$

where

$$\zeta_k := \frac{1/\sqrt{T}+1}{1/\sqrt{T}+k} \prod_{m=k+1}^{T} \frac{m-1}{1/\sqrt{T}+m}.$$

The inequality follows from that

$$\begin{aligned}
\zeta_k - \zeta_{k-1} &= \frac{(1/\sqrt{T})(1/\sqrt{T}+1)}{(k-1+1/\sqrt{T})(k+1/\sqrt{T})} \\
&\quad \times \prod_{m=k+1}^{T} \frac{m-1}{1/\sqrt{T}+m} \\
&= \frac{(1/\sqrt{T})(1/\sqrt{T}+1)}{(k-1+1/\sqrt{T})(k+1/\sqrt{T})} \\
&\quad \times \frac{\prod_{m=k+1}^{T}(m-1)}{\prod_{m=k+1}^{T}(1/\sqrt{T}+m)} \\
&= (1/\sqrt{T})(1/\sqrt{T}+1)\frac{\prod_{m=k}^{T-1} m}{\prod_{m=k-1}^{T}(1/\sqrt{T}+m)} \\
&= (1/\sqrt{T})(1/\sqrt{T}+1)\frac{\prod_{m=k}^{T-1} m}{\prod_{m=k}^{T+1}(1/\sqrt{T}+m-1)} \\
&= \frac{(1/\sqrt{T})(1/\sqrt{T}+1)}{T(T+1)} \prod_{m=k}^{T+1} \frac{m}{(1/\sqrt{T}+m-1)} \\
&\leq \frac{2}{\sqrt{T}} \frac{1}{T(T+1)}.
\end{aligned}$$

If f is L-strongly convex, the proof of the inequality in (12) follows from (15).