BYZANTINE-ROBUST AND COMMUNICATION-EFFICIENT DISTRIBUTED NON-CONVEX LEARNING OVER NON-IID DATA

Xuechao He[*] *Heng Zhu*[*†] *Qing Ling*[*]

[*]Sun Yat-Sen University [†]University of California, San Diego

ABSTRACT

Motivated by the emerging federated learning applications, we jointly consider the problems of *Byzantine-robustness* and *communication efficiency* in distributed *non-convex* learning over *non-IID* data. We propose a compressed robust stochastic model aggregation (C-RSA) method, which applies the idea of robust stochastic model aggregation to achieve *Byzantine-robustness* over *non-IID* data, while compresses the transmitted messages so as to achieve *communication efficiency*. Utilizing the tools of Moreau envelope and proximal point projection, we establish the convergence of C-RSA for distributed *non-convex* learning problems. Numerical experiments on training a large-scale neural network demonstrate the effectiveness of the proposed C-RSA method.

Index Terms— Federated learning, Byzantine-robustness, communication efficiency.

1. INTRODUCTION

The rapidly increasing distributed devices, such as intelligent sensors and mobile phones, are generating a large amount of data. Learning from this distributed data yields better models than learning from a single device, but in the mean time, brings concerns on data privacy. Federated learning addresses this issue by allowing the distributed devices (also termed as workers) to keep their private data and periodically communicate with a master node so as to jointly train a model [1]. In a federated learning system, the distributed data is often *non-IID* (independent and identically distributed). Besides, the underlying distributed learning problem is often *non-convex* due to the popularization of deep learning. In addition to data privacy, *communication efficiency* and *robustness* are two major challenges in federated learning [2, 3, 4].

Message transmissions between the master node and the workers form a bottleneck of a federated learning system. To tackle this problem, one approach is to reduce the communication frequency by letting every worker run multiple local iterations before a transmission [5], or skipping some relatively informative transmissions [6]. Another approach, which is the focus of this paper, is to reduce the communication size by compression, such as quantization and sparsification [7, 8].

On the other hand, the transmitted messages from the workers to the master node are not always reliable; they are subject to corruptions and/or even malicious attacks. We often characterize such unreliability with the worst-case Byzantine attacks model [9].

Most of the Byzantine-robust distributed learning methods modify the popular stochastic gradient descent (SGD) method by replacing the mean aggregation with robust aggregation rules to reduce the Byzantine interference, such as median, trimmed mean, geometric median, Krum, to name a few [10, 11, 12, 13]. However, these methods rely on the IID assumption on the data distribution that means the transmitted stochastic gradients are also IID, and do not fit for the federated learning applications. The work of [14] adopts robust stochastic model aggregation, instead of gradient aggregation, to handle this challenge. In [15] the transmitted stochastic gradients are resampled to reduce the heterogeneity prior to sending for robust aggregation. Note that although there are abundant works on federated learning with non-IID data distribution [16], most of them are not Byzantine-robust. Byzantine-robustness has also been investigated for distributed non-convex learning, generally under the IID assumption. Examples include [17, 18, 19]. The works of [20, 21] study how to escape saddle points and fake local minima created by Byzantine workers.

Motivated by the federated learning applications, in this paper we aim at handling the challenges of *Byzantine-robustness* and *communication efficiency* in distributed *non-convex* learning over *non-IID* data. To the best of our knowledge, there are no prior works that jointly consider these four factors. Byzantine-robust and communication efficient algorithms have been developed in [22, 23, 24], but they are limited to the convex, IID case. The most relevant work is [25], which combines robust stochastic model aggregation [14] to enhance Byzantine-robustness over non-IID data and lazy stochastic gradient updates [6] to improve communication efficiency. However, the analysis and numerical experiments in [25] are on convex problems. In addition, the idea of using lazy stochastic gradient updates for reducing the communication frequency is orthogonal to our idea of using compressions for reducing the communication size.

In this context, our contributions are as follows.

C1) We jointly consider *Byzantine-robustness* and *communication efficiency* in distributed *non-convex* learning over *non-IID* data. The investigated problem is novel, and of practical importance in federated learning systems.

C2) Motivated by [14], we propose a compressed robust stochastic model aggregation (C-RSA) method, which inherits the merit of RSA, namely, achieving *Byzantine-robustness* over *non-IID* data. Different from RSA, C-RSA compresses the transmitted messages, and is hence more *communication-efficient*.

C3) Utilizing the tools of Moreau envelope and proximal point projection, we establish the convergence of C-RSA for distributed *non-convex* learning problems.

C4) We conduct numerical experiments by train a neural network with one third million parameters to demonstrate the effectiveness of the proposed C-RSA method.

2. PROBLEM STATEMENT

Consider a distributed network consisting of one master node and $K = r + b$ workers, among which r workers are regular and b workers are Byzantine. The numbers and identities of regular and Byzan-

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

tine workers are not known by the master node. Denote $\mathcal{R}$ and $\mathcal{B}$ as the sets of regular and Byzantine workers, respectively. The goal is to solve a distributed non-convex learning problem in the form of

$$\tilde{x}^* = \arg\min_{\tilde{x}} \; F(\tilde{x}) \triangleq \sum_{k\in\mathcal{R}} \mathbb{E}[f(\tilde{x}, \xi_k)] + f_0(\tilde{x}). \tag{1}$$

Here $\tilde{x} \in \mathbb{R}^d$ is the optimization variable, $f(\tilde{x}, \xi_k)$ is the non-convex loss function of regular worker k with respect to the random variable ξ_k, and $f_0(\tilde{x})$ is the (possibly non-convex) regularization term held by the master node. We assume that the data on the regular workers are non-IID; that is, $\xi_k \sim \mathcal{D}_k$ where $\mathcal{D}_k$ are the data distributions of regular workers k and can be different with each other.

The main challenges in solving (1) are as follows: (i) During the learning process, the *Byzantine* workers will send malicious messages to bias the decisions of the master node; (ii) The data distributions of the regular workers are *non-IID*; (iii) The loss function and the regularization term are *non-convex*; and (iv) Message transmissions between the master node and the workers are subject to limited bandwidths, and thus the developed algorithms must be *communication-efficient*.

To enable communication-efficient learning, we introduce compressions [7], [8] for the message transmissions. A general compression operator is defined as follows.

Definition 1 (Compression operator). A function $C(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is called a compression operator, if there exists a constant $\gamma \in (0, 1]$, such that for any $\tilde{x} \in \mathbb{R}^d$, it holds

$$\mathbb{E}_C ||\tilde{x} - C(\tilde{x})||^2 \leq (1-\gamma)||\tilde{x}||^2. \tag{2}$$

In this paper, our focus is on the rand-l sparsification-based compression operator, which randomly set l out of d elements in the input vector as zeros, with l being an integer between 1 and d. In this case, $\gamma = \frac{l}{d}$. To transmit the output of rand-l sparsification, we only need to transmit l real values and a random seed to represent their indices.

Another typical sparsification operator is top-l, in which we sort the input vector, find the largest l elements, and then output these elements as well as their original indices. Although top-l sparsification keeps more information of the input vector, it requires extra computation for the sorting, which is remarkable when the dimension d is large. Besides, all the selected l indices must be transmitted, other than a single random seed in rand-l.

3. COMPRESSED BYZANTINE-ROBUST STOCHASTIC MODEL AGGREGATION

When the data distributions of the regular workers are IID and communication efficiency is not at consideration, distributed SGD with Byzantine-robust stochastic gradient aggregation is a powerful tool to solve (1). At time t, the master node broadcasts the global vector $\tilde{x}^t$ to all the workers. Each regular worker $k \in \mathcal{R}$ calculates the stochastic gradient $\nabla f(\tilde{x}^t, \xi_k^t)$, where $\xi_k^t \sim \mathcal{D}_k$ corresponds to a mini-batch of local samples, and sends to the master node. On the other hand, each Byzantine worker $k' \in \mathcal{B}$ generates a malicious message $z_{k'}^t \in \mathbb{R}^d$, and sends to the master node. Then, the master node aggregates the received messages in a Byzantine-robust manner, such as median or geometric median. The aggregated result and $\nabla f_0(\tilde{x}^t)$ are used to yield a descent direction so as to update $\tilde{x}^{t+1}$. However, the effectiveness of Byzantine-robust stochastic gradient aggregation relies on the IID assumption. For non-IID data distributions, the stochastic gradients from the regular workers can be very different, such that the aggregated result is far from the averaged stochastic gradient of the regular workers [14, 15, 25].

Byzantine-robust stochastic model aggregation (RSA) has been proven effective when the data distributions of the regular workers are non-IID [14]. In RSA, the master node maintains a local vector $x_0 \in \mathbb{R}^d$, and each regular worker $k \in \mathcal{R}$ maintains a local vector $x_k \in \mathbb{R}^d$. The insight behind RSA is that, even when the IID assumption is violated, the local models are still expected to be close to each other. Denote $x = [x_0; x_1; \cdots; x_r] \in \mathbb{R}^{(r+1)d}$. Then, (1) is equivalent to

$$\begin{aligned} x^* = \arg\min_{x} \; & \sum_{k\in\mathcal{R}} \mathbb{E}[f(x_k, \xi_k)] + f_0(x_0), \\ s.t. \; & x_0 = x_k, \; \forall k \in \mathcal{R}. \end{aligned} \tag{3}$$

By penalizing the consensus constraints $x_0 = x_k$ for all $k \in \mathcal{R}$ with the ℓ_1-norms, we approximate (3) by

$$\min_{x} \; \sum_{k\in\mathcal{R}} \mathbb{E}[f(\tilde{x}_k, \xi_k)] + f_0(\tilde{x}_0) + \lambda ||x_k - x_0||_1, \tag{4}$$

where $\lambda > 0$ is the penalty parameter. Intuitively, the ℓ_1-norm terms force the local models to be close to each other, but also allow outliers, which is critical for Byzantine-robustness.

Using SGD to solve (4) yields

$$x_0^{t+1} = x_0^t - \alpha \left[\nabla f_0(x_0^t) + \lambda \left(\sum_{k\in\mathcal{R}} \text{sign}(x_0^t - x_k^t) \right) \right], \tag{5}$$

$$x_k^{t+1} = x_k^t - \alpha \left[\nabla f(x_k^t, \xi_k^t) + \lambda \text{sign}(x_k^t - x_0^t) \right], \; \forall k \in \mathcal{R}, \tag{6}$$

where $\alpha > 0$ is the step size and $\text{sign}(\cdot)$ is the element-wise sign function that outputs 1 for positive input, -1 for negative input, and 0 for zero input. When the Byzantine workers are present, (6) remains the same but (5) becomes

$$\begin{aligned} x_0^{t+1} = x_0^t - \alpha \Bigg[& \nabla f_0(x_0^t) + \lambda \left(\sum_{k\in\mathcal{R}} \text{sign}(x_0^t - x_k^t) \right) \\ & + \lambda \left(\sum_{k'\in\mathcal{B}} \text{sign}(x_0^t - z_{k'}^t) \right) \Bigg], \end{aligned} \tag{7}$$

where we recall that $z_{k'}^t \in \mathbb{R}^d$ is the malicious message generated by Byzantine worker $k' \in \mathcal{B}$. Observe from (7) that any message from a worker, no matter regular or malicious, changes each element of x_0^{t+1} by at most λ. With this mechanism, RSA effectively controls the influence of Byzantine attacks.

Nevertheless, RSA is not communication-efficient. Transmitting $\text{sign}(x_0^t - x_k^t)$ from regular worker k to the master node involves d bits, while broadcasting x_0^t from the master node to all the workers involves d real numbers. Thus, the communication costs are remarkable for high-dimensional problems. To address this issue, we introduce the rand-l sparsification-based compression operator $C(\cdot)$. The master node broadcasts $C(x_0^t)$ to all the workers, which in practice means broadcasting the kept l elements of x_0^t, as well as a set ω_0^t that contains their indices. Define $P_{\omega_0^t}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ as the projection operator that keeps the input elements with indices in ω_0^t unaltered but sets the rest as zeros. It is obvious that $P_{\omega_0^t}(x_0^t) = C(x_0^t)$. Upon receiving $C(x_0^t)$, each regular worker $k \in \mathcal{R}$ updates its local vector as

$$x_k^{t+1} = x_k^t - \alpha \left[\nabla f(x_k^t, \xi_k^t) + \lambda P_{\omega_0^t}(\text{sign}(x_k^t - C(x_0^t))) \right]. \tag{8}$$

On the other hand, each regular worker $k \in \mathcal{R}$ sends to the master node $P_{\omega_0^t}(\text{sign}(C(x_0^t) - x_k^t))$, and each Byzantine worker $k' \in \mathcal{B}$

Algorithm 1 C-RSA at Time t

Master Node

Broadcast $C(x_0^t)$ to all workers;
Receive $P_{\omega_0^t}(\text{sign}(C(x_0^t) - x_k^t))$ from regular workers and
  $P_{\omega_0^t}(\text{sign}(C(x_0^t) - z_{k'}^t))$ from Byzantine workers;
Update x_0^{t+1} according to (9);

Regular Worker k

Receive $C(x_0^t)$ from master node;
Send $P_{\omega_0^t}(\text{sign}(C(x_0^t) - x_k^t))$ to master node;
Update x_k^{t+1} according to (8);

Byzantine Worker $k^{'}$

Receive $C(x_0^t)$ from master node;
Generate malicious message $z_{k'}^t$;
Send $P_{\omega_0^t}(\text{sign}(C(x_0^t) - z_{k'}^t))$ to master node;

sends $P_{\omega_0^t}(\text{sign}(C(x_0^t) - z_{k'}^t))$. Note that the Byzantine workers must send their messages following the compression rule; otherwise, their identities might be distinguished. The master node updates its local vector as

$$x_0^{t+1} = x_0^t - \alpha\Bigg[\nabla f_0(x_0^t) + \lambda\left(\sum_{k\in\mathcal{R}} P_{\omega_0^t}(\text{sign}(C(x_0^t) - x_k^t))\right) + \lambda\left(\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\text{sign}(C(x_0^t) - z_{k'}^t))\right)\Bigg]. \quad (9)$$

The proposed algorithm, compressed Byzantine-robust stochastic model aggregation (C-RSA), is outlined in Algorithm 1. At time t, the master node broadcasts l real numbers and one random seed, while each regular worker transmits l integers. As $l < d$, the communication cost is effectively reduced. It is also possible to introduce other compression rules, such as quantization, to C-RSA. We will investigate this valuable extension in our future work.

4. CONVERGENCE ANALYSIS

In this section, we analyze the convergence of the proposed C-RSA for distributed non-convex learning under Byzantine attacks. Note that now the cost function (4) is non-convex and non-smooth, which brings a major challenge in the analysis: We can use neither the optimality gap of function value or iterate for convex functions, nor the stationarity condition for non-convex smooth functions, as the measure of convergence. To handle this challenge, we rely on the weak convexity of cost function, and leverage the Moreau envelop and the proximal point projection to establish the convergence [27].

We leave the proofs to an extended version of this paper. Therein, as a byproduct, we also analyze the convergence of RSA, which is different to the analysis of [14] that assumes the strong convexity of cost function, and hence of independent interest.

4.1. Assumptions

For regular worker $k \in \mathcal{R}$, we define $f_k(x_k) = \mathbb{E}[f(x_k, \xi_k)]$ as its local cost function and $g_k^t = \nabla f(x_k^t, \xi_k^t)$ as its stochastic gradient at time t. We give the assumptions used in the analysis.

Assumption 1 (Lipschitz Continuous Gradients)**.** *The local cost functions $f_k(\tilde{x})$ of regular workers k and the regularization term $f_0(\tilde{x})$ have Lipschitz continuous gradients with a constant L. For any $\tilde{x}, \tilde{y} \in \mathbb{R}^d$, it holds that*

$$\|\nabla f_k(\tilde{x}) - \nabla f_k(\tilde{y})\| \le L\|\tilde{x} - \tilde{y}\|, \ \forall k \in \mathcal{R} \cup 0. \quad (10)$$

Assumption 1 is standard in analyzing optimization algorithms.

Assumption 2 (Weak Convexity)**.** *The local cost functions $f_k(\tilde{x})$ of regular workers k and the regularization term $f_0(\tilde{x})$ are ρ-weakly convex, which implies that $f_k(\tilde{x}) + \frac{\rho}{2}\|\tilde{x}\|^2$ and $f_0(\tilde{x}) + \frac{\rho}{2}\|\tilde{x}\|^2$ are convex. For any $\tilde{x}, \tilde{y} \in \mathbb{R}^d$, it holds that*

$$f_k(\tilde{y}) \ge f_k(\tilde{x}) + \langle \nabla f_k(\tilde{x}), \tilde{y} - \tilde{x}\rangle - \frac{\rho}{2}\|\tilde{y} - \tilde{x}\|^2, \quad \forall k \in \mathcal{R} \cup 0. \quad (11)$$

By Assumption 2, the functions are not arbitrarily non-convex, such that we can exploit their structures in the analysis. Fortunately, many typical cost functions in statistical learning and signal processing satisfy this assumption, such as nonlinear least squares, phase retrieval, robust principal component analysis, to name a few.

Assumption 3 (Bounded Gradients)**.** *For any regular worker $k \in \mathcal{R}$ and any $x_k^t \in \mathbb{R}^d$, the norm of its stochastic gradient is upper-bounded by*

$$\mathbb{E}\|g_k^t\|^2 \le M^2. \quad (12)$$

For any $x_0 \in \mathbb{R}^d$, the norm of gradient at master node is also upper-bounded by

$$\|\nabla f_0(x_0)\|^2 \le M^2. \quad (13)$$

Assumption 3 is also common in machine learning. Particularly, gradient clipping is a standard operation to control the norm of gradients for deep learning, such that this assumption naturally holds.

4.2. Moreau Envelope and Proximal Point Projection

Consider a continuous weakly convex function $h(\tilde{x})$. For any point $\tilde{x} \in \mathbb{R}^d$ and any constant $\beta > 0$, define the Moreau envelope $h_\beta(\tilde{x})$ and the proximal point projection $\text{prox}_{\beta h}(\tilde{x})$ as

$$h_\beta(\tilde{x}) := \min_{\tilde{y}} h(\tilde{y}) + \frac{1}{2\beta}\|\tilde{y} - \tilde{x}\|^2, \quad (14)$$

$$\text{prox}_{\beta h}(\tilde{x}) := \arg\min_{\tilde{y}} h(\tilde{y}) + \frac{1}{2\beta}\|\tilde{y} - \tilde{x}\|^2, \quad (15)$$

respectively [27]. For any $\tilde{x} \in \mathbb{R}^d$, its proximal point $\hat{x} = \text{prox}_{\beta h}(\tilde{x})$ satisfies

$$\|\partial h(\hat{x})\| \le \|\nabla h_\beta(\tilde{x})\|, \quad (16)$$

$$\|\hat{x} - \tilde{x}\| = \beta\|\nabla h_\beta(\tilde{x})\|, \quad (17)$$

where $\partial h(\hat{x})$ is any subgradient of $h(\cdot)$ at $\hat{x}$. Thus, a small gradient norm $\|\nabla h_\beta(\tilde{x})\|$ implies two facts: (i) $\hat{x}$ is close to a stationary point of $h(\cdot)$ by (16); and (ii) $\tilde{x}$ is close to its proximal point $\hat{x}$ by (17). Combining the two facts, we know that $\tilde{x}$ is close to a stationary point of $h(\cdot)$. As a consequence, we can use $\|\nabla h_\beta(\tilde{x})\|^2$ to measure the convergence of $\tilde{x}$ to a stationary point of $h(\cdot)$.

4.3. Convergence Analysis of Non-convex C-RSA

In C-RSA, the convergence is influenced by compressing x_0^t. For the rand-l sparsification-based compression operator $C(\cdot)$, we have

$$\mathbb{E}P_{\omega_0^t}(\text{sign}(x_k^t - C(x_0^t))) = \gamma\text{sign}(x_k^t - x_0^t), \quad (18)$$

$$\mathbb{E}P_{\omega_0^t}(\text{sign}(C(x_0^t) - x_k^t)) = \gamma\text{sign}(x_0^t - x_k^t), \quad (19)$$

where we recall $\gamma = \frac{l}{d}$ the expectation is taken with respect to $C(\cdot)$. Note that the ensuing analysis can be extended to other random compression operators that satisfy similar conditions.

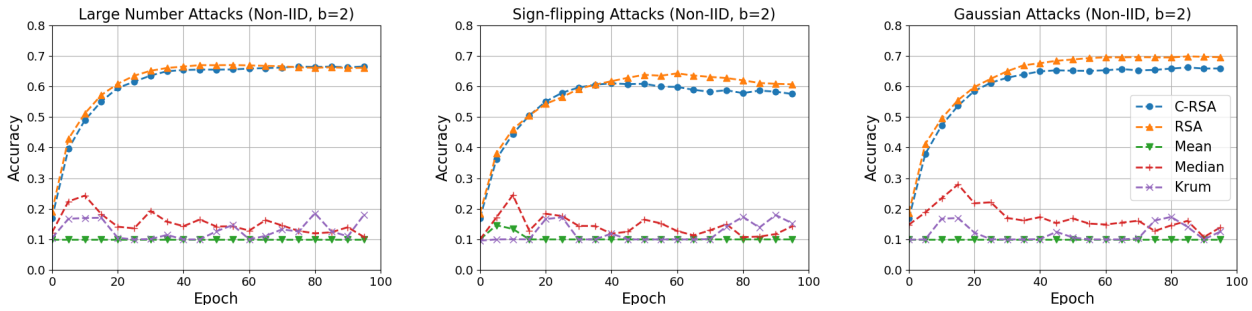

Fig. 1. Performance under different Byzantine attacks.

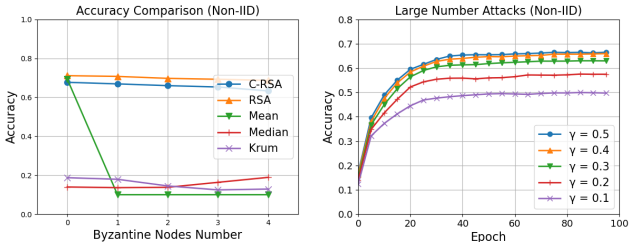

Fig. 2. Performance with different numbers of Byzantine workers b (L). Performance with different compression ratios γ of C-RSA (R).

Now we define

$$\tilde{h}_k(x_k) = f_k(x_k) + \gamma\lambda \left\|x_k - x_0\right\|_1, \tag{20}$$

$$\tilde{h}_0(x_0) = f_0(x_0) + \gamma\lambda \sum_{k\in\mathcal{R}} \left\|x_k - x_0\right\|_1. \tag{21}$$

Observe $\tilde{h}_k(x_k)$ and $\tilde{h}_0(x_0)$ are also ρ-weakly convex as $\|x_k - x_0\|_1$ is convex. We respectively define $\tilde{h}_{k,1/\bar{\rho}}(x_k)$ and $\tilde{h}_{0,1/\bar{\rho}}(x_0)$ as the Moreau envelops of $\tilde{h}_k(x_k)$ and $\tilde{h}_0(x_0)$, where $\bar{\rho} > 0$ is a constant. Let $\tilde{h}_{1/\bar{\rho}}(x) = \sum_{k\in\mathcal{R}} \tilde{h}_{k,1/\bar{\rho}}(x_k) + \tilde{h}_{0,1/\bar{\rho}}(x_0)$. The following theorem states that the iterate of C-RSA converges to a neighborhood of a stationary point of $\tilde{h}_{1/\bar{\rho}}(\cdot)$, and hence that of $\tilde{h}(\cdot)$.

Theorem 1 (Convergence of Non-convex C-RSA)**.** *Suppose that Assumptions 1, 2, 3 hold. The step size is constant and set to $\alpha = \alpha^t = \frac{1}{\sqrt{T}}$. For any constant $\bar{\rho} > \rho$, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla \tilde{h}_{1/\bar{\rho}}(x^t)\right\|^2 \leq \frac{\Delta_1'}{\sqrt{T}} + \Delta_2', \tag{22}$$

where Δ_1', Δ_2' are certain constants and $\Delta_2' = O(\lambda^2 b^2 \gamma)$.

Without the Byzantine workers, $\mathbb{E}\|\nabla \tilde{h}_{1/\bar{\rho}}(x^t)\|^2$ asymptotically converge to zero if $T \to \infty$. However, the malicious messages transmitted by the Byzantine workers lead the iterate to converge to a neighborhood of the stationary point of $\tilde{h}_{1/\bar{\rho}}(\cdot)$. The learning error is proportional to the squared number of Byzantine workers b^2, as well as the squared penalty parameter λ^2. A small λ means a small learning error in terms of $\mathbb{E}\|\nabla \tilde{h}_{1/\bar{\rho}}(x^t)\|^2$. But in the meantime, it means weak ℓ_1-norm penalties such that the consensus constraints $x_0 = x_k$ for all $k \in \mathcal{R}$ tend to be violated. The learning error is also proportional to the compression ratio γ. The proofs indicate that the impact of Byzantine attacks is on each and every element; updating less elements of x_0^t means that less Byzantine attacks are effective per time, although it may slow down the convergence.

5. NUMERICAL EXPERIMENTS

To evaluate the Byzantine-robustness and communication efficiency of the proposed C-RSA in distributed non-convex learning, we train a convolutional neural network (CNN) on the CIFAR10 dataset. CIFAR10 is a classification dataset with 10 categories. It has 50,000 training samples and 10,000 testing samples, where each sample is a 32×32 image. In the training stage, we set the batch size as 10. The CNN consists of 3 fully connected layers and 2 convolutional layers, with 368,052 parameters in total. The loss function is cross-entropy, and the regularization term is $f_0(\tilde{x}) := \frac{\mu}{2}||\tilde{x}||^2$, where $\mu = 0.01$.

We launch a distributed network with one master node and 10 workers. For the non-IID case, half of each label's training samples are allocated to a corresponding worker, while the rest half are randomly and evenly allocated to all the workers [26]. The number of Byzantine workers is set as $b = 2$ by default.

We consider three commonly used Byzantine attacks:

Large-number attacks. The Byzantine workers multiply each element of the true messages by a large number $10,000$.

Sign-flipping attacks. The Byzantine workers change each element of the true messages to its negative.

Gaussian attacks. The Byzantine workers change each element of the true messages following the standard normal distribution. Note that the elements of the true messages are generally in the order of $10^{-4} \sim 10^{-2}$ for these numerical experiments.

There are four baseline algorithms: SGD with mean aggregation that is unable to defend against Byzantine attacks, SGD with median aggregation [11], SGD with Krum aggregation [12], and RSA [14]. Note that Krum needs to exactly know the number of Byzantine workers. For C-RSA, by default we set the compression ratio as $\gamma = 0.5$, the step size as $\alpha = 0.001$, and the regularization parameter as $\lambda = 0.001$. All the parameters are hand-tuned to the best. Fig. 1 demonstrates that C-RSA has similar performance as RSA for all the attacks. The accuracies become slightly lower, but the communication costs are significantly reduced. In contrast, the other three algorithms fail for this distributed non-convex non-IID learning task. Fig. 2 shows the performance under the Gaussian attacks, with different numbers of Byzantine workers b. Here the accuracy refers to the average over the last 10 epochs. Although the performance degrades when b increases, C-RSA is still close to RSA. Fig. 2 also shows the performance of C-RSA under the Gaussian attacks with different compression ratios. The testing accuracy is higher for larger γ, which coincides with the intuition.

6. REFERENCES

[1] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," arXiv:1610.05492, 2016.

[2] 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.

[3] P. Kairouz and H. B. McMahan, "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, pp. 1–210, 2021.

[4] L. Zhou, K. H. Yeh, G. Hancke, Z. Liu, and C. Su, "Security and privacy for the industrial Internet of Things: An overview of approaches to safeguarding endpoints," IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 76–87, 2018.

[5] S. U. Stich, "Local SGD converges fast and communicates little," ICLR, 2019.

[6] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," NeurIPS, 2018.

[7] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, "Qsparse-Local-SGD: Distributed SGD with quantization, sparsification, and local computations," IEEE Journal on Selected Areas in Information Theory, vol. 1, no. 1, pp. 217–226, 2020.

[8] S. U. Stich, J. B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," NeurIPS, 2020.

[9] Z. Yang, A. Gang, and W. U. Bajwa, "Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the Byzantine threat model," IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 146–159, 2020.

[10] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," SIGMETRICS, 2017.

[11] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," arxiv: 1803.01498, 2018.

[12] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," NerIPS, 2017.

[13] X. Cao and L. Lai, "Distributed gradient descent algorithm robust to an arbitrary number of Byzantine attackers," IEEE Transactions on Signal Processing, vol. 67, no. 22, pp. 5850–5864, 2019.

[14] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q, Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," AAAI, 2019.

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

[16] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," MLSys, 2020.

[17] S. Bulusu, P. Khanduri, P. Sharma, and P. K. Varshney, "On distributed stochastic gradient descent for nonconvex functions in the presence of Byzantines," ICASSP, 2020.

[18] C. Xie, O. Koyejo, and I. Gupta, "Zeno++: Robust fully asynchronous SGD," ICML, 2020.

[19] S. P. Karimireddy, L. He, M. Jaggi, "Learning from history for Byzantine robust optimization," ICML, 2021.

[20] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Defending against saddle point attack in Byzantine-robust distributed learning," ICML, 2019.

[21] Z. Allen, F. Ebrahimian, J. Li, and D. Alistarh, "Byzantine-resilient non-convex stochastic gradient descent," arXiv:2012.14368, 2020.

[22] J. Bernstein, J. Zhao, K. Azizzadenesheli, and Anandkumar, "SignSGD with majority vote is communication efficient and fault tolerant," ICLR, 2019.

[23] A. Ghosh, R. K. Maityy, S. Kadhe, A. Mazumdary, and K. Ramachandran, "Communication efficient and Byzantine tolerant distributed learning," ISIT, 2020.

[24] H. Zhu, and Q. Ling, "BROADCAST: Reducing both stochastic and compression noise to robustify communication-efficient federated learning," arXiv:2104.06685, 2021.

[25] Y. Dong, G. B. Giannakis, T. Chen, J. Cheng, M. Hossain, and V. Leung, "Communication-efficient robust federated learning over heterogeneous datasets," arXiv:2006.09992, 2020.

[26] X. Cao, M. Fang, J. Liu, and N. Gong, "FLTrust: Byzantine-robust federated learning via trust bootstrapping," arXiv:2012.13995, 2020.

[27] D. Damek and D. Dmitriy, "Stochastic model-based minimization of weakly convex functions," SIAM Journal on Optimization, vol. 29, no. 1, pp. 207–239, 2019.