2020 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 21–24, 2020, ESPOO, FINLAND

SECURE FEDERATED AVERAGING ALGORITHM WITH DIFFERENTIAL PRIVACY

Yiwei Li[⋆] *Tsung-Hui Chang*[†] *Chong-Yung Chi*[⋆]

[⋆] Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan.
[†] School of Science and Engineering, Shenzhen Research Institute of Big Data,
The Chinese University of Hong Kong, Shenzhen, China.

Email: lywei0306@foxmail.com, tsunghui.chang@ieee.org, cychi@ee.nthu.edu.tw.

ABSTRACT

Federated learning (FL), as a recent advance of distributed machine learning, is capable of learning a model over the network without directly accessing the client's raw data. Nevertheless, the clients' sensitive information can still be exposed to adversaries via differential attacks on messages exchanged between the parameter server and clients. In this paper, we consider the widely used federating averaging (FedAvg) algorithm and propose to enhance the data privacy by the differential privacy (DP) technique, which obfuscates the exchanged messages by properly adding Gaussian noise. We analytically show that the proposed secure FedAvg algorithm maintains an $\mathcal{O}(1/T)$ convergence rate, where T is the total number of stochastic gradient descent (SGD) updates for local model parameters. Moreover, we demonstrate how various algorithm parameters can impact on the algorithm communication efficiency. Experiment results are presented to justify the obtained analytical results on the performance of the proposed algorithm in terms of testing accuracy.

Index Terms— Federated learning, Differential privacy, Convergence analysis, Model averaging

1. INTRODUCTION

With widespread government regulations and laws on privacy protection, privacy-preserving machine learning (ML) methods have attracted much attention [1]. One of the most promising methods is the so called federated learning (FL) [2] which can enable large numbers of data users to collaboratively learn a shared ML model without directly exposing any of their local data. Specially, in FL, a central parameter server (PS) coordinates the clients' operations and computes a global ML model based on the local models learned by the clients from their local private data. Nevertheless, a number of challenges arise in order to deploy the FL framework.

Firstly, FL typically involves a large number of clients, and therefore it is challenging to have reliable connectivity for all clients due to limited wireless communication resources [3]. Secondly, the amounts of data that the clients have as well as the data distribution can vary greatly from one client to another, which makes the FL algorithm design and theoretical analysis more challenging [4]. Among the existing methods, the federated averaging (FedAvg) algorithm proposed in [2] is one of the most popular algorithms. Within each communication round, FedAvg runs multiple steps of stochastic gradient descent (SGD) on a small randomly sampled subset of clients (i.e., partial participation), and averages the local models on the PS. As a result, FedAvg is suitable for large scale FL networks with better communication efficiency.

Although FL can protect clients' raw data from being directly accessed by the PS or other adversaries, the individual sensitive information can still be revealed if they apply differential attacks [5] to the messages exchanged between the PS and clients. One of the approaches to prevent differential attacks from breaching privacy is differential privacy (DP) [6], which can protect the privacy even when the PS/adversaries have full knowledge of the training mechanism and access the model parameters [7]. The DP has been considered in several FL algorithms. For example, the work [8] considered DP for a model averaging (MA) based FL algorithm and analyzed the impact of DP level on the algorithm convergence; the work [9] employed Bayesian DP to provide less privacy loss for general MA based FL systems; the work [10] applied DP to the FedAvg algorithm but considering local full gradient descent and full client participation; analogously, the work [11] considered DP for the distributed ADMM algorithm.

In this paper, we analyze the convergence of the secure FedAvg algorithm. A striking feature of our analysis is that we explicitly take into account the interplay between various system parameters (including mini-batch size, the local epoch length, and number of randomly selected clients) and achievable protection level based on the amplification privacy theorem [12]. The analysis results reveal insightful trade-off between the algorithm convergence speed and these system parameters. Experimental results are presented to verify our

The work of C.-Y. Chi is supported by the Ministry of Science and Technology under Grant MOST 108-2221-E-007-012, MOST 109-2221-E-007-088. The work of T.-H. Chang is supported in part by the NSFC, China, under Grant 61731018, and in part by the Shenzhen Fundamental Research Fund under Grant JCYJ20190813171003723 and No. KQTD2015033114415450.

978-1-7281-6662-9/20/$31.00

theoretical findings.

2. SECURE FEDERATED AVERAGING WITH DP

2.1. Federated Averaging Algorithm

We consider a FL network consisting of a parameter server (PS) and a total number of N clients. The PS aims to learn a ML model through collaboration with the clients without directly accessing their private raw data. Let $\mathcal{D}_k \triangleq \{(\boldsymbol{x}_{k,m}, y_{k,m})\}_{m=1}^{n_k}$ be the local dataset of client k for $k \in [N] \triangleq \{1, \ldots, N\}$, where n_k is the data size, $\boldsymbol{x}_{k,m}$ is the m-th training data sample and $y_{k,m}$ is the corresponding label. Assume that the ML task involves solving the following optimization problem [2]

$$\min_{\boldsymbol{w}} \left\{ F(\boldsymbol{w}) \triangleq \sum_{k=1}^{N} p_k F_k(\boldsymbol{w}) \right\}. \tag{1}$$

Here, $\boldsymbol{w} \in \mathbb{R}^M$ is the learning parameter vector, $p_k = n_k / \sum_{k=1}^{N} n_k$ is a weighting coefficient, $n = \sum_{k=1}^{N} n_k$ is the data size for all clients. $F_k(\boldsymbol{w}) = \frac{1}{n_k} \sum_{m=1}^{n_k} \ell(\boldsymbol{w}; \boldsymbol{x}_{k,m}, y_{k,m})$ is the local cost function, where $\ell(\cdot)$ is a user-specified loss function.

In this paper, we employ the FedAvg algorithm in [2, 13] to handle problem (1). FedAvg is based on the classical distributed stochastic gradient descent (SGD) algorithm [14]. In distributed SGD, at each iteration number t, each client k maintains a local model $\boldsymbol{w}_{t+1}^k$ by local SGD update

$$\boldsymbol{w}_{t+1}^k \leftarrow \boldsymbol{w}_t^k - \eta_t \nabla F_k\left(\boldsymbol{w}_t^k; \xi_t^k, b\right), \tag{2}$$

where η_t is a step size, b is the mini-batch size, ξ_t^k is a mini-batch data randomly sampled from $\lceil \frac{n_k}{b} \rceil$ mini-batches of $\mathcal{D}_k$, and $\nabla F_k\left(\boldsymbol{w}_t^k; \xi_t^k, b\right) = (1/b) \sum_{m \in \xi_k^t} \nabla \ell\left(\boldsymbol{w}_t^k; \boldsymbol{x}_{k,m}, y_{k,m}\right)$ is the mini-batch gradient. After local SGD update, the local models $\boldsymbol{w}_{t+1}^k$, $k \in [N]$, are uploaded to the PS for model averaging $\bar{\boldsymbol{w}}_{t+1} = \sum_{k=1}^{N} p_k \boldsymbol{w}_{t+1}^k$. Then, the PS broadcasts the averaged model $\bar{\boldsymbol{w}}_{t+1}$ to the clients, and the above steps repeat until certain stopping condition is satisfied.

The FedAvg algorithm in [2, 13] can achieve improved communication efficiency over the distributed SGD owing to the adoption of two strategies, namely 1) partial client participation, and 2) multiple local SGD updates. Specifically, by partial client participation, only a subset of clients $\mathcal{S}_t \subseteq [N]$ (with size $|\mathcal{S}_t| = K$) are randomly activated to perform local SGD update and upload the latest model to the PS. It is shown that partial client participation can reduce the required communication rounds for achieving a desired testing accuracy, especially when the data are non-independent identically distributed (non-i.i.d.) across the clients and a small mini-batch size is used [2]. Note that if the selection of $\mathcal{S}_t$ is modeled as a sampling process without replacement, then the PS should take a weighted model average $\bar{\boldsymbol{w}}_{t+1} = \frac{N}{K} \sum_{k \in \mathcal{S}_t} p_k \boldsymbol{w}_{t+1}^k$ for unbiased global model estimation [13].

Analogously, it is shown empirically in [2] and analytically in [13] that the algorithm convergence can be effectively expedited if the clients locally perform an appropriate number of SGD updates (say Q times where $Q > 1$) within each communication round. In particular, let t_0 be the iteration number such that $\mathrm{mod}(t_0, Q) = 0$. Then, each client k performs Q local SGD updates, i.e., $\boldsymbol{w}_{t+1}^k \leftarrow \boldsymbol{w}_t^k - \eta_t \nabla F_k\left(\boldsymbol{w}_t^k; \xi_t^k, b\right)$, for $t = t_0, \ldots, t_0 + Q - 1$, followed by uploading the latest local model to the PS. Thus, the clients upload the local models to the PS only at iteration number t satisfying $\mathrm{mod}(t+1, Q) = 0$. Suppose T is the total number of local SGD updates performed by each client in FedAvg. Then, the number of communication round is T/Q.

Although being communication efficient, FedAvg may not be secure since either the (honest-but-curious) PS or other adversaries may overhear the exchanged messages $\{\boldsymbol{w}_{t+1}^k\}_{k=1}^N$ and $\bar{\boldsymbol{w}}_{t+1}$, and attempt to crack the clients' data privacy through advanced attacks [5, 7, 15]. We propose to apply the DP technique to FedAvg for enhancing the data privacy of the clients.

2.2. Differential Privacy

Differential privacy (DP) is a strong criterion against differential attacks from adversaries [7].

Definition 1 (ϵ, δ)*-DP* [11]. *Consider two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$, which differ in only one data sample. For any deterministic query function $f : \mathcal{D} \to \mathbb{R}^M$ and a randomized mechanism $\mathcal{M} : \mathbb{R}^M \to \mathcal{O}$, we say $\mathcal{M} \circ f$ achieves (ϵ, δ)-DP if for any subset of outputs $S \subseteq \mathcal{O}$:*

$$\Pr[\mathcal{M}(f(\mathcal{D})) \in S] \leq e^{\epsilon} \cdot \Pr\left[\mathcal{M}\left(f(\mathcal{D}') \in S\right] + \delta. \right. \tag{3}$$

In (3), smaller ϵ represents stronger privacy protection level, and $\delta \in [0, 1]$ stands for the probability to break the $(\epsilon, 0)$-DP.

A common strategy to achieve DP is to obfuscate $f(\cdot)$ by properly adding random noise [7]. For example, the Gaussian noise mechanism is

$$\mathcal{M}(f(\mathcal{D})) = f(\mathcal{D}) + \mathbf{z},\ \mathbf{z} \in \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_M), \tag{4}$$

where $\boldsymbol{I}_M$ is the $M \times M$ identity matrix, and $\mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_M)$ is the i.i.d. multivariate Gaussian noise with zero mean and variance σ^2. According to [7], to achieve (ϵ, δ)-DP, the required noise variance is $\sigma^2 = 2(\triangle f)^2 \ln(1.25/\delta)/\epsilon^2$, where $\triangle f$ is so-called global sensitivity of function f

$$\triangle f \triangleq \max_{\mathcal{D}, \mathcal{D}'} \left\| f(\mathcal{D}) - f\left(\mathcal{D}'\right) \right\|, \tag{5}$$

where $\|\cdot\|$ denotes the 2-norm.

2.3. Secure FedAvg with DP

To protect the client data privacy, we apply the DP to the uploaded messages $\{\boldsymbol{w}_{t+1}^k\}_{k=1}^N$ when $\mathrm{mod}(t+1, Q) = 0$. It corresponds to (4) with the query function $f_{k,t+1}(\mathcal{D}_k) = \boldsymbol{w}_{t+1}^k$ for each client k. Note that according to the post-processing invariance property [7], the risk of privacy leakage would not

Algorithm 1 Secure FedAvg

1: **Input:** Initial model $\bar{\boldsymbol{w}}_0$ and step size η_0. The PS broadcasts $\bar{\boldsymbol{w}}_0$ to all clients ($\mathcal{S}_0 = [N]$).
2: **for** $t = 0, 1, \ldots, T-1$ **do**
3: **Client side:**
4: **for** $k \in \mathcal{S}_t$ in parallel **do**
5: **if** $\mod(t, Q) = 0$ **then**
6: Set $\boldsymbol{w}_t^k = \bar{\boldsymbol{w}}_t$.
7: **end if**
8: Sample a mini-batch ξ_t^k from $\mathcal{D}_k$ and calculate the local gradient $\boldsymbol{g}_t^k = \nabla F_k\left(\boldsymbol{w}_t^k; \xi_t^k, b\right)$.
9: **if** $\mod(t+1, Q) \neq 0$ **then**
10: $\boldsymbol{w}_{t+1}^k \leftarrow \boldsymbol{w}_t^k - \eta_t \boldsymbol{g}_t^k$,
11: **else if** $\mod(t+1, Q) = 0$ **then**
12: $\boldsymbol{w}_{t+1}^k \leftarrow (\boldsymbol{w}_t^k - \eta_t \boldsymbol{g}_t^k) + \mathbf{z}_t^k$, $\mathbf{z}_t^k \sim \mathcal{N}(\mathbf{0}, \sigma_{t,k}^2 \boldsymbol{I}_M)$.
13: Send $\boldsymbol{w}_{t+1}^k$ to the PS.
14: **end if**
15: **end for**
16: **for** $k \notin \mathcal{S}_t$ in parallel **do**
17: $\boldsymbol{w}_{t+1}^k = \boldsymbol{w}_t^k$.
18: **end for**
19: **Server side:**
20: **if** $\mod(t+1, Q) = 0$ **then**
21: $\bar{\boldsymbol{w}}_{t+1} = \frac{N}{K} \sum_{k \in \mathcal{S}_t} p_k \boldsymbol{w}_{t+1}^k$.
22: Select a subset of clients $\mathcal{S}_{t+1}$ by sampling without replacement, and broadcast $\bar{\boldsymbol{w}}_{t+1}$ to all clients.
23: **end if**
24: **end for**

be increased if the raw data are not accessed again. Therefore, given that the averaged model $\bar{\boldsymbol{w}}_{t+1}$ broadcasted by the PS is a linear combination of $\{\boldsymbol{w}_{t+1}^k\}_{k=1}^N$, which will be protected by DP, it is not essential to further protect the downlink message $\bar{\boldsymbol{w}}_{t+1}$ as in [8].

Algorithm 1 outlines the proposed secure FedAvg algorithm with DP. In particular, the only difference between Algorithm 1 and the vanilla FedAvg algorithm [16] lies in Step 11 and Step 12, where whenever $\mod(t+1, Q) = 0$, each client k adds Gaussian noise to the local model, i.e.,

$$\boldsymbol{w}_{t+1}^k \leftarrow (\boldsymbol{w}_t^k - \eta_t \boldsymbol{g}_t^k) + \mathbf{z}_t^k, \tag{6}$$

where $\mathbf{z}_t^k \sim \mathcal{N}(\mathbf{0}, \sigma_{t,k}^2 \boldsymbol{I}_M)$, and uploads the noisy local model to the PS.

Privacy amplification: According to the privacy amplification theorem [12], it is known that the DP mechanism run on a random sample of a dataset provides stronger privacy protection than when run on the entire dataset. It implies that the noise variance required for achieving a predefined DP level can be reduced if the protected query function entails randomly sampled data.

For Algorithm 1, because the uploaded local model $\boldsymbol{w}_{t+1}^k$ (Step 13) is obtained through Q times of mini-batch sampling (Step 8), we can apply the privacy amplification property to Algorithm 1. Specifically, suppose that the mini-batches are sampled without replacement. Then, according to [17], given the noise level $\sigma_{t,k}^2 = 2(\triangle f_{t,k})^2 \ln(1.25/\delta)/\epsilon^2$ (where $\triangle f_{t,k}$ is the global sensitivity of $f_{t,k}(\mathcal{D}_k)$), one can show that (6) in fact achieves $(\log(1 + (1 - (1 - b/n_k)^Q)(e^\epsilon - 1)), q\delta)$-DP for $\boldsymbol{w}_{t+1}^k$, where $q = Qb/n_k$. Since

$$(1 - (1 - b/n_k))^Q \leq Qb/n_k = q, \tag{7}$$

$$\log(1 + q(e^\epsilon - 1)) \leq q(e^\epsilon - 1) \leq 2q\epsilon, \tag{8}$$

one achieves at least $(2q\epsilon, q\delta)$-DP. In other words, if one aims to achieve an (ϵ, δ)-DP for $\boldsymbol{w}_{t+1}^k$, the required noise level can be reduced to

$$\begin{aligned} \sigma_{t,k}^2 &= \frac{2(\triangle f_{t,k})^2 \ln(1.25/(\delta/q))}{(\epsilon/2q)^2} \\ &= \frac{8q^2(\triangle f_{t,k})^2 \ln(1.25q/\delta)}{\epsilon^2}, \end{aligned} \tag{9}$$

where $\triangle f_{t,k}$ will be determined in Section 3.1.

It has been shown that proper large values of b and Q may benefit the convergence of vanilla distributed SGD [18]. However, as seen from (9), either increasing b or Q would request higher noise level for DP protection, which may slow down algorithm convergence. Therefore, (9) implies that, on the choice of b and Q, there is a trade-off between algorithm convergence speed and achieved protection level. To understand the trade-off, we analyze the convergence property of Algorithm 1 in the next section.

3. CONVERGENCE ANALYSIS

3.1. Assumptions

We first give some assumptions for problem (1).

Assumption 1 *Each F_k is L-smooth, i.e., for all $\boldsymbol{w}$ and $\boldsymbol{v}$, $F_k(\boldsymbol{v}) \leq F_k(\boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T \nabla F_k(\boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{v} - \boldsymbol{w}\|^2$, $k \in [N]$.*

Assumption 2 *Each F_k is μ-strong convex, i.e., for all $\boldsymbol{w}$, $\boldsymbol{v}$, $F_k(\boldsymbol{v}) \geq F_k(\boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T \nabla F_k(\boldsymbol{w}) + \frac{\mu}{2}\|\boldsymbol{v} - \boldsymbol{w}\|^2$, $k \in [N]$.*

Note that the strongly convex and smooth assumptions are typical examples for logistic regression, softmax regression and L_2-norm regularized linear regression problems.

Assumption 3 *The stochastic gradients for each client satisfy $\mathbb{E}[\nabla F_k\left(\boldsymbol{w}_t^k; \xi_t^k, b\right)] = \nabla F_k\left(\boldsymbol{w}_t^k\right)$ and $\mathbb{E}[\|\nabla F_k(\boldsymbol{w}_t^k; \xi_t^k, b) - \nabla F_k(\boldsymbol{w}_t^k)\|^2] \leq \gamma_k^2/b, \forall k \in [N]$.*

Assumption 4 *The gradient for all clients is bounded, i.e., $\left\|\nabla F_k\left(\boldsymbol{w}_t^k; \xi_t^k, b\right)\right\|^2 \leq G^2, \forall k \in [N]$.*

Given Assumption 4, one can compute the global sensitivity in (5).

Lemma 1 *Suppose that Assumption 4 holds. Let $\mod(t+1, Q) = 0$ and assume that $\eta_{t+1-Q+\tau} \leq 2\eta_{t+1}$ for $\tau = 0, 1, \ldots, Q-1$. Denote $f_{t+1,k}(\mathcal{D}_k) = \boldsymbol{w}_{t+1}^k(\mathcal{D}_k)$ where $\boldsymbol{w}_{t+1}^k(\mathcal{D}_k)$ represents the local model $\boldsymbol{w}_{t+1}^k$ obtained from dataset $\mathcal{D}_k$. Then, the global sensitivity is*

$$\triangle f_{t+1,k} = \max_{\mathcal{D}_k, \mathcal{D}_k'} \left\|\boldsymbol{w}_{t+1}^k(\mathcal{D}_k) - \boldsymbol{w}_{t+1}^k(\mathcal{D}_k')\right\| = 4QG\eta_{t+1},$$

where $\mathcal{D}_k'$ is a neighboring dataset of $\mathcal{D}_k$.

Proof: Since $\boldsymbol{w}_{t+1}^k(\mathcal{D}_k)$ is obtained by Q steps of local SGD starting from $\bar{\boldsymbol{w}}_{t+1-Q}$. Hence, we have

$$\boldsymbol{w}_{t+1}^k(\mathcal{X}) = \bar{\boldsymbol{w}}_{t+1-Q} - \sum_{\tau=0}^{Q-1} \eta_{t+1-Q+\tau} \boldsymbol{g}_{t+1-Q+\tau}^k(\mathcal{X}), \quad (10)$$

where $\boldsymbol{g}_{t+1-Q+\tau}^k(\mathcal{X})$ is the local gradient vector based on $\mathcal{X} \in \{\mathcal{D}_k, \mathcal{D}_k'\}$. Then, we have

$$\begin{aligned}
&\left\| \boldsymbol{w}_{t+1}^k(\mathcal{D}_k) - \boldsymbol{w}_{t+1}^k(\mathcal{D}_k') \right\|^2 = \\
&\left\| \sum_{\tau=0}^{Q-1} \eta_{t+1-Q+\tau} \boldsymbol{g}_{t+1-Q+\tau}^k(\mathcal{D}_k) - \sum_{\tau=0}^{Q-1} \eta_{t+1-Q+\tau} \boldsymbol{g}_{t+1-Q+\tau}^k(\mathcal{D}_k') \right\|^2 \\
&\leq 4\eta_{t+1}^2 \left\| \sum_{\tau=0}^{Q-1} \left(\boldsymbol{g}_{t+1-Q+\tau}^k(\mathcal{D}_k) - \boldsymbol{g}_{t+1-Q+\tau}^k(\mathcal{D}_k') \right) \right\|^2 \\
&\leq 4Q^2\eta_{t+1}^2 \sum_{\tau=0}^{Q-1} \left\| \boldsymbol{g}_{t+1-Q+\tau}^k(\mathcal{D}_k) - \boldsymbol{g}_{t+1-Q+\tau}^k(\mathcal{D}_k') \right\|^2 \\
&\leq 16Q^2G^2\eta_{t+1}^2, \quad (11)
\end{aligned}$$

where the first inequality is by $\eta_{t+1-Q} \leq 2\eta_{t+1}$ for $\tau = 0, 1, \ldots, Q-1$, and the last inequality is by Assumption 4. ■

3.2. Convergence Result

Let $\boldsymbol{w}^\star$ be the optimal solution to problem (1) and the convergence of Algorithm 1 is measured by $\varepsilon \triangleq \mathbb{E}\left[F(\bar{\boldsymbol{w}}_T) - F(\boldsymbol{w}^\star)\right]$. The convergence result is obtained as follows.

Theorem 1 *Assume the Assumption 1 - 4 hold. Let* $\kappa = \frac{L}{\mu} > 1$, $\alpha = 8\kappa$ *and the learning rate* $\eta_t = \frac{2}{\mu(\alpha+t)}$ *such that* $\eta_{t+1} \leq 2\eta_{t+Q}$. *Then, Algorithm 1 satisfies*

$$\mathbb{E}\left[F(\bar{\boldsymbol{w}}_T) - F(\boldsymbol{w}^\star)\right] \leq \frac{2\kappa}{T+\alpha} \left(\frac{A+B+C}{\mu} + 2L\|\boldsymbol{w}_0 - \boldsymbol{w}^\star\|^2 \right),$$

where

$$A = \sum_{k=1}^K p_k^2 \frac{\gamma_k^2}{b} + 8(Q-1)^2G^2 + 6L\Gamma, \quad (12a)$$

$$B = \frac{64MQ^4b^2G^2}{nL^2\epsilon^2} \log(n) \log(\frac{1.25^2Q^2b^2}{n\delta^2}) \left(8(Q-1)^2 + 1\right), \quad (12b)$$

$$C = \frac{N-K}{N-1} \frac{4}{K} Q^2G^2, \ \ \Gamma = F(\boldsymbol{w}^\star) - \sum_{k=1}^K p_k F_k^\star, \quad (12c)$$

where $F_k^\star = \min_{\boldsymbol{w}} F_k(\boldsymbol{w})$, $k \in [N]$.

Proof: The proof primarily follows that in [13], with additional consideration of added noise in (6). Due to limited space, details will be presented in the future publication. ■

In (12c), the term Γ reflects the heterogeneity of the data distribution across the clients, and it can impact the algorithm convergence. That is, if data in $\mathcal{D}_k$, $k \in [N]$ follow similar distributions, then Γ would be close to zero, whereas Γ could be large for non-i.i.d. data distribution [13].

We have the following remark regarding the impact of Q, b and (ϵ, δ)-DP on the algorithm convergence.

Remark 1 Let T_ε be the number of required iterations for Algorithm 1 to achieve an ε convergence accuracy. Then, by (12a), the number of required communication round T_ε/Q is

$$\begin{aligned}
\frac{T_\varepsilon}{Q} \propto \frac{Mb^2G^2\log(n)\log(Q^2b^2/n\delta^2)}{nL^2\epsilon^2} Q^5 + \left(1 + \frac{1}{K}\right) G^2Q \\
+ \frac{\sum_{k=1}^K p_k^2\gamma_k^2/b + L\Gamma}{Q}. \quad (13)
\end{aligned}$$

Note that when Q is small and the data size n is large, the first term in the right hand side (RHS) of (13) may become negligible, and thereby the impact of (ϵ, δ)-DP on the algorithm convergence could become minor. Under the same condition, one can see from (13) that increasing Q may benefit the convergence speed, but an over-large Q may slow down the algorithm convergence. So there exists an optimal Q such that the T_ε/Q is minimal.

For the mini-batch size b, one can see that when Q is large such that the first term in the RHS of (13) dominates, increasing b can deteriorate the convergence speed; otherwise, a large value of b may improve the algorithm convergence like the conventional FedAvg without DP. These observations will be verified by experimental results in the next section.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1. Experiment Setting

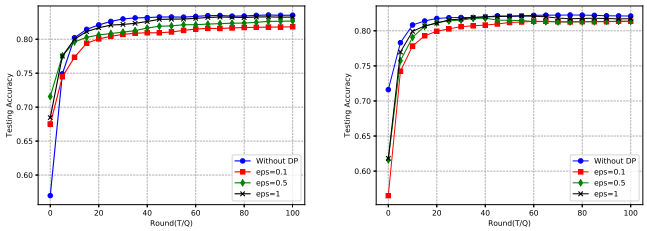
In this section, we evaluate the performance of the secure FedAvg algorithm (Algorithm 1) for a logistic regression problem. Denote the $\ell(\boldsymbol{w}; \boldsymbol{x}_i)$ as the prediction model with parameter $(\boldsymbol{w}, b)$. The loss function is Cross entropy, which is defined as

$$F_k(\boldsymbol{w}) = -\frac{1}{n_k} \sum_{i=1}^{n_k} \left[y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i) \right] + \lambda \|\boldsymbol{w}\|^2,$$

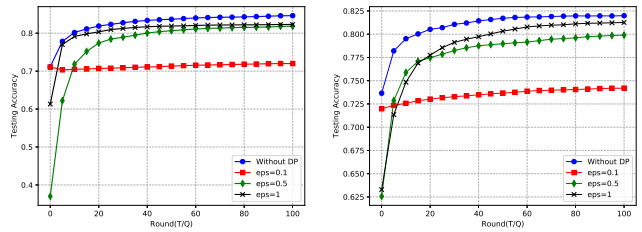
where $y_i \in \{0, 1\}$, $\hat{y}_i = 1/\left(1 + \exp(-\boldsymbol{w}^T\boldsymbol{x}_i + b)\right)$ and $\lambda = 10^{-4}$ is the regularization parameter. Note that, this is a strongly convex optimization problem.

Datasets: The benchmark dataset is Adult [19], which consists of 32561 training samples and 16281 testing samples. In data preprocessing, the feature is normalized and the missing values in the dataset are replaced by the most frequently occurring value in each feature. All the training samples are uniformly distributed among $N = 100$ clients. Beside the i.i.d. case, we also consider the non-i.i.d. case where training samples are distributed among 100 clients such that each client only contains one class.

Parameter setting: In all experiments, we set $\delta = 10^{-4}$ and assume the diminishing learning rate follows the scheme of $\eta_t = \frac{1/5}{1+0.01t}$. We set the maximal local gradient $G = 2$, which is determined by performing the gradient clipping [20] on each iteration. The required noise power for (ϵ, δ)-DP is obtained by (9) and Lemma 1.

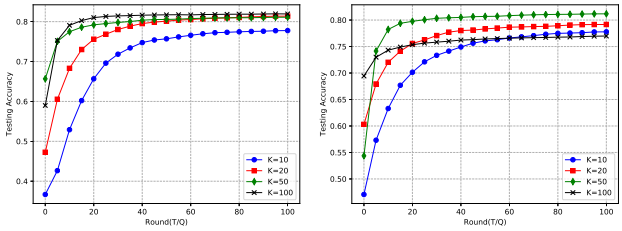


(a) iid data ($Q = 1$) (b) non-iid data ($Q = 1$)



(c) iid data ($Q = 10$) (d) non-iid data ($Q = 10$)

Fig. 1: Impact of ϵ.



(a) iid data (b) non-iid data

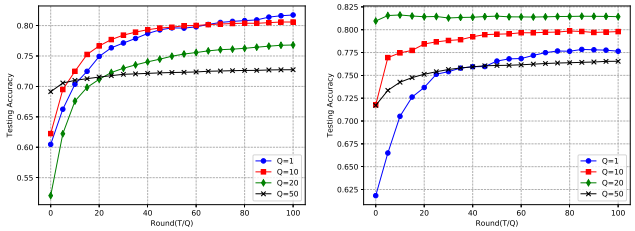
Fig. 2: Impact of K.

4.2. Comparison analysis on various parameters

Impact of the privacy protection level ϵ: We show in Fig.1 the testing accuracy versus communication round (T/Q) of Algorithm 1 for different values of ϵ, and $K = 50$, $Q \in \{1, 10\}$ and $b = 1$. One can see from this figure that the testing accuracy is better for lower protection level (i.e., larger ϵ) and one for the case "without DP" (i.e., $\sigma^2_{t,k} = 0$) is the best as expected. Moreover, these results are quite close for $Q = 1$, while they are quite different for $Q = 10$, implying that the testing accuracy is more sensitive to Q.

Impact of the number of chosen clients K: Figure 2 shows the testing accuracy versus T/Q of Algorithm 1 for different values of K, and $\epsilon = 0.5$, $Q = 10$ and $b = 1$. One can see from this figure that the testing accuracy is better for larger K for the i.i.d. data case, however, it is not necessarily true for the non-i.i.d. data case. The reason for this may be that more active clients impose noise for larger K to the FL system during the training.

Impact of the number of local SGD updates Q: Figure 3 shows the testing accuracy versus T/Q of Algorithm 1 for different values of Q, as well as $\epsilon = 0.5$, $K = 10$, $b = 1$. It



(a) iid data (b) non-iid data

Fig. 3: Impact of Q.

can be seen from this figure that the testing accuracy is better for smaller Q for the i.i.d. data case, but this does not apply to the case for the non-i.i.d. data case, implying that there may exist an optimal Q to balance the communication efficiency and the testing accuracy performance.

Impact of mini-batch size b: Figure 4 shows the testing accuracy of Algorithm 1 for different values of b, along with $\epsilon = 0.5$, $K = 50$, $Q \in \{1, 10\}$. It can be seen that its testing accuracy is better for larger b for the case of $Q = 1$, but this is reverse for $Q = 10$. The above simulations results are consistent with the analyses in Remark 1.

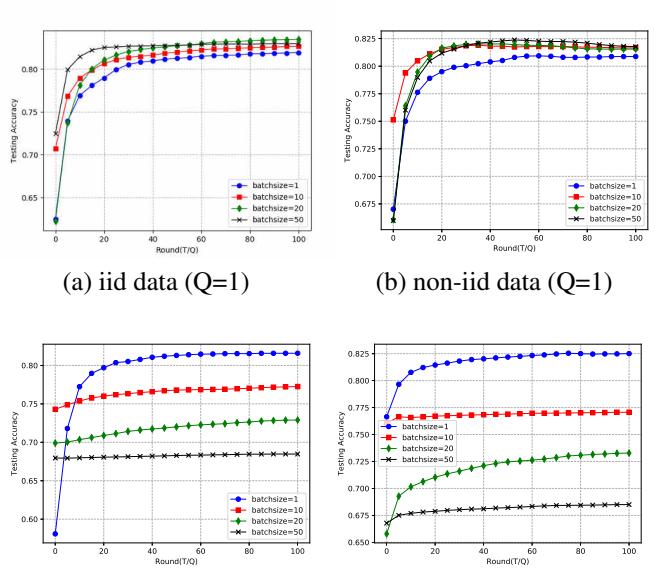
5. CONCLUSIONS

We have presented a secure FedAvg algorithm by employing the DP technique, and its convergence analysis. We have analytically shown that the secure FedAvg can maintain the $\mathcal{O}(1/T)$ convergence rate, together with the trade-off between the communication efficiency and the desired privacy protection level, and how its performance depends on all the designed parameters. Then we have provided some experimental results to support the effectiveness of the algorithm and all the analytical results. Specifically, the mini-batch size b and the number of local model parameters updates Q in every communication round are key parameters, that are essential to the proposed secure FedAvg algorithm in a non-trivial manner due to the DP applied, which is different from the conventional FedAvg algorithm.

6. REFERENCES

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

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



(a) iid data (Q=1) (b) non-iid data (Q=1)

(c) iid data (Q=10) (d) non-iid data (Q=10)

Fig. 4: Impact of b.

[3] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities and challenges," *arXiv preprint arXiv:1908.06847*, 2019.

[4] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," in *Proc. IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sept. 2019, pp. 1–6.

[5] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," *arXiv preprint arXiv:1807.00459*, 2018.

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

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

[8] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farhad, S. Jin, T. Q. Quek, and H. V. Poor, "Performance analysis on federated learning with differential privacy," *arXiv preprint arXiv:1911.00222*, 2019.

[9] A. Triastcyn and B. Faltings, "Federated learning with Bayesian differential privacy," *arXiv preprint arXiv:1911.10071*, 2019.

[10] Y. Kang, Y. Liu, and W. Wang, "Weighted distributed differential privacy ERM: Convex and non-convex," *arXiv preprint arXiv:1910.10308*, 2019.

[11] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Trans. Information Forensics and Security*, vol. 15, pp. 1002–1012, 2019.

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

[13] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[14] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Proc. ACM Neural Information Processing Systems (NIPS)*, Dec. 2010, pp. 2595–2603.

[15] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2015, pp. 1322–1333.

[16] S. U. Stich, "Local SGD converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

[17] B. Balle, G. Barthe, and M. Gaboardi, "Privacy amplification by subsampling: Tight analyses via couplings and divergences," in *Proc. ACM Neural Information Processing Systems (NIPS)*, Dec. 2018, pp. 6277–6287.

[18] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. ACM Neural Information Processing Systems (NIPS)*, Dec. 2018, pp. 4447–4458.

[19] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," 1998, [http://www.ics.uci.edu/rvmlearnIMLRepository.html]. Irvine. CA: University of California, Department of Information and Computer Science.

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