VARIANCE REDUCTION-BOOSTED BYZANTINE ROBUSTNESS IN DECENTRALIZED STOCHASTIC OPTIMIZATION

Jie Peng[*] *Weiyu Li*[†] *Qing Ling*[*]

[*]Sun Yat-Sen University [†]Harvard University

ABSTRACT

We consider the Byzantine-robust decentralized stochastic optimization problem, where every agent periodically communicates with its neighbors to exchange the local models, and then updates its own local model by stochastic gradient descent. However, an unknown number of the agents are Byzantine, and perform adversarially during the optimization process. Few works have considered this challenging scenario, and an existing method termed DECEMBER is unable to simultaneously achieve linear convergence speed and small learning error due to the stochastic noise. To eliminate the negative effect of the stochastic noise, we introduce two variance reduction methods, stochastic average gradient algorithm (SAGA) and loopless stochastic variance-reduced gradient (LSVRG), to Byzantine-robust decentralized stochastic optimization. The two resulting methods, DECEMBER-SAGA and DECEMBER-LSVRG, enjoy both linear convergence speeds and small learning errors. Numerical experiments demonstrate their effectiveness.

Index Terms— Decentralized stochastic optimization, Byzantine robustness, variance reduction

1. INTRODUCTION

In recent years, *decentralized stochastic optimization* has attracted immense research interest, and found applications in signal processing [1], machine learning [2], systems control [3], communications and networking [4], etc. Consider a decentralized network with N agents. The finite-sum form of the decentralized stochastic optimization problem is given by

$$\min_{\tilde{x}\in\mathbb{R}^p} F(\tilde{x}) := \frac{1}{N}\sum_{w=1}^{N} F_w(\tilde{x}), \tag{1}$$

where $\tilde{x}$ is the model parameter to optimize, $F(\tilde{x})$ is the global cost, and $F_w(\tilde{x}) := \frac{1}{J}\sum_{j=1}^{J} F_{w,j}(\tilde{x})$ is the local cost of agent w with $F_{w,j}(\tilde{x})$ being the cost associated with sample j at agent w. Without loss of generality, we assume that every agent has J samples. At every iteration, every agent communicates with its neighbors to obtain neighboring models, aggregates them, and updates its own model with one or a mini-batch of samples [5,6]. This is different to *decentralized deterministic optimization* in which all the samples are used in the computation step [7, 8], and hence fits for the scenarios where the number of samples on every agent is large.

However, during the operation of the decentralized system, some of the agents might malfunction or even behave adversarially due to data corruptions, network failures and malicious attacks. One popular way to describe such uncertainties is to use the Byzantine attacks model [9]. The goal of this paper is to develop provably Byzantine-robust decentralized stochastic optimization methods.

Existing *decentralized deterministic optimization* methods, such as the classical decentralized gradient descent (DGD) method [7], generally use weighted mean to aggregate neighboring models and are hence vulnerable to Byzantine attacks. The remedy is to replace weighted mean aggregation by robust aggregation rules. The works of [10–12] study Byzantine-robust decentralized deterministic optimization with scalar models, using trimmed mean to replace weighted mean. When the models are vectors, ByRDiE [13] performs coordinate gradient descent, and applies trimmed mean [10] to filter out potential malicious elements on every dimension. A centerpoint-based method is proposed in [14] to guarantee the aggregated model always lies in the convex hull of the regular neighbors' models. BRIDGE [15] extends ByRDiE [13], allowing all coordinates to be updated at one time with gradient descent. In [16], every agent aggregates the signs of the differences between its own model and the neighbors' such that the influence of Byzantine attacks is limited.

Although there are an increasing number of methods, including decentralized parallel stochastic gradient descent (DPSGD) [2,7], to solve the *decentralized stochastic optimization* problem, few of them have considered Byzantine robustness. Motivated by [16], [17] introduces a total variation (TV)-norm penalized formulation to force the local models to be close, but allow possible outliers. Applying the stochastic subgradient method to solve this TV-norm penalized formulation yields a Byzantine-robust method with sublinear convergence. Byzantine-robust multi-agent reinforcement learning is studied in [18], but the analysis of decentralized policy gradient is different to that of decentralized stochastic gradient.

On the other hand, a popular topic in the machine learning society is Byzantine-robust *distributed stochastic optimization*, where a central server maintains a global model and aggregates messages from the distributed agents. Most of the algorithms modify the distributed stochastic gradient descent (SGD) method by robustly aggregating the stochastic gradients sent from the agents [19,20]. Such robust aggregation rules include coordinate-wise median, geometric median, trimmed mean, Krum, etc. Byzantine-robust stochastic second-order methods have also been developed in [21]. To enhance Byzantine robustness, [22] assumes that the central server has extra clean data, and uses the clean data to distinguish malicious messages from the true stochastic gradients. However, it is difficult to generalize the above-mentioned works to the decentralized scenario, since there is no central server to collect and aggregate messages.

In stochastic optimization, the noise from calculating stochastic gradients plays a critical role and affects convergence. To reduce the negative effect of stochastic noise, variance reduction methods have been employed in *decentralized stochastic optimization* [23–25]. In Byzantine-robust *distributed stochastic optimization*,

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.

stochastic noise also significantly influences Byzantine robustness. With large stochastic noise, the regular stochastic gradients are statistically very different, which is conducive for the Byzantine agents to hide themselves and perform attacks [26]. In light of this observation, [26] proposes to use the stochastic average gradient algorithm (SAGA) to eliminate the impact of inner variation, namely, the stochastic noise caused by the heterogeneity of every regular agent's local samples. The works of [27,28] show that momentum acceleration can effectively reduce the impact of stochastic noise and enhance Byzantine robustness. Aiming at non-convex problems, [29] combines the stochastic variance-reduced gradient (SVRG) method and robust aggregation, and empirically demonstrates the performance. However, none of the existing works considers variance reduction in Byzantine-robust *decentralized stochastic optimization*.

Our contributions. Although the proposed algorithm in [17], which we call DECEMBER thereafter, is proven effective in handling Byzantine attacks for decentralized stochastic optimization, it has an unsatisfactory trade-off between convergence speed and learning error. With a diminishing step size, the learning error is small but the convergence speed is slow. Otherwise, with a constant step size, the convergence speed is linear but the learning error is large. The underlying reason is the existence of the stochastic noise. Therefore, we introduce two variance reduction methods, SAGA and loopless (L-) SVRG, to Byzantine-robust decentralized stochastic optimization to eliminate the negative effect of the stochastic noise. The two methods, DECEMBER-SAGA and DECEMBER-LSVRG, enjoy both linear convergence speeds and stochastic noise-independent small learning errors. The analysis shares similarities with that in [17], but is quite different because of the introduction of the variance reduction techniques. The analysis is also more challenging than those of the Byzantine-robust distributed variance-reduced stochastic optimization methods [26], due to the absence of the central server and the existence of the non-smooth TV-norm penalty term. We also conduct extensive numerical experiments to demonstrate the effectiveness of the proposed methods.

2. PROBLEM FORMULATION

Consider a decentralized undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of N agents where $\mathcal{V} := \{1, \cdots, N\}$ represents the set of agents and $\mathcal{E}$ represents the set of edges. If $e := (w, v) \in \mathcal{E}$, then agents w and v are neighbors and can communicate with each other. We consider the Byzantine attacks model where a number of agents might be Byzantine and behave arbitrarily. The number and identities of the Byzantine agents are unknown. Denote $\mathcal{R}$ as the set of regular agents with $|\mathcal{R}| = R$ and $\mathcal{B}$ as the set of Byzantine agents with $|\mathcal{B}| = B$. For agent w, denote $\mathcal{R}_w$ and $\mathcal{B}_w$ as the sets of regular neighbors and Byzantine neighbors, respectively. Denote $\mathcal{E}_R$ as the set of edges that are not attached to any Byzantine agents. The following assumption is necessary to enable collaboration between the regular agents [17].

Assumption 1. *(Network Connectivity) The network composed of all regular agents, denoted as $(\mathcal{R}, \mathcal{E}_R)$, is connected.*

Since the Byzantine agents can always arbitrarily manipulate their local samples, it is impossible to solve (1). A reasonable goal is to minimize the average of the regular agents' local costs, as

$$\tilde{x}^* = \arg\min_{\tilde{x}\in\mathbb{R}^p} \frac{1}{R} \sum_{w\in\mathcal{R}} F_w(\tilde{x}). \tag{2}$$

The main difficulty in solving (2) is that the Byzantine agents will send arbitrary, malicious messages to their neighbors, such that the optimization process is unstable. Let each regular agent $w \in \mathcal{R}$ have a local model $x_w \in \mathbb{R}^p$ and use $x \in \mathbb{R}^{pR}$ to stack all local models of the regular agents. With the connected regular network $(\mathcal{R}, \mathcal{E}_R)$, we can rewrite (2) to a consensus-constrained form, as

$$\begin{aligned} \min_x \ & \frac{1}{R} \sum_{w\in\mathcal{R}} F_w(x_w), \\ s.t.\ & x_w = x_v,\ \forall w \in \mathcal{R},\ \forall v \in \mathcal{R}_w. \end{aligned} \tag{3}$$

Given an optimal solution $\tilde{x}^*$ to (2), a longer vector that stacks R vectors $\tilde{x}^*$ is also an optimal solution to (3).

Prior work of DECEMBER. Following the line of [16,17], we penalize the consensus constraints in (3) by a TV-norm term, as

$$x^* = \arg\min_x \frac{1}{R} \sum_{w\in\mathcal{R}} \left(F_w(x_w) + \frac{\lambda}{2} \sum_{v\in\mathcal{R}_w} \|x_w - x_v\|_1 \right), \tag{4}$$

where $\lambda > 0$ is the penalty parameter. The TV-norm penalty term forces the local models to be close, but also allows possible outliers. Note that [16] considers *decentralized deterministic optimization*, while [17] considers *decentralized stochastic optimization* in the expectation minimization form, not in the finite-sum minimization form. Nevertheless, the DECEMBER algorithm in [17] can still be applied to solve (4). When the Byzantine agents are absent, applying the stochastic subgradient method to solve (4) yields the update $x_w^{k+1} = x_w^k - \alpha^k (F'_{w,i_w^k}(x_w^k) + \lambda \sum_{v\in\mathcal{R}_w} sign(x_w^k - x_v^k))$ for any regular agent $w \in \mathcal{R}$ at time k. Here $\alpha^k > 0$ is the step size, and i_w^k is the local sample selected by regular agent w at time k. To run the update, every regular agent $w \in \mathcal{R}$ needs to collect its neighboring models. However, when the Byzantine agents appear, they may send arbitrary, malicious messages for the sake of disturbing the optimization process. Denote $z_v^k \in \mathbb{R}^p$ as an arbitrary model that Byzantine agent $v \in \mathcal{B}$ sends to all neighbors. In fact, it can send different arbitrary models to different neighbors, but for notational convenience we use the same z_v^k, which does not affect algorithm development and theoretical analysis. For regular agent $w \in \mathcal{R}$, the update rule of DECEMBER is changed to

$$\begin{aligned} x_w^{k+1} = x_w^k - \alpha^k \Bigg(& F'_{w,i_w^k}(x_w^k) + \lambda \sum_{v\in\mathcal{R}_w} sign(x_w^k - x_v^k) \\ & + \lambda \sum_{v\in\mathcal{B}_w} sign(x_w^k - z_v^k) \Bigg). \end{aligned} \tag{5}$$

3. OUR PROPOSED METHODS

In DECEMBER-SAGA, every regular agent $w \in \mathcal{R}$ stores a stochastic gradient table where every item corresponds to a local sample, and updates an item in the gradient table when the corresponding sample is selected to calculate the stochastic gradient. Specifically, let $\phi_{w,j}^{k+1} = \phi_{w,j}^k$ if the selected sample index $i_w^k = j$ at time k and $\phi_{w,j}^{k+1} = x_w^k$ otherwise, where $\phi_{w,j}^k$ represents the most recent local model used in evaluating $F'_{w,j}$ on agent w, prior to time k. Therefore, $F'_{w,j}(\phi_{w,j}^k)$ refers to the most recent stochastic gradient for sample j on agent w, prior to time k.

Rather than using a stochastic gradient $F'_{w,i_w^k}(x_w^k)$ to update its local model, every regular agent $w \in \mathcal{R}$ calculates a corrected stochastic gradient g_w^k, which is defined as

$$g_w^k = F'_{w,i_w^k}(x_w^k) - F'_{w,i_w^k}(\phi_{w,i_w^k}^k) + \frac{1}{J} \sum_{j=1}^{J} F'_{w,j}(\phi_{w,j}^k). \tag{6}$$

The corrected stochastic gradient g_w^k modifies the stochastic gradient $F'_{w,i_w^k}(x_w^k)$ by subtracting the stored one $F'_{w,i_w^k}(\phi_{w,i_w^k}^k)$ that also corresponds to sample index i_w^k, and adding the average $\frac{1}{J}\sum_{j=1}^J F'_{w,j}(\phi_{w,j}^k)$ of all stored stochastic gradients. Such a variance reduction operation works on the local samples and is irrelevant with the TV-norm penalty. Thus, we replace the original stochastic gradient $F'_{w,i_w^k}(x_w^k)$ in (5) by g_w^k, and write the update rule of every regular agent $w \in \mathcal{R}$ in DECEMBER-SAGA as

$$x_w^{k+1} = x_w^k - \alpha\Bigg(g_w^k + \lambda \sum_{v\in\mathcal{R}_w} sign(x_w^k - x_v^k) \qquad (7)$$
$$+ \lambda \sum_{v\in\mathcal{B}_w} sign(x_w^k - z_v^k)\Bigg).$$

It is worth noting that we use a constant step size $\alpha > 0$ here, taking advantage of the variance reduction property of SAGA [30]. In section 4, we will show that DECEMBER-SAGA can linearly converge with a constant step size.

Inheriting the property of SAGA, DECEMBER-SAGA requires every regular agent to maintain a stochastic gradient table, which brings extra storage cost. For storage-limited applications, we replace SAGA by loopless SVRG (a.k.a. LSVRG) [31], which incurs no extra storage cost, although doubles the computation cost.

Let $y_w^{k+1} = x_w^k$ with probability $\frac{1}{J}$ and $y_w^{k+1} = y_w^k$ with probability $1 - \frac{1}{J}$, where y_w^{k+1} is a reference point at which the local full gradient F'_w is calculated. In DECEMBER-LSVRG, every regular agent $w \in \mathcal{R}$ also updates with (7), where the corrected stochastic gradient is defined as

$$g_w^k = F'_{w,i_w^k}(x_w^k) - F'_{w,i_w^k}(y_w^k) + \frac{1}{J}\sum_{j=1}^J F'_{w,j}(y_w^k). \qquad (8)$$

Observe that at every time, with probability $\frac{1}{J}$, y_w^{k+1} is updated to x_w^k, meaning that the local full gradient must be calculated. By doubling the computation cost, DECEMBER-LSVRG does not need to maintain a stochastic gradient table as in DECEMBER-SAGA.

4. THEORETICAL ANALYSIS

We make the following assumptions that are common in the analysis of decentralized stochastic optimization [2, 17] and variance reduction [30, 31]. Assumption 4 bounds the variance of the local stochastic gradients. It is worth noting that DECEMBER-SAGA and DECEMBER-LSVRG only need Assumptions 2 and 3, but do not need Assumption 4, which serves for the theoretical analysis of DECEMBER. The reason is that the adopted variance reduction methods are able to eliminate the effect of stochastic noise.

Assumption 2. ***(Strong Convexity)*** *For any model $\tilde{x} \in \mathbb{R}^p$ and every regular agent $w \in \mathcal{R}$, the local cost $F_w(\tilde{x})$ is strongly convex with constant μ_w.*

Assumption 3. ***(Lipschitz Continuous Gradients)*** *At time k, every regular agent $w \in \mathcal{R}$ randomly selects a sample with index $i_w^k \in \{1, \cdots, J\}$. For any model $\tilde{x} \in \mathbb{R}^p$, the local sample cost $F_{w,i_w^k}(\tilde{x})$ has Lipschitz continuous gradients with constant L_w.*

Assumption 4. ***(Bounded Variance)*** *At time k, every regular agent $w \in \mathcal{R}$ randomly selects a sample with random variable $i_w^k \in \{1, \cdots, J\}$. For any model $\tilde{x} \in \mathbb{R}^p$, the variance of $F'_{w,i_w^k}(\tilde{x})$ is upper bounded by δ_w^2.*

As the network topology plays a critical role in the anlaysis, we define $A \in \mathbb{R}^{R\times|\mathcal{E}_R|}$ as the oriented node-edge incidence matrix of $(\mathcal{R}, \mathcal{E}_R)$. For an edge $e = (w, v) \in \mathcal{E}_R$, the (w, e)-th entry of A is 1 and the (v, e)-th entry of A is -1. We review the following lemma.

Lemma 1. *([17], Theorem 1) Suppose that Assumptions 1 and 2 hold true. If $\lambda \geq \frac{\sqrt{R}}{\tilde{\sigma}_{\min}(A)} \max_{w\in\mathcal{R}} \|F'_w(\tilde{x}^*)\|$ where $\tilde{\sigma}_{\min}(A)$ is the minimum nonzero singular value of A, then for the optimal solution x^* of (4) and the optimal solution $\tilde{x}^*$ of (2), we have that x^* stacks R vectors $\tilde{x}^*$ in the form of $[\cdots; \tilde{x}^*; \cdots]$.*

As DECEMBER-SAGA, DECEMBER-LSVRG and DECEMBER all solve the TV norm-penalized problem (4), they share Lemma 1, which shows that (4) is equivalent to the original problem (2) when the penalty parameter λ is sufficiently large. Below, we only consider the regime of large λ and focus on the convergence to (4). The next theorem demonstrates that DECEMBER-SAGA and DECEMBER-LSVRG can achieve linear convergence, and the learning error is independent with the stochastic noise.

Theorem 1. *Suppose Assumptions 1, 2, 3 hold true. Denote $\eta = \min_{w\in\mathcal{R}}\{\frac{2\mu_w L_w}{3(\mu_w+L_w)}\}$ and $\bar{L} := \max_{w\in\mathcal{R}} L_w$. Set the step size of DECEMBER-SAGA and DECEMBER-LSVRG as $\alpha \leq \frac{\eta}{12\bar{L}^2 J}$. we have $\mathbb{E}[V^k] \leq (1-\eta\alpha)^k V^0 + \Delta_1$, where the Lyapunov function $V^k := \|x^k - x^*\|^2 + 8J\alpha^2\bar{L}^2 S^k$ and the learning error is*

$$\Delta_1 := \frac{\alpha}{\eta}\sum_{w\in\mathcal{R}}(32\lambda^2|\mathcal{R}_w|^2 p + 4\lambda^2|\mathcal{B}_w|^2 p) + \frac{1}{\eta^2}\sum_{w\in\mathcal{R}}\lambda^2|\mathcal{B}_w|^2 p. \qquad (9)$$

For DECEMBER-LSVRG, $S^k := \sum_{w\in\mathcal{R}}\|x_w^* - y_w^k\|^2$. For DECEMBER-SAGA, $S^k := \sum_{w\in\mathcal{R}}\frac{1}{J}\sum_{j=1}^J\|x_w^* - \phi_{w,j}^k\|^2$.

Theorem 1 asserts that DECEMBER-SAGA and DECEMBER-LSVRG can linearly converge to a neighborhood of the optimal solution x^* of (4) and the size of the neighborhood is determined by the penalty parameter λ, the summation of the squared numbers of Byzantine neighbors $\sum_{w\in\mathcal{R}}|\mathcal{B}_w|^2$, and the problem dimension p.

To see the advantage of variance reduction to Byzantine-robust decentralized stochastic optimization, we compare the theoretical results with that of DECEMBER.

Lemma 2. *[17, Theorem 2] Suppose Assumptions 1, 2, 3, 4 hold true. Denote $\eta = \min_{w\in\mathcal{R}}\{\frac{2\mu_w L_w}{3(\mu_w+L_w)}\}$. Set the step size of DECEMBER as $\alpha \leq \min_{w\in\mathcal{R}}\{\frac{1}{4(\mu_w+L_w)}\}$, we have $\mathbb{E}\|x^k - x^*\|^2 \leq (1-\eta\alpha)^k\|x^0 - x^*\|^2 + \Delta_2$, where the learning error is*

$$\Delta_2 := \frac{\alpha}{\eta}\sum_{w\in\mathcal{R}}(32\lambda^2|\mathcal{R}_w|^2 p + 4\lambda^2|\mathcal{B}_w|^2 p + 2\delta_w^2) \qquad (10)$$
$$+ \frac{1}{\eta^2}\sum_{w\in\mathcal{R}}\lambda^2|\mathcal{B}_w|^2 p.$$

Lemma 2 shows that with a constant step size, DECEMBER can also achieve a linear convergence rate, and the learning error is relative to the variance δ_w^2. Although the Lyapunov functions in Theorem 1 and Lemma 2 are different and the learning errors are not directly comparable, we can observe that the effect of stochastic noise is fully eliminated in DECEMBER-SAGA and DECEMBER-LSVRG. Note that from Theorem 2 in [17], with a diminishing step size, DECEMBER is also able to eliminate the stochastic noise from its learning error, but the convergence rate is sublinear. To the best of our knowledge, our proposed methods are the first two Byzantine-robust decentralized stochastic optimization methods with stochastic noise-independent learning errors and linear convergence rates.

5. NUMERICAL EXPERIMENTS

We consider an undirected Erdos-Renyi graph consisting of $N = 100$ agents, where every edge $e = (w, v)$ for all $w, v \in \mathcal{V}$ is connected with probability $q \in [0, 1]$. We set $q = 0.5$ for all experiments. All experiments are conducted on the MNIST dataset using softmax regression. The global cost is defined as

$$F(\tilde{x}) = -\frac{1}{J}\sum_{j=1}^{J}\sum_{m=0}^{M-1}\left(I(b^{(j)} = m)\ln\left(\frac{\exp((\tilde{x})_m^T a^{(j)})}{\sum_{l=0}^{M-1}\exp((\tilde{x})_l^T a^{(j)})}\right)\right).$$

Here J and M are the numbers of samples and categories, respectively. Sample j is represented by $(a^{(j)}, b^{(j)})$, where $a^{(j)} \in \mathbb{R}^{p/M}$ is the data and $b^{(j)} \in \mathbb{R}$ is the target. $I(b^{(j)} = m)$ is the indicator function; $I(b^{(j)} = m) = 1$ if $b^{(j)} = m$, and $I(b^{(j)} = m) = 0$ otherwise. The model variable is $\tilde{x} \in \mathbb{R}^p$ and $(\tilde{x})_m \in \mathbb{R}^{p/M}$ is the m-th block of $\tilde{x}$. The MNIST dataset contains $M = 10$ handwritten digits from 0 to 9, with 60,000 training images and 10,000 testing images whose dimensions are $\frac{p}{M} = 784$. We consider i.i.d. (independent and identically distribution) and non-i.i.d. data distributions across the agents. In the i.i.d. case, all N agents randomly and evenly partition the training samples. In the non-i.i.d. case, every 10 agents randomly and evenly partition the training samples of one digit.

We compare DECEMBER-SAGA and DECEMBER-LSVRG with several benchmarks, including DPSGD [2, 7], the stochastic version of ByRDiE [13] (called ByRDiE-S), the stochastic version of BRIDGE [15] (called BRIDGE-S), and DECEMBER [17]. For DPSGD, we choose the Metropolis weight [32] to ensure that the mixing matrix is doubly stochastic. For ByRDiE-S, we set the number of inner iterations is 1, and for fair comparison, refer one time step as all dimensions being updated once. All step sizes of the benchmark methods are hand-tuned to the best. For DECEMBER, the step size is set to $\alpha^k = O(\frac{1}{\sqrt{k}})$ as in [17] unless specified.

When the Byzantine agents exist, $B = 20$. Consider the following Byzantine attacks. The sample-duplicating attacks are applied to the non-i.i.d. case. The others are applied to the i.i.d. case.

Gaussian attacks. Every Byzantine agent $v \in \mathcal{B}$ sends malicious model x_v^k following multi-variate Gaussian distribution $\mathcal{N}(0, 100^2)$ to neighbors.

Sign-flipping attacks. Every Byzantine agent $v \in \mathcal{B}$ calculates its true model x_v^k, multiplies it with a negative constant c, and sends the malicious model $z_v^k = cx_v^k$ to neighbors. Here we set $c = -4$.

Sample-duplicating attacks. The Byzantine agents own two classes of samples. They collude to select a specific regular agent, copy the model of the selected agent, and send it to the neighbors. This amounts to that the Byzantine agents duplicate the samples of the selected regular agent.

When computing the accuracy over the testing samples, we randomly choose one regular model and compute the testing accuracy on this model in Fig. 1. To see the consensus error between the regular agents, we also show the variance of the regular models. Without Byzantine attacks, all methods perform well and we omit the figures. Under Gaussian attacks, DPSGD fails since it is unable to defend against Byzantine attacks. All other Byzantine-robust methods perform well. Under sign-flipping attacks, the results are similar to those of the Gaussian attacks. Under sample-duplicating attacks, it is worth noting that since the Byzantine agents own the samples of two classes, the best possible accuracy of the regular agents' model is 0.8. DECEMBER-SAGA, DECEMBER-LSVRG, and DECEMBER all perform well and achieve near-optimal accuracy. The TV-norm penalty forces the regular models to reach consensus, and thus these methods are insensitive to non-i.i.d. data distribution. In contrast, with majority voting, ByRDiE-S and BRIDGE-S fail.

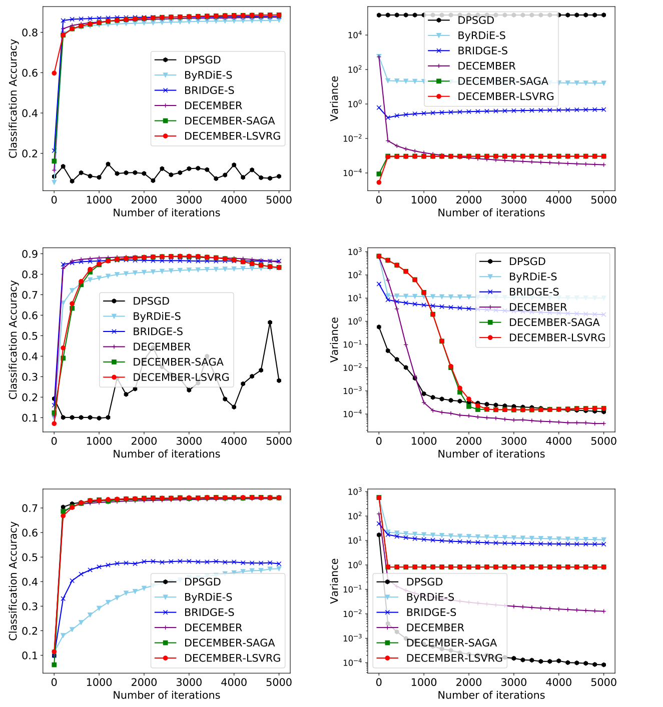
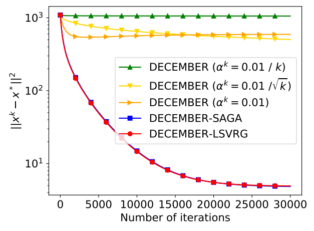
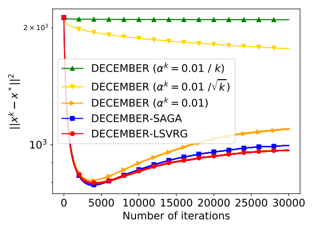


Fig. 1. Classification accuracy and variance of regular agents' local models. TOP: Gaussian attacks; MIDDLE: sign-flipping attacks; BOTTOM: sample-duplicating attacks. Penalty parameter $\lambda = 0.005, 0.0022, 0.02$, respectively.



(a) Sign-flipping attacks



(b) Sample-duplicating attacks

Fig. 2. Optimality gap $\|x^k - x^*\|^2$. Penalty parameter $\lambda = 0.0022, 0.02$, respectively.

We further compare DECEMBER-SAGA and DECEMBER-LSVRG with DECEMBER in terms of the optimality gap. Since the step sizes of DECEMBER are different in theory and in practice, we choose three step size rules for DECEMBER, $\alpha^k = 0.01/k$, $\alpha^k = 0.01/\sqrt{k}$ and $\alpha^k = 0.01$. The attacks are sign-flipping and sample-duplicating. The results are depicted in Fig. 2. Observe that DECEMBER-LSVRG and DECEMBER-SAGA, with the help of variance reduction, both achieve faster convergence speeds than DECEMBER with the diminishing step sizes, and achieve smaller learning errors than DECEMBER with a constant step size that suffers from the stochastic noise. These experimental results corroborate the theoretical analysis in Section 4 and show the superior performance of DECEMBER-SAGA and DECEMBER-LSVRG.

6. REFERENCES

[1] Tsung-Hui Chang, Mingyi Hong, Hoi-To Wai, Xinwei Zhang, and Songtao Lu, "Distributed learning in the nonconvex world: From batch data to streaming and beyond," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 26–38, 2020.

[2] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 5330–5340.

[3] Yubo Du and Keyou You, "Asynchronous stochastic gradient descent over decentralized datasets," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 3, pp. 1212–1224, 2021.

[4] Georgios B. Giannakis, Qing Ling, Gonzalo Mateos, Ioannis D. Schizas, and Hao Zhu, "Decentralized learning for wireless communications and networking," *Splitting Methods in Communication and Imaging, Science and Engineering (R. Glowinsky, S. Osher, and W. Yin, eds.)*, 2016.

[5] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu, "d^2: Decentralized training over decentralized data," in *International Conference on Machine Learning*, 2018, pp. 4848–4856.

[6] Aryan Mokhtari and Alejandro Ribeiro, "Dsa: Decentralized double stochastic averaging gradient algorithm," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2165–2199, 2016.

[7] Angelia Nedic and Asuman Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[8] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[9] Leslie Lamport, Robert Shostak, and Marshall Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[10] Lili Su and Nitin H Vaidya, "Fault-tolerant multi-agent optimization: Optimal iterative distributed algorithms," in *ACM Symposium on Principles of Distributed Computing*, 2016, pp. 425–434.

[11] Shreyas Sundaram and Bahman Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2018.

[12] Lili Su and Nitin H Vaidya, "Byzantine-resilient multiagent optimization," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2227–2233, 2020.

[13] Zhixiong Yang and Waheed U Bajwa, "Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 4, pp. 611–627, 2019.

[14] Jiani Li, Waseem Abbas, Mudassir Shabbir, and Xenofon Koutsoukos, "Resilient distributed diffusion for multi-robot systems using centerpoint," *Robotics: Science and Systems*, 2020.

[15] Zhixiong Yang and Waheed U Bajwa, "Bridge: Byzantine-resilient decentralized gradient descent," *arXiv preprint arXiv:1908.08098*, 2019.

[16] Walid Ben-Ameur, Pascal Bianchi, and Jeremie Jakubowicz, "Robust distributed consensus using total variation," *IEEE Transactions on Automatic Control*, vol. 61, no. 6, pp. 1550–1564, 2015.

[17] Jie Peng, Weiyu Li, and Qing Ling, "Byzantine-robust decentralized stochastic optimization over static and time-varying networks," *Signal Processing*, vol. 183, pp. 108020, 2021.

[18] Zhaoxian Wu, Han Shen, Tianyi Chen, and Qing Ling, "Byzantine-resilient decentralized td learning with linear function approximation," *IEEE Transactions on Signal Processing*, 2021.

[19] Yudong Chen, Lili Su, and Jiaming Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

[20] Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al., "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 119–129.

[21] Xinyang Cao and Lifeng Lai, "Distributed approximate Newton's method robust to Byzantine attackers," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6011–6025, 2020.

[22] Xinyang Cao and Lifeng 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.

[23] Kun Yuan, Bicheng Ying, Jiageng Liu, and Ali H. Sayed, "Variance-reduced stochastic learning by networked agents under random reshuffling," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 351–366, 2019.

[24] Ran Xin, Usman A Khan, and Soummya Kar, "Variance-reduced decentralized stochastic optimization with accelerated convergence," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6255–6271, 2020.

[25] Ran Xin, Soummya Kar, and Usman A Khan, "Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 102–113, 2020.

[26] Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B Giannakis, "Federated variance-reduced stochastic gradient descent with robustness to Byzantine attacks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4583–4596, 2020.

[27] El-Mahdi El-Mhamdi, Rachid Guerraoui, and Sébastien Rouault, "Distributed momentum for Byzantine-resilient learning," *arXiv preprint arXiv:2003.00010*, 2020.

[28] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi, "Learning from history for Byzantine robust optimization," in *International Conference on Machine Learning*, 2021, pp. 5311–5319.

[29] Prashant Khanduri, Saikiran Bulusu, Pranay Sharma, and Pramod K Varshney, "Byzantine resilient non-convex svrg with distributed batch gradient computations," *arXiv preprint arXiv:1912.04531*, 2019.

[30] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien, "Saga: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in Neural Information Processing Systems*, 2014, pp. 1646–1654.

[31] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik, "Dont jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop," in *Algorithmic Learning Theory*, 2020, pp. 451–467.

[32] Lin Xiao, Stephen Boyd, and Sanjay Lall, "Distributed average consensus with time-varying metropolis weights," *Automatica*, vol. 1, 2006.