28 Sep 2023

Is Performance Fairness Achievable In Presence Of Attackers Under Federated Learning?

Ashish Gupta

George Markowsky
Missouri University of Science and Technology, markov@mst.edu

Sajal K. Das
Missouri University of Science and Technology, sdas@mst.edu

Recommended Citation

A. Gupta et al., "Is Performance Fairness Achievable In Presence Of Attackers Under Federated Learning?," *Frontiers in Artificial Intelligence and Applications*, vol. 372, pp. 948 - 955, IOS Press, Sep 2023.
The definitive version is available at https://doi.org/10.3233/FAIA230365

ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230365

Is Performance Fairness Achievable in Presence of Attackers Under Federated Learning?

Ashish Gupta [a;*], **George Markowsky**[b] **and Sajal K. Das**[a]

[a]Missouri University of Science and Technology, Rolla, United States
[b]Kennesaw State University, Marietta, United States
ashish.gupta@mst.edu, markowsky@gmail.com, sdas@mst.edu

Abstract. In the last few years, Federated Learning (FL) has received extensive attention from the research community because of its capability for privacy-preserving, collaborative learning from heterogeneous data sources. Most FL studies focus on either average performance improvement or the robustness to attacks, while some attempt to solve both jointly. However, the performance disparities across clients in the presence of attackers have largely been unexplored. In this work, we propose a novel *Fair Federated Learning* scheme with *Attacker Detection* capability (abbreviated as FFL+AD) to minimize performance discrepancies across benign participants. FFL+AD enables the server to identify attackers and learn their malign intent (e.g., targeted label) by investigating suspected models via *top performers*. This two-step detection method helps reduce false positives. Later, we introduce fairness by regularizing the benign clients' local objectives with a variable boosting parameter that gives more emphasis on low performers in optimization. Under standard assumptions, FFL+AD exhibits a convergence rate similar to FedAvg. Experimental results show that our scheme builds a more fair and more robust model, under label-flipping and backdoor attackers, compared to prior schemes. FFL+AD achieves competitive accuracy even when 40% of the clients are attackers.

1 Introduction

The evolution of machine learning techniques led to the development of Federated Learning (FL), which enables collaborative training of a model from distributed data sources (users or clients) in a privacy-preserving manner [29]. Prior studies illustrated the usefulness of FL in applications such as next word prediction [14], emoji prediction [34], and visual object detection [27]. In general, FL aims to produce a highly accurate model by aggregating local models that are trained and fine-tuned over distributed clients' private data. However, due to heterogeneity and non-independent identically distributed (non-IID) data, it is important to satisfy fairness constraints to limit the performance disparities (variation) across clients [18, 43]. We present Figure 1 to give an example of high-performance variance in spite of the average accuracy being more than 85%. The underlying FL model (consisting of 4 deep layers) is essentially a digit classification model trained on the MNIST dataset [20]. Besides the fairness, the robustness against model poisoning attacks (e.g., label-flipping and backdoor [39, 33]) is also an important aspect to be taken care of, especially when FL is deployed in the wild.

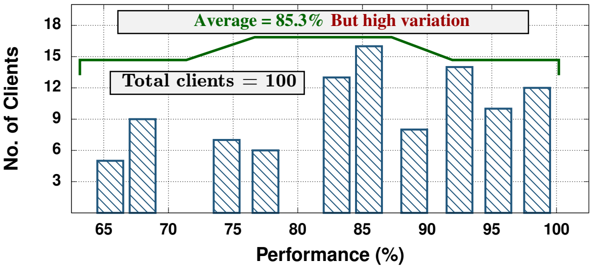


Figure 1: Illustration of the performance disparities of an FL model trained across 100 clients.

Prior FL studies have addressed both fairness and robustness issues, but separately. In particular, the agnostic federated learning (AFL) algorithms are proposed in [31, 9] that maximize the performance of worst performers to minimize disparities. The works in [7, 8] adopted a min-max optimization framework to achieve performance consistency. In the same line, q-Fair Federated Learning (q-FFL) [23] up-weighted empirical loss by its qth-power into the local objective of the clients. From the robustness standpoint, there also exist algorithms, such as Median [41], Krum and Multi-Krum [5], FoolsGold [11], Contra [1], and FedAvg-RLR [33], that aim to exclude poisonous models (from aggregation) uploaded by attackers.

Motivation: This research focuses mainly on producing a fair and robust model by addressing the following limitations of existing works: (i) Prior fairness schemes such as q-FFL and AFL, fail to reduce performance disparities when some participants are compromised by adversaries; in such cases, these schemes may accidentally boost up the attackers (if they are low-performers), which in turn would result in even higher discrepancies across benign clients. In fact, these schemes can easily overfit the adversary's objective, resulting in a highly corrupted model. (ii) On the flip side, the robust FL methods (Krum, FoolsGold, and FedAvg-RLR) can not produce a fair enough model as they aim to improve average performance (not to minimize variance) by mitigating the attackers' impact. (iii) Though the model personalization algorithms [22, 16, 28] have addressed both issues jointly to some extent, they impose high computational overheads at the client's devices as each device needs to compute two models (one personalized and one shared).

In this paper, we address the question: *Is fairness in FL achievable in the presence of attackers?* To answer, we propose a ***F****air* ***F****ederated* ***L****earning scheme with* ***A****ttacker* ***D****etection* capability

[*] Corresponding Author. Email: ashish.gupta@mst.edu

(FFL+AD). We clarify that our scheme tightly intertwines the attacker detection steps with FFL to address the problem that can not be otherwise solved by a linear combination of any existing fairness and defense solutions. The major contributions of this paper are:

- By proposing FFL+AD, we facilitate an FL framework to train a fair and accurate global model while detecting and isolating malign local models prior to aggregation at the server.
- FFL+AD incorporates a novel *two-step attacker detection* method that first identifies suspected models, by employing $\mathcal{K}-medoid$ clustering, and then they undergo an investigation (carried out by *top performers*) to ascertain their malign intent. The latter step helps minimize false detection and enables the server to learn attackers' common interests (i.e., targeted labels).
- We introduce a regularization-based scheme to minimize the performance disparities, thereby producing a fair model. This scheme essentially *boosts benign clients (excluding top performers) by a variable factor* computed with the help of top performers.
- Our theoretical analysis shows that the convergence rate of FFL+AD is similar to FedAvg under the standard assumptions. Our scheme builds a more fair and robust model compared to the prior algorithms, even when 40% of the total clients are malicious.

The organization of the paper is as follows. Section 2 discusses relevant prior works and Section 3 describes the problem setup including the challenge and objective. Section 4 proposes the FFL+AD scheme with its convergence analysis. In Section 5, we evaluate the solution empirically using two benchmark image classification datasets. Finally, Section 6 concludes the paper.

2 Related Work

FL has been an active area of research from its inception, and therefore, there exist several studies on fairness and attacker detection. We discuss some of the notable contributions here.

Fairness in FL: Though fairness received significant attention [4, 19, 44, 15] long before the inception of FL, its fundamental goal was restricted to alleviate model overfitting and improve generalization in the centralized learning framework. However, achieving fairness is even more crucial in FL due to the heterogeneous nature of the clients. To achieve this, prior approaches (q-FFL, AFL, etc.) employed min-max optimization [31, 8, 7, 9] and boosting worst-performers [23, 21, 43, 25]. In both types of approaches, FL training requires many global update rounds to reach an optimal model, and the convergence speed gets even slower in the presence of attackers because of unknowingly boosting the attackers' models.

Different from our goal, some previous studies revolve around other notions of fairness. For example, the research in [17, 45, 25] investigated fairness from a client selection perspective by formulating an optimization problem incorporating a trade-off between long-term fairness and short-term accuracy. The FairAvg [30] algorithm focused on fair aggregation by treating all clients equally regardless of the amount of data they possess.

Attacker detection in FL: In the literature on robust FL, there are broadly two types of insider attackers, targeted and untargeted, whose detection led to the existence of several solutions. As untargeted attackers aim to corrupt the whole model, via uploading random weights or sign-flipping, they can be easily defended by using strategic aggregation algorithms [41, 6, 5, 12] that either clip or remove the gradients which are far from the median model. Such an aggregation (e.g., Krum), however, may produce an unfair model by excluding informative updates. On the other hand, targeted attackers (e.g., label-flipping and backdoor) aim to corrupt only some targeted samples [37, 11, 35, 39], however, their detection is harder because they maintain the overall performance of the model on other classes [33]. FoolsGold [11] and FedAvg-RLR [33] can effectively mitigate the impact of targeted attackers but fail to ensure fairness.

Some efforts have also been made to jointly achieve fairness and robustness [16, 22]. Considering data heterogeneity as a root cause for unfairness, in [22] and [28], the authors have developed a personalized model for each client, which inherently inhibits both properties. However, such solutions do not produce any single global model, thus deviating from the design principles of FL. In contrast, we aim to train only one fair and robust global model for all clients.

3 Problem Description

Given an FL system with a set of clients $\mathcal{C} = \{c_1, c_2, \cdots, c_K\}$ and a local dataset with each client, the server aims to build a global model by aggregating local models $w_1, w_2, \cdots, w_K$ (received from K clients) over multiple rounds. Let $w_k \in \mathbf{W}$ be a set of weights defining a mapping function $h : \mathcal{X} \to \mathcal{Y}$, $k \in \{1, 2, \cdots, K\}$, and $\mathbf{W}$ is a parameter space. By minimizing the average loss over all clients, the server aims to learn an optimal model

$$w^* = \underset{w}{\text{argmin}}\, F(w) \triangleq \sum_{k=1}^{K} p_k F_k(w), \tag{1}$$

where $F_k(w)$ is the local objective function of client c_k, $p_k = n_k / \sum_k n_k$, $\sum_k p_k = 1$, and n_k denotes the number of samples with the client c_k. Let the dataset stored at c_k be $\mathcal{D}_k = \{(x_{k,j}, y_{k,j})\}_{j=1}^{n_k}$, where x and y respectively belongs to input space $\mathcal{X}$ and output space $\mathcal{Y}$. By using the dataset $\mathcal{D}_k$, $F_k(w)$ is given as

$$F_k(w) \triangleq \frac{1}{n_k} \sum_{j=1}^{n_k} l(h_w(x_{k,j}), y_{k,j}), \tag{2}$$

where $l(\cdot,\cdot)$ is an underlying loss function. At any round τ, once the global model is received by clients, they employ Stochastic Gradient Descent (SGD) with η^τ learning rate to update the model as

$$\forall k \in [K] \text{ in parallel, } w_k^{\tau+1} \leftarrow w^\tau - \eta^\tau \nabla F_k(w^\tau). \tag{3}$$

Later, using a standard FedAvg algorithm [29], the local models are aggregated at the server to compute a global model for the next round as $w^{\tau+1} = \sum_k p_k w_k^{\tau+1}$.

3.1 Defining Fairness

Due to the participation of unconditional clients with non-IID data, achieving a uniform test performance among the clients is a challenging task. To quantify fairness, we adopt the following definition.

Definition 1. *Let $\rho_k(w)$ denote the testing performance (i.e., average accuracy over all classes) of a global model w achieved by client c_k on its local dataset. Given two models w and w', the model w is said to be more fair than w' if $Var(\{\rho_k(w)\}_{1\leq k\leq K}) < Var(\{\rho_k(w')\}_{1\leq k\leq K})$, where Var is variance. It essentially means that a model that yields smaller performance disparities across participants is fair. We measure fairness only across benign devices.*

Note that in our implementation, $\rho_k(w)$ is a set of class-wise accuracies and we refer to the performance as an average accuracy unless mentioned specifically. According to FL literature [31, 23, 22], Definition 1 has been widely adopted for fairness. However, this notion of fairness is different from the studies [17, 45] where fairness refers to a fair selection of clients during aggregation, which has no relation with performance discrepancies. Another set of traditional definitions sought fairness from the lens of demographic disparities in centralized systems [10, 32] and in FL [7].

3.2 Fairness in the Presence of Attackers

When FL systems are deployed in the wild, they become susceptible to insider attackers who intend to corrupt the model. We consider only targeted attackers who are interested in reducing model performance on a specific task while keeping the overall performance almost unchanged [3], thereby they can avoid suspicion and gain trust from the server; consequently, their detection becomes challenging.
Attacker's capability: An attacker has access to the local data of a compromised client and is aware of the server's defense strategy. The attackers are *non-persistent*, meaning they do not play attack in every FL round, and thus the history-based defense mechanisms such as [11, 12] would not be useful anymore. In our FL setup, the attacker can inject poison through *label-flipping or backdoor attack* on the targeted samples in the local dataset before updating the model.
Challenge: Under the considered targeted attacks, the global model would learn wrong mapping for the targeted label, causing an overall accuracy drop on benign clients, and in turn, performance variance would increase across all the clients. Due to the non-persistent nature of attackers, they are hard to track and may elevate performance disparities even further if left undetected in early rounds.
Objective: We aim to learn a fair global model in the existence of insider attackers. Specifically, the model must satisfy the fairness constraint (stated in Definition 1) across benign clients as the training progresses, and meanwhile, the server should isolate adversaries and mitigate their influence on the model. Now, the objective of the server, expressed in Eq. 1, can be re-written as

$$w^* = \underset{w}{\text{argmin}}\ F(w) \triangleq \sum_{k=1}^{K_{ben}} p_k F_k(w), \tag{4}$$
$$s.t.\ \ Var\{\rho_k(w^*)\}_{k=1}^{K_{ben}} < Var\{\rho_k(w')\}_{k=1}^{K_{ben}}, \forall w'(\neq w^*) \in \mathbf{W}$$

where $K_{ben} \leq K$ denotes the number of benign client. Each client sends its local model to the server at scheduled intervals. The communication-related issues are beyond the scope of this work.

4 Proposed Solution: FFL+AD

This section presents our solution scheme, FFL+AD, to aid the FL server in learning a fair and robust model. There are two major tasks: (i) how to separate out poisonous local models? (ii) how to minimize performance disparities across benign clients? While accomplishing the tasks, our solution makes the following assumptions:

- Each client receives a boosting factor (denoted by β) along with the global model at every round.
- Before retraining, each client calculates the performance of the global model on the local data, which is exchanged with the server along with the updated local model and training loss.
- All the attackers inject corruption into the same class label or samples. The number of attackers is always less than benign ones, which is true in real-world scenarios.

Note that the above assumptions abide by the design principles of FL and do not incur any additional communication costs. For better understanding, we present Figure 2 illustrating a single round communication between the server and a client c_k. At round $\tau = 1$, the server initiates FL by sending a global model with random weights w^τ and $\beta_k = 0$ to client c_k for $k \in 1, 2, \cdots, K$. The client c_k would upload the performance $\rho_k(w^\tau)$, updated model w_k^τ, and training loss L_k to the server. The value of β_k is determined based on the spread of loss across benign clients. For notational brevity, we use ρ_k in place of $\rho_k(w^\tau)$, which essentially includes class-wise accuracy $\{a_1, a_2, \cdots, a_l\}$ for all l class labels.

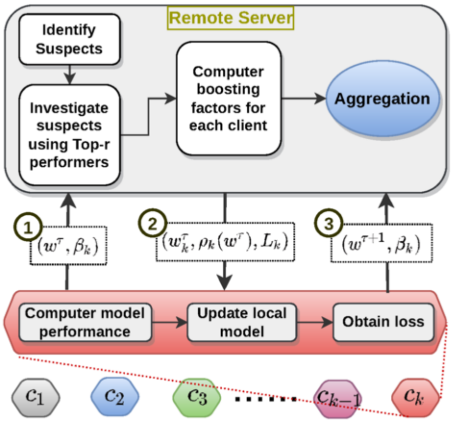


Figure 2: Overview of FFL+AD with its steps for any single round communication between the server and a client c_k.

4.1 Identify Suspicious Clients

Let $w_1^\tau, w_2^\tau, \cdots, w_K^\tau$ be the received models at round τ. By exploiting the fact that targeted attackers are closer to each other than the benign clients, we compute pair-wise Euclidean distance between every pair of clients c_i and c_j, and obtain a distance matrix D as

$$D[i,j] = D[j,i] = dist(w_i^\tau, w_j^\tau), \tag{5}$$

where $1 \leq i < j \leq K$. We preferred Euclidean distance over cosine similarity because the targeted attackers share a common target and their local models (i.e., the parameters) stay closer (in the Euclidean space) to each other compared to the benign clients. The cosine similarity could be a better choice if the untargeted attackers, such as sign-flipping, also participate in FL. We employ $\mathcal{K} - medoid$ clustering with $\mathcal{K} = 2$ to partition the models into two groups. Unlike traditional clustering, we supply pre-computed D and two farthest clients (found using D) as initial centroids to achieve faster convergence. We clarify that the matrix D is "pre-computed" because it does not need to be recomputed *during clustering in the same round*, however, it is still to be recomputed at the beginning of each round. By leveraging the stated assumptions, our scheme FFL+AD annotates the clients of the smaller group as suspicious participants.

Remark 1. *Since there are high chances of false positive attackers due to non-IID data, the suspected models are still to be investigated.*

4.2 Investigate Suspected Models

This step is crucial to avoid false positives that may occur due to the closeness of some benign clients to attackers. Thus, in FFL+AD, we recruit top-r performers (in experiments, $r = 10\%$ of the total clients) for verifying the suspected clients before excluding their models from aggregation. Top performers are picked from the pool of non-suspected clients based on their reported performance values. Note that the list of top-r clients is subject to change in every round.

Let S be the number of suspected models. The server investigates these models by executing the following steps:

1. At first, it assigns each of S models to a top performer in a round-robin manner. Some clients may receive more than one model when $S >$ top-r or may not receive any when $S <$ top-r.
2. Each top performer evaluates the model on its local dataset and uploads class-wise accuracy back to the server. For any model $s \in [S]$, let $\rho_s = \{a_1, a_2, \cdots, a_l\}$ be the class-wise performance reported by the suspected client and $\widehat{\rho}_s = \{\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_l\}$ be the performance reported by a top performer.
3. For class label j and model $s \in [S]$, if $a_j - \hat{a}_j > \phi$ then the class j is marked as a dirty label (which might be attacked). Such labels are verified across all s. Later, the class that was marked dirty by the majority of the models is treated as a targeted attacked label. And the model showing such label as suspicion is an attacker.

Computing ϕ: Intuitively, we know that the performance of a client can vary within the range from highest performer to lowest. From the pool of non-suspected clients, let c_k and $c_{k'}$ be the highest and lowest performing clients, respectively, and their respective reported class-wise accuracy values (uploaded along with the retrained model, see Fig. 2) are $\rho_k = \{a_1, a_2, \cdots, a_l\}$ and $\rho_{k'} = \{a_1^{'}, a_2^{'}, \cdots, a_l^{'}\}$. Now, we can compute a threshold

$$\phi = \max_{1 \le j \le l} \{|a_j^{'} - a_j|\}, \tag{6}$$

with $\phi \to 0$ as the training progresses.

Remark 2. *Our FFl+AD scheme is robust to the strategic attackers who also report fake high performance. However, this fake behavior reveals their malign nature when their models are found suspicious.*

Remark 3. *With two steps (Sections 4.1 and 4.2) process, FFL+AD does not rely on the history of clients, thus it can capture non-persistent attackers without fail, which makes our scheme more robust and unique.*

Remark 4. *By identifying the attacked class label, FFL+AD unfolds the attackers' interest or target, which helps to learn the susceptible class labels in the given context.*

4.3 Fairness through Variable Boosting

Upon isolating attackers, we present a novel variable-boosting strategy aiming to minimize disparities among participants without diminishing the overall accuracy of the global model. The idea is to boost low-performing clients dynamically through *regularization*.

Let K_{ben} denote the number of benign clients and $L_k \in \mathbb{R}$ is the training loss (error) obtained by client c_k where $k \in [K_{ben}]$. At every round $\tau \ge 2$, the server first computes mean loss across all top-r performers, and then for each client, it calculates a variable boosting factor as

$$\beta_k = \begin{cases} \left| \frac{\sum_{i=1}^{\text{top-r}} L_i}{\text{top-r}} - L_k \right| & \text{If } k \in [K_{ben}] \text{ and } k \notin [\text{top-r}] \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

These boosting factors are sent to the respective clients for introducing fairness by penalizing training loss. With this, the client c_k would attempt to optimize:

$$\min_{w^\tau} F_k(w^\tau) \triangleq \frac{1}{n_k} \sum_{j=1}^{n_k} l(h_{w^\tau}(x_{k,j}), y_{k,j}) + \lambda \beta_k, \tag{8}$$

where the term $\lambda\beta_k$ is a regularizer to penalize the spread of loss across the clients and $\lambda \in \mathbb{R}^+$ balances fairness and performance. Higher the value of λ, the more the fairness. We choose λ empirically from a range $[0, 5]$ with an increment of 0.5 where both performance and fairness are maximum.

Algorithm 1 FFL+AD steps for any round $\tau \ge 2$

```
Input: (i) Local models $w_1^\tau, w_2^\tau, \cdots, w_K^\tau$ at $\tau$
(ii) Performance $\rho_k = \{a_1, a_2, \cdots, a_l\}$ for $k \in \{1, 2, \cdots, K\}$
(iii) Training loss $L_k,\ \forall k$
Output: A fair global model ($w^{\tau+1}$) and $\beta_k,\ \forall k$
Compute a distance matrix $D$ using Eq. 5
Identify suspicious models using $\mathcal{K}-medoid$ with $\mathcal{K} = 2$
Set top-r $\leftarrow$ top $r\%$ performers from non-suspected clients
Compute $\phi$ using Eq. 6
/* Let $S$ denote the number of suspected models */
Assign each model $s \in [S]$ to a top-r client
/* Let $\rho_s = \{a_1, a_2, \cdots, a_l\}$ and $\widehat{\rho}_s = \{\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_l\}$ be the performance reported by the suspected client and predicted by a top performer, respectively */
Set dirty-labels $\leftarrow$ []
for each model $s \in [S]$ do
    dirty-labels[$s$] $\leftarrow$ []
    for $j = 1$ to $l$ do
        if $a_j - \hat{a}_j > \phi$ then
            dirty-labels[$s$] $\leftarrow$ dirty-labels[$s$] $\cup$ $j$
attacked-label = $\underset{j}{\text{argmax}}$ {count(dirty-labels[s][j])}, $\forall s, j$
for each $s \in [S]$ do
    if attacked-label $\in$ dirty-labels[$s$] then
        Mark the client (who sent model $s$) as an attacker.
/* Let $K_{ben}$ be the number of benign clients */
Compute aggregated model $w^{\tau+1} = \sum_{k=1}^{K_{ben}} p_k w_k^{\tau+1}$
for $k = 1$ to $K$ do
    Compute boosting factor $\beta_k$ for client $c_k$ using Eq. 7
return $w^{\tau+1}$ along with $\beta_k$ to $c_k$
Note: Each client $c_k$ obtains $w_k^{\tau+1}$ using Eq. 8 and uploads to the server for the next round $\tau + 1$.
```

Optimization: Given the boosting factors, at any round τ, the global objective (defined in Eq. 4) translates to the following:

$$\min_{w^\tau} F(w^\tau) \triangleq \sum_{k=1}^{K_{ben}} p_k F_k(w^\tau) + \sum_{i=1}^{K_{ben}-\text{top-r}} \lambda \beta_i. \tag{9}$$

By limiting the boosting only to benign clients, the suspected clients never get boosted, thus avoiding the adverse effect that may occur due to the boosting of likely attackers. The value of boosting term varies across the benign clients and it gets updated in every round.

Remark 5. *By sending the boosting factor (a single scalar value) as a piggyback with the global model, the fairness does not impose any communication overhead.*

Remark 6. *Higher the value of β, the more the emphasis on a client's model. As the objective of the server is to minimize the total empirical loss (defined in Eq. 9), the low-performing benign clients would receive more importance as they would incur a higher loss.*

Aggregation at round τ: Finally, the global model is computed as $w^{\tau+1} = \sum_{k=1}^{K_{ben}} p_k w_k^{\tau+1}$. By excluding the malicious models from aggregation, their effect is automatically eliminated from the global model. As τ increases, the attackers' models become apparent as they come very close to each other and their performances drop drastically on the targeted label. Algorithm 1 summarizes the steps of FFL+AD.

In a similar work, q-FFL [23], the fairness is also achieved by up-weighting the empirical loss of clients with a static factor q as $\sum_{k=1}^{K} \frac{p_k}{q+1} F_k^{q+1}(w)$. However, q-FFL may yield a worse model w^* if a low-performing client is an attacker. In addition, the reweighing of all the clients (including top-performers) may delay the convergence. Another work on the same track is AFL [31], which attempted to overcome this limitation by boosting only low-performers, however, it can not withstand insider attackers besides that it employed static boosting. In contrast, FFL+AD applies variable boosting and consistently reduces the disparities at every round, thereby producing an optimal model faster than q-FFL and AFL (see Section 5.3).

4.4 Convergence Guarantee

Considering FedAvg [29] for aggregation, the convergence guarantee of FFL+AD is similar to FedAvg with non-IID data [24] except one difference that is the influence of regularization term (see Eq. 8) on the training process for benign clients. For deriving our convergence results, we adopt Assumptions $1 - 3$ [24, 26, 42, 36], on the local objective functions F_k, where $1 \leq k \leq K$. To follow standard notations, we re-use β, but it does **not** refer to the boosting factor.

Assumption 1. *F_k is β-smooth, i.e., there exists a constant $\beta \in [0, \infty)$ such that, $\forall w, w' \in \mathbb{R}^d$,*

$$F_k(w) - F_k(w') \leq (w - w')^T \nabla F_k(w') + \frac{\beta}{2} \left\|w - w'\right\|_2^2 .$$

Assumption 2. *F_k is μ-strongly convex, i.e., there exists a constant $\mu \in [0, \infty)$ such that, $\forall w, w' \in \mathbb{R}^d$,*

$$F_k(w) - F_k(w') \geq (w - w')^T \nabla F_k(w') + \frac{\mu}{2} \left\|w - w'\right\|_2^2 .$$

Assumption 3. *For a uniformly random chosen data batch $\zeta_k^{(t)}$ from local data $\mathcal{D}_k$ at device k, the variance of stochastic gradients is bounded by a constant $\sigma > 0$, i.e.,*

$$\mathbb{E}_{\zeta_k^{(t)} \sim \mathcal{D}_k} \left\|\nabla F_k(w_k^{\tau,(t)}; \zeta_k^{(t)}) - \nabla F_k(w_k^{\tau,(t)})\right\|^2 \leq \sigma_k^2,$$

where t denotes a single local update within round τ. The norm of stochastic gradients is bounded by a constant $G > 0$,

$$\mathbb{E}_{\zeta_k^{(t)} \sim \mathcal{D}_k} \left\|\nabla F_k(w_k^{\tau,(t)}; \zeta_k^{(t)})\right\|^2 \leq G^2.$$

Degree of heterogeneity: Assume w^* and w_k^* be the optimized global and local model, respectively. The degree of heterogeneity across K devices can be quantified as

$$\Gamma = F(w^*) - \sum_{k=1}^{K} p_k F(w_k^*)$$

Large the value of Γ, the higher the degree of heterogeneity. Ideally, $\Gamma \to 0$ as the learning advances when the data is IID.

Theorem 1. *Let I denote the total number of local updates that each client performs at every FL round τ. Under Assumptions 1 to 3 with defined $\beta, \mu, \sigma_k, G, \kappa = \frac{\beta}{\mu}, \gamma = max\{8\kappa, I\}$, learning rate $\eta^{\tau,(t)} = \frac{2}{\mu(\gamma + \tau I + t)}$, and data heterogeneity Γ, the FedAvg with* **our scheme FFL+AD** *would satisfy the following*

$$\mathbb{E}[F(w^{\mathcal{T}})] - F(w^*) \leq \frac{2\kappa}{\gamma + \mathcal{T} I K} \left(\frac{A + \mathbb{E}\left\|\varphi^{\tau,(t)}\right\|_2^2 B}{\mu} + C \right),$$

$$\textit{where,} \quad A = \sum_{k=1}^{K} p_k^2 + 6L\Gamma + 8(I-1)^2 G^2,$$

$$B = \sum_{k=1}^{K} p_k^2 + 8(I-1)^2, \quad C = \frac{\mu\gamma}{4} \mathbb{E}\left\|w^1 - w^*\right\|^2 \tag{10}$$

Proof. Our proof has two main parts. The first part includes deriving new bounds similar to Assumption 3 for our scheme FFL+AD, and the latter presents the convergence guarantee using those bounds.
(i) New bounds with our scheme: Let $\varphi^{\tau,(t)}$ denote the amount of change in gradients due to the regularization term (defined in Eq. 8). Within τ-th FL round for t-th local update, the model on the client c_k is essentially updated (by expanding Eq. 3) as:

$$\begin{aligned} \nabla F_k'(w_k^{\tau,(t)}; \zeta_k^{(t)}) &= \nabla F_k(w_k^{\tau,(t)}; \zeta_k^{(t)}) + \varphi^{\tau,(t)} \\ w_k^{\tau,(t+1)} &= w_k^{\tau,(t)} - \eta^{\tau,(t)} \nabla F_k'(w_k^{\tau,(t)}; \zeta_k^{(t)}), \end{aligned} \tag{11}$$

Now, the expected distance between raw gradients (without FFL+AD) and the gradients with FFL+AD is

$$\begin{aligned} &\mathbb{E}\left\|\nabla F_k'(w_k^{\tau,(t)}; \zeta_k^{(t)}) - \nabla F_k(w_k^{\tau,(t)}; \zeta_k^{(t)})\right\|_2^2 \\ &= \mathbb{E}\left\|\varphi^{\tau,(t)}\right\|_2^2 \end{aligned} \tag{12}$$

By using Eq. 12 and norm triangle inequality, the new bound on the variance of stochastic gradients for Assumption 3 is

$$\begin{aligned} &\mathbb{E}\left\|\nabla F_k'(w_k^{\tau,(t)}; \zeta_k^{(t)}) - \nabla F_k(w_k^{\tau,(t)})\right\|^2 \\ \leq &\mathbb{E}\left\|\nabla F_k'(w_k^{\tau,(t)}; \zeta_k^{(t)}) - \nabla F_k(w_k^{\tau,(t)}; \zeta_k^{(t)})\right\|^2 \\ &+ \mathbb{E}\left\|\nabla F_k(w_k^{\tau,(t)}; \zeta_k^{(t)}) - \nabla F_k(w_k^{\tau,(t)})\right\|^2 \\ \leq &\mathbb{E}\left\|\varphi^{\tau,(t)}\right\|_2^2 + \sigma_k^2 \end{aligned} \tag{13}$$

Similarly, the new bound on the norm of gradients is

$$\begin{aligned} &\mathbb{E}\left\|\nabla F_k'(w_k^{\tau,(t)}; \zeta_k^{(t)})\right\|^2 \\ \leq &\mathbb{E}\left\|\nabla F_k'(w_k^{\tau,(t)}; \zeta_k^{(t)}) - \nabla F_k(w_k^{\tau,(t)}; \zeta_k^{(t)})\right\|^2 \\ &+ \mathbb{E}\left\|\nabla F_k(w_k^{\tau,(t)}; \zeta_k^{(t)})\right\|^2 \\ \leq &\mathbb{E}\left\|\varphi^{\tau,(t)}\right\|_2^2 + G^2 \end{aligned} \tag{14}$$

(ii) Convergence guarantee: From the FedAvg convergence (Theorem 1 in [24]), we have

$$\begin{aligned} &\mathbb{E}[F(w^{\mathcal{T}})] - F(w^*) \\ \leq &\frac{\kappa}{\gamma + T} \left(\frac{2D}{\mu} + \frac{\mu\gamma}{2} \mathbb{E}\left\|w^1 - w^*\right\|^2 \right) \end{aligned} \tag{15}$$

where, T is the total number of gradient updates across all clients to reach an optimal solution, which is equivalent to $\mathcal{T} I K$, $D = \sum_{k=1}^{K} p_k^2 \sigma_k^2 + 6L\Gamma + 8(I-1)^2 G^2$, and w^1 is an initial model at the server. Now, by substituting the value of D and applying the new bounds from Eq. 13 and 14, we can re-write Eq. 15 as

$$\begin{aligned} \leq &\frac{\kappa}{\gamma + \mathcal{T} I K} \left(\frac{2\left\{\sum_{k=1}^{K} p_k^2 (\mathbb{E}\left\|\varphi^{\tau,(t)}\right\|_2^2 + \sigma_k^2) + 6L\Gamma\right\}}{\mu} \right. \\ &\left. + \frac{2\left\{8(I-1)^2 (\mathbb{E}\left\|\varphi^{\tau,(t)}\right\|_2^2 + G^2)\right\}}{\mu} + \frac{\mu\gamma}{2} \mathbb{E}\left\|w^1 - w^*\right\|^2 \right) \\ \leq &\frac{2\kappa}{\gamma + \mathcal{T} I K} \left(\frac{A + \mathbb{E}\left\|\varphi^{\tau,(t)}\right\|_2^2 B}{\mu} + C \right), \end{aligned} \tag{16}$$

$$\text{where,}\quad A = \sum_{k=1}^{K} p_k^2 + 6L\Gamma + 8(I-1)^2G^2,$$

$$B = \sum_{k=1}^{K} p_k^2 + 8(I-1)^2, \quad C = \frac{\mu\gamma}{4}\mathbb{E}\left\|w^1 - w^*\right\|^2$$

□

Under Assumptions 1, 2, 3 and **Theorem** 1, the global model converges to a fair optimal solution w^* at a rate of $O(\frac{I}{T})$. We get this rate upon simplifying Eq. 10.

Remark 7. *Theorem 1 shows a convergence rate* $O(\frac{I}{T})$ *similar to* $O(\frac{I^2}{T})$ *in FedAvg (*T *is the total number of gradient updates across all clients); however, it is affected by the amount of change in gradients (denoted by* $\varphi^{\tau,(t)}$*) due to boosting.*

5 Experimental Evaluation

In this section, we empirically demonstrate that *fairness is achievable in the presence of insider targeted attackers*. Our scheme offers better fairness than the strong baseline algorithms, with almost no compromise on average accuracy.

5.1 *Datasets and Implementation Details*

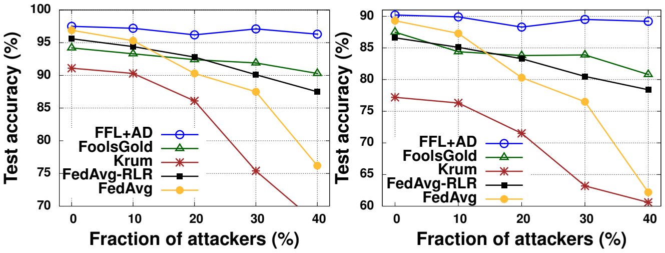
We evaluate the effectiveness of our scheme on image classification tasks using two benchmark datasets: MNIST [20] and Fashion-MNIST [40]. These datasets have been extensively used in the FL [13, 23, 31, 12]. MNIST consists of 70,000 labeled images (with dimension 28×28) of handwritten digits from $0 - 9$. We train a digit classification model comprising of 2 convolutional neural network (CNN) followed by 2 fully connected layers. Fashion-MNIST dataset also consists of 70,000 images (with dimension 28×28) belonging to 10 different categories such as “Tshirt”, “Trouser”, “Bag”, *etc.* Since its classification is more complex than MNIST, we employ a deeper model with 6 CNN and 1 fully connected layers.

Implementation details: We simulated our FL framework with a server and 100 clients, in Python programming language using PyTorch libraries. The classification model was trained for 30 communication rounds with 5 local epochs at each client in every round. The model adopts an SGD optimizer with a learning rate = 0.01 and momentum = 0.5. For non-IID data distribution, we employ Dirichlet distribution [38, 2] with a hyper-parameter $\alpha = 0.9$. The fairness balance parameter λ, used in Eq. 8, is fine-tuned over the range $[0.5, 1, 1.5, \cdots, 4.5, 5]$ for both datasets. We performed multiple experiments to choose an appropriate value of λ and set it to 3 for MNIST and 4.5 for Fashion-MNIST (see Appendix here: https://drive.google.com/file/d/1YaGpiqDvZoQexwQUpS_autOwpqF4Y91N/view?usp=sharing). The source code is available at https://github.com/agupta582/FFL-AD.

Attack simulation: For *label-flipping attackers*, we flip (replace) the labels of the images of digit “2” by “8” in MNIST and images of “Trouser” by label “Bag” in Fashion-MNIST. For *backdoor attackers*, we embedded a plus (“+”) pattern of size 5×5 pixels into the images (see Appendix). Each attacker injects corruption into 50% of the total images and modifies their labels to the targeted labels (“8” and “Bag” respectively for MNIST and Fashion-MNIST dataset.).

5.2 *Analyzing Robustness (without Fairness)*

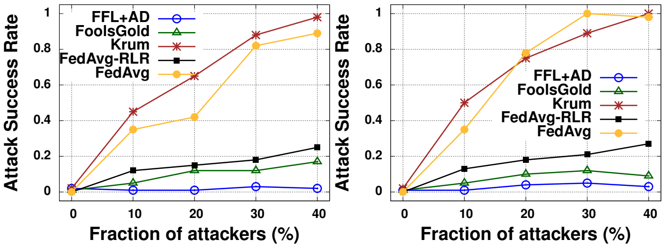
With a varying fraction of attackers (including both label-flipping and backdoor), we conduct experiments to quantify the impact of corruption on test accuracy and report the results[1] for FFL+AD, FedAvg, and strong prior defenses including FoolsGold [11], Krum [5], and FedAvg-RLR [33], in Figure 3. FFL+AD achieves steady performance regardless of the fraction of attackers. At 40% fraction, with an absolute accuracy gain of 6.5% for MNIST and 8.3% for Fashion-MNIST, FFL+AD is more robust than a highly robust scheme FoolsGold. It is important to note that FedAvg is still the best option when the corruption level of low.



(a) MNIST dataset (b) Fashion-MNIST dataset

Figure 3: Impact of attackers on the test accuracies.

Next, we report the attack success rate (**ASR**) [38] in Figure 4. ASR is calculated as $\frac{\text{\# successfully attacked samples}}{\text{\# attacked samples}}$. The lower the ASR, the better the robustness. FFL+AD consistently outperforms the prior schemes with almost negligible ASR, indicating the effectiveness at mitigating the impact of attackers on the targeted labels (i.e., “2” and “Trouser”). Although FedAvg-RLR and FoolsGold perform quite similarly to ours in some cases, they never gave better ASR. Besides that, FFL+AD also improves the fairness of the model (see Section 5.4). Though FedAvg reaches a good average accuracy up to 10% fraction of attackers, it failed on ASR badly for both datasets.



(a) On attacked label “2” (b) On attacked label “Trouser”

Figure 4: Attack success rate on the attacked label for both datasets.

5.3 *Analyzing Fairness (without Attackers)*

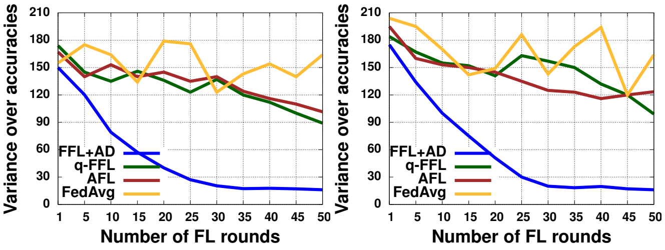
We analyze the fairness of the global model, trained over 50 rounds, based on the variance of test performances (i.e., accuracy) achieved by all 100 clients. The results, reported in Figure 5, also include two state-of-the-art algorithms, q-FFL (specifically q-FedAvg with $q = 5$ for both the datasets) [23], AFL [31], and a baseline scheme FedAvg. For the q-FFL and AFL implementation, we reused the source code from https://github.com/litian96/fair_flearn. The results clearly indicate that FFL+AD continuously improves the fairness by reducing the variance (*Var*) as training progresses and it converges after 30^{th} round with variance 17.6, whereas q-FFL and AFL could hardly drop till 112.4 and 106.5, respectively for Fashion-MNIST dataset in no-attacker case. While FFL+AD, q-FFL, and AFL show a downward

[1] Implementation of the prior defense methods including Krum and FoolsGold, is borrowed from [38].

Table 1: Accuracy and fairness comparison in the presence of targeted attackers. [Acc: Accuracy, *Var*: variance over the clients' performances]

Schemes	MNIST dataset						Fashion-MNIST dataset					
	no-attacker		**20% attackers**		**40% attackers**		**no-attacker**		**20% attackers**		**40% attackers**	
	Acc. (%)	*Var*	Acc. (%)	*Var*	Acc. (%)	*Var*	Acc. (%)	*Var*	Acc. (%)	*Var*	Acc. (%)	*Var*
q-FFL	94.6	91.5	89.3	135.7	78.1	185.3	86.2	112.4	78.9	147.8	68.1	181.6
AFL	94.2	103.2	81.9	122.6	69.3	166.5	85.2	106.5	75.3	152.5	62.3	191.6
FoolsGold	95.9	170.2	94.2	205.6	93.8	193.2	87.2	188.3	85.9	219.4	85.3	229.5
FedAvg-RLR	96.2	176.2	92.2	197.5	91.6	210.1	85.5	182.6	86.5	211.4	84.8	221.3
FedAvg	96.6	195.1	91.8	210.4	76.2	227.4	88.9	186.5	85.8	212.3	63.2	267.2
FFL+AD (Ours)	**97.2**	**20.2**	**96.9**	**24.1**	**96.9**	**28.7**	**89.2**	**17.6**	**88.9**	**22.1**	**89.4**	**23.5**

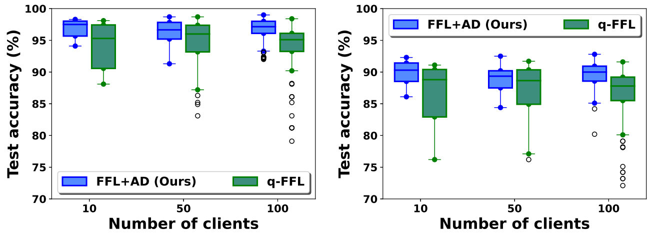
trend, FedAvg could not stabilize because it does not incorporate any fairness improvement strategy and the clients have non-IID data.



(a) MNIST dataset (b) Fashion-MNIST dataset

Figure 5: Variance of the test performances over the FL rounds. [Lower the variance, the better the scheme]

Additionally, we carried out experiments with a *varying number of clients* to analyze the spread of test accuracies, as depicted via box plots in Figure 6. Besides the gain in average accuracy, our scheme outperforms q-FFL by achieving much lesser disparities among the clients. For instance, with 100 clients for MNIST dataset, the difference between the lowest and highest performer is ≈ 8.6% for FFL+AD, which is more than 20% for q-FFL.



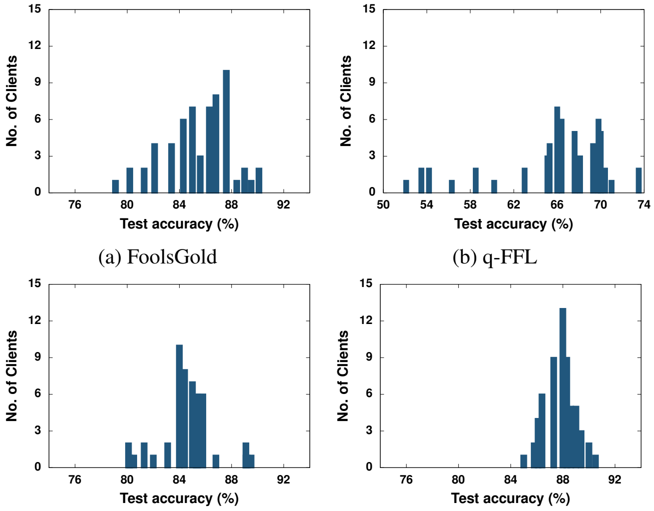
(a) MNIST dataset (b) Fashion-MNIST dataset

Figure 6: Spread of test accuracies with a varying number of clients.

5.4 *Fairness in the Presence of Attackers*

Finally, we carry out the experiments for analyzing fairness in the presence of the attackers (including both label-flipping and backdoor) and report the average test accuracy (*Acc*) and variance (*Var*) of the accuracies across all clients, in Table 1. FFL+AD outperforms (substantially) all the competitive schemes, specifically when the percentage of attackers increases. In spite of that q-FFL and AFL were originally developed to deal with the fairness issue, they show large variance values (e.g., more than 150 with 40% attackers). It is because of boosting the attackers' models unknowingly. On the other side, while focusing on robustness, FoolsGold, and FedAvg-RLR achieved comparatively good accuracy in all the cases, but they failed to yield a fair model as *Var* $>$ 190 for both datasets when the corruption level is high.

Additionally, we plot the histogram of test accuracy distribution across 60 benign clients (excluding all 40 attackers) for Fashion-MNIST dataset in Figure 7. It is clear that our scheme can achieve almost uniform distribution by squeezing the spread between 84.8% and 91.1%, though Ditto algorithm [22] (the code is borrowed from https://github.com/litian96/ditto) also managed to reduce the variance but the average accuracy dropped by around 4%. In contrast, the fair scheme q-FFL shows a much higher variation (from 52.2% to 73.7%) than other algorithms. See Appendix for more results.



(c) Ditto (d) FFL+AD (Ours)

Figure 7: Distribution of the test accuracies across 60 benign clients (excluding 40 attackers) for Fashion-MNIST dataset.

6 Conclusion

We proposed a novel fair federated learning scheme with an attacker detection method, abbreviated as FFL+AD. By isolating the targeted attackers with the help of $\mathcal{K} - medoid$ and top performers, our scheme is adept at reducing the performance disparities (across participants) to a large extent by strategically boosting the model weights of low-performers (only benign clients). We presented a novel regularization framework for the local objective function to improve fairness. Under standard assumptions, FFL+AD follows a convergence rate similar to FedAvg with new bounds. With non-IID data across 100 clients including 40% attackers, our experimental results demonstrated that FFL+AD can outperform the strong prior algorithms on fairness and attack success rate while achieving a competitive accuracy on image classification tasks. Currently, the proposed solution can detect only the attackers who share a common malign objective and must corrupt the same target label, which we plan to address in the future besides scaling the solution for non-targeted attackers and more diverse datasets such as CIFAR-100 and ImageNet.

Acknowledgements

This work is partially supported by the NSF grant award #2008878 (FLINT: Robust Federated Learning for Internet of Things) and the NSF award #2030624 (TAURUS: Towards a Unified Robust and Secure Data Driven Approach for Attack Detection in Smart Living).

References

[1] Sana Awan, Bo Luo, and Fengjun Li, 'Contra: Defending against poisoning attacks in federated learning', in *European Symposium on Research in Computer Security*, pp. 455–475. Springer, (2021).

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov, 'How to backdoor federated learning', in *Proc. AISTATS*, pp. 2938–2948, (2020).

[3] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo, 'Analyzing federated learning through an adversarial lens', in *Proc. ICML*, pp. 634–643, (2019).

[4] Peter J Bickel, Eugene A Hammel, and J William O'Connell, 'Sex bias in graduate admissions: Data from berkeley', *Science*, **187**(4175), 398–404, (1975).

[5] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer, 'Machine learning with adversaries: Byzantine tolerant gradient descent', in *Proc. NeurIPS*, pp. 118–128, (2017).

[6] Yudong Chen, Lili Su, and Jiaming Xu, 'Distributed statistical machine learning in adversarial settings: Byzantine gradient descent', *Proc. POMACS*, **1**(2), 1–25, (2017).

[7] Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang, 'Addressing algorithmic disparity and performance inconsistency in federated learning', *Advances in Neural Information Processing Systems*, **34**, 26091–26102, (2021).

[8] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi, 'Distributionally robust federated averaging', *Proc. NeurIPS*, **33**, (2020).

[9] Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong, 'Fairness-aware agnostic federated learning', in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 181–189. SIAM, (2021).

[10] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian, 'Certifying and removing disparate impact', in *Proc. SIGKDD*, pp. 259–268, (2015).

[11] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh, 'The limitations of federated learning in sybil settings', in *Proc. RAID*, pp. 301–316, (2020).

[12] Ashish Gupta, Tie Luo, Mao V Ngo, and Sajal K Das, 'Long-short history of gradients is all you need: Detecting malicious and unreliable clients in federated learning', in *European Symposium on Research in Computer Security*, pp. 445–465. Springer, (2022).

[13] Weituo Hao, Mostafa El-Khamy, Jungwon Lee, Jianyi Zhang, Kevin J Liang, Changyou Chen, and Lawrence Carin Duke, 'Towards fair federated learning with zero-shot data augmentation', in *Proc. CVPR*, pp. 3310–3319, (2021).

[14] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage, 'Federated learning for mobile keyboard prediction', *arXiv preprint arXiv:1811.03604*, (2018).

[15] Moritz Hardt, Eric Price, and Nati Srebro, 'Equality of opportunity in supervised learning', *Proc. NeurIPS*, **29**, 3315–3323, (2016).

[16] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu, 'Fedmgda+: Federated learning meets multi-objective optimization', *arXiv preprint arXiv:2006.11489*, (2020).

[17] Tiansheng Huang, Weiwei Lin, Wentai Wu, Ligang He, Keqin Li, and Albert Y Zomaya, 'An efficiency-boosting client selection scheme for federated learning with fairness guarantee', *IEEE Transactions on Parallel and Distributed Systems*, **32**(7), 1552–1564, (2020).

[18] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al., 'Advances and open problems in federated learning', *arXiv preprint arXiv:1912.04977*, (2019).

[19] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma, 'Fairness-aware learning through regularization approach', in *Proc. ICDM Workshops*, pp. 643–650, (2011).

[20] Yann LeCun, 'The mnist database of handwritten digits', *http://yann.lecun. com/exdb/mnist/*, (1998).

[21] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith, 'Tilted empirical risk minimization', in *Proc. ICLR*, (2020).

[22] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith, 'Ditto: Fair and robust federated learning through personalization', in *Proc. ICML*, pp. 6357–6368, (2021).

[23] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith, 'Fair resource allocation in federated learning', in *Proc. ICLR*, (2019).

[24] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang, 'On the convergence of fedavg on non-iid data', in *Proc. ICLR*, (2019).

[25] Zhidu Li, Yujie Zhou, Dapeng Wu, Tong Tang, and Ruyan Wang, 'Fairness-aware federated learning with unreliable links in resource-constrained internet of things', *IEEE Internet of Things Journal*, (2022).

[26] Frank Po-Chen Lin, Christopher G Brinton, and Nicolo Michelusi, 'Federated learning with communication delay in edge networks', in *Proc. IEEE GlobeCom*, pp. 1–6, (2020).

[27] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang, 'Fedvision: An online visual object detection platform powered by federated learning', in *Proc. AAAI*, volume 34, pp. 13172–13179, (2020).

[28] Jun Luo and Shandong Wu, 'Adapt to adaptation: Learning personalization for cross-silo federated learning', in *IJCAI*, pp. 1–1, (2022).

[29] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas, 'Communication-efficient learning of deep networks from decentralized data', in *Proc. AISTATS*, pp. 1273–1282, (2017).

[30] Umberto Michieli and Mete Ozay, 'Are all users treated fairly in federated learning systems?', in *Proc. CVPR*, pp. 2318–2322, (2021).

[31] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh, 'Agnostic federated learning', in *Proc. ICML*, pp. 4615–4625, (2019).

[32] Fatemeh Nargesian, Abolfazl Asudeh, and HV Jagadish, 'Tailoring data source distributions for fairness-aware data integration', *Proceedings of the VLDB Endowment*, **14**(11), 2519–2532, (2021).

[33] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel, 'Defending against backdoors in federated learning with robust learning rate', in *Proc. AAAI*, volume 35, pp. 9268–9276, (2021).

[34] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays, 'Federated learning for emoji prediction in a mobile keyboard', *arXiv preprint arXiv:1906.04329*, (2019).

[35] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein, 'Poison frogs! targeted clean-label poisoning attacks on neural networks', in *Proc. NeurIPS*, pp. 6106–6116, (2018).

[36] Sebastian U Stich, 'Local sgd converges fast and communicates little', *arXiv preprint arXiv:1805.09767*, (2018).

[37] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan, 'Can you really backdoor federated learning?', *arXiv preprint arXiv:1911.07963*, (2019).

[38] Ching Pui Wan and Qifeng Chen, 'Robust federated learning with attack-adaptive aggregation', *arXiv preprint arXiv:2102.05257*, (2021).

[39] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy yong Sohn, Kangwook Lee, and Dimitris S. Papailiopoulos, 'Attack of the tails: Yes, you really can backdoor federated learning', in *Proc. NeurIPS*, (2020).

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

[41] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett, 'Byzantine-robust distributed learning: Towards optimal statistical rates', in *Proc. ICML*, pp. 5650–5659, (2018).

[42] Hao Yu, Sen Yang, and Shenghuo Zhu, 'Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning', in *Proc. AAAI*, volume 33, pp. 5693–5700, (2019).

[43] Xubo Yue, Maher Nouiehed, and Raed Al Kontar, 'Gifair-fl: A framework for group and individual fairness in federated learning', *INFORMS Journal on Data Science*, (2022).

[44] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork, 'Learning fair representations', in *Proc. ICML*, pp. 325–333, (2013).

[45] Qunsong Zeng, Yuqing Du, Kaibin Huang, and Kin K Leung, 'Energy-efficient radio resource allocation for federated edge learning', in *Proc. ICC Workshops*, pp. 1–6, (2020).