The Cost of Privacy in Asynchronous Differentially-Private Machine Learning

Farhad Farokhi, Nan Wu, David Smith, and Mohamed Ali Kaafar

***Abstract*—We consider training machine learning models using data located on multiple private and geographically-scattered servers with different privacy settings. Due to the distributed nature of the data, communicating with all collaborating private data owners simultaneously may prove challenging or altogether impossible. We consider differentially-private asynchronous algorithms for collaboratively training machine-learning models on multiple private datasets. The asynchronous nature of the algorithms implies that a central learner interacts with the private data owners one-on-one whenever they are available for communication without needing to aggregate query responses to construct gradients of the entire fitness function. Therefore, the algorithm efficiently scales to many data owners. We define the cost of privacy as the difference between the fitness of a privacy-preserving machine-learning model and the fitness of trained machine-learning model in the absence of privacy concerns. We demonstrate that the cost of privacy has an upper bound that is inversely proportional to the combined size of the training datasets squared and the sum of the privacy budgets squared. We validate the theoretical results with experiments on financial and medical datasets. The experiments illustrate that collaboration among more than 10 data owners with at least 10,000 records with privacy budgets greater than or equal to 1 results in a superior machine-learning model in comparison to a model trained in isolation on only one of the datasets, illustrating the value of collaboration and the cost of the privacy. The number of the collaborating datasets can be lowered if the privacy budget is higher.**

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

I. Introduction

Unprecedented abundance of data has ignited a machine learning (ML) race that aims to boost productivity and spur economic growth globally. However, the data required for training such ML models is often distributed across multiple independent competing entities, e.g., financial or energy data is often scattered across servers for several service providers with competing interests. Regulatory frameworks, such as the GDPR, are increasingly restricting migration of private data across companies or even geographical boundaries for possible merger and training. This might restrict ML techniques from accessing the data in its entirety for training models, which motivates the development of distributed ML techniques with privacy guarantees.

Training data for machine learning can be located on multiple private geographically-scattered servers with different privacy settings. For instance, the training data can be gathered by Internet of Things (IoT) devices or hosted locally on smart devices with privacy settings enforced by users. Another example is cross-sector or -services ML with cross-governance datasets. In these cases, communicating with all private data owners simultaneously when training ML models is unpractical, if not impossible. A learner (i.e., a central agent responsible for training ML models) needs to resort to asynchronous communication with the different data owners. This implies that the learner can communicate with the data owners on a one-on-one basis without needing to wait for all data owners to respond. When using a gradient descent algorithm for training the ML model, the asynchronous communication raises an important challenge: the learner no longer knows the direction for the best model update based on all the training dataset; it can only infer the best update direction for the communicating data owner.

In this paper, we investigate the fitness of asynchronous ML learning algorithms. The learner updates the model based on differentially-private (DP) [1], [2] gradient of only the part of the fitness that depends on the data possessed by the communicating data owner. To address the challenge of not knowing the direction for the best model update, the learner updates the ML model with small, yet constant, learning rates. The learner also shows inertia in updating its ML model so that it does not change the model significantly because of the gradient of just one data owner. These choices are motivated by that the learner is not overly confident that an update that is good for one data owner is also good for the others. The constant learning rate and the inertia of the learner allow the gradients of all the data owners to get mixed with across time so that the learner follows the direction for the best model update.

In this paper, we focus on ϵ-differential privacy instead of the relaxed notion of (ϵ, δ)-differential privacy, which is known to admit less conservative composition bounds [3]. These advanced composition bounds have been used to develop moment accountants for differentially-private machine learning [4]. The composition bounds stem from the relationship between Rényi differential privacy [5] and approximate differential privacy [6]. The improved composition bounds can have significant impact in practice as they can allow running more iterations under

F. Farokhi is with the Dept of Electrical and Electronic Engineering at the University of Melbourne, Australia. E-mail: Farhad.Farokhi@unimelb.edu.au

N. Wu is with the Macquarie University and CSIRO's Data61, Australia. E-mail: Nan.Wu@data61.csiro.au

D. Smith is with the CSIRO's Data61 and the Australian National University, Australia. E-mail: David.Smith@data61.csiro.au

M. A. Kaafar is with the Macquarie University and CSIRO's Data61, Australia. E-mail: Dali.Kaafar@data61.csiro.au

Manuscript received January 4, 2021.

The work has been funded 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 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. Aspects of this work were done while F. Farokhi held a joint appointment with CSIRO's Data61.

the same privacy budget, and this can help to improve the performance of the trained models. This has motivated many researchers to focus on improving the composition bounds [7]. Although very powerful, the improved composition bounds do not change the way the performance degradation is related to the privacy budget ϵ and, at best, improve the performance degradation by a *constant factor* [8]. To see this, consider the centralized privacy-preserving machine learning in [8]. In this case, the performance degradation caused by privacy-preserving noise is $\mathcal{O}(\epsilon^{-1})$ for both $(\epsilon, 0)$-differential privacy and (ϵ, δ)-differential privacy (when the cost function is smooth enough). However, the constant term behind ϵ^{-1} changes from being a linear function of the dimension of the model parameters to the square root of the dimension. Since, in this work, we are interested in investigating the effect of the privacy budget and the size of the training dataset (not the dimension of the ML model parameters) on the performance of privacy-preserving ML models trained by asynchronous distributed communication, we focus on ϵ-differential privacy to simplify the algebraic derivations behind the presented performance bounds. An important direction for future work is to utilize (ϵ, δ)-differential privacy and investigate the trade-off further.

Note that, in this paper, we only investigate honest-but-curious threats in which the data owners do not trust the learner or each other for sharing private datasets while they trust that the learner trains the model correctly based on a specified algorithm. For instance, in a financial sector, a central bank or a government organisation may be trusted for training ML models from distributed datasets but financial organisations prefer not to share their original data with the bank nor with each other. Likewise, in the health sector, a government organisation may be trusted to play the role of the central learner. For more general settings, incentives must be provided to ensure that the learner follows the training algorithm [9], [10].

The difference between the fitness function evaluated for privacy-preserving ML model and the fitness function evaluated for trained ML model without privacy concerns, or the degradation caused in the performance of ML models by the presence of differential privacy noise, captures the cost of privacy. In this paper, we prove that the cost of privacy is inversely proportional to the combined size of the training datasets squared and the privacy budgets squared. We validate the theoretical results on experiments on financial data. We use linear regression on a dataset of loan information from the Lending Club, a peer-to-peer lending platform, for setting interest rates of loans based on attributes, such as loan size and credit rating. We also use regression models on a dataset of hospital visits by patients in the U.S for determining the length of stay based on attributes, such as age, gender, and diagnosis. We show that, for collaboration among large numbers of private data owners, i.e., more than 10 data owners with at least 10,000 records, and with relatively large privacy budgets, i.e., privacy budgets greater than 1, the performance of the private ML model can beat the performance of a model that is trained with no collaboration. Therefore, we establish the value of collaboration in ML between multiple private data owners. Finally, we use a credit card fraud dataset to demonstrate the applicability of the results to support vector machines. We show that, if the privacy budget is 10 times larger, the performance degradation caused by the privacy-preserving noise becomes 10 times smaller (i.e., if the data owners are less conservative in terms of sharing private data, we can train far superior ML models). This illustrates the inverse relationship between the performance degradation and the privacy budget, which is also theoretically proved in the paper. Interestingly, the theoretical bounds regarding the relationship between the performance degradation and the size of the training dataset are not as tight for private support vector machines trained distributedly (i.e., in experiments, the performance degradation reduces more than 100 folds or more if the training dataset is only 10 times larger). This is perhaps caused by non-smooth nature of the fitness function (i.e., the fitness is not differentiable everywhere).

II. Related Work

ML with Differential Privacy: Previous work [11]–[15] studied ML training under the differential privacy framework. These approaches require merging the private datasets for training and rely on obfuscating the generated ML model using DP once the training on the aggregated data is performed. Alternatively, an ML model based on the obfuscated, yet merged data is trained. These studies do not consider the need for privacy preservation prior to merging the data. In addition they *do not consider the asynchronous nature of the communication between the learner and the data owners* by only requiring responses to some queries on the private dataset.

Distributed/Collaborative Privacy-Preserving ML: Distributed privacy-preserving ML proposes the use of DP gradients for training ML models [4], [16]–[24]. Noisy DP gradients can be used to train ML models with convex and non-convex fitness functions [19]–[21]. An important aspect of these studies is that they sometimes use better DP composition methods, such as moment accountant, for reducing the scale of the DP noise [4]. *These studies however propose synchronous updates* in which the ML model must be updated according to the contributions of all the data owners simultaneously (rather than a subset of them). This assumption can prohibit the use of the above distributed or collaborative ML training algorithms in the presence of numerous data owners. They also *do not provide a forecast for the performance of privacy-preserving trained models*. Note that, although on the surface the problem formulation in this paper seems a similar to [19], in this paper, we consider asynchronous communication by allowing the learner to communicate with the datasets in a one-on-one basis whenever they are available, which was not considered in [19]. Also, the techniques behind the proofs are entirely different and novel to this paper. Here, we also use health data in addition to financial datasets in [19] to further experimentally validate the theoretical results in naturally distributed privacy-aware environments. The availability for communication is particularly modeled using Poisson point processes. These processes are often utilized for analysis of asynchronous multi-agent systems and are shown to mimic practical scenarios [25]–[27].

Asynchronous Distributed Optimization and ML: Distributed asynchronous optimization algorithms can be used for training ML models [25], [28]–[32]. This is because we can formulate

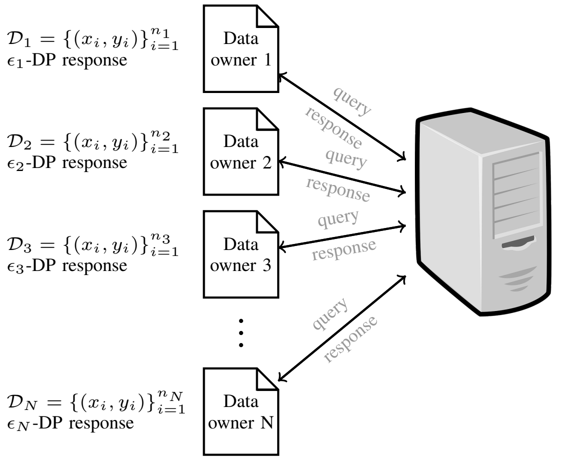

Fig. 1. Communication structure between a central learner and multiple data owners with private datasets.

distributed ML training as a distributed optimization problem with private datasets represented as parts of the fitness function. These algorithms are however generic and do not address the issue of selecting learning rate for ML training with DP gradients and forecasting the quality of the trained ML model based on dataset sizes and privacy budgets. Forecasting the performance of privacy-preserving ML algorithms can be used to understand the value of collaboration between distributed private datasets. Without such forecasts the private data owners might need to forgo their private datasets so that a trusted third-party can compare the performance of the private ML model with the ML model trained in absence of privacy concerns (as otherwise there is no ground truth for comparison in general). Asynchronous optimization has been also utilized in the past for ML purposes; see, e.g., [33]–[35]. These studies however do not consider additive DP noises and their impact on quality of trained ML models. Another asynchronous ML training algorithm was presented in [36]. Although that paper provides a rigorous privacy analysis, it only uses numerical results to investigate the performance degradation caused by privacy-preserving noise and bounds on forecasting the quality of trained ML models based on dataset sizes and privacy budgets are missing.

III. Asynchronous ML Training with DP

We consider $N \in \mathbb{N}$ private data owners connected to a central learning node, referred to as learner, responsible for training a ML model.

Figure 1 depicts the communication structure between the learner and the private data owners. The set of data owners is denoted by $\mathcal{N} := \{1, \dots, N\}$. The data owners possess a private training dataset composed of inputs x_i and outputs y_i. The dataset is denoted by $\mathcal{D}_i := \{(x_i, y_i)\}_{i=1}^{n_i} \subseteq \mathbb{X} \times \mathbb{Y} \subseteq \mathbb{R}^{p_x} \times \mathbb{R}^{p_y}$.

Informally, an ML model is a meaningful relationship between inputs and outputs in a training dataset. The ML model is $\mathfrak{M}(\cdot;\theta)$ for some mapping $\mathfrak{M} : \mathbb{X} \times \mathbb{R}^{p_\theta} \to \mathbb{Y}$ with $\theta \in \mathbb{R}^{p_\theta}$ denoting the parameters of the ML model. The learner in Figure 1 aims to train the ML model $\mathfrak{M}(\cdot;\theta)$ based on the available training datasets $\mathcal{D}_i$, $\forall i \in \mathcal{N}$, by solving the optimization problem in

$$\theta^* \in \operatorname*{arg\,min}_{\theta \in \Theta} f(\theta), \tag{1}$$

where $\Theta := \{\theta \in \mathbb{R}^{p_\theta} \mid \|\theta\|_\infty \leq \theta_{\max}\}$ and $f : \mathbb{R}^{p_\theta} \to \mathbb{R}$ is the fitness for ML model parameter θ, i.e., the fitness of ML model $\mathfrak{M}(\cdot;\theta)$ for relating the inputs and outputs in the training dataset $\cup_{j \in \mathcal{N}} \mathcal{D}_j$, given by

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

In the fitness (2), $g(\theta)$ is a regularizing term, $\ell(\mathfrak{M}(x;\theta), y)$ is a loss function capturing the distance between the output of the ML model $\mathfrak{M}(x;\theta)$ and the true output y, and $n = \sum_{j \in \mathcal{N}} n_j$. Finally, note that we can select a large enough $\theta_{\max}$ so that, if desired, training on Θ does not add any conservatism (in comparison to the unconstrained case).

Remark 1. *Note that, in this paper, the training data is not required to be i.i.d.*[1]*. Therefore, the proved performance degradation bounds due to privacy-preserving noise are valid for unbalanced and non-i.i.d datasets. This is because we are considering the performance degradation in terms of the training dataset fitness. Non-i.i.d datasets can reduce the performance in terms of generalization error, or test dataset fitness. Investigating the effect of privacy-preserving noise in generalization error is left as future work. In this case, we particularly need to focus on non-i.i.d. or unbalanced datasets to develop results are relevant to practice.*

In what follows, we present our ML learning algorithm for solving (1). To do so, the learner must update the ML model based on the gradient of the fitness function (2). Noting that the learner might not have the communication and computational capacities required to interact with all the data owners at the same time, we consider the design constraint of having an asynchronous interaction between the data owners and the learner. The asynchronous setup implies that the learner can only communicate with one of the data owners at each given iteration and thus can only access the gradient of the part of the fitness that depends on the data possessed by the communicating data owner. This makes the task of updating the ML model challenging as the learner would not know the direction for the best model update. In fact, an update direction that is good for one data owner might not be good for all the others. To alleviate this problem, the learner updates the ML model with small, yet constant, learning rates. It also shows inertia in updating the ML model to avoid significant changes because of the gradient of just one data owner. The constant learning rate and the inertia of the learner allow the gradients of all the data owners to get mixed with across time so that the learner follows the direction for the best model update. In the remainder of this section, we clarify all the steps in the algorithm.

We model the internal clock of the data owner by Poisson point processes with rates of one. These clocks are not in any

[1] independently and identically distributed

form synchronized and equal rate (of one) for the clocks simply implies that the data owners communicate with the learner with equal probability (not at equal times). At random times, the Poisson processes instigate communication between the data owners and the learner on a one-on-one basis. The Poisson process model is often utilized for analysis of asynchronous multi-agent systems [25]–[27]. Let the time instants in which the data owners communicate with the learner be given by

$$0 = t_1 \leq t_2 \leq \cdots \leq t_k \leq \cdots \leq t_T.$$

At each time instant t_k, $k \in \mathbb{N}$, one the data owners at random communicates with the learner. We use the notation $i_k \in \mathbb{N}$ to denote the index of that data owner that is communicating with the learner at time instance t_k.

Two approaches can be utilized in the asynchronous communication. One approach is **broadcasting by the learner**. In this scenario, the learner, in regular time intervals, broadcasts gradient queries to all data owners (some might be listening while others not). Whenever one of the data owners responds, the index k is incremented. Let t_k denote the time at which the communication takes place and i_k denote the index of the communicating data owner. Another approach is **requesting for update by the data owner**. In this scenario, the leaner is constantly listening for requests of update. Whenever a data owner submits a request, the index k is incremented with t_k denoting the time and i_k denoting the index of the data owner. At this point, the learner only communicates with that data owner until the update is over.

Remark 2. *In some applications, there might be a large number of data owners, such as mobile phones, and it would be inefficient for the the central learner to communicate with only one device at each time. In this case, the central learner can choose to communicate with multiple data owners simultaneously. If the central learner does not change the sets of data owners with which it simultaneously communicates and averages their gradients together, we can model the sets of data owners as larger fictitious data owners that communicate with the central learner on a one-by-one basis. In this scenario, the results of the paper would still be relevant and the efficiency of the communication significantly increases.*

At each iteration, the learner submits a gradient query of the form

$$\mathcal{Q}_{i_k}(\mathcal{D}_{i_k};\theta) := \frac{1}{n_{i_k}} \sum_{\{x,y\}\in\mathcal{D}_{i_k}} \nabla_\theta \ell(\mathfrak{M}(x;\theta),y) \in \mathcal{Q} \tag{3}$$

to the communicating data owner $i_k \in \mathcal{N}$. Here, $\mathcal{Q}$ is the output space of the queries. The communicating data owner $i_k \in \mathcal{N}$ provides the DP response

$$\overline{\mathcal{Q}}_{i_k}(\mathcal{D}_{i_k};\theta) = \mathcal{Q}_{i_k}(\mathcal{D}_{i_k};\theta) + w_{i_k}(k) \tag{4}$$

to the gradient query $\mathfrak{Q}_{i_k}(\mathcal{D}_{i_k};\theta)$. Here, $w_{i_k}(k)$ is a privacy-preserving additive noise to ensure DP.

Algorithm 1 Asynchronous ML learning using DP gradients for strongly-convex smooth fitness cost.

```
Require: $T \in \mathbb{N}$, $\rho \in \mathbb{R}_{\geq 0}$
Ensure: $(\theta_{1,k}, \theta_{2,k}, \dots, \theta_{N,k}, \theta_{L,k})_{k=1}^T$
1: Learner: Initialize $\theta_{1,0} = \cdots = \theta_{N,0} = \theta_{L,0} = 0$
2: for $k = 1, \dots, T$ do
3:     Randomly at uniform select data owner $i_k$
4:     Learner: Compute $\bar{\theta}_k$ according to (6)
5:     Learner: Submit gradient query $\mathcal{Q}_{i_k}(\mathcal{D}_{i_k};\bar{\theta}_k)$ to data owner $i_k$ according to (3)
6:     Data owner $i_k$: Provide DP response according to (4)
7:     Learner: Update ML models according to (5) and (7)
8: end for
```

Definition 1 (Differential Privacy)**.** *Responses of data owner $\ell \in \mathcal{N}$ are ϵ_ℓ-differentially private (or ϵ_ℓ-DP) over the horizon T if*

$$\mathbb{P}\left\{(\overline{\mathfrak{Q}}_\ell(\mathcal{D}_\ell;k))_{k:i_k=\ell} \in \mathcal{Y}\right\} \leq \exp(\epsilon_\ell)\mathbb{P}\left\{(\overline{\mathfrak{Q}}_\ell(\mathcal{D}'_\ell;k))_{k:i_k=\ell} \in \mathcal{Y}\right\},$$

where $\mathcal{Y}$ is any Borel-measurable subset of $\mathcal{Q}^{|\{k:i_k=\ell\}|}$, and $\mathcal{D}_\ell$ and $\mathcal{D}'_\ell$ are two adjacent datasets differing at most in one entry, i.e., $|\mathcal{D}_\ell \setminus \mathcal{D}'_\ell| = |\mathcal{D}'_\ell \setminus \mathcal{D}_\ell| \leq 1$.

Remark 3. *Note that (ϵ,δ)-differential privacy admits less conservative composition bounds for differential privacy; see moment accountant [4]. However, these composition bounds do not change the way the performance degradation is related to the privacy budget ϵ and, at best, improve the performance degradation by a constant. For instance, if we consider the centralized privacy-preserving machine learning [8], the performance degradation caused by privacy-preserving noise is $\mathcal{O}(\epsilon^{-1})$ for both $(\epsilon,0)$-differential privacy and (ϵ,δ)-differential privacy (when the cost function is smooth enough). Therefore, in this work, we focus on ϵ-differential privacy to simplify the algebraic derivations behind the presented performance bounds.*

We make the following standing assumptions throughout the paper for the purpose of theoretical analysis.

Assumption 1. *$g(\theta)$ is σ strongly convex in θ and $\ell(\mathfrak{M}(x;\theta),y)$ is convex in θ.*

Assumption 2. *The following properties hold:*

1) $\Xi_g := \sup_{\theta\in\Theta} \|\nabla_\theta g(\theta)\|_1 < \infty$*;*
2) $\Xi := \sup_{\theta\in\Theta} \sup_{(x,y)\in\mathbb{X}\times\mathbb{Y}} \|\nabla_\theta \ell(\mathfrak{M}(x;\theta),y)\|_1 < \infty.$

Note that, because $\|x\|_2 \leq \|x\|_1$ for all vectors x [37, Lemma 7.1, p. 121], Assumption 2 also implies that $\|\nabla_\theta g(\theta)\|_2 \leq \Xi_g$ and $\|\nabla_\theta \ell(\mathfrak{M}(x;\theta),y)\|_2 \leq \Xi$. We use these inequalities extensively in the proofs.

Assumption 3. *$T \in \mathbb{N}$ is the maximum number of iterations for communication between data owners and learner.*

Theorem 1. *The policy of data owners in line 6 of Algorithm 1 for responding to the queries over the horizon $\{1,\dots,T\}$ is*

ϵ_i-DP, $\forall i \in \mathcal{N}$, if $w_i(k)$ are statistically independent Laplace noises with scale $2\Xi T/(n_i\epsilon_i)$.

Proof. See Appendix A. □

We consider an approach in which the leaner keeps track of a central ML model, i.e., $\theta_{L,k}$, and N copies of it for each data owners, i.e., $\theta_{i,k}$ for each $i = 1, \dots, N$. This is motivated by the algorithm in [29] that forms the basis of our ML training algorithm. The local copies are only updated when the corresponding data owner is communicating with the learner. This is to keep track of the updates for each data owner. The update for the local ML model is given by

$$\theta_{i_k,k} = \Pi_\Theta \Bigg[\bar{\theta}_k - \frac{N\rho}{T^2\sigma} \Bigg(\frac{1}{2N} \nabla_\theta g(\bar{\theta}_k) + \frac{n_{i_k}}{n} \overline{\mathcal{Q}}_{i_k}(\mathcal{D}_{i_k}; \bar{\theta}_k) \Bigg) \Bigg], \quad (5)$$

where

$$\bar{\theta}_k = \frac{1}{2}(\theta_{L,k-1} + \theta_{i_k,k-1}). \quad (6)$$

Note that the learner updates the ML model with small, yet constant, learning rates. The learner also shows inertia in updating the central ML model so that it does not change the model significantly because of the gradient of just one data owner. The update for the central ML model is given by

$$\theta_{L,k} = \Pi_\Theta \left[\bar{\theta}_k - \frac{(N-1)\rho}{NT^2\sigma} \nabla_\theta g(\bar{\theta}_k) \right]. \quad (7)$$

The constant learning rate and the inertia of the learner allow the gradients of all the data owners to get mixed with each other across time so that the learner follow the direction for the best model update. All the steps of the learner and the data owners for generating queries, responding to the queries, and using the DP responses for updating the ML model are summarized in Algorithm 1. Finally, note that in step 3 of Algorithm 1 it states that the data owners are selected uniformly at random. This is compatible with the Poisson process clocks. The first data owner whose Poisson clock ticks communicates with the learner and because of the symmetry this happens with equal probability between the data owners (hence, in each iteration, one of the agents with uniform probability communicates with the learner).

IV. Performance of Private ML Models

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

Theorem 2. *For any N, there exist constants[2] $c_1, c_2, c_1', c_2' > 0$ such that the iterates of Algorithm 1 satisfy*

$$\mathbb{E}\{\|\theta_{L,T} - \theta^*\|_2^2\} \leq c_1 \sqrt{\frac{1}{T^2} + N \sum_{i \in \mathcal{N}} \left(\frac{1}{T} + \frac{2\sqrt{2}}{n\epsilon_i} \right)^2} + c_2 \left(\frac{1}{T^2} + N \sum_{i \in \mathcal{N}} \left(\frac{1}{T} + \frac{2\sqrt{2}}{n\epsilon_i} \right)^2 \right). \quad (8)$$

and

$$\mathbb{E}\{f(\theta_{L,T})\} - f(\theta^*) \leq c_1' \sqrt{\frac{1}{T^2} + N \sum_{i \in \mathcal{N}} \left(\frac{1}{T} + \frac{2\sqrt{2}}{n\epsilon_i} \right)^2} + c_2' \left(\frac{1}{T^2} + N \sum_{i \in \mathcal{N}} \left(\frac{1}{T} + \frac{2\sqrt{2}}{n\epsilon_i} \right)^2 \right). \quad (9)$$

Proof. See Appendix B. □

For large enough learning horizon T, the upper bound (8) takes the form of

$$\mathbb{E}\{\|\theta_{L,T} - \theta^*\|_2^2\} \leq \frac{\bar{c}_1}{n} \sqrt{\sum_{i \in \mathcal{N}} \frac{1}{\epsilon_i^2}} + \frac{\bar{c}_2}{n^2} \left(\sum_{i \in \mathcal{N}} \frac{1}{\epsilon_i^2} \right), \quad (10)$$

where $\bar{c}_1 = \sqrt{8N} c_1$ and $\bar{c}_2 = 8Nc_2$. Similarly, for large T, the upper bound (8) takes the form of

$$\mathbb{E}\{f(\theta_{L,T})\} - f(\theta^*) \leq \frac{\bar{c}_1'}{n} \sqrt{\sum_{i \in \mathcal{N}} \frac{1}{\epsilon_i^2}} + \frac{\bar{c}_2'}{n^2} \left(\sum_{i \in \mathcal{N}} \frac{1}{\epsilon_i^2} \right), \quad (11)$$

where again $\bar{c}_1' = \sqrt{8N} c_1'$ and $\bar{c}_2' = 8Nc_2'$. This takes the form of the performance bound in [19]. Under the assumption that all the data owners have equal privacy budgets $\epsilon_i = \epsilon$, $\forall i$, the bound in (11) scales as ϵ^{-2}. This bound matches the lower and upper bounds in [8] for strongly convex loss functions. The same outcome also holds if $N = 1$ and $\epsilon_1 = \epsilon$ which captures centralized privacy-preserving learning.

Remark 4. *In practice, the data owners might drop out and reconnect due to communication failures [38] and, hence, some data owners communicate with the learner less frequently. In that case, if we scale the gradients of each datasets inversely proportional to its frequency of communication, we get the same bounds. If we do not scale accordingly, the datasets with higher communication frequency get a larger weight in the updates. We have commented on this in the revised paper.*

We can introduce the cost of privacy (CoP) as the difference of the fitness for privacy-preserving ML model and the fitness for trained ML model in the absence of privacy concerns. The inequalities in (10) and (11) show that CoP is inversely proportional to the combined size of the training datasets squared and the sum of the privacy budgets squared.

[2] See the proof of the theorem in the supplementary materials for the exact constants.

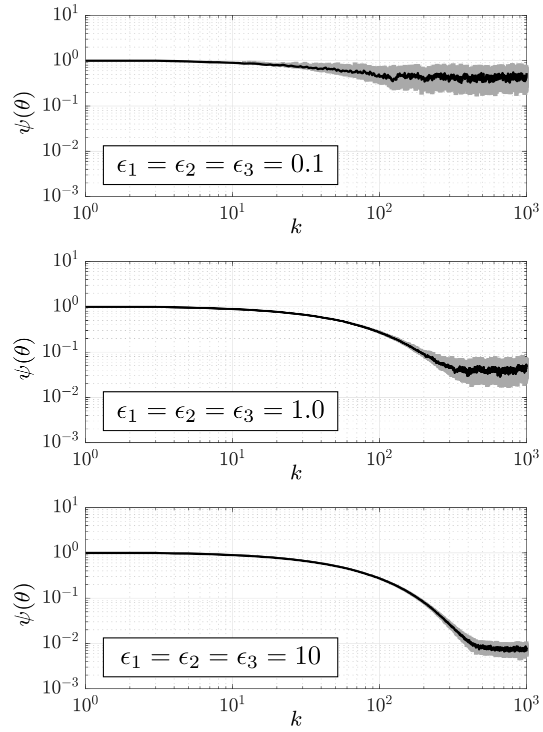

Fig. 2. Percentile statistics of relative fitness of 100 runs of Algorithm 1 for learning lending-interest-rates versus the iteration number k for a learning horizon of $T = 1,000$ iterations with three choices of privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The gray area illustrates the range of 25% to 75% percentiles and the black line shows the median of relative fitness.

V. Experimental Validation

In this section, we investigate the performance of Algorithm 1 on real datasets from the financial and health domains. In our experiments, the datasets have significantly different sizes and the size of the training datasets influence the performance of both non-private and private ML models. Hence, we factor out the effects of the size of the training datasets on the performance of the learning by only considering the relative fitness, defined as $\psi(\theta) := f(\theta)/f(\theta^*) - 1$. This measure captures the quality of any ML model θ in comparison to the non-private ML model θ^* in terms of the fitness in (2). By definition, $\psi(\theta) \geq 0$ for any ML model θ. The larger $\psi(\theta)$, the worse the performance of ML model θ in comparison with the non-private ML model θ^*.

A. Lending Dataset (Financial)

We first train a linear regression model on lending datasets as an example of automating banking processes requiring access to sensitive private datasets.

1) Dataset Description and Pre-Processing: We use a dataset of anonymized loan application information from roughly 890,000 individuals [39]. We remove unique identifiers, such as id and member id, and irrelevant attributes, such as URL addresses. We endeavour to train a linear regression model on this dataset. The input to the regression model are loan information, such as loan size, and applicant information, such as credit rating, state of residence and age. The model estimates the annual interest rate for the loans. We encode categorical attributes, such as state of residence and loan grade, with integer numbers.

In order to improve the numerical stability of the algorithm, we use Principal Component Analysis (PCA) to perform feature selection. We select the top ten important features. For this step, we only use the last ten-thousand entries of the dataset. We can assume that these entries are known to the learner and thus do not violate the distributed nature of the algorithm. This would have been a restrictive assumption if the learner used the entire dataset for the PCA (because the data owners must have agreed to perform PCA in collaboration without privacy concerns, which is contradiction with their original interest for privacy-preserving ML). Using the PCA, the learner can construct a dictionary for feature selection and communicate it to private data owners.

2) Experiment Setup and Results: We start with an experiment evaluating the outcome of collaborations between $N = 3$ banks. We use the linear regression model $y = \mathfrak{M}(x;\theta) := \theta^\top x$ with θ denoting the model parameters. The fitness function is given by $g_2(\mathfrak{M}(x;\theta), y) = \|y - \mathfrak{M}(x;\theta)\|_2^2$, and $g_1(\theta) = 10^{-5}\theta^\top\theta$. The first data owner is assumed to possess the first n_1 entries of the dataset. The second data owner owns entries ranging from $n_1 + 1$ to $n_1 + n_2$. Finally, the third data owner has access to entries between $n_1 + n_2 + 1$ to $n_1 + n_2 + n_3$ as its private dataset.

We start with demonstrating the convergence of Algorithm 1 when $n_1 = n_2 = n_3 = 250,000$. Figure 2 illustrates the percentile statistics of the relative fitness $\psi(\theta_{L,k})$ for 100 runs of Algorithm 1 versus the iteration number k for the learning horizon $T = 1,000$. Note that, in Algorithm 1, only one of the data owners communicates with the learner in each iteration. Figure 3 illustrates an example of communication timing for the asynchronous learning in Algorithm 1, illustrating i_k versus the iteration number k. Recalling the stochastic nature of the algorithm, due to the DP noise in query responses, the relative fitness varies for each run of the algorithm. The gray area in Figure 2 shows 25%–75% percentiles of the relative fitness. The black solid lines in Figure 2 shows the median of relative fitness versus the iteration number. The median decreases across time until the algorithm converges to a neighbourhood of the solution of (1). The relative fitness of the trained model also improves as $\epsilon_1 = \epsilon_2 = \epsilon_3$ increases. Note that smaller privacy budgets also increase the variations in the relative fitness (i.e., larger gray area).

Figure 4 illustrates the average relative fitness of the trained ML model using Algorithm 1 after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the size of the private datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The mesh surface shows the bound in (11) with $\bar{c}_1' = 0$ and $\bar{c}_2' = 2.1 \times 10^9$. This figure clearly shows the tightness of the result of Theorem 2. Note that, as expected, the relative fitness rapidly improves as the sizes of the datasets $n_1 = n_2 = n_3$ or the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$ increase.

Let us isolate the effects of the size of the datasets and the privacy budgets. Figure 5 shows the average relative fitness of the trained ML model using Algorithm 1 after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$ [top] and the size of the datasets $n_1 = n_2 = n_3$ [bottom].

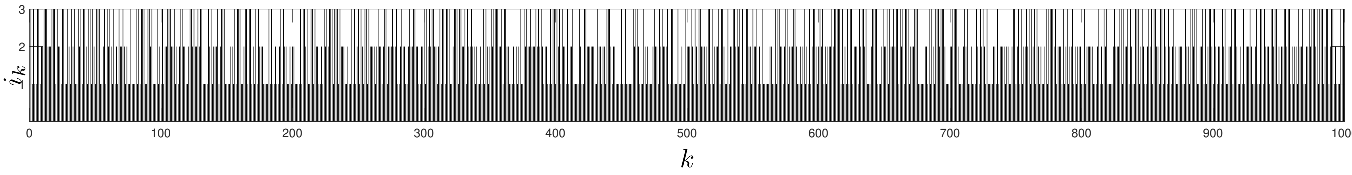

Fig. 3. Example of communication timing for the asynchronous learning in Algorithm 1 for learning lending-interest-rates, illustrating i_k versus the iteration number k.

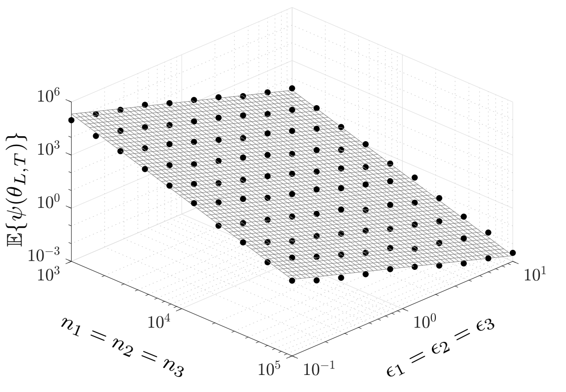

Fig. 4. Relative fitness of Algorithm 1 for learning lending-interest-rates after $T = 1,000$ iterations versus the size of the datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The mesh surface illustrates the bound in (11).

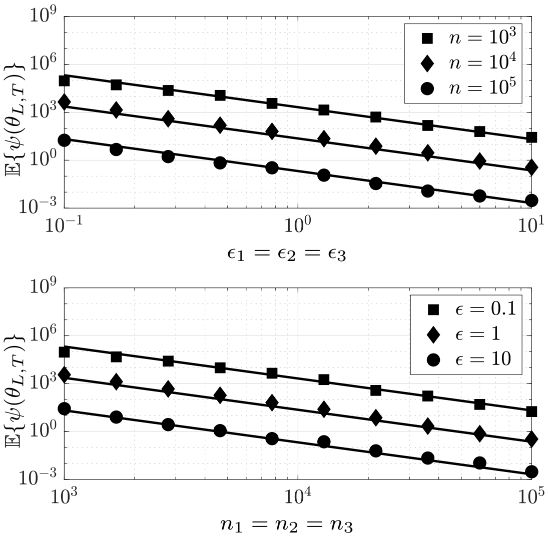

Fig. 5. Relative fitness of Algorithm 1 for learning lending-interest-rates after $T = 1,000$ iterations versus the privacy budget [top] and the size of the datasets [bottom]. The solid line illustrates the bound in (11).

In this figure, the markers (i.e., ■, ◆, and ●) are from the experiments and the solid show the bound in (11). For both these cases, the bounds in Theorem 2 are tight fits. Therefore, the theoretical results in Theorem 2 match the experiments.

Let us also demonstrate the value of collaboration between among many banks. Consider an experiment with N banks each with $n_i = 10,000$ records collaborating to train a regression model. Figure 6 shows the average relative fitness of Algorithm 1 for learning lending-interest-rates after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the privacy budgets ϵ_i, $\forall i$, and the number of the collaborating data owners N. The solid gray surface shows the relative fitness of the non-private ML model θ_1^*, $\psi(\theta_1^*)$, constructed based on only the private data of the first data owner. Note that $\psi(\theta_1^*)$ is not random (as its construction does not require DP noise) and is not a function of ϵ_i. If the relative fitness of Algorithm 1 is smaller than the relative fitness of the non-private ML model θ_1^*, collaboration benefits the first data owner, which is illustrated by the black region at the bottom of the figure. Evidently, the first data owner benefits from collaboration if there are more than 5 data owners with privacy budgets greater than or equal to 10 or if there are more than 100 data owners with privacy budgets greater than or equal to 2.5.

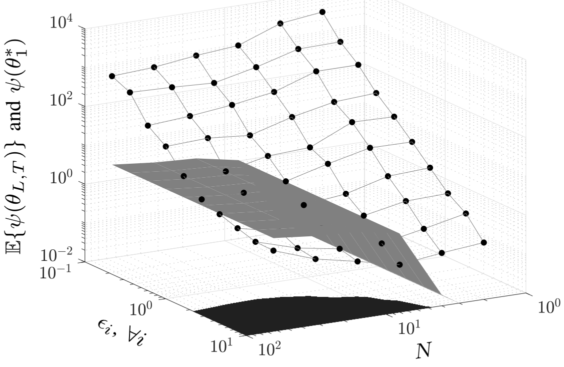

Fig. 6. Relative fitness of Algorithm 1 for learning lending-interest-rates after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the privacy budgets ϵ_i, $\forall i$, and the number of collaborating data owners N. The solid gray surface shows the relative fitness of the non-private ML model θ_1^*, $\psi(\theta_1^*)$, constructed based on only the private data of the first data owner. If the relative fitness of Algorithm 1 is smaller than the relative fitness of the non-private ML model θ_1^*, collaboration benefits the first data owner (illustrated by the black region at the bottom of the figure).

B. Health-related Data

Now, we use the hospital admission and discharge dataset from the New York State to validate the theoretical results.

1) Dataset Description and Pre-Processing: The dataset contains hospital visit and discharge information from nearly 2,350,000 de-identified patients including information, such as characteristics, diagnoses, treatments, services, and charges. This dataset is made public by the Bureau of Health Informatics [40]. We train a linear regression model, as in the previous subsection, with inputs, such as age, gender, race, ethnicity,

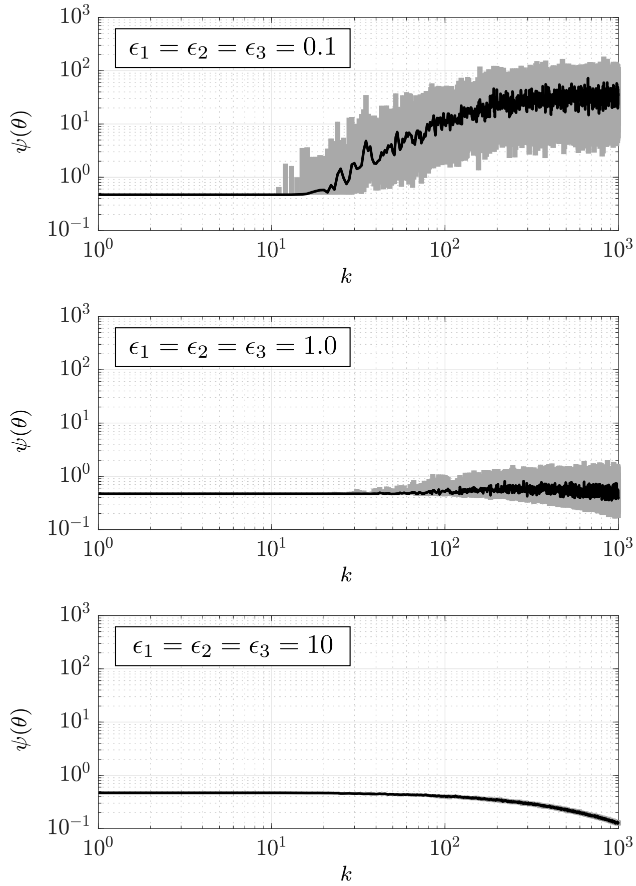

Fig. 7. Percentile statistics of relative fitness of 100 runs of Algorithm 1 for learning length of stay at hospital versus the iteration number k for a learning horizon of $T = 1,000$ iterations with three choices of privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The gray area illustrates the range of 25% to 75% percentiles for the relative fitness and the black line shows the median of relative fitness.

diagnosis code, procedure code, and drug code, to automatically determine the length of stay. This can be used as a tool for determining the capacity of hospitals in the future based on currently admitted patients. Similarly, we encode categorical attributes, such as gender and ethnicity, with integer numbers. We also remove attributes, such as total charges and costs, as well as irrelevant attributes, such as the postcode. Similar to the lending data, in order to improve the numerical stability of the algorithm, we perform the PCA to balance the features. We do so based on the last fifty-thousand entries of the dataset to ensure that the feature selection does not violate the distributed nature of the algorithm.

2) Experiment Setup and Results: The data in [40] is tagged by the hospital name and code. There are 213 hospitals in the dataset. We focus on 86 hospital with at least 10,000 records. Experiments on the convergence of the algorithms and the tightness of theoretical bounds are similar to the lending data and are therefore eliminated due to space constraints. They are however presented as supplementary material.

We demonstrate the performance of the iterates of Algorithm 1. Figure 7 illustrates the percentile statistics of the relative fitness $\psi(\theta_{L,k})$ for 100 runs of Algorithm 1 versus the iteration number k for the learning horizon $T = 1,000$. Figure 8 illustrates an example of communication timing for the asynchronous learning in Algorithm 1, illustrating i_k versus the iteration number k. At each iteration, only one of the 86 data owners communicates with the learner. The gray area in Figure 7 shows 25%–75% percentiles of the relative fitness and the black solid lines in show the median of relative fitness versus the iteration number. For large privacy budgets, the median decreases across time until the algorithm converges to a neighbourhood of the solution of (1). The relative fitness of the trained model also improves as $\epsilon_1 = \epsilon_2 = \epsilon_3$ increases.

Figure 9 shows the average relative fitness of the trained ML model using Algorithm 1 after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, versus the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. The markers (i.e., ■) show the experiments. Evidently, the relative fitness rapidly improves as the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$ increase.

Figure 10 illustrates the relative fitness of Algorithm 1 for learning length of stay at hospital after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, for three choices of privacy budgets $\epsilon_i = 0.1$ (black line), $\epsilon_i = 1$ (dashed line), $\epsilon_i = 10$ (dash-dotted line). The markers show the relative fitness of the non-private ML model θ_i^*, $\psi(\theta_i^*)$, constructed based on only the private data of the i-th data owner versus the size of the data set owned by the i-th data owner. For $\epsilon = 10$, eight hospitals (i.e., Women And Children's Hospital Of Buffalo, Crouse Hospital, St Peters Hospital, White Plains Hospital Center, Westchester Medical Center, Memorial Hospital for Cancer and Allied Diseases, Long Island Jewish Schneiders Children's Hospital Division, St Francis Hospital) benefit from collaboration. The relative fitness of the non-private ML model θ_i^* for these eight hospitals are above the dash-dotted line.

C. Credit Card Fraud Data

In this subsection, we illustrate the theoretical results by asynchronous training of a linear support vector machine classifier on a credit card dataset.

1) Dataset Description and Pre-Processing: The datasets contains transactions made by European credit card holders in September 2013 [41]. The dataset contains the amount of the transaction and vectors extracted by PCA (to avoid confidentiality issues). We are interested in determining whether a transaction was fraudulent or not.

2) Experiment Setup and Results: The experiments demonstrate the outcome of collaborations among $N = 3$ entities for training a linear support vector machine classifier to detect credit card fraud. The linear support vector machine model is $\mathfrak{M}(x;\theta) := \theta^\top [x^\top \; 1]^\top$, and $g(\theta) := (10^{-5}/2)\theta^\top\theta$, and $\ell(\mathfrak{M}(x;\theta), y) := \max(0, 1 - \mathfrak{M}(x;\theta)y)$.

Figure 11 shows the expectation of the relative fitness of the method in Algorithm 1 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 expected, the fitness improves by increasing the size of the datasets $n_1 = n_2 = n_3$ and the privacy budgets $\epsilon_1 = \epsilon_2 = \epsilon_3$. However, the theoretical bounds illustrated in Figure 11 [bottom] are not as tight as the previous experiments when $n_1 = n_2 = n_3$ gets large. This is perhaps caused by non-smooth nature of the fitness function (i.e., the fitness is not differentiable everywhere).

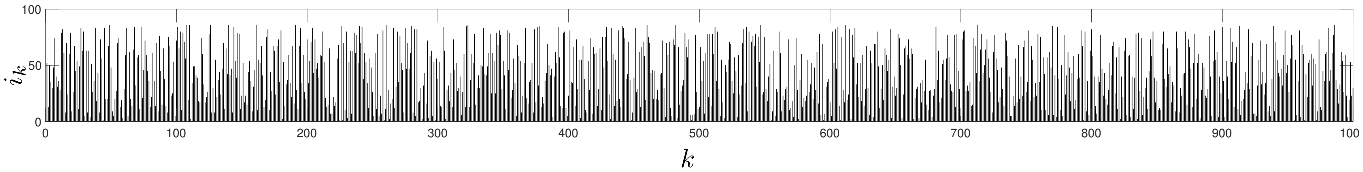

Fig. 8. Example of communication timing for the asynchronous learning in Algorithm 1 for learning length of stay at hospital, illustrating i_k versus the iteration number k.

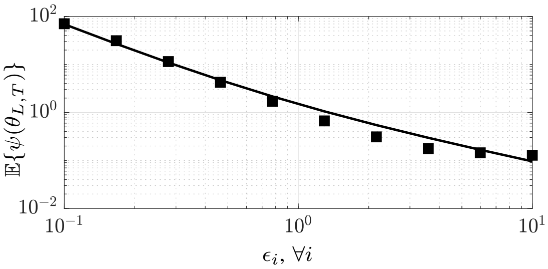

Fig. 9. Relative fitness of Algorithm 1 for learning length of stay at hospital after $T = 1,000$ iterations versus the privacy budget ϵ_i, $\forall i$. The solid line illustrates the bound in (11).

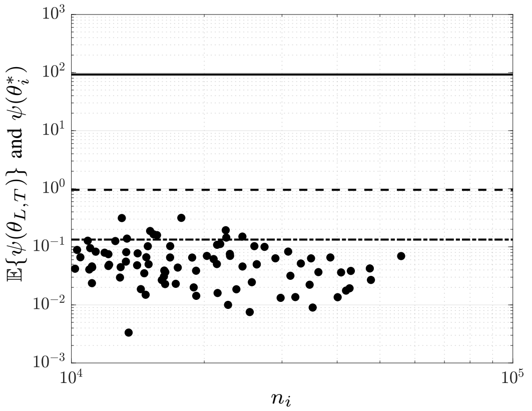

Fig. 10. Relative fitness of Algorithm 1 for learning length of stay at hospital after $T = 1,000$ iterations, $\mathbb{E}\{\psi(\theta_{L,T})\}$, for three choices of privacy budgets $\epsilon_i = 0.1$ (black line), $\epsilon_i = 1$ (dashed line), $\epsilon_i = 10$ (dash-dotted line). The markers show the relative fitness of the non-private ML model θ_i^*, $\psi(\theta_i^*)$, constructed based on only the private data of the i-th data owner versus the size of the data set owned by the i-th data owner. For $\epsilon = 10$, eight hospitals benefit from collaboration. The relative fitness of the non-private ML model θ_i^* for these eight hospitals are above the dash-dotted line.

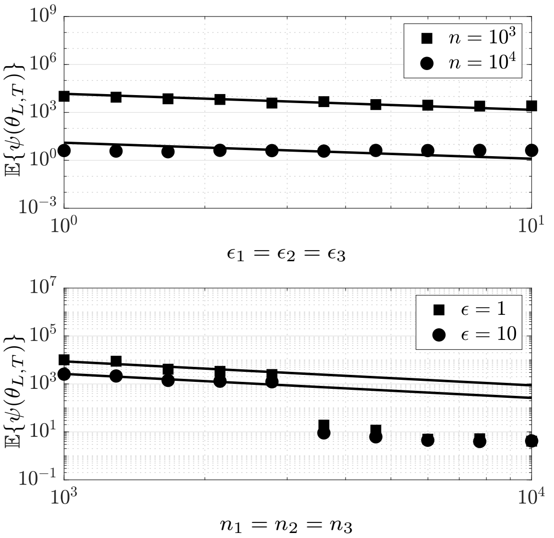

Fig. 11. Relative fitness of Algorithm 1 for detecting credit card fraud after $T = 100$ iterations versus the privacy budget [top] and the size of the datasets [bottom]. The solid line illustrates the bound in (11).

VI. Conclusions and Future Research

In this paper, we developed an asynchronous DP algorithm for training ML models on multiple private datasets. We proved that, by following the asynchronous algorithm, the cost of privacy is inversely proportional to the combined size of the training datasets squared and the privacy budgets squared. Finally, we validated the theoretical results on experiments on financial data. Although, in this paper, the training data is not required to be i.i.d. as we are considering the performance degradation in terms of the training dataset fitness, it is of extreme importance to investigate the effect of privacy-preserving noise in generalization error in the future. For practical results, we also need to also focus on the effect of non-i.i.d. and unbalanced data. Future work can also focus on multiple directions. An interesting extension is to consider multiple learners training separate ML models. This would be more similar to the distributed ML on arbitrary connected graphs. This way, we can extend the results to more general communication structures with the learner not necessarily at the center. We can investigate the behaviour of private data owners and learners in a data market. The cost of privacy in this paper can be used as a guide for developing compensation mechanisms for private data owners to increase their privacy budgets. The developed algorithm is particularly of use as the data owners and the learners in the data market can predict the performance of privately-trained ML models during negotiation for setting privacy budgets and compensating data owners. Finally, we can extend the results to adversarial ML with more sophisticated adversaries.

References

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference*, pp. 265–284, 2006.

[2] 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.

[3] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60, IEEE, 2010.

[4] 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.

[5] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, IEEE, 2017.

[6] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 486–503, Springer, 2006.

[7] S. Asoodeh, J. Liao, F. P. Calmon, O. Kosut, and L. Sankar, "A better bound gives a hundred rounds: Enhanced privacy guarantees via f-divergences," *arXiv preprint arXiv:2001.05990*, 2020.

[8] 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, 2014.

[9] D. C. Parkes and J. Shneidman, "Distributed implementations of Vickrey-Clarke-Groves mechanisms," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Vol 1*, pp. 261–268, 2004.

[10] T. Tanaka, F. Farokhi, and C. Langbort, "Faithful implementations of distributed algorithms and control laws," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 2, pp. 191–201, 2015.

[11] 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.

[12] 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.

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

[14] 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.

[15] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 429–438, IEEE, 2013.

[16] 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.

[17] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "Dp-admm: Admm-based distributed learning with differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1002–1012, 2019.

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

[19] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafar, "The value of collaboration in convex machine learning with differential privacy," in *Proceedings of the 41st IEEE Symposium on Security and Privacy*, 2020.

[20] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *International Conference on Learning Representations (ICLR)*, 2018.

[21] 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, 2015.

[22] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson, "Scalable private learning with PATE," in *International Conference on Learning Representations*, 2018.

[23] N. Papernot, M. Abadi, Úlfar Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *International Conference on Learning Representations*, 2017.

[24] A. Smith, A. Thakurta, and J. Upadhyay, "Is interaction necessary for distributed private learning?," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 58–77, IEEE, 2017.

[25] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Asynchronous gossip algorithms for stochastic optimization," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 3581–3586, 2009.

[26] P. Heidelberger and K. S. Trivedi, "Queueing network models for parallel processing with asynchronous tasks," *IEEE Transactions on Computers*, vol. C-31, no. 11, pp. 1099–1109, 1982.

[27] R. Lagunoff and A. Matsui, "Asynchronous choice in repeated coordination games," *Econometrica*, pp. 1467–1477, 1997.

[28] K. Srivastava and A. Nedic, "Distributed asynchronous constrained stochastic optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011.

[29] B. Touri, A. Nedič, and S. S. Ram, "Asynchronous stochastic convex optimization over random networks: Error bounds," in *2010 Information Theory and Applications Workshop (ITA)*, pp. 1–10, 2010.

[30] N. Aybat, Z. Wang, and G. Iyengar, "An asynchronous distributed proximal gradient method for composite convex optimization," in *International Conference on Machine Learning*, pp. 2454–2462, 2015.

[31] A. S. Bedi and K. Rajawat, "Asynchronous incremental stochastic dual descent algorithm for network resource allocation," *IEEE Transactions on Signal Processing*, vol. 66, no. 9, pp. 2229–2244, 2018.

[32] M. Hong and T.-H. Chang, "Stochastic proximal gradient consensus over random networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 11, pp. 2933–2948, 2017.

[33] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, pp. 1928–1937, 2016.

[34] P. Smyth, M. Welling, and A. U. Asuncion, "Asynchronous distributed learning of topic models," in *Advances in Neural Information Processing Systems*, pp. 81–88, 2009.

[35] B. McMahan and M. Streeter, "Delay-tolerant algorithms for asynchronous distributed online learning," in *Advances in Neural Information Processing Systems*, pp. 2915–2923, 2014.

[36] L. Xie, I. M. Baytas, K. Lin, and J. Zhou, "Privacy-preserving distributed multi-task learning with asynchronous updates," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1195–1204, 2017.

[37] W. Ford, *Numerical Linear Algebra with Applications: Using MATLAB*. Elsevier Science, 1 ed., 2015.

[38] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, *et al.*, "Towards federated learning at scale: System design," in *Proceedings of Machine Learning and Systems (MLSys 2019)*, 2019.

[39] Kaggle, "Lending club loan data: Analyze lending club's issued loans," 2018. https://www.kaggle.com/wendykan/lending-club-loan-data.

[40] New York State Department of Health, "Hospital Inpatient Discharges (SPARCS De-Identified): 2015," 2015. https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/82xm-y6g8.

[41] 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.

Appendix A
Proof of Theorem 1

Since there are at most T rounds of communication, the privacy budget in each step should be set as ϵ_i/T for all i. Now, note that

$$\begin{aligned}
\|\mathcal{Q}_{i_k}&(\mathcal{D}_{i_k};\bar{\theta}_k)-\mathcal{Q}_{i_k}(\mathcal{D}'_{i_k};\bar{\theta}_k)\|_1 \\
&=\frac{1}{n_{i_k}}\Bigg\|\sum_{\{x,y\}\in\mathcal{D}_{i_k}}\nabla_\theta\ell(\mathfrak{M}(x;\theta),y) \\
&\qquad-\sum_{\{x,y\}\in\mathcal{D}'_{i_k}}\nabla_\theta\ell(\mathfrak{M}(x;\theta),y)\Bigg\|_1 \\
&=\frac{1}{n_{i_k}}\Big\|\nabla_\theta\ell(\mathfrak{M}(x;\theta),y)|_{\{x,y\}\in\mathcal{D}_{i_k}\setminus\mathcal{D}'_{i_k}} \\
&\qquad-\nabla_\theta\ell(\mathfrak{M}(x;\theta),y)|_{\{x,y\}\in\mathcal{D}'_{i_k}\setminus\mathcal{D}_{i_k}}\Big\|_1 \\
&=\frac{2\Xi}{n_{i_k}}.
\end{aligned}$$

Therefore, the scale of the noise must be $2\Xi T/(n_{i_k}\epsilon_{i_k})$.

Appendix B
Proof of Theorem 2

We start by casting the problem of privacy-aware learning in the framework of asynchronous distributed optimization in [29]. For any $\eta < 1/N$, we can define $f_i(\theta) = \eta g(\theta) + \frac{1}{n}\sum_{\{x,y\}\in\mathcal{D}_i} \ell(\mathfrak{M}(x;\theta),y), \forall i \in \mathcal{N}$, and $f_L(\theta) = (1-\eta N)g(\theta)$. We can think of f_i as the cost functions of data owners and f_L as the cost function of the learner. By construct, f_L is σ_L strongly convex with $\sigma_L = (1-\eta N)\sigma$ and f_i is σ_i strongly convex with $\sigma_i = \eta\sigma$. Note that

$$\begin{aligned}\|\nabla_\theta f_i(\theta)\|_2 =& \left\| \eta\nabla_\theta g(\theta) + \frac{1}{n}\sum_{\{x,y\}\in\mathcal{D}_i} \nabla_\theta \ell(\mathfrak{M}(x;\theta),y) \right\|_2 \\ \leq& \eta\Xi_g + \frac{n_i}{n}\Xi \\ \leq& \Xi_g + \Xi,\end{aligned}$$

and $\|\nabla_\theta f_L(\theta)\|_2 = \|(1-\eta N)\nabla_\theta g(\theta)\|_2 \leq (1-\eta N)\Xi_g \leq \Xi_g$. Therefore, $\|\nabla_\theta f_i(\theta)\|_2 \leq C$, $\forall i$, and $\|\nabla_\theta f_L(\theta)\|_2 \leq C$ with $C = \Xi_g + \Xi$.

In each iteration, one of the data owners at random is selected and follows the gossip algorithm (see [29]) for exchanging information in learning and updating the decision variables. In this paper, however, we assume that the learner takes care of all the updates and storing the iterates. Therefore, the learner submits a gradient query to the selected data owner and receives a DP response for updating the decision variables. Let i denote the index of the randomly-selected data owner at iteration k; note that i_k is used in Algorithm 1 for denoting the index. We use $\mathcal{G} = (\mathcal{V},\mathcal{E})$ to denote a graph with the vertex set $\mathcal{V} = \{1,\dots,N,N+1\}$, in which node $N+1$ is the learner L, and the edge set $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$. By the methodology of [29], we get

$$W_k = I - \frac{1}{2}(e_i - e_{N+1})(e_i - e_{N+1})^\top,$$

and $U_k = \{L, i\}$. It is evident that the probability of selecting the learner at each round is equal to one, i.e., $\gamma_L = 1$, and the probability of selecting any data owner is $\gamma_i = 1/N$ in the notation of [29],. We get

$$\begin{aligned}\overline{W} =& \mathbb{E}\{W_k\} \\ =& I - \begin{bmatrix} \frac{1}{2N} & 0 & \cdots & 0 & -\frac{1}{2N} \\ 0 & \frac{1}{2N} & \cdots & 0 & -\frac{1}{2N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{2N} & \frac{1}{2N} \\ -\frac{1}{2N} & -\frac{1}{2N} & \cdots & -\frac{1}{2N} & \frac{1}{2} \end{bmatrix}.\end{aligned}$$

We meet all the conditions of Assumption 2 in [29]. Furthermore, using Theorem 1 in [29], we can see that

$$\lambda = \left\| W_k - \frac{1}{N+1}\mathbb{1}\mathbb{1}^\top W_k \right\|_2^2 < 1.$$

The updates in (2) in [29] can be rewritten as

$$\bar{\theta}_k = \frac{1}{2}\theta_{L,k-1} + \frac{1}{2}\theta_{i,k-1},$$

with the notation substitution of $\bar{\theta}_k$ instead of $v_{i,k} = v_{L,k}$, $\theta_{i,k}$ instead of $x_{i,k}$, and $\theta_{L,k}$ instead of $x_{L,k}$. The updates in (3) in [29] can also be rewritten as

$$\begin{aligned}\theta_{i,k} =& \Pi_\Theta\left[\bar{\theta}_k - \alpha_i\eta\nabla_\theta g(v_k) + \alpha_i\frac{n_i}{n}\overline{\mathcal{Q}}_i(\bar{\theta}_k;k)\right] \\ =& \Pi_\Theta\left[\bar{\theta}_k - \alpha_i\left(\eta\nabla_\theta g(v_k) + \frac{n_i}{n}\left(\mathcal{Q}_i(\bar{\theta}_k;k) + w_i(k)\right)\right)\right] \\ =& \Pi_\Theta\left[\bar{\theta}_k - \alpha_i(\nabla_\theta f_i(\bar{\theta}_k) + \bar{w}_i(k))\right],\end{aligned}$$

with $\bar{w}_i(k) = w_i(k)n_i/n$ and

$$\begin{aligned}\theta_{L,k} =& \Pi_\Theta\left[\bar{\theta}_k - \alpha_L\nabla f_L(\bar{\theta}_k)\right] \\ =& \Pi_\Theta\left[\bar{\theta}_k - (1-\eta N)\alpha_L\nabla_\theta g(\bar{\theta}_k)\right],\end{aligned}$$

where $w_i(k)$ is the additive i.i.d. DP noise and

$$\mathcal{Q}_i(\bar{\theta}_k;k) = \frac{1}{n_i}\sum_{\{x,y\}\in\mathcal{D}_i} \nabla_\theta\ell(\mathfrak{M}(x;\bar{\theta}_k),y).$$

Note that, here, we are using i instead of i_k to reduce the complexity of the notation and for conforming to the notation of [29]. We have

$$\begin{aligned}\mathbb{E}\{\bar{w}_i(k)|\mathcal{F}_k\} =& 0, \\ \mathbb{E}\{\|\bar{w}_i(k)\|_2^2|\mathcal{F}_k\} \leq& \nu_i^2,\end{aligned}$$

where $\mathcal{F}_k$ is the filtration generated by the entire history of Algorithm 1 up to iteration k. Using Theorem 1, we can see that

$$\nu_i = \frac{2\sqrt{2}\Xi T}{n\epsilon_i}.$$

Extending Lemma 3 in [29] results in

$$\begin{aligned}\mathbb{E}\{\|\nabla_\theta f_i(\bar{\theta}_k) + w_i(k)\|_2^2|\mathcal{F}_{k-1},W_k\} \leq& C^2 + \nu_i^2 \leq (C+\nu_i)^2, \\ \mathbb{E}\{\|\nabla_\theta f_L(\bar{\theta}_k)\|_2^2|\mathcal{F}_{k-1},W_k\} \leq& C^2.\end{aligned}$$

Therefore, we can upgrade the right-hand side of (22) in [29] to

$$\mathbb{E}\{\alpha_i^2(C+\nu_i)^2\} + \alpha_L^2C^2 = \alpha_L^2C^2 + \frac{1}{N}\sum_{i\in\mathcal{N}}\alpha_i^2(C+\nu_i)^2.$$

Note that, in the case of this paper, the summation only contains two terms because, in each iteration, only the learner and another data owner update their decision variables. This implies that, in Proposition 1 in [29], ε_{net} must be updated to

$$\varepsilon_{\text{net}} = \frac{C\sqrt{N+1}}{1-\sqrt{\lambda}}\sqrt{\alpha_L^2 + \frac{1}{N}\sum_{i\in\mathcal{N}}\alpha_i^2\left(1+\frac{\nu_i}{C}\right)^2}.$$

With the same line of reasoning, we can improve the bound in Proposition 2 in [29] to get

$$\begin{aligned}\limsup_{k\to\infty}&\left[\mathbb{E}\{\|\theta_{L,k}-\theta^*\|_2^2\} + \sum_{i\in\mathcal{N}}\mathbb{E}\{\|\theta_{i,k}-\theta^*\|_2^2\}\right] \\ &\leq \frac{\varepsilon + 2\alpha_{\max}C\varepsilon_{\text{net}}}{1-q}, \qquad (12)\end{aligned}$$

where

$$\begin{aligned}\varepsilon =&2(N+1)(1-\gamma_{\min})\delta_{\alpha,\sigma}\operatorname{diam}(\Theta)^2\\&2(N+1)\delta_{\alpha,\gamma}C\operatorname{diam}(\Theta)\\&+C^2\left(\alpha_L^2+\frac{1}{N}\sum_{i\in\mathcal{N}}\alpha_i^2\left(1+\frac{\nu_i}{C}\right)^2\right),\end{aligned}\tag{13}$$

and

$$\begin{aligned}\alpha_{\max}=&\max_i \alpha_i,\\ \gamma_{\min}=&1/N,\\ q=&1-2\gamma_{\min}\min\left\{\alpha_L\sigma_L,\min_{i\in\mathcal{N}}\alpha_i\sigma_i\right\},\\ \delta_{\alpha,\sigma}=&\max\left\{\alpha_L\sigma_L,\max_{i\in\mathcal{N}}\alpha_i\sigma_i\right\}-\min\left\{\alpha_L\sigma_L,\min_{i\in\mathcal{N}}\alpha_i\sigma_i\right\},\\ \delta_{\alpha,\gamma}=&\max\left\{\alpha_L\gamma_L,\max_{i\in\mathcal{N}}\alpha_i\gamma_i\right\}-\min\left\{\alpha_L\gamma_L,\min_{i\in\mathcal{N}}\alpha_i\gamma_i\right\}.\end{aligned}$$

Therefore, for any $\varsigma>0$, there exists large enough $T\in\mathbb{N}$ such that

$$\begin{aligned}&\left[\mathbb{E}\{\|\theta_{L,T}-\theta^*\|_2^2\}+\sum_{i\in\mathcal{N}}\mathbb{E}\{\|\theta_{i,T}-\theta^*\|_2^2\}\right]\\&\qquad\leq\varsigma+\frac{\varepsilon+2\alpha_{\max}C\varepsilon_{\text{net}}}{1-q},\end{aligned}\tag{14}$$

Selecting $\eta=1/(2N)$ and $\alpha_L=\alpha_i/N=\alpha/\sigma$ for some constant $\alpha\in(0,1)$, we get $\delta_{\alpha,\sigma}=\delta_{\alpha,\gamma}=0$. Therefore, we can simplify (13) to get

$$\varepsilon=\frac{\alpha^2C^2}{\sigma^2}\left(1+N\sum_{i\in\mathcal{N}}\left(1+\frac{\nu_i}{C}\right)^2\right).\tag{15}$$

We will also get

$$2\alpha_{\max}C\varepsilon_{\text{net}}=\frac{2N\alpha^2C^2\sqrt{N+1}}{\sigma^2(1-\sqrt{\lambda})}\sqrt{1+N\sum_{i\in\mathcal{N}}\left(1+\frac{\nu_i}{C}\right)^2}.\tag{16}$$

Furthermore,

$$1-q=2\gamma_{\min}\min\left\{\alpha_L\sigma_L,\min_{i\in\mathcal{N}}\alpha_i\sigma_i\right\}=\frac{\alpha}{N}.\tag{17}$$

Combining (14) with (15)–(17), we get

$$\begin{aligned}&\mathbb{E}\{\|\theta_{L,T}-\theta^*\|_2^2\}\\&\leq\left[\mathbb{E}\{\|\theta_{L,T}-\theta^*\|_2^2\}+\sum_{i\in\mathcal{N}}\mathbb{E}\{\|\theta_{i,T}-\theta^*\|_2^2\}\right]\\&\leq\frac{N\alpha C^2}{\sigma^2}\left(1+N\sum_{i\in\mathcal{N}}\left(1+\frac{2\sqrt{2}\Xi T}{n\epsilon_i(\Xi_g+\Xi)}\right)^2\right)\\&\quad+\frac{2N^2\alpha C^2\sqrt{N+1}}{\sigma^2(1-\sqrt{\lambda})}\sqrt{1+N\sum_{i\in\mathcal{N}}\left(1+\frac{2\sqrt{2}\Xi T}{n\epsilon_i(\Xi_g+\Xi)}\right)^2}.\end{aligned}$$

Define $c_1=NC^2/\sigma^2$, $c_2=2N^2C^2\sqrt{N+1}/(\sigma^2(1-\sqrt{\lambda}))$. We have

$$\begin{aligned}\mathbb{E}\{\|\theta_{L,T}-\theta^*\|_2^2\}\leq&c_1\alpha\left(1+N\sum_{i\in\mathcal{N}}\left(1+\frac{2\sqrt{2}\Xi T}{n\epsilon_i(\Xi_g+\Xi)}\right)^2\right)\\&+c_2\alpha\sqrt{1+N\sum_{i\in\mathcal{N}}\left(1+\frac{2\sqrt{2}\Xi T}{n\epsilon_i(\Xi_g+\Xi)}\right)^2}.\end{aligned}$$

Selecting $\alpha=\rho/T^2$ and noting that $\Xi\leq\Xi_g+\Xi$, the upper bound can be further simplified (8). Following the same modifications in the proof of Proposition 3 in [29] results in (9).

Farhad Farokhi (S'10–M'15–SM'20) received B.Sc. and M.Sc. degrees from Sharif University of Technology in 2008 and 2010, respectively, the Ph.D. degree from the KTH Royal Institute of Technology in 2014.

In 2014, he joined The University of Melbourne, where he is currently a Lecturer (equivalent to Assistant Professor in North America). From 2018–2020, he was also a Research Scientist at the CSIRO's Data61. In 2013, he held a Visiting Researcher position at the University of California, Berkeley.

Dr. Farokhi has been the recipient of the VESKI Victoria Fellowship from the Victoria State Government, Australia, and the McKenzie Fellowship, the 2015 Early Career Researcher Award, and 2020 MSE Excellence Award for Early Career Research from The University of Melbourne. He has been involved in multiple projects on data privacy and cyber-security funded by the Australian Research Council, the Defence Science and Technology Group, the Department of the Prime Minister and Cabinet, the Department of Environment and Energy, and the CSIRO. He is the associate editor for IET Smart Grid and Results in Control and Optimization.

Nan Wu received the B.S. degrees(Hons.) in electronic and communication systems from both the Australian National University, Australia, and the Beijing Institute of Technology, China, in 2015 and 2016, and the MRes degree in computer science from Macquarie University, Australia, in 2019. She is currently pursuing the Ph.D. degree at Macquarie University and is now also with CSIRO's Data61. Her research interests include privacy-preserving machine learning, privacy in data matching, security and privacy, and game theory.

David Smith (S'01–M'04) received the B.E. degree in electrical engineering from the University of New South Wales, Sydney, NSW, Australia, in 1997, and the M.E. (research) and Ph.D. degrees in telecommunications engineering from the University of Technology, Sydney, NSW, Australia, in 2001 and 2004, respectively. Since 2004, he was with National Information and Communications Technology Australia (NICTA, incorporated into Data61 of CSIRO in 2016), and the Australian National University (ANU), Canberra, ACT, Australia, where he is currently a Principal Research Scientist with CSIRO Data61, and an Adjunct Associate Professor with ANU. He has a variety of industry experience in electrical and telecommunications engineering. His current research interests include data privacy, privacy for networks, IoT, game theory for distributed signal processing, disaster tolerant networks, 5G networks, distributed optimization for smart grid and electric vehicles. He has published over 150 technical refereed papers. He has made various contributions to IEEE standardisation activity in personal area networks. He is an area editor for IET Smart Grid and has served on the technical program committees of several leading international conferences in the fields of communications and networks. Dr. Smith was the recipient of four conference Best Paper Awards.

Dali Kaafar is a Full Professor of Privacy Preserving Technologies at the Faculty of Science and Engineering at Macquarie University and the Executive Director of the Optus Macquarie University Cyber Security Hub. Dali Kaafar is also the founder of the Information Security and Privacy group at CSIRO Data61. Prior to that, Dali was the group leader of the Information Security and Privacy and Mobile systems Research groups at CSIRO Data61 and the networked systems at NICTA. He received his PhD from University of Nice Sophia Antipolis and INRIA in France where he pioneered research in the security of Internet Coordinate Systems.

Author/s:
Farokhi, F; Wu, N; Smith, D; Kaafar, MA

Title:
The Cost of Privacy in Asynchronous Differentially-Private Machine Learning

Date:
2021-01-01

Citation:
Farokhi, F., Wu, N., Smith, D. & Kaafar, M. A. (2021). The Cost of Privacy in Asynchronous Differentially-Private Machine Learning. IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, 16, pp.2118-2129. https://doi.org/10.1109/TIFS.2021.3050603.

Persistent Link:
http://hdl.handle.net/11343/268323

File Description:
Accepted version