A BYZANTINE-RESILIENT DUAL SUBGRADIENT METHOD FOR VERTICAL FEDERATED LEARNING

Kun Yuan[*] *Zhaoxian Wu*[†] *Qing Ling*[†]

[*]DAMO Academy, Alibaba Group [†]Sun Yat-Sen University

ABSTRACT

Federated learning (FL) raises new challenges on security risks, especially when the FL system involves Byzantine clients that send corrupted or adversarial messages to the central server for deteriorating the training paradigm. While there is an extensive research on robust algorithms for horizontal or data-partitioned FL problems, the exploration in Byzantine-resilient vertical or feature-partitioned FL is quite limited. In this paper, we provide a problem formulation of vertical FL in the presence of Byzantine attacks, and propose a Byzantine-resilient dual subgradient method. Convergence analysis is established, and the influence of the Byzantine clients is also clarified. Numerical experiments show the proposed algorithm is robust to various Byzantine attacks on vertical FL.

Index Terms— Vertical federated learning, Byzantine-resilience, dual subgradient method

1. INTRODUCTION

Federated learning (FL) [1–4] is a prevalent machine learning framework in which a central server and multiple clients collaborate to train a machine learning model. In FL paradigms, the data samples are allowed to stay inside local clients such as cellphones, vehicles, laptops, or the wearable devices. With the coordination of the central server, all clients will work together to complete learning tasks.

Compared with existing distributed learning settings, FL raises new challenges on security risks. In FL applications, a number of clients may be highly unreliable [5, 6]. These clients, which are referred to as Byzantine clients, may send corrupted or even adversarial messages to the server [7]. The existence of Byzantine clients can severely deteriorate the learning paradigm. For example, stochastic gradient descent (SGD) [8], a popular method for large-scale federated learning, is vulnerable to even one Byzantine client [9].

There has been an extensive study on Byzantine-resilient learning algorithms in FL. Most of existing Byzantine-resilient learning methods consider the scenario where each client has a different set of data samples but all clients share common features. This setting is also referred to as data-partitioned or *horizontal* FL [1, 2, 10]. However, there is limited exploration in Byzantine-resilient learning methods for *vertical* FL [11, 12], in which all clients own the same dataset but each client has a unique set of features. Vertical FL is very common in e-commerce, financial, and healthcare applications, which are more sensitive to privacy leakage and Byzantine attacks.

This paper is devoted to Byzantine-resilient algorithms for vertical FL. The contributions of this paper are summarized as follows:

- We provide a mathematical formulation for the vertical FL problem under Byzantine attacks. Prior to this work, the research on Byzantine-resilient vertical FL is quite limited.
- We develop a Byzantine-resilient dual subgradient method to solve the vertical FL problem under Byzantine attacks. We also analyze its performance in terms of the convergence rate as well as the error caused by the Byzantine attacks.
- We conduct numerical experiments to show the effectiveness of the proposed method under various Byzantine attacks.

Related work. Byzantine-resilient optimization has been an emerging paradigm for large-scale distributed learning. Most of the existing works consider the horizontal FL scenario in which they extend SGD to the Byzantine-resilient setting. In this scenario, an unbiased stochastic gradient based on local data samples will be sent to the server by each regular client, while messages sent by Byzantine clients could be arbitrary or even adversarial. To avoid the deviation caused by Byzantine clients, the central server has to aggregate the stochastic gradients in a robust manner. Commonly-used gradient aggregation policies include geometric median [9, 13], marginal trimmed mean [14], dimensional median [15], and Bulyan [16]. In a recent Krum algorithm [17], the server utilizes a gradient with minimal summation of Euclidean distances from several nearest gradients. Building on these methods, more advanced techniques such as variance reduction [18] and momentum [19] are proposed to further promote the robustness of the above aggregation approaches.

The major difficulty in the development of the robust horizontal FL methods lies in the data heterogeneity between local clients. Most of the above mentioned algorithms are designed for tasks with independent and identically distributed (IID) data. When local data follow non-IID distributions, some of the above methods will collapse and do not have performance guarantee. Various methods have been proposed to handle the non-IID data distributions. Robust stochastic model aggregation (RSA) [20] suggests performing model aggregation rather than gradient aggregation. Another useful algorithm is to resample the received gradients [21] so that the variance incurred by non-IID data distributions can be significantly reduced. However, such an approach is at the cost of tolerating a smaller number of Byzantine clients [22]. It is worth noting that all methods discussed above are for robust horizontal FL. The exploration for robust vertical FL is rather limited. While some recent works [10, 23, 24] have developed effective methods for vertical FL or learning with distributed features, they are not robust to Byzantine attacks.

2. PROBLEM FORMULATION

Notation. Given a vector x, we let $x_{[i]} \in \mathbb{R}$ be its i-th element. We use $\mathrm{col}\{x_{[1]}, \cdots, x_{[d]}\}$ to denote a vector formed by stacking $x_{[1]}, \cdots, x_{[d]}$. Given an integer N, we define $[N] := \{1, \cdots, N\}$.

Qing Ling is supported by NSF China Grant 61973324, Guangdong Basic and Applied Basic Research Foundation Grant 2021B1515020094, and Guangdong Provincial Key Laboratory of Computational Science Grant 2020B1212060032.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

ℓ_2-regularized finite-sum minimization. We consider the following ℓ_2-norm regularized finite-sum minimization problem with a linear classifier

$$\min_{w\in\mathbb{R}^d} \quad \sum_{n=1}^{N} \ell(x_n^T w; y_n) + \frac{\rho}{2}\|w\|^2, \tag{1}$$

where $w \in \mathbb{R}^d$ is the unknown variable (also known as the model parameter) to optimize, $\{x_n, y_n\}_{n=1}^N$ is the training dateset in which $x_n \in \mathbb{R}^d$ is a feature vector while $y_n \in \mathbb{R}$ is the corresponding label. The notion $\ell(z; y)$ refers to the loss function and is assumed to be differentiable and convex over $z \in \mathbb{R}$. Typical examples of the loss function are the linear regression in which $\ell(z; y) = \frac{1}{2}(z - y)^2$, and the logistic regression $\ell(z; y) = \log(1 + \exp(-yz))$. The regularization term $\|w\|^2$ is to promote the strong convexity in problem (1), and reduce the chance of overfitting during the training stage. The constant $\rho > 0$ will control the magnitude of the ℓ_2-norm.

Vertical FL formulation. We consider solving problem (1) in the vertical FL setting. Suppose the dataset $\{x_n, y_n\}_{n=1}^N$ is maintained by a set of M clients $\mathcal{M} := \{1, \cdots, M\}$. Each client m maintains a unique feature fraction $x_{n,m} \in \mathbb{R}^{d_m}$ and the label y_n for $n = 1, \cdots, N$. Throughout this paper, we assume there is no feature overlap between any two different fractions and hence $d = \sum_{m=1}^M d_m$ holds. If we let $w_m \in \mathbb{R}^{d_m}$ be the m-th model fraction and $\ell_n(x_n^T w) := \ell(x_n^T w; y_n)$, then problem (1) can be rewritten as

$$\min_{w\in\mathbb{R}^d} \quad \sum_{n=1}^{N} \ell_n\big(\sum_{m=1}^{M} x_{n,m}^T w_m\big) + \frac{\rho}{2}\|w\|^2. \tag{2}$$

The M clients will collaborate to solve problem (2). Problem (2) is in the vertical FL form because all clients use the same dataset while each client maintains a different fraction of the feature vectors.

Robust vertical FL formulation. Now we consider a scenario in which Byzantine clients exist among the set of M clients. Since the Byzantine clients can prevent the server from accessing their local feature fractions, it is meaningless to solve problem (2), which utilizes the feature fractions from all clients without distinguishing regular and Byzantine clients. Instead, a reasonable goal is to find a solution that solves the empirical minimization problem with feature fractions from regular clients, given by

$$\min_{w\in\mathbb{R}^d} \quad \sum_{n=1}^{N} \ell_n\big(\sum_{m\in\mathcal{R}} x_{n,m}^T w_m\big) + \frac{\rho}{2}\sum_{m\in\mathcal{R}}\|w_m\|^2, \tag{3}$$

where we respectively denote $\mathcal{B}$ and $\mathcal{R}$ as the sets of Byzantine and regular clients. We let $|\mathcal{B}| = B$ and $|\mathcal{R}| = R$ so that $B + R = M$.

3. BYZANTINE-RESILIENT DUAL SUBGRADIENT

3.1. Robust vertical FL problem in the dual domain

We examine problem (3) in the dual domain. We introduce an auxiliary vector $z := \mathrm{col}\{z_{[n]}\} \in \mathbb{R}^N$ and rewrite the problem as

$$\begin{aligned} \min_{w,z} \quad & \sum_{n=1}^{N} \ell_n(z_{[n]}) + \frac{\rho}{2}\sum_{m\in\mathcal{R}}\|w_m\|^2 \\ \text{s.t.} \quad & \sum_{m\in\mathcal{R}} x_{n,m}^T w_m = z_{[n]}, \quad \forall n \in [N]. \end{aligned} \tag{4}$$

Next we introduce the dual variable $\theta := \mathrm{col}\{\theta_{[n]}\} \in \mathbb{R}^N$, with each $\theta_{[n]}$ associated with the n-th constraint. The following lemma establishes the dual problem of problem (4).

Lemma 1. *The dual problem of problem* (4) *is given by*

$$\min_{\theta\in\mathbb{R}^N} \quad \sum_{n=1}^{N} \ell_n^*(\theta_{[n]}) + \frac{1}{2\rho}\sum_{m\in\mathcal{R}}\|X_m\theta\|^2. \tag{5}$$

where $\ell_n^*(\theta_{[n]}) := \sup_{z_{[n]}}\{\theta_{[n]} z_{[n]} - \ell_n(z_{[n]})\}$ is the conjugate function of $\ell_n(z_{[n]})$, and $X_m := [x_{1,m}, \cdots, x_{N,m}] \in \mathbb{R}^{d_m\times N}$ is the set of feature fractions maintained by client m. When the global solution $\theta^\star$ is achieved, client m can recover its primal optimal solution by $w_m^\star = -\frac{1}{\rho}X_m\theta^\star$.

Proof. The proof simply follows the definition of the dual problem, and is omitted due to the page limit. □

For notation simplicity, we rewrite the dual problem (5) as

$$\theta^\star = \arg\min_{\theta\in\mathbb{R}^N} \big\{ \sum_{m\in\mathcal{R}} F_m(\theta) + \rho F_0(\theta)\big\}, \tag{6}$$

where $F_0(\theta) := \sum_{n=1}^N \ell_n^*(\theta_{[n]})$ and $F_m(\theta) := \frac{1}{2}\|X_m\theta\|^2$.

Remark 1 (PROPERTY OF $F_0(\theta)$). *Since each $\ell_n(z_{[n]})$ is differentiable but not necessarily strongly-convex (e.g., the logistic regression loss), it is known from [25, 26] that its conjugate function $\ell_n^*(\theta_{[n]})$, and hence $F_0(\theta)$, is strongly convex but not necessarily differentiable. Furthermore, it is easy to verify that*

$$\partial_{\theta_{[n]}}\ell_n^*(\theta_{[n]}) := \big\{z_{[n]}^+ | z_{[n]}^+ = \arg\min_{z_{[n]}}\{\ell_n(z_{[n]}) - \theta_{[n]} z_{[n]}\}\big\}.$$

If this set is a singleton, then $\ell_n^*(\theta_{[n]})$ is differentiable in terms of $\theta_{[n]}$. We further define

$$\partial F_0(\theta) := \mathrm{col}\big\{\partial_{\theta_{[n]}}\ell_n^*(\theta_{[n]})\big\} \in \mathbb{R}^N \tag{7}$$

as the subgradient of $F_0(\theta)$. Since $F_0(\theta)$ is strongly-convex, it holds that $\theta^\star$ is the unique global solution to problem (6)*.*

Remark 2 (PROPERTY OF $F_m(\theta)$). *Recall $F_m(\theta) = \frac{1}{2}\|X_m\theta\|^2$, it holds that $F_m(\theta)$ is convex and differentiable with gradient given y $\nabla F_m(\theta) = X_m^T X_m\theta$. Note that $X_m \in \mathbb{R}^{d_m\times N}$ where N is the sample size and d_m is the number of features maintained by client m. If the problem is over-parameterized so that $d_m > N$ for any $m \in \mathcal{R}$, then $F_m(\theta)$ will be strongly convex. Otherwise, $F_m(\theta)$ is not strongly convex in the general scenarios.*

3.2. Approximate constrained dual problem

Since problem (6) is non-differentiable, one standard approach to solving it is the *subgradient method.* To avoid the unbounded iterates generated by the subgradient method, we consider an approximate constrained dual problem as

$$\tilde{\theta}^\star = \arg\min_{\theta\in\mathcal{C}} \big\{ \sum_{m\in\mathcal{R}} \big(F_m(\theta) + \frac{\beta}{2}\|\theta\|^2\big) + \rho F_0(\theta)\big\}. \tag{8}$$

HEre $\beta > 0$ is to promote the strong convexity, and $\mathcal{C}$ is a compact box constraints defined as

$$\mathcal{C} = \{\theta \in \mathbb{R}^N | -c\mathbb{1}_N \le \theta \le c\mathbb{1}_N\}. \tag{9}$$

where $c > 0$ and $\mathbb{1}_N \in \mathbb{R}^N$ is a vector with all elements being 1. Note that $\tilde{\theta}^\star$ is also unique to problem (8) due to the strong convexity. The distance between $\tilde{\theta}^\star$ and $\theta^\star$ can be characterized as follows.

Lemma 2. *We let $\theta^\star$ and $\tilde{\theta}^\star$ be the unique global solutions to problem* (6) *and* (8)*, respectively. It holds that*

$$\|\tilde{\theta}^\star - \theta^\star\|^2 = \Delta_{c,\beta}, \tag{10}$$

where $\Delta_{c,\beta}$ is a constant dependent on c and β, and $\Delta_{c,\beta} \to 0$ as $\beta \to 0$ and c becomes sufficiently large so that $\theta^\star \in \mathcal{C}$.

This lemma implies that the distance between $\tilde{\theta}^\star$ and $\theta^\star$ can be controlled by adjusting parameters β and c. In the following context, we will focus on solving the approximate constrained dual problem (8) instead of the original dual problem (6).

3.3. Byzantine-resilient dual subgradient method

Now we develop a Byzantine-resilient dual subgradient method to solve problem (8). Inspired by the RSA algorithm [20], we rewrite problem (8) into

$$\begin{aligned} \min_{\{\theta_m,\theta_0\}\in\mathcal{C}} \quad & \sum_{m\in\mathcal{R}} \left(F_m(\theta_m) + \frac{\beta}{2}\|\theta_m\|^2\right) + \rho F_0(\theta_0), \\ \text{s.t.} \quad & \theta_m = \theta_0, \quad \forall m \in \mathcal{R}. \end{aligned} \tag{11}$$

where θ_m is the local variable held by client m while θ_0 is held by the server. Next we introduce an ℓ_1-norm regularized form of (11), given by

$$\min_{\{\theta_m,\theta_0\}\in\mathcal{C}} \sum_{m\in\mathcal{R}} \Big(F_m(\theta_m) + \frac{\beta}{2}\|\theta_m\|^2 + \lambda\|\theta_m - \theta_0\|_1\Big) + \rho F_0(\theta_0), \tag{12}$$

in which $\lambda > 0$ is a penalty parameter. The above problem can be solved via the projected subgradient method, given by

$$\begin{aligned} \theta_m^{k+1} &= \mathrm{Proj}_{\mathcal{C}}\{\theta_m^k - \alpha^k\Big(\nabla F_m(\theta_m^k) \\ &\qquad + \beta\theta_m^k + \lambda\,\mathrm{sign}(\theta_m^k - \theta_0^k)\Big)\}, \ \forall m \in \mathcal{R}, \qquad (13) \\ \theta_0^{k+1} &= \mathrm{Proj}_{\mathcal{C}}\{\theta_0^k - \alpha^k\Big(\rho\partial F_0(\theta_0^k) \\ &\qquad + \lambda\sum_{m\in\mathcal{R}} \mathrm{sign}(\theta_0^k - \theta_m^k)\Big)\}, \qquad (14) \\ w_m^{k+1} &= -\frac{1}{\rho}X_m\theta_m^{k+1}, \ \forall m \in \mathcal{R}, \qquad (15) \end{aligned}$$

where $\alpha^k > 0$ is the learning rate, $\mathrm{sign}(\cdot)$ is the element-wise sign function, and $\mathrm{sign}(0)$ can be any value within $[-1, 1]$. Notation $\mathrm{Proj}_{\mathcal{C}}(\theta)$ denotes the projection onto set $\mathcal{C}$. Gradient $\nabla F_m(\theta) := X_m^T X_m\theta$ while subgradient $\partial F_0(\theta)$ is defined in (7).

Next we consider how the updates in (13)–(14) behave in the presence of Byzantine clients. The update of regular client is the same as in (13). However, due to the unknown identities of the Byzantine clients, the server will update as

$$\begin{aligned} \theta_0^{k+1} = \mathrm{Proj}_{\mathcal{C}}\{\theta_0^k - \alpha^k\Big(\rho\,\partial F_0(\theta_0^k) + \lambda\sum_{m\in\mathcal{R}} \mathrm{sign}(\theta_0^k - \theta_m^k) \\ + \lambda\sum_{m\in\mathcal{B}} \mathrm{sign}(\theta_0^k - \xi_m^k)\Big)\}, \end{aligned} \tag{16}$$

where $\xi_m^k \in \mathbb{R}^N$ is an arbitrary vector sent by Byzantine client m. The Byzantine-resilient method for vertical FL is in Algorithm 1.

Algorithm 1 Byzantine-Resilient Dual Subgradient Method

```
Central Server:
Input: $\theta_0^0, \rho, c, \lambda$, and learning rate $\alpha^k$. At iteration $k+1$:
Broadcast its current iterates $\theta_0^k$ to all clients;
Receive all local iterates $\theta_m^k$ sent by regular clients or faulty values $\xi_m^k$ sent by Byzantine clients;
Update $\theta_0^{k+1}$ via (16).

Regular client $m$:
Input: $\theta_m^0, \rho, c, \beta, \lambda$, and learning rate $\alpha^k$. At iteration $k+1$:
Send the current local iterate $\theta_m^k$ to the server;
Receive the local iterate $\theta_0^k$ from the server;
Update $\theta_m^{k+1}$ and $w_m^{k+1}$ via (13) and (15), respectively.
```

4. PERFORMANCE ANALYSIS

This section will examine the convergence property of the Byzantine-resilient dual subgradient method listed in Algorithm 1. To this end, we first give several standard assumptions.

Assumption 1 (SMOOTHNESS)**.** *Each loss function $\ell_n(z_{[n]})$ is convex and differentiable in terms of $z_{[n]}$, and its gradient is Lipschitz continuous with constant L for any $n = 1, \cdots, N$.*

Remark 3 (STRONG CONVEXITY OF $F_0(\theta)$)**.** *Under Assumption 1, it holds that $\ell_n^*(\theta_{[n]})$ is strongly-convex with constant $1/L$ [25, 26]. This implies that $F_0(\theta) = \sum_{n=1}^N \ell_n^*(\theta_{[n]})$ is also strongly convex with constant $1/L$ within the constraint set $\mathcal{C}$. In other words, for every $\theta \in \mathcal{C}$ and $g \in \partial F_0(\theta)$, it holds that*

$$F_0(\phi) \geq F_0(\theta) + g^T(\phi - \theta) + \frac{1}{2L}\|\phi - \theta\|^2, \quad \forall \phi \in \mathcal{C}. \tag{17}$$

Assumption 2 (BOUNDED SUBGRADIENT)**.** *We assume $\|g_\theta\| \leq G$ for any $g_\theta \in \partial F_0(\theta)$ and $\theta \in \mathcal{C}$.*

Proof highlights in performance analysis. The major idea to prove the convergence of Algorithm 1 is as follows. We first need to establish two facts:

- The solution to the ℓ_1-regularized approximate problem (12) is consensual and identical to that of (8) (Theorem 1).
- The iterates generated via Algorithm 1 can converge towards the solution to problem (12) (Theorem 2).

Combining the above two facts, we can conclude that the iterates generated via Algorithm 1 can converge towards $\tilde{\theta}^\star$, the solution to problem (8). This result together with Lemma 2 and algorithm update (15) can lead to the bound for $\|w_m^k - w_m^\star\|^2$ (Corollary 1). Due to page limits, we omit the proof details of all these theorems.

Theoretical results. The following theorem establishes the condition under which the optimal solution to (12) is consensual and identical as that to (8). Note that the proof is different from Theorem 1 in [20] due to the existence of the box constraint $\mathcal{C}$.

Theorem 1. *Suppose Assumption 1 holds. If*

$$\lambda \geq \lambda_0 = \max_{m\in\mathcal{R}} \|\nabla F_m(\tilde{\theta}^\star)\|_\infty$$

in which $\tilde{\theta}^\star$ is the global solution to (8)*, it holds that the solution to problem* (12) *follows that $\tilde{\theta}_m^\star = \tilde{\theta}^\star, \forall m \in \mathcal{R}$ and $\tilde{\theta}_0^\star = \tilde{\theta}^\star$.*

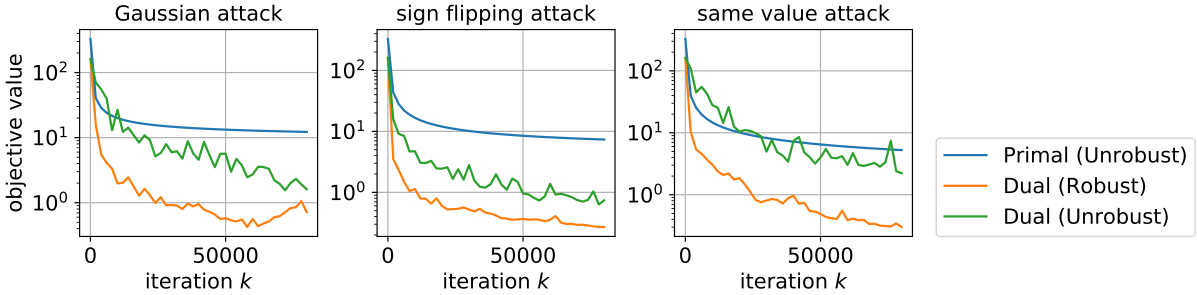

Fig. 1. Performance of different methods on linear regression under various Byzantine attacks

The following theorem establishes the convergence performance of Algorithm 1 to the optimal solution to problem (8).

Theorem 2. *Under Assumptions 1 and 2, if constant learning rate $\alpha^k = \alpha$ satisfies $\alpha < 1/(\beta+\delta)$, it holds that*

$$\sum_{m\in\mathcal{R}} \|\theta_m^k - \tilde{\theta}_m^\star\|^2 + \|\theta_0^k - \tilde{\theta}_0^\star\|^2 \le \left(1-\alpha\underline{\eta}\right)^k \Delta_0 + \frac{\alpha\Delta_1}{\underline{\eta}} + \frac{\Delta_2}{\underline{\eta}}, \tag{18}$$

where $\delta := \max_m\{\lambda_{\max}(X_m^T X_m)\} + \beta$, and

$$\underline{\eta} := \min\{\frac{2\beta\delta}{\beta+\delta}, \frac{\rho}{L}\}, \tag{19}$$

$$\Delta_0 := \sum_{m\in\mathcal{R}} \|\theta_m^0 - \tilde{\theta}_m^\star\|^2 + \|\theta_0^0 - \tilde{\theta}_0^\star\|^2, \tag{20}$$

$$\Delta_1 := 8NR\lambda + 12\lambda^2R^2N + 3\lambda^2B^2N + 6\rho^2G^2, \tag{21}$$

$$\Delta_2 := \frac{\lambda^2B^2NL}{\rho}. \tag{22}$$

Corollary 1. *Under Assumptions 1 and 2, if diminishing learning rate $\alpha^k = O(\ln(k)/k)$ and k is sufficiently large such that $\alpha^k < 1/(\beta+\delta)$, it holds that*

$$\sum_{m\in\mathcal{R}} \|w_m^k - w_m^\star\|^2 \le \frac{\Delta_0'}{k} + \frac{\ln(k)\Delta_1'}{k} + \Delta_2', \tag{23}$$

where $\gamma := \max_m\{\lambda_{\max}(X_m^T X_m)\}$ and

$$\Delta_0' := \frac{2\gamma}{\rho}\Delta_0 = O(R), \tag{24}$$

$$\Delta_1' := \frac{2\gamma}{\rho}\frac{\Delta_1}{\underline{\eta}} = O\Big((R^2+B^2)N + G^2\Big), \tag{25}$$

$$\Delta_2' := \frac{2\gamma}{\rho}\frac{\Delta_2}{\underline{\eta}} + \frac{2\gamma R}{\rho}\Delta_{c,\beta} = O\Big(B^2N + R\Delta_{c,\beta}\Big). \tag{26}$$

Remark 4. *When there exists no Byzantine clients, Corollary 1 implies that $\sum_{m=1}^M \|w_m^k - w_m^\star\| \to 0$ as $k \to \infty$. When Byzantine clients exist, the proposed algorithm can still converge to a neighborhood around the true solution without diverging, and the neighborhood is on the order of $O(B^2N)$ when $\Delta_{c,\beta} = 0$. This implies the proposed algorithm is applicable to over-parameterized scenarios where $d \gg N$.*

5. NUMERICAL EXPERIMENTS

Experiment setting. This section examines the performance of the Byzantine-resilient dual subgradient method in presence of various Byzantine attacks. We consider the simple linear regression setting, in which the loss function is $\ell(z;y) = \frac{1}{2}(z-y)^2$ and its conjugate function is $\ell_n^*(\theta_{[n]}) = \frac{1}{2}\theta_{[n]}^2 - \theta_{[n]}y_n$. We randomly construct a synthetic dataset with $N = 50$ samples and $d = 200$ features. All feature vectors x_n and the optimal model w^* are sampled from the Gaussian distribution $\mathcal{N}(\mu\mathbb{1}_d, 1)$, where $\mu \sim \mathcal{U}(0,1)$. In addition, the label y_n is generated by $y_n = x_n^T w^* + \sigma_n$ with noise $\sigma_n \sim \mathcal{N}(0, 0.01)$. In the experiments, we set $R = 8$ regular clients and $B = 2$ Byzantine clients. The number of iterations for each tested algorithm is $80,000$ with hyper-parameters as $\lambda = 0.06$, $\beta = 0.001$ and $\rho = 0.005$. The diminishing learning rate $\alpha^k = \alpha^0/(k+k_0)$ is utilized for each algorithm, where α^0 and k_0 are chosen such that the initial learning rate is 0.0005 and it will decrease to $1/100$ of the initial value at iteration $20,000$. The experiments are conducted on a computer with AMD Ryzen 7 4800U CPU@1.80 GHz.

Baseline algorithms. We compare Algorithm 1 with two commonly-used approaches to solving problem (2) in Byzantine-free scenarios: the *primal gradient descent* method and the *dual gradient ascent* method. Note that these two approaches are not robust to Byzantine attacks. The goal of the experiments is to demonsrate the performance difference between Algorithm 1 and these non-Byzantine-resilient algorithms for vertical FL.

Byzantine attacks. We consider three different Byzantine attacks in the experiments: *Gaussian*, *sign flipping* and *same value* attacks. We use notation u_m^k to represent the message sent from client m at time k. For regular clients, u_m^k equals $x_{n,m}^T w_m^k$ for primal gradient descent, $\nabla F_m(\theta^k) + \beta\theta^k$ for dual gradient ascent, and θ_m^k for Byzantine-resilient dual subgradient in Algorithm 1. For Byzantine clients, u_m^k can be an arbitrary vector. In addition, we denote the mean of messages from regular clients as $\bar{u}^k := \frac{1}{R}\sum_{m\in\mathcal{R}} u_m^k$. For the Gaussian, sign flipping and same value attacks, Byzantine client will draw its message from a Gaussian distribution with variance 0.25, the mean $\bar{u}^k$, $-0.5\bar{u}^k$, $\omega^k\mathbb{1}_p/\sqrt{p}$, respectively, where p is the dimension of message u_m^k and $\omega^k := \text{mean}(\bar{u}^k) \in \mathbb{R}$.

Experimental Results. The results are shown in Fig. 1. The performance metric is the objective value of (3). In presence of Byzantine attacks, the performance of both the primal and dual gradient methods degrades rapidly while the proposed Algorithm 1 remains nearly unaffected and still performs well. This illustrates the robustness of Algorithm 1 to Byzantine attacks.

6. CONCLUSION AND FUTURE WORK

This paper studies robust learning methods for vertical FL. The proposed Byzantine-resilient dual subgradient method is both theoretically and empirically effective in the presence of Byzantine clients. However, the current algorithm is a deterministic approach that needs to sample all data to finish one round of training. We will develop variants that only need mini-batch sampling in the future.

7. REFERENCES

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

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, 2017.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[4] 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.

[5] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[6] Z. Yang, A. Gang, and W. U. Bajwa, "Adversary-resilient inference and machine learning: From distributed to decentralized," *arXiv preprint arXiv: 1908.08649*, 2019.

[7] N. A. Lynch, *Distributed algorithms*. Elsevier, 1996.

[8] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT*, pp. 177–186, Springer, 2010.

[9] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

[10] T. Chen, X. Jin, Y. Sun, and W. Yin, "Vafl: a method of vertical asynchronous federated learning," *arXiv preprint arXiv:2007.06081*, 2020.

[11] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *arXiv preprint arXiv:1711.10677*, 2017.

[12] Y. Liu, Y. Kang, L. Li, X. Zhang, Y. Cheng, T. Chen, M. Hong, and Q. Yang, "A communication efficient vertical federated learning framework," *arXiv preprent arXiv: 1912.11187*, 2019.

[13] C. Xie, O. Koyejo, and I. Gupta, "Generalized Byzantine-tolerant SGD," *arXiv preprint arXiv: 1802.10116*, 2018.

[14] C. Xie, O. Koyejo, and I. Gupta, "Phocas: Dimensional Byzantine-resilient stochastic gradient descent," *arXiv preprint arXiv: 1805.09682*, 2018.

[15] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *nternational Conference on Machine Learning*, pp. 5650–5659, 2018.

[16] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *International Conference on Machine Learning*, pp. 3521–3530, 2018.

[17] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, pp. 118–128, 2017.

[18] Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis, "Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4583–4596, 2020.

[19] S. P. Karimireddy, L. He, and M. Jaggi, "Learning from history for Byzantine robust optimization," in *International Conference on Machine Learning*, pp. 5311–5319, 2021.

[20] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1544–1551, 2019.

[21] L. He, S. P. Karimireddy, and M. Jaggi, "Byzantine-robust learning on heterogeneous datasets via resampling," *arXiv preprint arXiv: 2006.09365*, 2020.

[22] J. Peng, Z. Wu, and Q. Ling, "Byzantine-robust variance-reduced federated learning over distributed non-iid data," *arXiv preprint arXiv: 2009.08161*, 2020.

[23] V. Smith, S. Forte, M. Chenxin, M. Takáč, M. I. Jordan, and M. Jaggi, "Cocoa: A general framework for communication-efficient distributed optimization," *Journal of Machine Learning Research*, vol. 18, pp. 1–49, 2018.

[24] B. Ying, K. Yuan, and A. H. Sayed, "Supervised learning under distributed features," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 977–992, 2018.

[25] J.-B. Hiriart-Urruty and C. Lemarechal, "Convex analysis and minimization algorithms II," 1993.

[26] X. Zhou, "On the fenchel duality between strong convexity and lipschitz continuous gradient," *arXiv preprint arXiv:1803.06573*, 2018.