VALID: a Validated Algorithm for Learning in Decentralized Networks with Possible Adversarial Presence

Mayank Bakshi* Sara Ghasvarianjahromi† Yauhen Yakimenka† Allison Beemer‡ Oliver Kosut* Jörg Kliewer†

*Arizona State University †New Jersey Institute of Technology ‡University of Wisconsin-Eau Claire

2024 IEEE International Symposium on Information Theory (ISIT) | 979-8-3503-8284-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISIT57864.2024.10619330

***Abstract*—We introduce the paradigm of *validated decentralized learning* for undirected networks with heterogeneous data and possible adversarial infiltration. We require *(a)* convergence to a global empirical loss minimizer when adversaries are absent, and *(b)* either detection of adversarial presence or convergence to an admissible consensus model in their presence. This contrasts sharply with the traditional byzantine-robustness requirement of convergence to an admissible consensus irrespective of the adversarial configuration. To this end, we propose the VALID protocol which, to the best of our knowledge, is the first to achieve a validated learning guarantee. Moreover, VALID offers an $O(1/T)$ convergence rate (under pertinent regularity assumptions), and computational and communication complexities comparable to non-adversarial distributed stochastic gradient descent. Remarkably, VALID retains optimal performance metrics in adversary-free environments, sidestepping the robustness penalties observed in prior byzantine-robust methods. A distinctive aspect of our study is a heterogeneity metric based on the norms of individual agents' gradients computed at the global empirical loss minimizer. This not only provides a natural statistic for detecting significant byzantine disruptions but also allows us to prove the optimality of VALID in wide generality. Lastly, our numerical results reveal that, in the absence of adversaries, VALID converges faster than state-of-the-art byzantine robust algorithms, while when adversaries are present, VALID terminates with each honest agent either converging to an admissible consensus or declaring adversarial presence in the network.**

I. INTRODUCTION

Machine learning is increasingly reliant on data from a variety of distributed sources. As such, it may be difficult to ensure that the data which originates from these sources is trustworthy. Thus, there is a need to develop distributed and decentralized learning strategies that can respond to bad or even malicious data. However, worst-case or *Byzantine* resilience is an extremely strong requirement, that performance be maintained if a malicious adversary controls a subset of the processing nodes and takes any conceivable action. In practice, an adversary launching such an attack against a learning process requires tremendous resources which may not be worth the cost to influence the learned model. Thus, even though malicious adversaries are a threat, for the vast majority of the time, they are not present. An algorithm that maintains Byzantine robustness necessarily sacrifices performance when no adversaries are present. Instead, we seek a middle ground in which there is no performance loss at all when adversaries are not present, but they can still be detected if they are.

We consider decentralized learning in a network of agents — in contrast to federated learning [1], [2], there is no trusted central processing entity. Each agent in the network may communicate with its neighbors in a topology graph, and it has access to data samples from a certain distribution. The goal is to learn a consensus model at each agent that minimizes a total loss function from combining all agents' individual data. Rather than Byzantine robustness, we require the weaker assumption of *validation*. In particular, when no adversaries are present, there should be no performance degradation compared to the completely honest setting. When one or more adversaries are present, one of two outcomes may occur for each honest (non-adversarial) agent: either (i) the model is learned with minimal deviation from the completely honest setting, or (ii) the agent signals an alarm, indicating that it has detected the presence of the adversary. In the latter case, the honest agents signaling the alarm do not converge on a model; however, once they signal the alarm, they may fall back to a more conservative Byzantine resilient algorithm, or even abandon the learning task in favor of ridding the network of adversaries.

Outline of the paper: We formally introduce our problem setting in Section II. Next, in Sections III and IV, we describe our main results and give an overview of the VALID protocol. Finally, in Section V, we give experimental results to verify the performance of VALID on practical datasets and compare it against relevant benchmarks. In the interest of space, we present the detailed descriptions of our protocols and the proof arguments in the extended version [3]. We begin with a review of related work below.

Related work: Early works such as the Byzantine Generals Problem [4], [5], [6], [7] laid the groundwork for Byzantine fault tolerance. [8] introduced 'early stopping' algorithms allowing adaptivity to the actual number of Byzantine faults. Along this direction, [9], [10], [11] study algorithms that optimize performance in fault-free scenarios while retaining worst-case robustness. Cachin *et al.*'s work [12] on broadcast primitives for asynchronous Byzantine consensus is particularly relevant to our validation focus.

Work in decentralized learning may be traced back to distributed optimization methods (*c.f.*, [13], [14]). Subsequent work has examined both deterministic and stochastic optimization techniques in this context [15], [16]. See [17], [18] for a

This material is based upon work supported by the National Science Foundation under Grant No. CCF-1908725, CCF-2107526, CCF-2107370, and CCF-2107488. Author emails: mayank.bakshi@ieee.org, sg273@njit.edu, yauhen.yakimenka@njit.edu, beemera@uwec.edu, okosut@asu.edu, jkliewer@njit.edu.

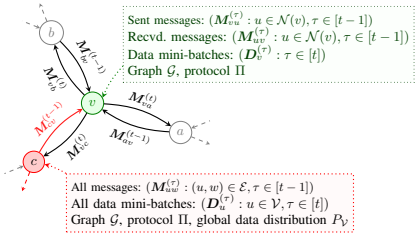

Fig. 1. This figure depicts the t-th round of a validated learning protocol (for $t = 1, 2, \ldots, T$). Agent v is an honest agent and agent c is a Byzantine agent. The rectangles next to agents v and agent c denote the available information to them at the beginning of the t-th round.

comprehensive survey of various methods in this area.

Byzantine-resilience in distributed optimization setups was first examined in the context of distributed estimation [19], [20], [21], [22] and Federated Learning [23], [24], [25], [26], [27]. The issue of Byzantine adversaries on distributed optimization was first examined by Su *et al.* [28] and has led to a substantial area of research, *c.f.* [29], [30], [31], [32], [33], [34], [35], [36]. We refer the reader to [37] for a comprehensive survey of these areas. In the context of decentralized learning, two broad approaches have emerged – *screening-based protocols* that involve outlier-robust aggregation of other agents' models by each agent [38], [39], [40], [41] and *performance-based protocols* that involve using each agent's local data to detect and eliminate outliers [42], [43]. Our validation requirement is similar to that in [22] that examines the problem of distributed estimation with an option to raise a "flag" if adversaries are detected.

II. Preliminaries

A. Learning Problem

We consider the problem of distributed empirical risk minimization across a set of agents $\mathcal{V}$. Each agent $v \in \mathcal{V}$ observes data sequentially, drawn independently and identically according to its *local distribution* $P_v \in \mathcal{P}(\mathcal{D})$, where the class of data distributions $\mathcal{P}(\mathcal{D})$ under consideration is known *a priori*. We employ a loss function $f : \mathbb{R}^d \times \mathcal{D} \to \mathbb{R}^+$ that evaluates the goodness-of-fit between the model vector $\boldsymbol{x} \in \mathbb{R}^d$ and the data $D \in \mathcal{D}$ with smaller loss values indicating a better fit.

For any subset $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ of $\mathcal{V}$, the joint data distribution is a product distribution $P_{\mathcal{U}} = P_{u_1} \times P_{u_2} \times \ldots \times P_{u_m} \in (\mathcal{P}(\mathcal{D}))^{|\mathcal{U}|}$. The joint distribution of all agents' data $P_{\mathcal{V}}$ is termed the *global data distribution*. The ideal learning goal[1] is to find a global empirical loss minimizer

$$\boldsymbol{x}^*(P_{\mathcal{V}}) = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbb{E}_{D \sim P_v} f(\boldsymbol{x}, D), \tag{1}$$

that attains the loss value $f^*(P_{\mathcal{V}}) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbb{E}_{D \sim P_v} f(\boldsymbol{x}^*(P_{\mathcal{V}}), D)$.

B. Network, Protocols, and Adversaries

1) Agent Graph: Agents are interconnected via an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ denotes the set of edges that specify which pairs of agents can exchange information noiselessly in each round. An agent v is considered a neighbor of agent u, denoted as $v \in \mathcal{N}(u)$, if and only if $(v, u) \in \mathcal{E}$.

[1]We refer to this as the "ideal" learning goal as finding $\boldsymbol{x}^*(P_{\mathcal{V}})$ (and hence, achieving $f^*(P_{\mathcal{V}})$) is, in general, too optimistic in the presence of adversaries. We elaborate on this further in Section II-C

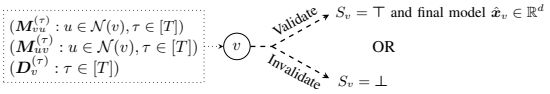

Fig. 2. At the conclusion of the T-th round, agent v either sets its validation state S_v to $\top$ to indicate a valid consensus or to $\perp$ to declare Byzantine presence.

2) Validated Learning Protocol: A Validated Learning Protocol Π is a decentralized protocol that operates over a fixed number of rounds, say T. As depicted in Figure 1, in each round $t \in [T]$, each agent v observes a mini-batch of data $\boldsymbol{D}_v^{(t)} = (D_{v,1}^{(t)}, D_{v,2}^{(t)}, \ldots, D_{v,K}^{(t)})$, where $D_{v,k}^{(t)} \in \mathcal{D}$ for all $k \in [K]$. Subsequently, each agent v exchanges messages with its neighbors based on the mini-batch $\boldsymbol{D}_v^{(t)}$ and accumulated information from previous rounds. At the conclusion of round T, agents independently compute their *validation state* S_v to be either "$\top$" or "$\perp$". If an agent v computes its final validation state to be $\top$, then it further outputs its final model vector $\hat{\boldsymbol{x}}_v$. We describe the desired outcome of a validated learning protocol in Section II-D.

3) Adversaries: An unknown subset $\mathcal{B}$ of agents, termed *Byzantine agents*, may exhibit adversarial behavior. We assume that Byzantine agents have a *causal* global knowledge, *i.e.*, at the outset of the t-th round, such agents know all other agents' sent transmissions and observed local data upto (and including) the t-th round. Additionally, Byzantine agents have full knowledge of the graph $(\mathcal{V}, \mathcal{E})$, the global data distribution $P_{\mathcal{V}}$, and the protocol Π. Armed with this knowledge, each Byzantine agent may produce arbitrary outputs (*i.e.*, they may deviate arbitrarily from Π) when exchanging messages with their neighbors. Agents in the subset $\mathcal{H} \triangleq \mathcal{V} \setminus \mathcal{B}$ are referred to as *honest agents*. Honest agents are unaware of the presence or identity of Byzantine agents.

C. Admissible Consensus Models

While the local distributions are not known *a priori*, we assume that it is common knowledge among the agents that $P_{\mathcal{V}}$ satisfies bounded heterogeneity. In this paper, the specific constraint defined below is of particular interest.

Definition 1 (δ-Heterogeneous Distributions)**.** *We say that the global data distribution $P_{\mathcal{V}} \in (\mathcal{P}(\mathcal{D}))^{|\mathcal{V}|}$ is δ-heterogeneous with respect to the loss function f if*

$$\frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \left\| \mathbb{E}_{D_v \sim P_v} \left[\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*(P_{\mathcal{V}}), D_v) \right] \right\|_2^2 \leq \delta,$$

where $\boldsymbol{x}^*(P_{\mathcal{V}})$ is as specified in (1)*. We denote the set of all δ-heterogeneous distributions in $(\mathcal{P}(\mathcal{D}))^{|\mathcal{V}|}$ as $\mathcal{P}_\delta$.*

Remark 1. *In the presence of Byzantine agents and heterogeneity, converging to $\boldsymbol{x}^*(P_{\mathcal{V}})$ is generally infeasible even if honest agents are able to detect if other agents deviate from honest behavior. To see this, consider a "benign" attack strategy orchestrated by Byzantine agents. Armed with the knowledge of $P_{\mathcal{V}}$ and δ, an adversary commandeers a subset $\mathcal{B}$ of nodes and prescribes specific data distributions Q_v for each $v \in \mathcal{B}$, while ensuring $P_{\mathcal{V}\setminus\mathcal{B}} \times Q_{\mathcal{B}} \in \mathcal{P}_\delta$. Each Byzantine agent v then simulates the actions of an honest agent, drawing data from Q_v instead of P_v. This renders the situation indistinguishable from a case where all agents, including those in $\mathcal{B}$, are honest and the actual global data*

distribution is $P_{\mathcal{V}\setminus\mathcal{B}} \times Q_{\mathcal{B}}$. Note that $f^*(P_{\mathcal{H}} \times Q_{\mathcal{B}})$ is an upper bound on the true global loss value $f^*(P_{\mathcal{V}})$.

Remark 1 underscores the need for including adversarial choices in attainable consensus models. We introduce the notion of *admissible consensus models* that capture this notion.

Definition 2 (($\mathcal{U}, P_{\mathcal{V}}, \delta$)-Admissible Consensus Models)**.** *Given a set of agents $\mathcal{U}$, a global data distribution $P_{\mathcal{V}}$, and a $\delta > 0$, a model vector $\hat{\boldsymbol{x}} \in \mathbb{R}^d$ is considered an $(\mathcal{U}, P_{\mathcal{V}}, \delta)$-admissible consensus model if there exist vectors $(\hat{\boldsymbol{g}}_v : v \in \mathcal{V} \setminus \mathcal{U})$ such that:*
a) $\sum_{u\in\mathcal{U}} \mathbb{E}_{D\sim P_u} \nabla_{\boldsymbol{x}} f(\hat{\boldsymbol{x}}, D) + \sum_{v\in\mathcal{V}\setminus\mathcal{U}} \hat{\boldsymbol{g}}_v = 0$ *and*
b) $\frac{1}{|\mathcal{V}|}\left[\sum_{u\in\mathcal{U}} \|\mathbb{E}_{D\sim P_u} \nabla_{\boldsymbol{x}} f(\hat{\boldsymbol{x}}, D)\|_2^2 + \sum_{v\in\mathcal{V}\setminus\mathcal{U}} \|\hat{\boldsymbol{g}}_v\|_2^2\right] \leq \delta$.
We denote the set of all $(\mathcal{U}, P_{\mathcal{V}}, \delta)$-admissible models by $\mathcal{A}(\mathcal{U}, P_{\mathcal{V}}, \delta)$.[2]

D. Protocol objectives

The goal of a validated learning protocol is that each honest agent either converges to an admissible consensus model or correctly identifies that there is at least one Byzantine agent in the network. Formally, let $\mathcal{H}_\top$ (resp. $\mathcal{H}_\perp$) be the subset of honest agents v that set their validation states S_v to $\top$ (resp. $\perp$) when the protocol terminates. Let ϵ (typically $o(1)$) be a non-negative tolerance parameter. We say a validated learning protocol Π terminates *ϵ-successfully* if one of the following outcomes is reached.

a) $\mathcal{B} = \phi$, $\mathcal{H}_\top = \mathcal{V}$, and $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} \|\boldsymbol{x}^*(P_{\mathcal{V}}) - \hat{\boldsymbol{x}}_v\|_2^2 < \epsilon$.
b) $\mathcal{B} \neq \phi$, $\mathcal{H}_\top \neq \phi$, and there exists $\hat{\boldsymbol{x}} \in \mathcal{A}(\mathcal{H}_\top, P_{\mathcal{V}}, \delta)$ such that $\frac{1}{|\mathcal{H}_\top|}\sum_{v\in\mathcal{H}_\top} \|\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}_v\|_2^2 < \epsilon$.
c) $\mathcal{B} \neq \phi$ and $\mathcal{H}_\perp = \mathcal{H}$.

E. Assumptions

Throughout this paper, assume that the Byzantine set $\mathcal{B}$ satisfies Assumption 1, the loss function f satisfies Assumptions 2, 3 and 4, the local data distributions P_v satisfy Assumptions 5 and 6 with respect to f, and the global data distribution $P_{\mathcal{V}}$ satisfies Assumption 7 with respect to f.

Assumption 1 (Existence of a source component)**.** *The graph $\mathcal{G}_{\mathcal{B}^c} \triangleq (\mathcal{V}\setminus\mathcal{B}, \mathcal{E}\setminus\mathcal{E}_{\mathcal{B}})$ is connected, where $\mathcal{E}_{\mathcal{B}} = \mathcal{E} \cap (\mathcal{B}\times\mathcal{V} \cup \mathcal{V}\times\mathcal{B})$ is the set of edges that are incident on nodes in $\mathcal{B}$.*

Assumption 2 (Finite loss)**.** $\sup_{D\in\mathcal{D}} f(\boldsymbol{x}, D) < \infty \; \forall\, \boldsymbol{x} \in \mathbb{R}^d$.

Assumption 3 (β-smooth)**.** *There exists $\beta > 0$ such that for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ and $D \in \mathcal{D}$,*

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, D) - \nabla_{\boldsymbol{x}} f(\boldsymbol{x}', D)\|_2 \leq \beta \|\boldsymbol{x} - \boldsymbol{x}'\|_2 .$$

Assumption 4 (μ-strongly convex)**.** *There exists $\mu > 0$ such that for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ and $D \in \mathcal{D}$,*

$$f(\boldsymbol{x}', D) \geq f(\boldsymbol{x}, D) + \langle \nabla_{\boldsymbol{x}} f(\boldsymbol{x}, D), \boldsymbol{x}' - \boldsymbol{x}\rangle + \frac{\mu}{2}\|\boldsymbol{x}' - \boldsymbol{x}\|_2^2 .$$

Assumption 5 (Bounded loss variance)**.** *There exists $\sigma_f > 0$ such that, for each agent v and for every $\boldsymbol{x} \in \mathbb{R}^d$,*

$$\mathbb{E}_{D\sim P_v}(f(x, D))^2 - (\mathbb{E}_{D\sim P_v} f(x, D))^2 \leq \sigma_f^2$$

Assumption 6 (Bounded gradient variance)**.** *There exists $\sigma_g > 0$ such that, for each agent v and for every $\boldsymbol{x} \in \mathbb{R}^d$,*

$$\mathbb{E}_{D\sim P_v} \|\nabla_{\boldsymbol{x}} f(x, D)\|_2^2 - \|\mathbb{E}_{D\sim P_v} \nabla_{\boldsymbol{x}} f(x, D)\|_2^2 \leq \sigma_g^2$$

[2]Note that when f is convex, for any $P_{\mathcal{V}} \in \mathcal{P}_\delta$, $\mathcal{A}(\mathcal{V}, P_{\mathcal{V}}, \delta) = \{\boldsymbol{x}^*(P_{\mathcal{V}})\}$. This follows from the uniqueness of the global empirical loss minimizer for convex loss functions.

Assumption 7 (δ-heterogeneous distribution)**.** *There exists $\delta > 0$ such that $P_{\mathcal{V}}$ is δ-Heterogeneous with respect to f.*

The following assumption is used in Theorem 2

Assumption 8 (f-completeness)**.** *We say that $\mathcal{P}_\delta$ is f-complete if for every $P_{\mathcal{V}} \in \mathcal{P}_\delta$, $\hat{\boldsymbol{x}} \in \mathbb{R}^d$, and $(\hat{\boldsymbol{g}}_v : v \in \mathcal{B}) \in \mathbb{R}^{d\times|\mathcal{B}|}$ satisfying*

$$\sum_{u\in\mathcal{H}} \mathbb{E}_{D\sim P_u} \nabla_{\boldsymbol{x}} f(\hat{\boldsymbol{x}}, D) + \sum_{v\in\mathcal{B}} \hat{\boldsymbol{g}}_v = 0 \text{ and}$$

$$\frac{1}{|\mathcal{V}|}\left[\sum_{u\in\mathcal{H}} \|\mathbb{E}_{D\sim P_u} \nabla_{\boldsymbol{x}} f(\hat{\boldsymbol{x}}, D)\|_2^2 + \sum_{v\in\mathcal{B}} \|\hat{\boldsymbol{g}}_v\|_2^2\right] \leq \delta,$$

there exist distributions $(Q_v : v \in \mathcal{B})$ such that $P_{\mathcal{H}} \times Q_{\mathcal{B}} \in \mathcal{P}_\delta$ and $\mathbb{E}_{D\sim Q_v} \nabla_{\boldsymbol{x}} f(\hat{\boldsymbol{x}}, D) = \hat{\boldsymbol{g}}_v$ for each $v \in \mathcal{B}$.

III. Main Results

Theorem 1 (The VALID protocol)**.** *Suppose that Assumptions 1-7 are satisfied. There exists a validated learning protocol* VALID *over T rounds with the following guarantees under any Byzantine attack,*

A. ***Successful termination:*** VALID *terminates $O(1/T)$-successfully with probability $1 - \exp(-O(KT))$.*
B. ***Convergence rate:*** *When $\mathcal{H}_\top \neq \phi$, there exists $\hat{\boldsymbol{x}} \in \mathcal{A}(\mathcal{H}_\top, P_{\mathcal{V}}, \delta)$ such that the final model vectors $(\hat{\boldsymbol{x}}_v : v \in \mathcal{H}_\top)$ satisfy $\sum_{v\in\mathcal{H}_\top} \|\hat{\boldsymbol{x}}_v - \hat{\boldsymbol{x}}\|_2^2 = O(|\mathcal{E}|\delta/T)$.*
C. ***Computational and Communication complexity:*** *With respect to the model dimension d, the number of iterations T, and the mini-batch size K, for each honest agent, the complexities of* VALID *scale as:*
   i) *Computational complexity: $O(c(d)TK + |\mathcal{V}||\mathcal{E}|dT)$, where $c(d)$ is the scaling (w.r.t. d) of the complexity of evaluating $\nabla_{\boldsymbol{x}} f(\cdot,\cdot)$.*
   ii) *Communication complexity: $O(dT + |\mathcal{E}|^2)$.*

Remark 2. *Even though honest agents do not know* a priori *whether or not $\mathcal{B} = \phi$, the rate of convergence of the model iterates $\boldsymbol{x}_v^{(t)}$ to the global optimal consensus model in the setting with $\mathcal{B} = \phi$ is identical to that of non-Byzantine stochastic gradient descent (where it is known that $\mathcal{B} = \phi$). Further, regardless of $\mathcal{B}$, the computational and communication complexities of* VALID *scale comparably to those for non-Byzantine distributed stochastic gradient descent [14].*

Theorem 2 (Optimality of VALID)**.** *Suppose $(f, \mathcal{P}_\delta)$ satisfy Assumptions 3- 8. Let Π be any validated learning protocol such that under no Byzantine attack, Π terminates ϵ-successfully with probability $1-\gamma$. For any $\mathcal{B} \subsetneq \mathcal{V}$, $P_{\mathcal{V}} \in \mathcal{P}_\delta$, and $\hat{\boldsymbol{x}} \in \mathcal{A}(\mathcal{V}\setminus\mathcal{B}, P_{\mathcal{V}}, \delta)$, there exists a Byzantine attack such that, with probability at least $1 - \gamma - o(1)$, Π terminates $O(1/T)$-successfully with $S_v = \top$ for each $v \in \mathcal{V}$ and $\sum_{v\in\mathcal{V}\setminus\mathcal{B}} \|\hat{\boldsymbol{x}}_v - \hat{\boldsymbol{x}}\|_2^2 = O(|\mathcal{V}\setminus\mathcal{B}|\delta/T)$.*

IV. The VALID Protocol

In this section, we describe our proposed VALID protocol. Due to space constraints, we prove Theorems 1 and 2 as well as provide additional algorithmic descriptions in [3]. VALID consists of two phases, the learning phase consisting of the LEARNMODEL sub-protocol and the validation phase consisting of the VALIDATEMODEL sub-protocol

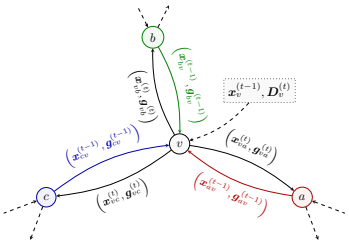

Fig. 3. The above figure shows the messages passed to node v by its neighbors in round $t-1$ and the messages passed from node v to its neighbor in round t of the LEARNMODEL protocol. Our LOCALVALIDATION protocol verifies if these messages are consistent with each other. For instance, if agent v performs all computations honestly, the messages for a single round must satisfy the property that $\boldsymbol{x}_{va}^{(t)} = \boldsymbol{x}_{vb}^{(t)} = \boldsymbol{x}_{vc}^{(t)}$ and $\boldsymbol{g}_{va}^{(t)} = \boldsymbol{g}_{vb}^{(t)} = \boldsymbol{g}_{vc}^{(t)}$ for all t. Further, the messages passed in different rounds are related by the equality $\boldsymbol{x}_{va}^{(t)} = (1-3\eta^{(t)})\boldsymbol{x}_{va}^{(t-1)} + \eta^{(t)}(\boldsymbol{x}_{av}^{(t-1)} + \boldsymbol{x}_{bv}^{(t-1)} + \boldsymbol{x}_{cv}^{(t-1)}) - \boldsymbol{g}_{va}^{(t)}$.

A. Learning phase

The learning phase is based on distributed stochastic gradient for non-Byzantine networks (*c.f.* [14]). In each round $t = 1, 2, \ldots, T$, each agent v computes its model iterate $\boldsymbol{x}_v^{(t)}$ and the gradient iterate $\boldsymbol{g}_v^{(t)}$ and transmits these to all its neighbors.

B. Validation phase

The validation phase comprises of local validation, global validation, and validation state agreement sub-protocols. Local validation examines message consistency in relation to the inputs, as shown in Figure 3. Global validation confirms the agents' model and gradient iterates are consistent with a δ-heterogeneous data distribution. Roughly speaking, successful local and global validations indicate that any Byzantine agents' iterates align with a $(\mathcal{V} \setminus \mathcal{B}, P_{\mathcal{V}}, \delta)$-admissible model. If either of these checks fails, the detecting agent's alarm state $S_v^{(0)}$ flips to $\perp$. The validation state agreement sub-phase ensures unanimous $\perp$ final states among honest agents if even one alarm state is $\perp$. However, even when the alarm state of all agents is $\top$, Byzantine activity during the validation state agreement phase may still lead some honest agents to set $\perp$. In this scenario, agents in $\mathcal{H}_\top$ converge to a valid consensus model, while those in $\mathcal{H}_\perp$ finalize their states as $\perp$.

1) Primitives: We introduce the validated broadcast primitive and the hash function employed in our protocol.

a) Validated Broadcast: This sub-protocol aims to uniformly broadcast a source agent's message, ensuring that honest agents either receive it intact or flag their validation state as $\perp$. Initially, only the source agent has the target message; others have a null message '$\varnothing$'. Agents exchange messages with neighbors, updating their own only if it was null or flagging $\perp$ upon discrepancies. After $\min\{|\mathcal{V}|, |\mathcal{E}|\}$ exchanges, all honest agents either possess the unaltered message or have flagged $\perp$.

b) Polynomial Hashing: Let $\mathbb{F}$ be a large enough finite field. Our hashing schemes are based on the following polynomial hash. For any vector $\xi \in \mathbb{R}^{d(T-2)}$ and $s \in \mathbb{F}$, define the hash $\text{HASH} : \mathbb{F} \times \mathbb{R}^{d(T-2)} \to \mathbb{R}$ as $\text{HASH}(s, \xi) = \sum_{i=1}^{d(T-2)} \xi_i \text{int}(s^{i-1})$, where, $\text{int}(s^{i-1})$ is integer representation of the finite field element s^{i-1}. Note that for a fixed key s, $\text{HASH}(s, \xi)$ is a linear function of ξ.

2) Local Validation: If agent v is honest, the transcripts on the edges incoming and outgoing from it must satisfy consistency and boundedness properties. Specifically, let $(\boldsymbol{x}_{uv}^{(t)}, \boldsymbol{g}_{uv}^{(t)})$ be the pair of vectors transmitted by agent u to agent v at the end of round t of the learning phase. By Lemma 4, an honest agent's transmissions must be bounded in magnitude. Next, let $\boldsymbol{X}_{uv}^{\text{out}} \triangleq [\boldsymbol{x}_{uv}^{(t)\top} : t = 1, 2, \ldots, T]^\top$, $\boldsymbol{X}_{uv}^{\text{in}} \triangleq [\boldsymbol{x}_{uv}^{(t)\top} : t = 0, 1, \ldots, T-1]^\top$, $\boldsymbol{X}_{uv}^{\text{in},\eta} \triangleq [\eta^{(t)}\boldsymbol{x}_{uv}^{(t)\top} : t = 0, 1, \ldots, T-1]^\top$, and and $\boldsymbol{\Gamma}_{uv} = [\alpha^{(t)}\boldsymbol{g}_{uv}^{(t)\top} : t = 1, 2, \ldots, T]^\top$ denote $\mathbb{R}^{dT}$-valued partial transcripts on the edge (u, v). An honest agent v implies that there exist $\overline{\boldsymbol{X}}_v^{\text{out}}, \overline{\boldsymbol{X}}_v^{\text{in}}, \overline{\boldsymbol{X}}_v^{\text{in},\eta}$, and $\overline{\boldsymbol{\Gamma}}_v$ such that $\overline{\boldsymbol{X}}_v^{\text{out}} = \boldsymbol{X}_{vu}^{\text{out}}$, $\overline{\boldsymbol{X}}_v^{\text{in}} = \boldsymbol{X}_{vu}^{\text{in}}$, $\overline{\boldsymbol{X}}_v^{\text{in},\eta} = \boldsymbol{X}_{vu}^{\text{in},\eta}$, and $\overline{\boldsymbol{\Gamma}}_v = \boldsymbol{\Gamma}_{vu}$ for $u \in \mathcal{N}(v)$. Further, if every agent performs communication and computations with these variables honestly, these variables further satisfy $\overline{\boldsymbol{X}}_v^{\text{out}} = \overline{\boldsymbol{X}}_v^{\text{in}} + \sum_{u \in \mathcal{N}(v)} (\overline{\boldsymbol{X}}_u^{\text{in},\eta} - \overline{\boldsymbol{X}}_v^{\text{in},\eta}) - \overline{\boldsymbol{\Gamma}}_v$.

The LOCALVALIDATE protocol first checks whether the norms of the messages transmitted by each neighbouring agent satisfy the bounds of Lemma 4 and then efficiently confirms transcript vector consistency for each agent v by using hash values instead of transmitting entire transcripts, thereby reducing communication cost. Agents first draw a private key s_v to compute and broadcast hash values of their incident transcript vectors. Following this, private keys are broadcast. This sequence prevents Byzantine agents from knowing other agents' keys before sharing their own hash values. Agents then recompute and broadcast hash values of their incident transcript vectors using all received keys. Broadcasts are executed using the validated broadcast protocol. Any discrepancies in consistency checks (or during the validated broadcast) prompt agents to set their alarm state $S_v^{(0)}$ to $\perp$.

3) Global Validation: The intuition behind the global validation phase is that when $\mathcal{B} = \phi$, for some collection of vectors $(\overline{\boldsymbol{g}}_v^* : v \in \mathcal{V})$ taken from $\mathbb{R}^d$, the gradient vectors must satisfy $\left\| \mathbb{E}_{\boldsymbol{D}^{(t)}} \boldsymbol{g}_{vu}^{(t)} - \overline{\boldsymbol{g}}_v^* \right\|_2^2 = O(1/t)$ for all $(v, u) \in \mathcal{E}$, $\sum_{v \in \mathcal{V}} \overline{\boldsymbol{g}}_v^* = 0$, and $\sum_{v \in \mathcal{V}} \|\overline{\boldsymbol{g}}_v^*\|_2^2 \leq \delta$.

a) Final gradient estimation and broadcast: Let γ be small enough (as specified in [3]). For each neighbor $u \in \mathcal{N}(v)$, node $v \in \mathcal{V}$ maintains estimates $\hat{\boldsymbol{g}}_{uv}$ of $\overline{\boldsymbol{g}}_u^*$, $\hat{\ell}_{uv}$ of $\|\overline{\boldsymbol{g}}_u^*\|_2$ using the following weighted sums:

$$\hat{\boldsymbol{g}}_{uv} = \frac{1-\gamma^{T-1}}{1-\gamma} \sum_{i=1}^{T-1} \gamma^{T-i-1} \boldsymbol{g}_{uv}^{(i)}, \text{ and}$$

$$\hat{\ell}_{uv} = \frac{1-\gamma^{T-1}}{1-\gamma} \sum_{i=1}^{T-1} \gamma^{T-i-1} \left\| \boldsymbol{g}_{uv}^{(i)} \right\|_2.$$

Next, every node v broadcasts $((\hat{\boldsymbol{g}}_{uv}, \hat{\ell}_{uv}) : u \in \mathcal{N}(v))$ using the validated broadcast algorithm.

b) Heterogeneity and optimality test: Set a slack parameter $\epsilon > 0$. Upon receiving the estimates $((\hat{\boldsymbol{g}}_{uv}, \hat{\ell}_{uv}) : (u, v) \in \mathcal{E})$, the following verification is performed by every agent.

1) **Consistency check:** If $(\hat{\boldsymbol{g}}_{uv}, \hat{\ell}_{uv}) \neq (\hat{\boldsymbol{g}}_{uv'}, \hat{\ell}_{uv'})$ for some $v, v' \in \mathcal{N}(u)$, declare the presence of an adversary. Else, set $(\hat{\boldsymbol{g}}_u^*, \hat{\ell}_u^*) = (\hat{\boldsymbol{g}}_{uv}, \hat{\ell}_{uv})$ for any $v \in \mathcal{N}(u)$.
2) **Optimality check:** If $\left\| \frac{1}{|\mathcal{V}|} \sum_{u \in \mathcal{V}} \hat{\boldsymbol{g}}_u^* \right\|_2 > \epsilon$, declare the presence of an adversary.
3) **Heterogeneity check:** If $\frac{1}{|\mathcal{V}|} \sum_{u \in \mathcal{V}} \hat{\ell}_u^{*2} > \delta + \epsilon$, declare the presence of an adversary.

4) The Validation State Agreement sub-phase: The goal of this sub-protocol is to ensure that if any honest agent detects a Byzantine presence, all others recognize this alert. Agents

exchange their validation states with neighbors, updating to $\perp$ if any neighboring state is $\perp$. After a set number of exchanges (up to $\min\{|\mathcal{V}|, |\mathcal{E}|\}$ times), agents concur on $\top$ if $\mathcal{B} = \phi$, or on $\perp$ if Byzantine presence was detected before this sub-protocol.[3]

V. Experiments

We test the performance of VALID and compare it with two state-of-the-art Byzantine-resilient algorithms for decentralized training, UBAR [42] and Bridge-median (Bridge-M) [40]. UBAR employs a combination of distance- and performance-based algorithms to mitigate the effect of adversaries. It checks the models sent by neighboring agents and discards outliers (based on the distances between vectors of parameters). Next, it tests the remaining models against the validation dataset, and takes the average of those with the best performances. Bridge-M screens the incoming models and applies a coordinate-wise median on the received parameters.

1) Network: We consider an undirected graph with $n = 20$ agents. The graph consists of two fully connected subgraphs of 10 agents each. These two subgraphs are connected with two edges chosen randomly. This ensures that there is a benign path between any two benign agents, provided there is at most one adversary.

2) Dataset and hyperparameters: We evaluate our validation scheme by training a Neural Network (NN) with three fully connected layers. Each of the first two fully connected layers is followed by ReLU, while softmax is used at the output of the third layer. We used a batch size of 300 and a decaying learning rate of $\frac{lr_{init}}{lr_{init}+T}$, where $lr_{init} = 5$ denotes the initial learning rate.

We consider two settings of data distributions - i.i.d. and moderately non-i.i.d., both based on the MNIST dataset [44]. To simulate independent and identically distributed (i.i.d.) data of every agent, the whole training dataset is shuffled randomly and partitioned equally among the agents. To model moderately non-i.i.d data of the agents, we first distribute i.i.d. data, as described above. Next, each of the 10 agents in the first subgraph randomly and independently picks four label classes out of "0", "1", "2", "3", and "4", and rotates the images from these four classes by 90 degrees. The procedure is repeated for the 10 agents in the second subgraph, but now the four label classes are chosen from "5", "6", "7", "8", and "9". As a result, the data distributions among the agents within each subgraph are closer to each other than to the data distribution of the agents in the other subgraph. Also, each image in the test dataset is rotated with a probability 0.4.

We now present our results for the moderately non-i.i.d. setting. Due to limited space, the results for i.i.d data distribution on MNIST are presented in [3].

A. No-adversary case

Fig.4 compares the performances of UBAR[4], Bridge-M, and VALID schemes in non-i.i.d. setting. The negative effect of non-i.i.d. data is more pronounced for UBAR and Bridge-M schemes. As UBAR is a performance-based scheme, it is hard for the agents to distinguish whether a high loss value for a given model sent by a neighboring agent can be attributed to a potential attack or to the heterogeneous nature of the data. Performance of the Bridge-M scheme also degrades in non-i.i.d setting, due to the coordinate-wise median choice of the parameters. To the contrary, in VALID, agents take the weighted average of all models received from their neighboring agents.

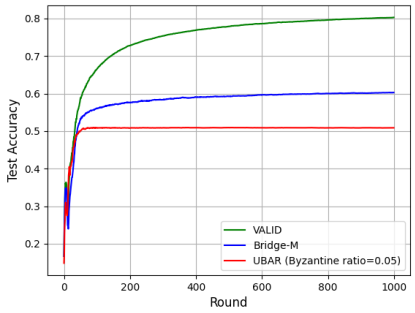

Fig. 4. Performance of VALID compared with UBAR and Bridge-M under non-i.i.d. setting.

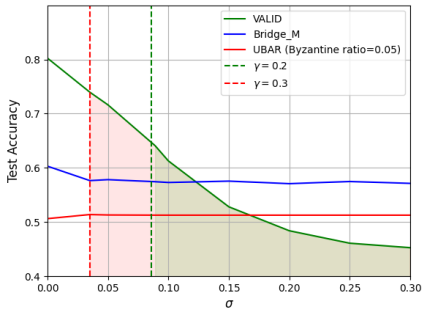

Fig. 5. Performance of VALID, UBAR, and Bridge-M under Gaussian attack (1 adversarial agent, non-i.i.d. setting).

B. Effect of an attack

We consider the following attack model: an adversary adds Gaussian noise $\mathcal{N}(0, \sigma^2)$ to each of its model parameters before sending them to its neighboring agents. While VALID demonstrates good performance in non-i.i.d setting, it is not a Byzantine-resilient scheme and its performance degrades in the presence of an adversary as the variance of the noise σ increases. In contrast, UBAR and Bridge-M are Byzantine-resilient schemes and their performances remain almost the same in the presence of an adversary with different values of σ as depicted in Fig. 5. However, VALID is able to detect the attack using the heterogeneity and optimality checks presented in section IV-B3. As depicted in Fig. 5, the sensitivity of VALID to detect the attack increases with γ. Therefore, it can detect weak attacks with the appropriate choice of γ. We experimentally select $\gamma \leq \gamma_{max}$, where $\gamma_{\max}$ is the largest value that can successfully pass the heterogeneity check without triggering a false alarm in the non-i.i.d. setting with no attack.

[3]Note that even without prior Byzantine detection, some honest agents (*i.e.*, those in $\mathcal{H}_\perp$) might adopt the validation state $\perp$ if Byzantine agents start intervening during this sub-protocol. In such cases, all honest agents may not reach consensus on the validation state since the alert raised by an honest agent detecting this Byzantine activity may not propagate to all agents before this sub-phase terminates (in a fixed number of time steps).

[4]Note that UBAR scheme is not specifically designed for non-i.i.d. setting.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, Apr. 2017, pp. 1273–1282.

[2] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[3] M. Bakshi, S. Ghasvarianjahromi, Y. Yakimenka, A. Beemer, O. Kosut, and J. Kliewer, "VALID: a Validated Algoritm for Learning in Decentralized Networks with Possible Adversarial Presence," arXiv preprint, May 2024.

[4] M. Pease, R. Shostak, and L. Lamport, "Reaching Agreement in the Presence of Faults," *J. ACM*, vol. 27, no. 2, pp. 228–234, Apr. 1980.

[5] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[6] D. Dolev, "The Byzantine generals strike again," *Journal of Algorithms*, vol. 3, no. 1, pp. 14–30, Mar. 1982.

[7] D. Dolev and H. R. Strong, "Authenticated Algorithms for Byzantine Agreement," *SIAM J. Comput.*, vol. 12, no. 4, pp. 656–666, Nov. 1983.

[8] D. Dolev, R. Reischuk, and H. R. Strong, "Early stopping in Byzantine agreement," *J. ACM*, vol. 37, no. 4, pp. 720–741, Oct. 1990.

[9] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, ser. OSDI '99. USA: USENIX Association, Feb. 1999, pp. 173–186.

[10] I. Abraham and D. Dolev, "Byzantine Agreement with Optimal Early Stopping, Optimal Resilience and Polynomial Complexity," in *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, ser. STOC '15. New York, NY, USA: Association for Computing Machinery, Jun. 2015, pp. 605–614.

[11] J.-P. Martin and L. Alvisi, "Fast Byzantine consensus," in *2005 International Conference on Dependable Systems and Networks (DSN'05)*, Jun. 2005, pp. 402–411.

[12] C. Cachin, K. Kursawe, F. Petzold, and V. Shoup, "Secure and Efficient Asynchronous Broadcast Protocols," in *Advances in Cryptology — CRYPTO 2001*, ser. Lecture Notes in Computer Science, J. Kilian, Ed. Berlin, Heidelberg: Springer, 2001, pp. 524–541.

[13] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

[14] A. Nedic and A. Ozdaglar, "Distributed Subgradient Methods for Multi-Agent Optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[15] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Incremental Stochastic Subgradient Algorithms for Convex Optimization," *SIAM J. Optim.*, vol. 20, no. 2, pp. 691–717, Jan. 2009.

[16] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained Consensus and Optimization in Multi-Agent Networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[17] A. Nedić, J.-S. Pang, G. Scutari, and Y. Sun, *Multi-Agent Optimization: Cetraro, Italy 2014*, ser. Lecture Notes in Mathematics, F. Facchinei and J.-S. Pang, Eds. Cham: Springer International Publishing, 2018, vol. 2224.

[18] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, Jan. 2019.

[19] A. Vempaty, L. Tong, and P. K. Varshney, "Distributed Inference with Byzantine Data: State-of-the-Art Review on Data Falsification Attacks," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 65–75, Sep. 2013.

[20] B. Kailkhura, Y. S. Han, S. Brahma, and P. K. Varshney, "Distributed Bayesian Detection in the Presence of Byzantine Data," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5250–5263, Oct. 2015.

[21] K. A. Lai, A. B. Rao, and S. Vempala, "Agnostic Estimation of Mean and Covariance," Aug. 2016.

[22] Y. Chen, S. Kar, and J. M. F. Moura, "Resilient Distributed Estimation Through Adversary Detection," *IEEE Trans. Signal Process.*, vol. 66, no. 9, pp. 2455–2469, May 2018.

[23] 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*, vol. 30. Curran Associates, Inc., 2017.

[24] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine Stochastic Gradient Descent," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[25] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 5650–5659.

[26] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed Stochastic Gradient Descent with Suspicion-based Fault-tolerance," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, pp. 6893–6901.

[27] ——, "Zeno++: Robust Fully Asynchronous SGD," in *Proceedings of the 37th International Conference on Machine Learning*. PMLR, Nov. 2020, pp. 10 495–10 503.

[28] L. Su and N. Vaidya, "Byzantine Multi-Agent Optimization: Part I," Jul. 2015.

[29] ——, "Byzantine Multi-Agent Optimization: Part II," Jul. 2015.

[30] ——, "Fault-Tolerant Multi-Agent Optimization: Part III," Sep. 2015.

[31] W. Xu, Z. Li, and Q. Ling, "Robust decentralized dynamic optimization at presence of malfunctioning agents," *Signal Processing*, vol. 153, pp. 24–33, Dec. 2018.

[32] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault, "The Hidden Vulnerability of Distributed Learning in Byzantium," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 3521–3530.

[33] S. Sundaram and B. Gharesifard, "Distributed Optimization Under Adversarial Nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, Mar. 2019.

[34] K. Kuwarabancharoen, L. Xin, and S. Sundaram, "Byzantine-Resilient Distributed Optimization of Multi-Dimensional Functions," in *2020 American Control Conference (ACC)*, Jul. 2020, pp. 4399–4404.

[35] L. Su and N. H. Vaidya, "Byzantine-Resilient Multiagent Optimization," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2227–2233, May 2021.

[36] M. Yemini, A. Nedić, S. Gil, and A. J. Goldsmith, "Resilience to Malicious Activity in Distributed Optimization for Cyberphysical Systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, Dec. 2022, pp. 4185–4192.

[37] 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, May 2020.

[38] E. M. El-Mhamdi, S. Farhadkhani, R. Guerraoui, A. Guirguis, L.-N. Hoang, and S. Rouault, "Collaborative Learning in the Jungle (Decentralized, Byzantine, Heterogeneous, Asynchronous and Nonconvex Learning)," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 25 044–25 057.

[39] Z. Yang and W. 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, Dec. 2019.

[40] C. Fang, Z. Yang, and W. U. Bajwa, "BRIDGE: Byzantine-Resilient Decentralized Gradient Descent," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 610–626, 2022.

[41] J. Hou, F. Wang, C. Wei, H. Huang, Y. Hu, and N. Gui, "Credibility Assessment Based Byzantine-Resilient Decentralized Learning," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–12, 2022.

[42] S. Guo, T. Zhang, H. Yu, X. Xie, L. Ma, T. Xiang, and Y. Liu, "Byzantine-Resilient Decentralized Stochastic Gradient Descent," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 4096–4106, Jun. 2022.

[43] A. R. Elkordy, S. Prakash, and S. Avestimehr, "Basil: A Fast and Byzantine-Resilient Approach for Decentralized Training," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2694–2716, Sep. 2022.

[44] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[45] D. Jakovetic, D. Bajovic, A. K. Sahu, and S. Kar, "Convergence Rates for Distributed Stochastic Optimization Over Random Networks," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec. 2018, pp. 4238–4245.

[46] A. K. Sahu, D. Jakovetic, D. Bajovic, and S. Kar, "Communication-Efficient Distributed Strongly Convex Stochastic Optimization: Non-Asymptotic Rates," Sep. 2018.