Byzantine-robust Decentralized Federated Learning via Dual-domain Clustering and Trust Bootstrapping

Peng Sun[†,♯], Xinyang Liu[§,‡,♯], Zhibo Wang[≀], Bo Liu[‡,*]
[†]College of Computer Science and Electronic Engineering, Hunan University, China
[§]Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, China
[≀]School of Cyber Science and Technology, Zhejiang University, China
[‡]Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), China
psun@hnu.edu.cn, codex.lxy@gmail.com, zhibowang@zju.edu.cn, liubo@cuhk.edu.cn

Abstract

Decentralized federated learning (DFL) facilitates collaborative model training across multiple connected clients without a central coordination server, thereby avoiding the single point of failure in traditional centralized federated learning (CFL). However, DFL exhibits increased susceptibility to Byzantine attacks owing to the lack of a responsible central server. Furthermore, a benign client in DFL may be dominated by Byzantine clients (more than half of its neighbors are malicious), posing significant challenges for robust model training. In this work, we propose DFL-Dual, a novel Byzantine-robust DFL method through dual-domain client clustering and trust bootstrapping. Specifically, we first propose to leverage both data-domain and model-domain distance metrics to identify client discrepancies. Then, we design a trust evaluation mechanism centered on benign clients, which enables them to evaluate their neighbors. Building upon the dual-domain distance metric and trust evaluation mechanism, we further develop a two-stage clustering and trust bootstrapping technique to exclude Byzantine clients from local model aggregation. We extensively evaluate the proposed DFL-Dual method through rigorous experimentation, demonstrating its remarkable performance superiority over existing robust CFL and DFL schemes.

1. Introduction

Federated learning (FL) is a popular distributed machine learning paradigm that enables collaborative model training across multiple clients without centralizing their raw training data [7, 15, 20, 39]. The traditional centralized federated learning (CFL) framework relies on a central server to coordinate the distributed model training process [3]. This dependence on a central entity may incur a single point of failures [16, 22]. Specifically, the normal model training process can be disrupted in cases where the central server experiences a crash or is hacked. Decentralized federated learning (DFL) [10, 12, 26] facilitates collaborative model training among connected clients without a central coordination server, thereby avoiding the single-point-of-failure issue. DFL has given rise to a new wave of distributed learning methods [17, 25, 35] that achieve comparable model accuracy to state-of-the-art CFL approaches while offering several significant advantages (e.g., fault tolerance, scalability, and flexibility) [18].

However, similar to CFL, DFL remains vulnerable to Byzantine attacks due to the inaccessibility of peer clients' local training data and the uninspectable local training process [9, 23, 24]. Specifically, malicious clients may tamper with local training data (i.e., data poisoning attacks) or falsify model parameters (i.e., model poisoning attacks) to craft malicious models to disrupt the model training process. Furthermore, DFL exhibits increased susceptibility to Byzantine attacks owing to the lack of a responsible central server. Consequently, effective Byzantine-robust DFL schemes (i.e., defense mechanisms) are highly desired to attain satisfactory DFL model training performance.

Thus far, researchers have developed diverse defense mechanisms against Byzantine attacks in CFL [5, 28, 34, 38]. The basic idea of these Byzantine-robust CFL approaches is that the central server tries to identify and exclude malicious local models from aggregation. However, their direct application to DFL is impeded due to the absence of coordination from a central entity. Furthermore, a benign client in DFL may be overwhelmed by Byzantine clients (i.e., most of its neighbors are malicious), which exacerbates the difficulty in identifying malicious clients.

[♯]Authors with equal contribution.
[*]Bo Liu is the corresponding author, Email: liubo@cuhk.edu.cn.

While recent studies have focused on the development of Byzantine-robust DFL schemes [6, 8, 27, 35], it is worth noting that, to the best of our knowledge, they have not explicitly accounted for the practical and crucial problem setting where malicious neighbors may dominate benign clients, and the data distribution among clients is highly non-independent and identically distributed (non-IID).

In this work, we propose DFL-Dual, a novel Byzantine-robust DFL method through dual-domain client clustering and trust bootstrapping. DFL-Dual employs multiple distance metrics in both model-domain (cosine similarity and Euclidean distance) and data-domain (Wasserstein distance) to distinguish benign clients from Byzantine ones. Hence, even under a rigorous adversary setting where the data is highly non-IID and Byzantine clients dominate benign ones, DFL-Dual remains resilient. Specifically, we first propose to leverage both model-domain Euclidean distance and data-domain Wasserstein distance to identify disparities among clients. Then, we establish a trust evaluation mechanism centered on benign clients, leveraging cosine similarities of their local models with those of their neighbors for assessment. Building upon the dual-domain distance metrics and trust evaluation mechanism, we further devise a two-stage clustering and trust bootstrapping technique. The first stage generates a divergence rate for each client, while the second stage excludes malicious local models from model aggregation. The main contributions of this work are summarized as follows: 1) ***A Novel Byzantine-robust DFL Framework***: To our best knowledge, DFL-Dual is the first Byzantine-robust DFL framework that can effectively defend against both untargeted and targeted Byzantine attacks under a rigorous adversary setting with **exceeding** 50% Byzantine clients and highly non-IID data distributions; 2) ***Multi Distance Metric Utilization***: We leverage multiple distance metrics in both model-domain and data-domain to identify disparities among clients. This multi-metric combination enables accurate discrimination between Byzantine clients and benign ones; 3) ***Two-stage Clustering and Trust Bootstrapping***: We design a two-stage clustering and trust bootstrapping technique. The first stage generates a divergence rate for each client, while the second stage excludes malicious local models from model aggregation; and 4) ***Extensive Performance Evaluation***: We thoroughly evaluate DFL-Dual through extensive experiments on various datasets, models, adversary settings, and Byzantine attacks. The results validate its significant performance superiority over existing schemes.

2. Preliminaries and Related Work

2.1. Decentralized Federated Learning

Consider a DFL system that consists of a set $\mathcal{N} = \{1, 2, \dots, N\}$ of clients. Each client $i \in \mathcal{N}$ has a private training dataset $\mathcal{D}_i$ containing $|\mathcal{D}_i|$ data samples and holds a local model $\boldsymbol{\theta}_i$. Formally, DFL aims to find a model $\boldsymbol{\theta}$ that minimizes the weighted average of losses among N clients:

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} F_i\left(\boldsymbol{\theta}; \mathcal{D}_i\right), \tag{1}$$

where $F_i\left(\boldsymbol{\theta}; \mathcal{D}_i\right) = \frac{1}{|\mathcal{D}_i|} \sum_{\zeta \in \mathcal{D}_i} F\left(\boldsymbol{\theta}; \zeta\right)$ is the local loss function of client i. DFL usually involves $\mathcal{T} = \{1, 2, \dots, T\}$ rounds. In each round $t \in \mathcal{T}$, the following procedures are sequentially executed.

- *Local Model Training:* Each client i samples a mini-batch of training samples from its local training dataset and computes a stochastic gradient $\boldsymbol{g}_i^t$. Then, client i updates its local model as

$$\boldsymbol{\theta}_i^{t+\frac{1}{2}} = \boldsymbol{\theta}_i^t - \eta \boldsymbol{g}_i^t, \tag{2}$$

  where η denotes the learning rate, $\boldsymbol{\theta}_i^t$ represents the local model of client i at the beginning of the t-th global training round, and $\boldsymbol{\theta}_i^{t+\frac{1}{2}}$ stands for the *pre-aggregation local model* of client i in round t.
- *Model Exchange and Aggregation:* Each client i sends its *pre-aggregation local model* $\boldsymbol{\theta}_i^{t+\frac{1}{2}}$ to its connected neighbors and receives their counterparts. Then, each client i aggregates the received *pre-aggregation local models* (including its own) to update its local model as

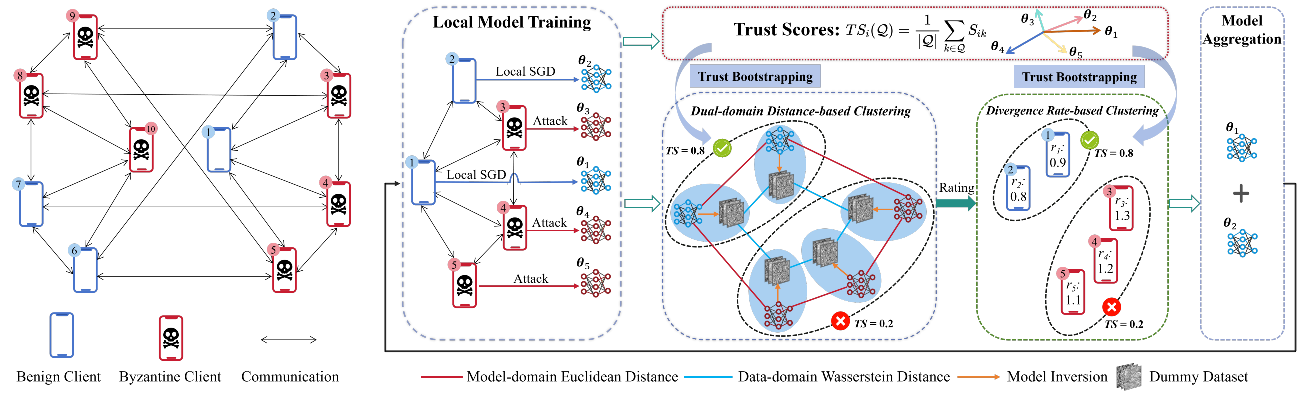
$$\boldsymbol{\theta}_i^{t+1}\left(\mathcal{G}_i\right) = \mathrm{Agg}\left(\left\{\boldsymbol{\theta}_k^{t+\frac{1}{2}} : k \in \mathcal{G}_i\right\}\right), \tag{3}$$

  where $\mathcal{G}_i$ denotes the sub-graph centered on client i (including client i and its neighbors), $\mathrm{Agg}\left(\cdot\right)$ represents the adopted aggregation rule (e.g., the consensus update rule in [19]), and the resulting $\boldsymbol{\theta}_i^{t+1}$ is the *post-aggregation local model* of client i in round t.

2.2. Byzantine-Robust CFL and DFL

In CFL, a commonly used aggregation rule is FedAvg [20]. However, the *post-aggregation local model* of a benign client can be easily manipulated by a malicious local model crafted by Byzantine clients in FedAvg [4]. To thwart such Byzantine attacks and achieve secure model training, researchers have developed various Byzantine-robust aggregation rules [4–6, 9, 27, 33, 36, 37].

For example, `Krum` [4] aggregates a client's received local models by selecting the one with the smallest sum of Euclidean distances to its subset of neighboring local models. `Median` and `Trimmed Mean` [36] are two robust aggregation rules based on coordinate-wise statistics. They compute the coordinate-wise median and trimmed average as the aggregated value for each model parameter among all received local models of a client. By employing `FLtrust` [5], a benign client can assign a low trust score to



(a) Communication topology

(b) Schematic diagram of DFL-Dual

Figure 1. Framework of DFL-Dual.

a neighbor's *pre-aggregation local model* if it significantly deviates from the client's own *pre-aggregation local model*. The recently proposed `FLdetector` [37] investigated the defenses from a new perspective, i.e., detecting malicious clients by checking their model-updates consistency. Thus, each client can only aggregate shared local models from neighbors detected as benign. The work in [27] proposed `Bristle`, which enables each client to securely update its model by designing a fast distance-based prioritize and a novel performance-based integrator.

The work in [8] applied `Median`, `Trimmed Mean`, and `Krum` aggregation rules to DFL. An iterative filtering rule is designed for DFL in [30], where a benign client repeatedly discards the model with the largest Euclidean distance to the average of its neighbors' models. Nevertheless, existing studies leave a notable gap in understanding how to effectively establish a Byzantine-robust DFL framework for practical and essential problem settings where *malicious neighbors may overwhelm benign clients and the data distribution among clients is highly non-IID*.

3. Threat Model

- *Attacker's Goal:* The attacker aims to send well-crafted poisoned *pre-aggregation local models* via compromised clients to benign clients to disrupt the DFL model training process. We consider both untargeted attacks (aiming to ruin the model performance indiscriminately) and targeted attacks (aiming to manipulate the model behavior on specific attacker-chosen inputs) in this work.
- *Attacker's Capability:* We consider a rigorous scenario where the attacker can compromise over 50% of the entire client population. Moreover, the compromised clients can strategically cluster around benign clients and dominate them. That is, more than half of a benign client's neighbors can be Byzantine clients.
- *Attacker's Background Knowledge:* We consider two cases of attacker's background knowledge (i.e., full knowledge and partial knowledge). Besides the local training data and models at compromised clients in the partial knowledge scenario, the attacker also knows the *pre-aggregation local models* on every benign client in the full knowledge scenario. Note that the full knowledge scenario has limited applicability in practice as we cannot ensure any two clients are connected. We use it to evaluate our defensive performance against adaptive attacks.

4. Methodology

4.1. Overview of DFL-Dual

DFL-Dual relies on benign clients to identify and filter out malicious *pre-aggregation local models* crafted by compromised neighbors in each training round. Without loss of generality, we take a benign client i and its connected neighbors (forming a sub-graph $\mathcal{G}_i$) as a concrete example to illustrate how DFL-Dual works. The framework of DFL-Dual is presented in Figure 1, and the workflow in each round of model aggregation at benign client i is as follows:

- *Dual-Domain Distance Computation:* After receiving all *pre-aggregation local models* from its neighbors, the benign client i computes pairwise Euclidean distances between any two *pre-aggregation local models* in $\mathcal{G}_i$. Besides, benign client i performs privacy-respecting model inversion on all *pre-aggregation local models* of clients in $\mathcal{G}_i$ to synthesize a corresponding dummy dataset for each client. Then, it computes pairwise Wasserstein distances among all synthesized dummy datasets. The weighted sum of these two distances constitutes the *dual-domain distance*, which will be used for client clustering.
- *Cosine Similarity Computation:* Benign client i computes the cosine similarity between its own *pre-aggregation local model* and its neighbors' counterparts, which will be used to obtain the *trust score*.

- *Two-stage Clustering and Trust Bootstrapping:* For each client $j \in \mathcal{G}_i$, benign client i clusters remaining clients $\mathcal{G}_i \backslash j$ into two groups based on their dual-domain distances to client j. The trust score of each group (defined as the average cosine similarity of the *pre-aggregation local models* in the group *w.r.t* client i) bootstraps the group selection, which allows to determine a divergence rate for each client $j \in \mathcal{G}_i$. Then, benign client i clusters all clients in $\mathcal{G}_i$ into two groups based on the generated divergence rates, with the trust score of each group bootstrapping the local model aggregation for client i.

4.2. Dual-Domain Distance Computation

Unlike existing studies (e.g., [4, 5, 8, 30]) that rely on single-domain distances to detect Byzantine clients, DFL-Dual utilizes dual-domain distances to enable each benign client to identify its Byzantine neighbors more accurately.

4.2.1 Model-Domain Distance Computation

Following prior works (e.g., [4, 30]), we employ the Euclidean distance (ED) metric to measure the discrepancies between benign and Byzantine clients in the model domain. Formally, the Euclidean distance $E_{ij}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ between two local models $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ is computed as

$$E_{ij}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|_2, \tag{4}$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm of a vector. Generally, a larger Euclidean distance means greater discrepancy.

However, the Euclidean distance metric suffers from the curse of dimensionality [11]. Specifically, deep models can be viewed as high-dimensional vectors, and usually, the Euclidean distance is unable to distinguish poisoned models from benign ones in high-dimensional space. Hence, we further introduce the data-domain distance metric below.

4.2.2 Data-Domain Distance Computation

In DFL, no client has access to others' private training data. Consequently, a natural question arises: how can we obtain data-domain distances to reveal the disparities between Byzantine and benign clients? To answer this question, we introduce a privacy-respecting model inversion method to obtain a dummy dataset for each client.

1) Privacy-Respecting Model Inversion. Inspired by Deep Leakage from Gradients (`DLG`) [40] that infers private training data of clients from their shared gradients in FL, we introduce a privacy-respecting model inversion method to obtain a dummy dataset for each client. The basic idea of `DLG` is randomly generating dummy data samples and iteratively updating them by matching the dummy gradients derived from dummy data with clients' shared actual gradients. However, to avoid privacy leakage of clients as in `DLG` and highlight the discrepancies among different clients' underlying data distributions, we make the following threefold adaptations to `DLG`.

- First, we allow each client i to perform E epochs of local training via mini-batch SGD with a batch size of B. Then, each client shares the *pre-aggregation local model rather than raw gradient* to its neighbors along with the epoch number E and mini-batch size B. In this way, client i shares an equivalent gradient $\left(\boldsymbol{\theta}_i^{t+1/2} - \boldsymbol{\theta}_i^t\right) / (E|\mathcal{D}_i|/B)$ in each training round t.
- Second, we introduce a scaling factor $s^{(t)}$ (i.e., s to the power of t) in each round t to amplify the differences among the equivalent gradients of different clients, especially in later training rounds where the equivalent gradients start to cancel out (approaching $\mathbf{0}$).
- Third, unlike `DLG` that optimizes both feature x_i' and label y_i' of each dummy data sample, we propose to optimize only x_i', while y_i' is sampled uniformly at random from all possible labels of the dataset and fixed.

Therefore, the dummy dataset for client i in round t is generated by solving the following optimization problem:

$$x_i'^* = \arg\min_{x_i'} \| \frac{\partial \ell\left((x_i', y_i'); \boldsymbol{\theta}_i^t\right)}{\partial \boldsymbol{\theta}_i^t} - \frac{s^{(t)}\left(\boldsymbol{\theta}_i^{t+1/2} - \boldsymbol{\theta}_i^t\right)}{E|\mathcal{D}_i|/B} \|_2^2, \tag{5}$$

where x_i' is the dummy feature to be optimized and $\ell(\cdot)$ is the loss function. We will show that the optimal solution $x_i'^*$ is close to the original data feature x_i from the perspective of distribution (without disclosing pixel-level private information), thus enabling each benign client to assess its neighbors in the data domain.

2) Wasserstein Distance Determination. We use the Wasserstein distance (WD) [29] between generated dummy datasets to capture the data-domain divergences among clients. The Wasserstein distance between any two dummy datasets (x_i', y_i') and (x_j', y_j') of clients i and j is given as

$$W_{ij}\left(x_i', x_j'\right) = \sum_{c=1}^{l} \sum_{d=1}^{m} Wass\left(x_i'^{,c,d,}, x_j'^{,c,d,}\right), \tag{6}$$

where $Wass(\cdot,\cdot)$ is the WD between any two vectors, $x_i'^{,c,d,}$ and $x_j'^{,c,d,}$ denotes the vector of all samples with feature d and label c in the dummy datasets of clients i and j, respectively, and l and m are the total number of labels and features of generated dummy datasets, respectively.

4.2.3 Dual-domain Distance Calculation

After receiving the *pre-aggregation local models* from its neighbors, benign client i clips the weighted sum of Euclidean distance and Wasserstein distance of any two clients in the sub-graph $\mathcal{G}_i$ to obtain the pairwise dual-domain distance as follows:

$$D_{ij} = \min\left(W_{ij} + \alpha E_{ij}, C_1\right), \forall (i,j) \in \mathcal{G}_i, \tag{7}$$

where D_{ij} is the dual-domain distance between clients i and j, with α and C_1 being tunable empirical parameters.

4.3. Trust Score Determination

To facilitate the accurate identification of Byzantine neighboring clients for benign clients, we further introduce the cosine similarity (CS) distance metric. It is a dimensionless metric with values falling within $[-1, 1]$, which helps achieve fair and robust evaluation of models under different attacks and environments. Formally, the cosine similarity between two models $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ is computed as

$$S_{ij}\left(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j\right) = \frac{\langle \boldsymbol{\theta}_i, \boldsymbol{\theta}_j \rangle}{\|\boldsymbol{\theta}_i\|_2 \cdot \|\boldsymbol{\theta}_j\|_2}. \tag{8}$$

Clearly, a smaller cosine similarity value $S_{ij}\left(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j\right)$ means the two models deviate from each other more significantly.

Then, we introduce the trust score (TS) of a client group from a benign client's perspective. Specifically, we define the trust score of a client group as the average cosine similarity values of the corresponding *pre-aggregation local models w.r.t.* that of a benign client. Mathematically, the trust score of benign client i to one group $\mathcal{Q}$ of its neighbors is computed as

$$TS_i\left(\mathcal{Q}\right) = \frac{1}{|\mathcal{Q}|} \sum_{k \in \mathcal{Q}} S_{ik}. \tag{9}$$

The trust score will bootstrap the selection of a benign group of neighbors for benign clients, as elaborated below.

4.4. Two-stage Clustering and Trust Bootstrapping

Large divergences in models and data among clients are common in DFL, especially when the data distribution among clients is highly non-IID. Hence, instead of simply rejecting *pre-aggregation local models* with large divergences, we propose a two-stage clustering and trust bootstrapping (TB) mechanism, whose workflow is as follows:

- **Stage 1**: Dual-domain Distance-based Clustering and Trust Bootstrapping. For each client $j \in \mathcal{G}_i$, benign client i clusters remaining clients $\mathcal{G}_i \backslash j$ into two groups M_{j1} and M_{j2} based on their dual-domain distances to client j. That is,

$$M_{j1}, M_{j2} = \text{2-Median}\left(\{D_{kj}, k \in \mathcal{G}_i \backslash j\}\right). \tag{10}$$

Then, benign client i bootstraps the selection of the group M_j^* for client j with a higher trust score, which is

$$M_j^* = \begin{cases} M_{j1}, & TS_i\left(M_{j1}\right) > TS_i\left(M_{j2}\right), \\ M_{j2}, & \text{otherwise.} \end{cases} \tag{11}$$

Algorithm 1: DFL-Dual

Inputs: Client number N, communication topology $\mathcal{G}$, global training rounds T, Clipping parameters C_1 and C_2.
Outputs: Local models $\boldsymbol{\theta}_i^T$ for each client $i \in \mathcal{N}$.
Initialization: Local models $\boldsymbol{\theta}_i^0$ for each client $i \in \mathcal{N}$.
for $t \in \{1, 2, \dots, T\}$ **do**
  for *benign client* $i \in \mathcal{N}$ ***in parallel*** **do**
    $\boldsymbol{\theta}_i^{t+1/2} \leftarrow$ Local update by (2).
  end
  for *benign client* $i \in \mathcal{N}$ ***in parallel*** **do**
    $E_{kj}, \forall (k,j) \in \mathcal{G}_i \leftarrow$ Compute ED by (4).
    $W_{kj}, \forall (k,j) \in \mathcal{G}_i \leftarrow$ Compute WD by (6).
    $D_{kj}, \forall (k,j) \in \mathcal{G}_i \leftarrow$ Compute dual-domain distance by (7).
    $S_{ij}, \forall j \in \mathcal{G}_i \leftarrow$ Compute CS by (8).
    $M_{j1}, M_{j2}, j \in \mathcal{G}_i \leftarrow$ Clustering by (10).
    $M_j^*, j \in \mathcal{G}_i \leftarrow$ Bootstraps selection by (11).
    $r_j, j \in \mathcal{G}_i \leftarrow$ Obtain divergence rate by (12) and (13).
    $N_{i1}, N_{i2}, j \in \mathcal{G}_i \leftarrow$ Clustering by (14).
    $N_i^*, j \in \mathcal{G}_i \leftarrow$ Bootstraps selection by (15).
    $\boldsymbol{\theta}_i^{t+1}(N_i^*) \leftarrow$ Model aggregation by (3).
  end
end

Furthermore, benign client i computes the divergence rate r_j for client j based on the selected groups M_j^* and M_i^* as follows:

$$r_j = \min\left(q_j / q_i, C_2\right) \tag{12}$$

where

$$q_j = \sum_{k \in M_j^*} D_{jk}, q_i = \sum_{k \in M_i^*} D_{ik}, \tag{13}$$

and C_2 is a tunable empirical parameter.

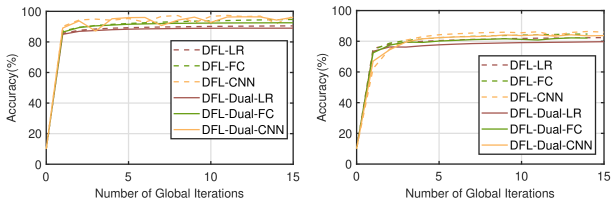
- **Stage 2**: Divergence Rate-based Clustering and Trust Bootstrapping. Based on the divergence rates of client $j \in \mathcal{G}_i$, benign client i first clusters all clients $j \in \mathcal{G}_i$ into two groups N_{i1} and N_{i2}, i.e.,

$$N_{i1}, N_{i2} = \text{2-Median}\left(\{r_j, j \in \mathcal{G}_i\}\right). \tag{14}$$

Benign client i then bootstraps the selection of the group N_i^* with a higher trust score as follows:

$$N_i^* = \begin{cases} N_{i1}, & TS_i\left(N_{i1}\right) > TS_i\left(N_{i2}\right), \\ N_{i2}, & \text{otherwise.} \end{cases} \tag{15}$$

The *pre-aggregation local models* in the finally selected group N_i^* are aggregated to obtain the *post-aggregation lo-*



(a) MNIST (b) Fashion-MNIST

Figure 2. The performance comparison between DFL and DFL-Dual without Byzantine attacks.

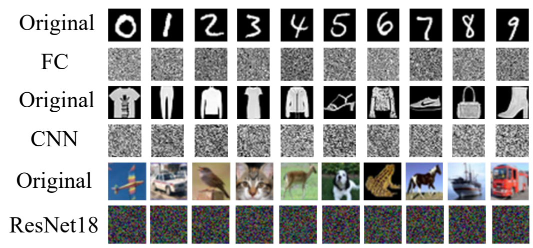


Figure 3. Illustration of original and dummy data samples.

cal model for client i. The details of the proposed DFL-Dual method are summarized in Algorithm 1.

5. Experiments

5.1. Experimental Setup

Taking Figure 1 (a) as an example of the decentralized communication topology, we evaluate DFL-Dual on different datasets and various models with two performance metrics of Accuracy (ACC) and Attack Success Rate (ASR). Specifically, we evaluate DFL-Dual on MNIST [14] and Fashion-MNIST [31] using Logistic Regression (LR), Fully Connected (FC), and Convolutional Neural Network (CNN), and on CIFAR-10 [13] using ResNet-18. We adopt the same method in [5, 38] to simulate different non-IID data distribution degrees. Specifically, the non-IID degree is captured by a sample allocation probability p, with larger p indicating a higher non-IID degree. We consider both untargeted and backdoor attacks. The untargeted attacks include Label Flipping Attack, Krum Attack [9], and Back-Gradient Attack [21], while the targeted attacks include Scaling Attack [1], DBA Attack [32], and A little is Enough Attack [2]. We take 6 aggregation methods (i.e., DFL [19], DFLTrust [5], DFLDetector [37], Multi-Krum [4], BridgeM [8], and IOS [30]) as baselines. Notably, for those designed for CFL, *we trim them to fit in the DFL scenario*. All experiments are conducted using PyTorch 2.0 on a machine with 2 RTX 4090 GPUs. The detailed experimental settings and parameters are provided in the supplementary material.

5.2. Convergence Performance of DFL-Dual

We first consider an ideal case that all 10 clients in Figure 1 (a) are benign with the non-IID degree being 0.8. Figure 2 shows the model performance via DFL-Dual and vanilla DFL, and we find DFL-Dual converges as nicely as vanilla DFL when no Byzantine attacks happen.

5.3. Privacy-respecting Property of DFL-Dual

In the model inversion process, DFL-Dual generates a dummy dataset (with 10 samples for MNIST and Fashion-MNIST, and 5 samples for CIFAR10, for each class) based on a client's *pre-aggregation local model*. Figure 3 illustrates the original images and the generated dummy samples (images) by the model inversion process in DFL-Dual. We find from this figure that it is nearly impossible to infer any private information from the generated dummy data samples, and thus verifies the privacy-respecting property of DFL-Dual.

5.4. Defense against Untargeted Attacks

The averaged accuracy of different models (CNN, FC, and LR) trained on various datasets using different aggregation methods is shown in Table 1. It is seen from the table that DFL-Dual consistently exhibits the highest accuracy under different untargeted attacks on almost all of the training tasks compared to other baselines. This verifies the effectiveness and robustness of the proposed DFL-Dual method.

Src / Def		MNIST		Fashion		CIFAR10
		CNN	LR	CNN	FC	ResNet18
DFL (No Attack)		95.39	89.84	84.85	82.67	49.96
Label Flipping	DFLTrust	18.28	1.11	12.8	53.21	10
	DFLDetector	33.84	89.9	84.06	36.36	29.58
	Multi-Krum	36.45	**89.83**	**84.37**	60.4	25.09
	DFL	26.05	15.40	31.78	20.85	25.3
	BridgeM	50.31	44.76	61.12	66.17	34.89
	IOS	0.24	0.95	0.51	0.57	20.59
	DFL-Dual	**96.64**	88.97	83.98	**82.03**	**49.06**
Krum	DFLTrust	20.01	1.11	14.76	64.33	10
	DFLDetector	22.01	17.66	32.91	31.5	22.08
	Multi-Krum	30.35	24.01	27.42	32.08	10
	DFL	71.14	71.88	49.76	58.88	10
	BridgeM	26.12	42.44	27.66	37.91	10
	IOS	77.08	77.59	50.44	68.11	10
	DFL-Dual	**96.14**	**89.05**	**83.69**	**81.91**	**49.84**
Back-Gradient	DFLTrust	9.8	9.8	10	10	10
	DFLDetector	9.8	14.74	10	11.75	10
	Multi-Krum	9.8	15.61	10	10.17	11.72
	DFL	25.81	56.05	19.41	27.39	10.03
	BridgeM	22.17	47.72	28.23	35.65	15.70
	IOS	10.52	42.17	12.33	34.59	19.29
	DFL-Dual	**95.14**	**88.99**	**83.73**	**81.99**	**49.1**

Table 1. Accuracies (%) under Untargeted Attacks.

5.5. Defense against Targeted Attacks

The averaged accuracy and ASR of different models (CNN, FC, and LR) trained on various datasets using different aggregation methods are shown in Table 2. The results validate that DFL-Dual consistently exhibits higher accuracy on benign testing data and lower ASR on testing data with backdoor triggers than other baselines.

Attack	Defence \ Source	MNIST CNN	Fashion CNN	CIFAR10 ResNet18
	DFL (No Attack)	95.39	84.85	49.96
Scaling	DFLTrust	9.8/100	10/100	19.45/100
	DFLDetector	67.06/99.75	69/91.94	20.85/85.66
	Multi-Krum	96.92/99.99	98.44/2.55	31.62/53.49
	DFL	49.61/100	61.78/98.91	18.7/79.14
	BridgeM	72.02/99.96	57.3/98.34	26.23/65.86
	IOS	11.01/97.43	81.74/91.85	30.78/53.64
	DFL-Dual	**96.21/0.50**	**84.83/1.70**	**49.01/4.44**
DBA	DFLTrust	9.8/100	10/100	18.64/82.27
	DFLDetector	34.06/70.46	84.97/4.21	17.53/82.59
	Multi-Krum	96.89/0.43	84.93/3.34	26.28/58.76
	DFL	9.8/100	10/100	17.87/87.49
	BridgeM	22.17/100	28.23/91.69	15.7/80.08
	IOS	**97.04/0.29**	82.13/1.91	25.87/62.34
	DFL-Dual	96.54/0.48	**83.38/2.53**	**48.79/4.35**
A Little is Enough	DFLTrust	92.29/99.75	80.1/98.91	33.59/100
	DFLDetector	92.34/8.74	83.08/14.74	36.94/100
	Multi-Krum	95.25/0.72	84.38/7.42	38.62/97.64
	DFL	95.01/85.35	81.55/86.25	45.66/99.67
	BridgeM	95.66/5.60	83.27/28.74	48.44/89.38
	IOS	**96.95/0.53**	84.31/5.9	45.05/89.82
	DFL-Dual	95.59/0.55	**83.88/2.16**	**50.01/4.78**

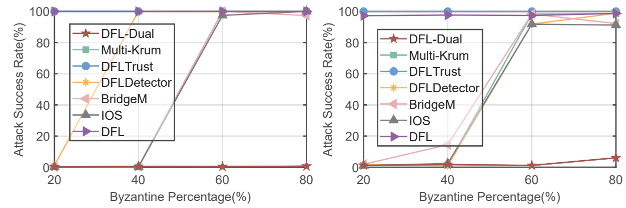
Table 2. Accuracies/ASRs (%) under Targeted Attacks.

5.6. Impact of Adversary Parameters

To further assess the effectiveness of DFL-Dual, we systematically compare its defensive performance on various configurations, including different percentages of Byzantine clients and various degrees of non-IID data distribution. We take the scaling attack as an example for the above comparison study, with the default Byzantine percentage and non-IID degree being 60% and 0.8, respectively.

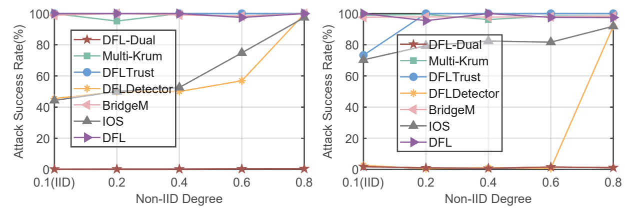
5.6.1 ASR versus Byzantine Percentage

Figure 4 depicts the ASRs for the scaling attack across a spectrum of Byzantine client percentages, ranging from



(a) MNIST (b) Fashion-MNIST

Figure 4. ASR versus Byzantine client percentage.



(a) MNIST. (b) Fashion-MNIST.

Figure 5. ASR versus non-IID degree.

20% to 80% (their topologies are in supplementary due to page limitation). It is evident that the ASRs of baseline schemes exhibit a clear upward trend, which reveals their inadequacy in effectively identifying and excluding a relatively substantial number of Byzantine clients. In contrast, DFL-Dual consistently maintains a low ASR, even when 80% of clients are compromised (note that the remaining two benign clients are connected in this extreme scenario). This resilient performance highlights DFL-Dual's superior capability in navigating adversarial environments characterized by a multitude of malicious clients.

5.6.2 ASR versus Non-IID Degree

Figure 5 illustrates the relationship between the ASRs against the considered defense schemes and the degree of non-IID data distribution (as indicated by the probability value p). The results reveal that, for baseline methods, the ASR increases with a higher degree of non-IID. This is attributed to the amplified divergences between benign local model updates, making it challenging to distinguish whether the outlying local model updates stem from Byzantine attacks or non-IID data distribution. Surprisingly, the ASR remains low under DFL-Dual, even when $p = 0.8$. In all cases, our proposed DFL-Dual consistently outperforms the baselines, achieving a lower ASR.

5.7. Ablation Study

To comprehensively assess the significance of considering both model-domain and data-domain distances in clustering clients and incorporating the trust bootstrapping mechanism for guiding cluster selection to identify malicious local

Atk / Def	Scaling			A Little is Enough			DBA		
	MNIST	Fashion	CIFAR10	MNIST	Fashion	CIFAR10	MNIST	Fashion	CIFAR10
w/o ED	**96.82/0.36**	**85.16/1.49**	36.85/31.61	93.16/1.07	85.65/57.12	50.01/4.29	37.21/18.99	62.14/48.91	41.33/9.01
w/o WD	86.49/0.48	14.73/21.02	49.69/4.59	95.05/95.76	79.84/56.93	**50.75/3.83**	12.38/52.92	54.56/47.84	**49.64/3.76**
w/o TB	75.60/95.23	30.99/46.31	49.74/4.67	96.62/31.15	84.42/42.68	50.43/4.08	50.97/74.23	69.40/49.19	47.78/4.42
DFL-Dual	96.21/0.50	84.83/1.70	**49.01/4.44**	**95.59/0.55**	**83.88/2.16**	50.01/4.78	**96.54/0.48**	**83.38/2.53**	48.79/4.35

Table 3. Ablation Study on Accuracies/ASRs (%) under Targeted Attacks.

models, we conduct ablation studies on our proposed DFL-Dual framework. We examine three variants:

- DFL-Dual-w/o-ED, where only data-domain WD are employed for client clustering in the first stage;
- DFL-Dual-w/o-WD, where only model-domain ED are utilized for client clustering in the first stage;
- DFL-Dual-w/o-TB, omitting the trust bootstrapping mechanism for cluster selection. Instead, the cluster with a lower average dual-domain distance is selected.

Table 3 presents the accuracies and ASRs of DFL-Dual and its variants against three targeted attacks. DFL-Dual consistently exhibits high accuracies and low ASRs against the evaluated targeted attacks. In contrast, each of the three variants fails to defend against at least one targeted attack. Thus, our findings affirm the efficacy of each technical design individually, emphasizing that their combination yields a more robust defense against adversarial scenarios.

5.8. Defense against Adaptive Attacks

Finally, we consider a more practical and rigorous adversarial scenario where each Byzantine client has access to all benign clients' *pre-aggregation local models* in each training round and knows the adopted distance metrics in DFL-Dual. Hence, they can conduct adaptive attacks. In this work, we formulate an adaptive attack by adding a regularization term to the loss function of Byzantine clients, which enables them to launch stealthy attacks from the perspectives of the three distance metrics (i.e., ED, CS, and WD). Specifically, we modify the loss function of each Byzantine client k in round t as:

$$\min_{\boldsymbol{\theta}_k^t} \beta \mathcal{L}_k + (1-\beta)\left(E(\hat{\boldsymbol{\theta}}^t, \boldsymbol{\theta}_k^t) + S(\hat{\boldsymbol{\theta}}^t, \boldsymbol{\theta}_k^t) + E(\hat{g}^t, g_k^t)\right), \tag{16}$$

where $\mathcal{L}_k$ is the original loss of Byzantine client k, β is the adaptive factor to balance attack strength and stealthiness. $E\left(\hat{\boldsymbol{\theta}}^t, \boldsymbol{\theta}_k^t\right)$ and $S\left(\hat{\boldsymbol{\theta}}^t, \boldsymbol{\theta}_k^t\right)$ are the ED and CS between the average of all benign clients' local models $\hat{\boldsymbol{\theta}}^t$ and the malicious model $\boldsymbol{\theta}_k^t$, and $E\left(\hat{g}^t, g_k^t\right)$ is the ED between the average of estimated benign gradient $\hat{g}^t \approx \hat{\boldsymbol{\theta}}^t - \hat{\boldsymbol{\theta}}^{t+1/2}$ and the Byzantine gradient g_k^t. Based on (5), we use $E\left(\hat{g}^t, g_k^t\right)$ to regularize WD between the corresponding generated dummy datasets indirectly. Given its three adopted metrics (ED, CS, and WD), these three terms are incorporated to bypass DFL-Dual. Figure 6 shows the accuracy and ASR

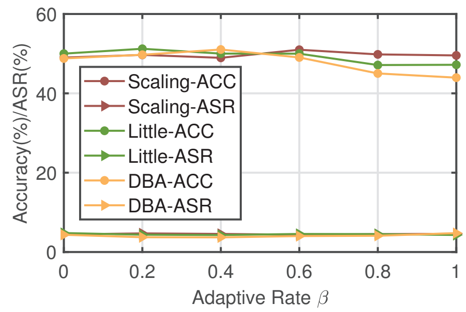


Figure 6. Accuracy/ASR versus adaptive rate.

of DFL-Dual on CIFAR-10 versus adaptive rate β of three adaptive targeted attacks, where we can find a consistent solid performance. This further verifies the robustness and effectiveness of the proposed DFL-Dual method.

6. Conclusion

This paper presented DFL-Dual, a novel Byzantine-robust DFL framework through dual-domain client clustering and trust bootstrapping. DFL-Dual leverages multiple distance metrics in the model domain (cosine similarity and Euclidean distance) and the data domain (Wasserstein distance) to identify client disparities. This multi-metric combination enables accurate discrimination between Byzantine and benign clients, even under a rigorous adversary setting with highly non-IID data distribution and exceeding 50% Byzantine clients dominating both the entire client population and a benign client's neighbors. We conduct an extensive experimental evaluation of DFL-Dual. The results validate its superior defensive performance against untargeted and targeted Byzantine attacks over existing schemes.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grants No. 62102337, 62203309, 62122066, and U20A20182), Young Elite Scientists Sponsorship Program by CAST (Grant No. 2023QNRC001), the Natural Science Foundation of Hunan Province (Grant No. 2023JJ40174), Shenzhen Science and Technology Program (Grant No. RCBS20221008093312031), the Fundamental Research Funds for the Central Universities, and Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020. 6

[2] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019. 6

[3] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state-of-the-art, frameworks, trends, and challenges. *arXiv preprint arXiv:2211.08413*, 2022. 1

[4] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017. 2, 4, 6

[5] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLtrust: Byzantine-robust federated learning via trust bootstrapping. In *NDSS*, 2020. 1, 2, 4, 6

[6] Jin-Hua Chen, Min-Rong Chen, Guo-Qiang Zeng, and Jia-Si Weng. Bdfl: A byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle. *IEEE Transactions on Vehicular Technology*, 70(9):8639–8652, 2021. 2

[7] Jian-hui Duan, Wenzhong Li, Derun Zou, Ruichen Li, and Sanglu Lu. Federated learning with data-agnostic distribution fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8074–8083, 2023. 1

[8] Cheng Fang, Zhixiong Yang, and Waheed U Bajwa. Bridge: Byzantine-resilient decentralized gradient descent. *IEEE Transactions on Signal and Information Processing over Networks*, 8:610–626, 2022. 2, 3, 4, 6

[9] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622, 2020. 1, 2, 6

[10] Chaoyang He, Emir Ceyani, Keshav Balasubramanian, Murali Annavaram, and Salman Avestimehr. Spreadgnn: Decentralized multi-task federated learning for graph neural networks on molecular data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6865–6873, 2022. 1

[11] Siquan Huang, Yijiang Li, Chong Chen, Leyu Shi, and Ying Gao. Multi-metrics adaptively identifies backdoors in federated learning. *arXiv preprint arXiv:2303.06601*, 2023. 4

[12] Shivam Kalra, Junfeng Wen, Jesse C Cresswell, Maksims Volkovs, and HR Tizhoosh. Decentralized federated learning through proxy model sharing. *Nature Communications*, 14 (1):2899, 2023. 1

[13] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4):1, 2010. 6

[14] Yann LeCun, Corinna Cortes, and Chris Burges. Mnist handwritten digit database, 1998. *URL http://www. research. att. com/yann/ocr/mnist*, 7, 1998. 6

[15] Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10132–10142, 2022. 1

[16] Frank Po-Chen Lin, Seyyedali Hosseinalipour, Sheikh Shams Azam, Christopher G Brinton, and Nicolo Michelusi. Semi-decentralized federated learning with cooperative d2d local model aggregations. *IEEE Journal on Selected Areas in Communications*, 39(12):3851–3869, 2021. 1

[17] Bo Liu and Zhengtao Ding. Distributed heuristic adaptive neural networks with variance reduction in switching graphs. *IEEE Transactions on Cybernetics*, 51(7):3836–3844, 2021. 1

[18] Bo Liu and Zhengtao Ding. A consensus-based decentralized training algorithm for deep neural networks with communication compression. *Neurocomputing*, 440:287–296, 2021. 1

[19] Bo Liu, Zhengtao Ding, and Chen Lv. Distributed training for multi-layer neural networks by consensus. *IEEE Transactions on Neural Networks and Learning Systems*, 31(5): 1771–1778, 2020. 2, 6

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 1, 2

[21] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 27–38, 2017. 6

[22] Yuben Qu, Haipeng Dai, Yan Zhuang, Jiafa Chen, Chao Dong, Fan Wu, and Song Guo. Decentralized federated learning for uav networks: Architecture, challenges, and opportunities. *IEEE Network*, 35(6):156–162, 2021. 1

[23] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*, 2021. 1

[24] Yifan Shi, Yingqi Liu, Kang Wei, Li Shen, Xueqian Wang, and Dacheng Tao. Make landscape flatter in differentially private federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24552–24562, 2023. 1

[25] Zhenheng Tang, Shaohuai Shi, Bo Li, and Xiaowen Chu. Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication. *IEEE Transactions on Parallel and Distributed Systems*, 34(3):909–922, 2022. 1

[26] Nurbek Tastan and Karthik Nandakumar. Capride learning: Confidential and private decentralized learning based on encryption-friendly distillation loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8084–8092, 2023. 1

[27] Joost Verbraeken, Martijn de Vos, and Johan Pouwelse. Bristle: Decentralized federated learning in byzantine, non-iid environments. *arXiv preprint arXiv:2110.11006*, 2021. 2, 3

[28] Wei Wan, Shengshan Hu, Minghui Li, Jianrong Lu, Longling Zhang, Leo Yu Zhang, and Hai Jin. A four-pronged defense against byzantine attacks in federated learning. *arXiv preprint arXiv:2308.03331*, 2023. 1

[29] Lilian Weng. From gan to wgan. *arXiv preprint arXiv:1904.08994*, 2019. 4

[30] Zhaoxian Wu, Tianyi Chen, and Qing Ling. Byzantine-resilient decentralized stochastic optimization with robust aggregation rules. *IEEE Transactions on Signal Processing*, 2023. 3, 4, 6

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[32] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019. 6

[33] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*, pages 6893–6901. PMLR, 2019. 2

[34] Gang Yan, Hao Wang, Xu Yuan, and Jian Li. Defl: Defending against model poisoning attacks in federated learning via critical learning periods awareness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10711–10719, 2023. 1

[35] Hao Ye, Le Liang, and Geoffrey Ye Li. Decentralized federated learning with unreliable communications. *IEEE Journal of Selected Topics in Signal Processing*, 16(3):487–500, 2022. 1, 2

[36] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018. 2

[37] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2545–2555, 2022. 2, 3, 6

[38] Bo Zhao, Peng Sun, Tao Wang, and Keyu Jiang. Fedinv: Byzantine-robust federated learning by inversing local model updates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9171–9179, 2022. 1, 6

[39] Joshua C Zhao, Ahmed Roushdy Elkordy, Atul Sharma, Yahya H Ezzeldin, Salman Avestimehr, and Saurabh Bagchi. The resource problem of using linear layer leakage attack in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2023. 1

[40] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32, 2019. 4