FLAIR: Defense against Model Poisoning Attack in Federated Learning

Atul Sharma
Purdue University
West Lafayette, IN, USA
sharm438@purdue.edu

Wei Chen
Purdue University
West Lafayette, IN, USA
chen2732@purdue.edu

Joshua Zhao
Purdue University
West Lafayette, IN, USA
zhao1207@purdue.edu

Qiang Qiu
Purdue University
West Lafayette, IN, USA
qqiu@purdue.edu

Saurabh Bagchi
Purdue University
West Lafayette, IN, USA
sbagchi@purdue.edu

Somali Chaterji
Purdue University
West Lafayette, IN, USA
schaterji@purdue.edu

ABSTRACT

Federated learning—multi-party, distributed learning in a decentralized environment—is vulnerable to model poisoning attacks, more so than centralized learning. This is because malicious clients can collude and send in carefully tailored model updates to make the global model inaccurate. This motivated the development of Byzantine-resilient federated learning algorithms, such as Krum, Bulyan, FABA, and FoolsGold. However, a recently developed untargeted model poisoning attack showed that all prior defenses can be bypassed. The attack uses the intuition that simply by changing the sign of the gradient updates that the optimizer is computing, for a set of malicious clients, a model can be diverted from the optima to increase the test error rate. In this work, we develop **FLAIR**—a defense against this directed deviation attack (DDA), a state-of-the-art model poisoning attack. FLAIR is based on our intuition that in federated learning, certain patterns of gradient flips are indicative of an attack. This intuition is remarkably stable across different learning algorithms, models, and datasets. FLAIR assigns reputation scores to the participating clients based on their behavior during the training phase and then takes a weighted contribution of the clients. We show that where the existing defense baselines of FABA [IJCAI '19], FoolsGold [Usenix '20], and FLTrust [NDSS '21] fail when 20-30% of the clients are malicious, FLAIR provides byzantine-robustness upto a malicious client percentage of 45%. We also show that FLAIR provides robustness against even a white-box version of DDA.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**.

KEYWORDS

Federated learning; Model poisoning; Byzantine-robust aggregation

ASIA CCS '23, July 10–14, 2023, Melbourne, VIC, Australia
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0098-9/23/07.
https://doi.org/10.1145/3579856.3582836

ACM Reference Format:
Atul Sharma, Wei Chen, Joshua Zhao, Qiang Qiu, Saurabh Bagchi, and Somali Chaterji. 2023. FLAIR: Defense against Model Poisoning Attack in Federated Learning. In *ACM ASIA Conference on Computer and Communications Security (ASIA CCS '23), July 10–14, 2023, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3579856.3582836

1 INTRODUCTION

Federated learning (FL) [13, 17] offers a way for multiple clients on heterogeneous platforms to learn collaboratively without sharing their local data. The clients send their local gradients to the parameter server that aggregates the gradients and updates the global model for the local clients to download. FL can be attacked during the training phase by compromising a set of clients that then send maliciously crafted gradients. The attack can be targeted against particular data instances or can be untargeted. The latter brings down the overall accuracy by affecting *all* classes. In this work, we use the state-of-the-art (SOTA) class of untargeted model poisoning attacks called *directed deviation attacks (DDA)* – Fang attack [5] and Shejwalkar attack [12]. *Both have been shown to bypass all existing Byzantine-robust aggregation techniques,* e.g., *Krum, Bulyan, Trimmed mean, and Median.* In our experiments, the Shejwalkar attack has been found to decrease the test accuracy from 70% to a low 10% on ResNet-18 trained on the CIFAR-10 dataset, and from 92% to 10% on a DNN trained on MNIST using FedSGD, distributed among clients, only 20% of which are under the attacker's control, and the server is secure. We describe the relevant details of this attack in Section 2.2.

Our solution. We propose a novel defense called **FLAIR** - **F**ederated **L**earning with Embedded **A**dversarial **I**njection **R**obustness against untargeted model poisoning attacks (Figure 1), which uses a stateful model to reduce the contributions by suspicious clients to the global model update. We show that where all prior Byzantine-resilient federated learning approaches fail against the directed-deviation attack, FLAIR is able to recover the test accuracy of the trained model. This benefit applies even when the attack knows the algorithm and all the parameters of our defense, *i.e.*, an adaptive white-box attack. FLAIR is based on a simple intuition that for a well-chosen small learning rate, as the model approaches an optima in a benign setting, a large number of gradients do not flip their direction with large magnitudes, that is, a degree of inertia is maintained. Our intuition is shown graphically through a sample

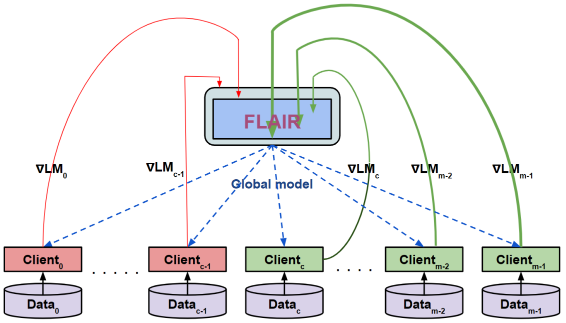

Figure 1: FLAIR's architecture where c out of m clients are malicious and send carefully crafted values of their local models to throw the global model off convergence. FLAIR weighs the gradients, received from the clients, by their reputation scores before aggregation represented by varying thicknesses of arrows from the clients.

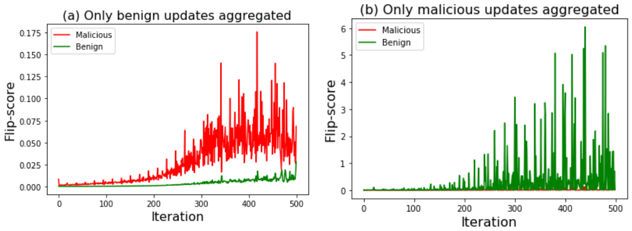

Figure 2: The average flip-score across malicious clients in red and across benign clients in green shown over time as a motivating experiment where a DNN is trained on MNIST for 500 iterations with 80 benign and 20 malicious clients. In (a), only benign updates were aggregated into the global model using FedSGD, and in (b), only malicious updates were aggregated, depicting the two extreme cases of federated learning in a malicious setting. These results show that when the global model update is benign (as in (a)), the malicious clients send gradients with high flip-scores to deviate the model from reaching convergence. However, when the global model update itself was poisoned (as in (b)), the benign clients send high flip-score gradients for recovery whereas the malicious clients maintain the direction of the already-poisoned model. Thus the intuition is that too high and too low flip-scores in a coordinated manner are red flags.

experiment in Figure 2. We capture this quantitatively in a metric that we introduce, called ***flip-score***, in Eqn. 1, which is calculated for each client in every round. It is the sum of square of gradient magnitudes of all those parameter updates which suggest a flip in the gradient direction from their previous global update. Our defense is secure by design against *any* possible untargeted model poisoning attack. The fundamental principle underlying such attacks is to generate gradients that move the model away from the estimated optima. It invariably requires coordinated flipping the gradient direction of the model parameters by multiple clients. We strike at the root cause of this class of attacks by preventing gradient flips that push the global model away from optima. We find that our intuition and correspondingly our defense FLAIR holds for all untargeted model poisoning attacks, as well as against label flipping attacks (Appendix § A.2, Figure 8). In our evaluation, we focus on the directed deviation, untargeted model poisoning attack as that has been shown in multiple publications to be the most damaging of the set mentioned above [5, 12].

In summary, FLAIR makes the following contributions.

(1) We use our simple intuition to detect malicious clients that attack federated learning using the SOTA attack model. Our intuition is that certain patterns of flips of the signs of gradients across multiple parameters and multiple clients are rare under benign conditions.
(2) We use a stateful suspicion model to keep the history of every client's activity and use that as a weighting factor in the aggregation. We theoretically prove the convergence of FLAIR that uses the weighted averaging and establish a convergence rate.
(3) We evaluate FLAIR on DNNs trained on MNIST and FEMNIST, ResNet-18 on CIFAR-10, and GRU on the Shakespeare dataset. We comparatively evaluate our defense against six baselines, including the most recent ones, FABA [16], FoolsGold [6], and FLTrust [4] and show that FLAIR remains robust even against an adaptive white-box attacker. While several of the existing defenses shine under specific configurations (combination of attacks and datasets/models), FLAIR is the only one whose protection transfers well across configurations. For example, all existing defenses fail when more than 30% of the clients are malicious, where FLAIR remains robust upto a malicious client percentage of 45%.

We release the source code, the attack scripts, the trained models, and the test harness for the evaluation at https://github.com/icanforce/federated-learning-flair. The rest of the paper is organized as follows. We describe in Sec 2 the threat model, the SOTA attack, and why all existing Byzantine-resilient federated learning approaches are susceptible. We present FLAIR's design in Sec 3 and convergence analysis in Sec 4. We describe the baselines and the datasets in Sec 5, and evaluation in Sec 6.

2 BACKGROUND

Our formulation of federated learning consists of m clients, each with its own local data, but with the same model architecture and SGD optimizer, out of which c are malicious. The parameter server is benign and secure, and assumes that a maximum of c_{max} number of clients can be malicious, $c \leq c_{max}$ in any given round. The clients run one local iteration, send their gradients (in unencrypted form) to the server, which updates the global model for the clients to download in a synchronous manner.

2.1 Byzantine-resilient Federated Learning

Here we describe the leading defenses briefly, which are all shown recently to be vulnerable to the SOTA untargeted model poisoning attacks.

(1) **FedSGD** [9] is the simplest aggregation technique and does a simple weighted mean aggregation of the gradients, weighted by the number of data samples each client holds. FedSGD can be attacked by a *single* malicious client that sends boosted malicious gradients.
(2) **Trimmed mean and Median** [18] aggregate the parameters independently, where the former trims c_{max} number of values each at the lower and higher extremes of every parameter and the latter takes the median of every parameter update across the gradients received from all the clients. The Full-Trim attack [5] is specifically designed toward these aggregations rules.
(3) **Krum** [2] selects one local model as the next global model. The client that has the lowest Euclidean distance from its closest $(m - c_{max} - 2)$ neighbors is chosen as the local model. The full-Krum attack [5] is tailored to attack Krum.
(4) **Bulyan** [10] combines the above approaches by running Krum iteratively to select a given number of models, and then running Trimmed Mean over the selected ones. The Full-Trim attack is also transferable to Bulyan.
(5) **FABA** [16] iteratively filters out models farthest away from the mean of the remaining unfiltered models, c_{max} number of times before returning the mean of the remaining gradients.
(6) **FoolsGold** [6] was motivated to defend against poisoning attacks by Sybil clones, and thus, it finds clients with similar cosine similarity to be malicious, penalizes their reputation, and returns a weighted mean of the gradients, weighed by their reputation.
(7) **FLTrust** [4] bootstraps trust in the clients by assuming that the server has access to a clean validation dataset, albeit small, and returns a weighted mean of the gradients weighed by this trust. In our setting, we do not see a realistic method to access such a clean dataset, especially considering the non-iid nature of the local datasets at the clients.

2.2 Threat model: State-of-the-Art Model Poisoning Attack

Our threat model consists of a scenario where an adversary compromises a fraction of all clients participating in federated learning. We assume that having compromised the clients, the adversary also has access to the gradient vector sent by the benign clients to the server. An attacker agnostic to the server aggregation technique, is not as powerful as the one that knows what aggregation algorithm the server is running. In our threat model, we allow the attacker to know this piece of information. We focus on the SOTA untargeted model poisoning attacks, the Fang attack [5] and the Shejwalkar attack [12], both of which can be classified as directed deviation attacks that bypass *all known defenses* [1] The DDA changes the local models on the compromised worker devices. This change is done strategically (through solving a constrained optimization problem) such that the global model deviates the most toward the *inverse of the direction* along which the benign global model would have changed. Their intuition is that the deviations accumulated over multiple iterations would make the learned global model differ from the benign one significantly.

The **Fang attack** has two variants, one specialized to poison Krum (transferable to Bulyan, we call this the *Full-Krum attack*), the other specialized for Trimmed Mean (transferable to Median, we call this the *Full-Trim attack*), respectively. We assume a full-knowledge (white-box) attack where the attackers have access to the current benign gradients. They themselves compute the benign gradients on their local data (on the compromised devices) as well, and thus estimate, for each parameter, the benign direction by averaging the benign gradients across all clients. This value is stored in a vector s of size equal to the number of model parameters.

$$s(t, \cdot) = sign(\underset{i}{sum}(\nabla LM_i(t, \cdot))),$$

where $\nabla LM_i(t, \cdot)$ is the gradient updates of client i at time t.

Full-Krum attack: Having estimated s, the attackers send gradients in the opposite direction, all with a magnitude λ, with some added noise to appear different but still maintaining a small Euclidean distance from one another. The upper bound of λ is computed in every iteration as a function of $m, c, |P|, GM(t, \cdot), \nabla LM_i(t+1, \cdot)$, where c out m participating clients are malicious, $|P|$ is the number of parameters in the global model GM. λ is then iteratively decreased until the attackers make sure (using a local estimate) that the parameter server would have chosen the attacked model, using the Krum aggregation technique.

Full-Trim attack: The Trim attack while following the same fundamental principle of flipping the gradient direction, attempts to skew the distribution of every parameter j toward the inverse of the direction that $s(t, j)$ suggests in order to attack a mean-like aggregation. It does so by randomly sampling gradient magnitudes from a range that has been computed by the attackers, guaranteed to skew the gradient distribution of every parameter, without appearing as obvious outliers (which would be caught by a method such as Trimmed Mean). Therefore, the attacked gradients here look more diverse than those in the Full-Krum attack.

The **Shejwalkar attack** is basically a more optimized DDA than the Fang attack. It computes the malicious gradient as $\nabla^b + \gamma \nabla^p$ where ∇^b is the reference benign aggregate, ∇^p is a perturbation vector, and γ is a scaling coefficient that is optimized for maximum damage as well as stealth. [12] describes three types of perturbation, namely, inverse unit vector, inverse standard deviation, and inverse sign. All of these are based on the general principle of opposing the direction of the benign gradients, and differ only in the gradient magnitudes. They present a white-box attack that is tailored to specific aggregation techniques, and a gray-box attack that is agnostic to the aggregation algorithm being used. While the Fang attack is not too effective when the aggregation algorithm is unknown, the Shejwalkar-agnostic attack is powerful enough to increase the error rate to completely random levels (10% accuracy for 10 image classes), as reported in Table 3. In our work, we evaluate FLAIR against the more powerful attacker that has the knowledge of aggregation. We create the worst case scenario and experimentally show the success of FLAIR even in this worst case. Overall, DDA at its core takes advantage of the fact that none of the existing aggregation techniques such as FABA and FoolsGold looks at the change in gradient directions to identify malicious gradients in a

[1] There is no overlap between the authors of this current submission and of the SOTA attacks [5, 12].

robust way. FLTrust compares the cosine similarity of a client's gradient with a ground truth gradient vector generated on a trusted root dataset, but such a root dataset cannot be practically collected, given the privacy concerns of the clients. Even estimating a root dataset is difficult because of the non-IIDness in the data distribution among the clients in any realistic setting. FLAIR highlights the importance of gradient direction during the aggregation stage and shows that byzantine-robustness can be achieved without the need of a ground truth root dataset. This makes FLAIR different from all existing defenses. We describe this in more details in the next section.

3 DESIGN

FLAIR uses a reputation-based scheme to compute the aggregation weights of the participating clients. Reputation-based schemes have been widely used in the literature. For example, [7] and [21] make use of reputation score as an incentive mechanism for clients to remain in the system. [6] and [1] compute reputation score from the pairwise cosine similarity of the gradients between clients, [4] does that by computing the cosine similarity of the client gradients with reference to trusted gradients calculated on a clean validation dataset at the server. We compute reputation-score in a different way by using flip-score of the local gradients, as described below. FLAIR assumes that a maximum of $c_{max}(< \frac{m}{2})$ clients can be malicious, where m is the total number of clients participating. It penalizes $2c_{max}$ clients (with too large and too small flip-score) and rewards the rest in every iteration by an amount $\mathcal{W}(i,t)$ based on their flip-score (details described below) and updates their reputation score, where i is the client ID and t is time. We present the pseudocode in Algorithm 1.

$$\mathcal{W}(i,t) = \begin{cases} -(1 - \frac{2c_{max}}{m}), & if \quad penalized. \\ \frac{2c_{max}}{m}, & if \quad rewarded. \end{cases}$$

These values make sure that that the expectation of the reputation score of a client is zero if their flip-scores belong to a uniform random distribution (shown later in this section). This means that in an ideal benign setting with IID data distribution, all clients have the same expected reputation score after a sufficiently large number of rounds. It is interesting to observe that, for $c_{max} < \frac{m}{4}$ the absolute value of penalty higher than the reward. This means that recovery for penalized clients is difficult when the benign clients are in a larger majority (> 75%). Even with a higher fraction of malicious nodes, although the recovery of a penalized client is relatively faster, FLAIR remains robust as long as $c \leq c_{max} < \frac{m}{2}$. This is because the rewarded clients are always prioritized as we will see in the defense policy below. The reputation score of a client i is initialized and updated as follows -

$$RS(i,0) = 0.$$
$$RS(i,t) = \mu_d RS(i,t-1) + \mathcal{W}(i,t), \quad t > 0.$$

where $0 \leq \mu_d \leq 1$ is the decay parameter, and RS is the reputation score. A low μ_d gives more importance to the present ranking of a client based on its flip-score and a high μ_d gives significant importance to the past performance of a client. This decay operation also helps bound the maximum and minimum reputation score for the clients as $\mu_d < 1$ (proven later in the section). We choose a default value of $\mu_d = 0.99$ to give significant importance to the past history of the clients. We normalize this reputation score using softmax into reputation weights W_R so that the sum of these weights equal one across all the clients. Softmax normalization is chosen so as to map all positive and negative reputaion scores to a value between 0 and 1. It also helps the server minimize the contribution of clients with large negative reputation scores while increasing the contribution of clients with positive reputation. We use this W_R as weights to do a weighted mean aggregation. However, a user can also choose to use reputation score or even the flip-score to directly filter out the suspicious gradients and use an aggregation rule of one's choice to aggregate the benign-classified clients in that training round. Hence, our method of computing flip-score is not restricted to any single aggregation technique.

Flip-score. We compare the present updates $\Delta LM_i(t+1,\cdot) = \eta\nabla LM_i(t+1,\cdot) = LM_i(t+1,\cdot) - GM(t,\cdot)$ sent by local model i with the gradient direction of the global model at time t, $s_g(t,\cdot) = sign(GM(t,\cdot) - GM(t-1,\cdot))$. We define flip-score as the sum of square of the gradient magnitudes of all parameters that experience a change in their gradient direction, that is,

$$FS_i(t+1) = \sum_{j=0}^{|P|-1} (\Delta LM_i(t+1,j))^2 \times \\ (sign(\Delta LM_i(t+1,j)) \neq s_g(t,j)), \tag{1}$$

where $|P|$ is the total number of trainable parameters in the model being used. A low flip-score thus suggests that the gradient updates are approximately in the same direction as the previous iteration. In contrast, a high flip-score suggests a deviation from the previous update. This could mean either a large number of parameters have flipped direction, or a small number of parameters have flipped direction with large magnitudes, or both. It is to be noted that the flip-score is proportional to the square of the learning rate used by a client. Therefore, the relative flip-score, and thereby the ranking of clients based on their current flip-score values is more important than the absolute value of the same. We have made use of this fact in our defense policy by trimming out the low and high-ranked clients in every round.

As observed in Figure 2, if the previous global update was benign, a malicious client will tend to have a high flip-score. However, if the previous update itself was poisoned, the flip-score of benign clients will be high and those of malicious clients will be low showing support to the already poisoned direction. Therefore, we penalize c_{max} number of clients at either end of the current flip-score distribution. This allows our system to have a higher detection coverage irrespective of whether the global model was poisoned in one iteration or not. If there is a trusted root dataset available at the server that can be used to generate the benign gradients with certainty, we only need to trim out the gradients with high flip-score as compared to the trusted updates. However, when is is not know what gradients are benign, the uncertainty requires penalizing both high and low flip-score updates, and rewarding updates with flip-score close to the median value in that given round. Based on the previous global update being benign or malicious, the c malicious nodes will all occupy the higher end or the lower end of the flip-score distribution respectively in the given iteration. The $m - c$ benign clients occupy the rest of the spectrum. FLAIR allows a benign node to redeem itself whenever it has a non-extreme flip-score, *i.e.*, it is among the $m - 2c_{max}$ clients that are away from the extremes and closer to the median value. It is also to be noted that we do not discourage low or high absolute values of flip-score moves as we do not impose any hard threshold on permissible flip-score but use the current median to identify the suspicious updates. When required, for example, at the time of convergence, the entire flip-score distribution shifts to a lower range, and the aggregated update will therefore also have a low flip-score and favor convergence. On the other extreme, when it is needed for the global model to escape a local minima, and such a move is supported by a majority of the clients, the entire distribution, and thereby the median will shift to a higher range to favor a shift in the gradient direction to come out of the minima.

Algorithm 1 Federated learning with FLAIR

```
Output: Global model $GM(t+1, \cdot)$
Input: Local model updates $w = \Delta LM_i(t+1, \cdot)$
Parameters: $m$, $c_{max}$, $\mu_d$
0 : Initialize reputation $RS_i(0) = 0$ for every client $i$
1 : Initialize global direction $s_g(0)$ to a zero vector
2 : for every client $i$ compute flip-score:
3 :     $FS_i(t+1) = \sum_{j=0}^{|P|-1} (\Delta LM_i(t+1, j))^2 \times$
             $(sign(\Delta LM_i(t+1, j)) \neq s_g(t, j))$
4 : Penalize $c_{max}$ clients on either end of FS spectrum as:
    $RS(i, t+1) = \mu_d RS(i, t) - (1 - \frac{2c_{max}}{m})$
6 : Reward the rest of the clients as:
    $RS(i, t+1) = \mu_d RS(i, t) + \frac{2c_{max}}{m}$
7 : Normalize reputation weights: $W_R = \frac{e^{RS}}{\sum e^{RS}}$
8 : Aggregate gradients: $\Delta GM(t+1, \cdot) = w^T W_R$
9 : Update global direction: $s_g(t+1, \cdot) = sign(\Delta GM(t+1, \cdot))$
10: Update global model and broadcast:
    $GM(t+1, \cdot) = GM(t, \cdot) + \Delta GM(t+1, \cdot)$
```

Penalty and reward selection Our design policy penalizes $2c_{max}$ out of m clients in every iteration. Considering a completely benign scenario, we want the expected value of the reputation score of a client that has been penalized e fraction of times to be zero, where $e = \frac{2c_{max}}{m}$. Let a client i be penalized en number of times in n iterations. There are $\binom{n}{en}$ ways to select the iterations where the client is penalized. After n iterations, the reputation score of client i is given by:

$$RS(i, n) = \sum_{t=0}^{n} \mu_d^{n-t} \mathcal{W}(i, t). \tag{2}$$

where $\mathcal{W}(i, t)$ is a sequence of penalty and reward over time for client i. Let p and r denote the absolute penalty and reward values. The expected value of this reputation score over all possible sequences $j \in \binom{n}{en}$ is

$$\begin{aligned}
\underset{j}{\mathbb{E}}[RS(i, n)] &= \frac{1}{\binom{n}{en}} \sum_j RS(i, n) \\
&= \frac{1}{\binom{n}{en}} \sum_j \sum_t \mu_d^{n-t} \mathcal{W}(i, t) \\
&= \frac{1}{\binom{n}{en}} \sum_t \mu_d^{n-t} \sum_j \mathcal{W}(i, t) \\
&= \frac{1}{\binom{n}{en}} \sum_t \mu_d^{n-t} (-(pen\binom{n}{en}) + (r(1-e)n\binom{n}{en}))
\end{aligned}$$

Our setting with $r = e = \frac{2c_{max}}{m}$ and $p = 1 - r$ makes the above quantity to be zero thus ensuring that its expected reputation score increment is zero. This proof assumes that it is a random process through which (benign) clients generate their flip scores. Thus, if a client is penalized $\frac{2c_{max}}{m}$ fraction of times, they are expected to have a net neutral reputation score.

Reputation score bounds From the above expression, it is obvious that if $\mu_d = 0$, that is, onlythe current flip-score is considered by ignoring the past, then $-p \leq RS \leq r$. When $0 < \mu_d < 1$, the upper and lower bounds can be computed by assuming the extreme cases where a client was only rewarded or penalized respectively in every iteration. Assuming that the number of iterations tends towards infinity, equation (1) forms an infinite geometric sequence, that can be solved to obtain $\frac{-p}{1-\mu_d} \leq RS \leq \frac{r}{1-\mu_d}$. It should be noted that these reputation scores are normalized using softmax to compute the reputation weights. If the absolute value of the lower bound is not large enough (if μ_d is set to be too small), then even after perfect detection, a malicious client can still have a significant reputation weight after softmax normalization. If μ_d is set to a value closer to 1, then the absolute value of the lower and upper bounds increase, bringing down the contribution of malicious clients to almost zero. At the same time, redemption becomes difficult for a client in this case. This trade-off needs to be kept in mind when setting the decay parameter. We have used $\mu_d = 0.99$ in our experiments in order to remain conservative and make recovery difficult for a client that has been penalized a lot of times. However, this is a design parameter that the user can decide.

Defense instantiation FLAIR helps in maintaining byzantine-robustness at all phases of federated training. Therefore, we recommend instantiating FLAIR right from the beginning of the training. At $t = 0$, we initialize the reference global direction s_g as a vector of zeros. The $(m - 2c_{max})$ clients with flip-score closer to the median at time $t = 1$ are classified as benign by FLAIR. The mean of the updates from these clients at $t = 1$ will set the global reference direction for the next round. As long $c \leq c_{max}$, the reference direction is expected to represent the correct benign direction, and one could ideally work with FLAIR by trimming only the high flip-score values. However, for extra caution against any false-negative malicious detection of clients, we remain conservative anticipating an adaptive whitebox attack and trim from both the ends. This keeps FLAIR robust as it can help the training process recover even after the attackers succeed in a certain round.

4 CONVERGENCE ANALYSIS

We denote the local objective function of client k by F_k, $k = 0, 1, 2, ...m - 1$ and make the following assumptions on it, part of which are adapted from [8].

(1) F_k are all **L-smooth**, that is, for all v and w, $F_k(v) \leq F_k(w) + (v-w)^T \nabla F_k(w) + \frac{L}{2}\|v-w\|_2^2$.
(2) F_k are all **μ-strongly convex**, that is, for all v and w, $F_k(v) \geq F_k(w) + (v-w)^T \nabla F_k(w) + \frac{\mu}{2}\|v-w\|_2^2$.
(3) Let ξ_t^k be sampled uniformly at random from the local data of the $k-th$ client, then the **variance of stochastic gradients** of each client is **bounded**, that is, $\mathbb{E}\,\|\nabla F_k(\mathbf{w}_t^k, \xi_t^k) - \nabla F_k(\mathbf{w}_t^k)\|^2 \leq \sigma_k^2$ for $k = 1, 2, ...m$.
(4) The expected **squared norm of stochastic gradients** is uniformly bounded, that is, $\mathbb{E}\,\|\nabla F_k(\mathbf{w}_t^k, \xi_t^k)\|^2 \leq G^2$ for $k = 1, ...m$, and $t = 0, ..T - 1$.
(5) Within K iterations, the reputation score of malicious clients drop at least by δ_{mal}, and reputation score of benign clients increase at least by δ_{ben}, that is, $|RS_{mal}^t - RS_{mal}^{t-K}| \geq \delta_{mal}$ and $|RS_{ben}^t - RS_{ben}^{t-K}| \geq \delta_{ben}$, for $t = 0, ..T - 1$. We empirically show this in Figure 4.

Assumptions #1 and #2 are standard and apply to l_2-norm regularized linear regression, logistic regression, and softmax classifier. Assumptions #3 and #4 have been also made by [14, 15, 19, 21]. In our problem setting, Assumption #3 claims that the gradient with a subset of local data is bounded from the gradient with whole batch for all clients. Assumption #5 means that after every K iterations in our algorithm, the reputation scores of malicious and benign clients will diverge further.

Theorem 1 - *Let Assumptions (1)-(5) hold and $L, \mu, \sigma_k, G, K\delta_{mal}, \delta_{ben}$ be defined therein. Choose $\gamma = max\{8\frac{L}{\mu}, 1\}$, and $\eta_t = \frac{2}{\mu(\gamma+t)}$. Let GM denote the global objective function and let GM^* and F_k^* be the minimum value of GM and F_k respectively, then:*

$$\mathbb{E}[GM(\mathbf{w}_T)] - GM^* \leq \frac{2}{\mu^2} \cdot \frac{L}{\gamma + T}\Big(\sum_{k=1}^{m} p_k^2 \sigma_k^2 + 6L\Gamma + 8G^2 + 8G^2 \sum_{k=1}^{c} p_{k,0} + \frac{\mu^2}{4}\|w_0 - w^*\|^2\Big).$$

where T is the total number of iterations, $p_{k,0}$ is the initial weight for the clients, and $\Gamma = GM^* - \sum_{k=1}^{m} p_k F_k^*$, that effectively quantifies the degree of non-iidness according to [8]. This shows that a weighted mean aggregation is guaranteed to converge in federated learning as long as a defense mechanism ensures that Assumption 5 holds, that is, the weights of the malicious and benign clients keep diverging with time. The convergence speed is $O(\frac{1}{T})$. The proof for the theorem is available in Appendix § A.1.

5 IMPLEMENTATION

We have simulated the federated learning on a single machine with a Tesla P100 PCIe GPU with 16GB memory, using PyTorch, with as many clients as can be handled by our machine. The data was distributed with a non-IID bias of 0.5 (default), except for Shakespeare where the data was distributed sequentially among the clients and FEMNIST where the data was distributed by the writer, both of which constitute non-IID distributions. In our simulation, all clients run one local iteration on a batch of its local data before communicating with the parameter server in a synchronous manner. The clients sample their local data in a round-robin manner, send their local gradients to the parameter server, and download the updated global model before running the next local iteration. The malicious clients attack every iteration of training. We assume $c = c_{max}$, that is, the extreme case of $c \leq c_{max}$ to test the limits of our defense. For the SHEJWALKAR attack, the perturbation type is chosen to be unit vector.

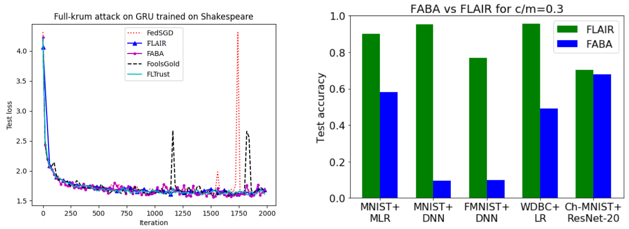

Figure 3: *Left* shows the test loss curve comparing FLAIR with the benchmark aggregation algorithms against the Full-Krum attack. We see that the attack generates sporadic spikes in training, best handled by FLAIR, evident from its smooth test loss curve. *Right* shows the comparison of FABA and FLAIR across diverse datasets for $c/m = 0.3$. FABA begins to fail with a higher fraction of malicious clients, while FLAIR remains robust.

Baselines and datasets. The baseline aggregation rules used are Krum, Bulyan, Trimmed Mean, and Median. We also compare FLAIR with the recent defense techniques of FABA [16], FoolsGold [6], and FLTrust [4]. We evaluate FLAIR on 4 different datasets (Table 1). MNIST, CIFAR-10, and FEMNIST are image datasets, Shakespeare is an NLP dataset. The DNN trained on MNIST has 2 conv layers with 30 and 50 channels respectively, each followed by a 2 × 2 maxpool layer, then by a fully connected layer of size 200, and an output layer of size 10. We use a constant learning rate, except for CIFAR-10 where we start with zero, reach the peak at one-fourth of the total iterations, and slowly get down to zero again. The CNN trained on the FEMNIST dataset follows the same network architecture as [3].

Non-IID data generation. We distribute the data among the clients as described in [5]. This method requires a non-IID bias $b, 0 < b < 1$. Given m clients and a dataset with C classes, we group the clients into C groups uniformly. The complete dataset is iterated through, one data sample at a time. A data sample with class label l is sent to group l with probability b and to any other group with a uniform probability $\frac{1-b}{l-1}$. Within a group, all data samples are distributed among the clients uniformly. This non-IID distribution is thus based on the label skew and is very useful in analyzing federated learning algorithms on skewed dataset distributions.

Table 1: Datasets with the number of classes (n_c) and training samples (n_s), the models and model parameters (P), training rounds (n_r), batch_size (b), learning rate (lr), total number of clients (m), number of malicious ones (c), and decay parameter (μ) used in FLAIR. (*: variable learning rate, peaks at 0.1.)

Dataset	n_c	n_s	Model	P	n_r	b	lr	m	c	μ_d
MNIST	10	60k	DNN	0.51M	500	32	0.01	100	20	0.99
CIFAR-10	10	50k	ResNet-18	5.2M	2000	128	0.1^*	10	2	0.99
Shakespeare	100	-	GRU	0.14M	2000	100	0.01	10	2	0.99
FEMNIST	62	805k	DNN	6.6M	2000	32	0.1	35	7	0.99

Table 2: Attack impact - Test accuracy for Directed Deviation model poisoning attacks (Full-Krum; Full-Trim), on different datasets with $c/m = 0.2$. For the Shakespeare dataset, test loss has been reported. We verify the damaging impact of the Full-Trim attack on mean-like aggregations (FedSGD, Trimmed mean, Median) and Full-Krum attack on Krum-like aggregations (Krum, Bulyan). We also observe that the existing defenses—FABA, FoolsGold, and FLTrust—seem to defend against this attack in some cases, and fail in others, whereas FLAIR consistently shines in all cases.

Attack	Defense	Test accuracy (%) / Test loss (only for Shakespeare)			
		MNIST+ DNN	CIFAR-10+ ResNet-18	Shakespeare+ GRU	FEMNIST+ DNN
None	FedSGD	92.45	**71.17**	**1.62**	83.60
	FLAIR	**92.52**	66.92	1.64	83.58
	FABA	91.77	69.94	1.76	82.69
	FoolsGold	91.20	70.71	1.63	**83.80**
	FLTrust	87.70	68.08	**1.62**	82.72
Full-Krum	FedSGD	82.97	39.68	1.62	29.87
	Krum	8.92	9.81	11.98	5.62
	Bulyan	10.14	13.24	9.23	9.91
	FLAIR	**87.73**	61.26	1.64	**80.19**
	FABA	86.99	55.96	1.75	55.61
	FoolsGold	47.12	42.28	**1.63**	0.07
	FLTrust	82.50	**65.25**	1.67	79.53
Full-Trim	FedSGD	65.25	47.32	1.74	32.34
	Trim	36.36	55.25	3.28	13.03
	Median	28.37	50.54	3.30	45.6
	FLAIR	90.55	67.65	1.66	82.51
	FABA	**91.84**	67.31	**1.64**	79.66
	FoolsGold	91.61	**69.24**	1.66	**83.09**
	FLTrust	34.20	64.23	1.68	79.28

Table 3: Test accuracy for FLAIR under Shejwalkar attack with and without the knowledge of the aggregator as compared with the baseline FedSGD performance on MNIST and CIFAR-10.

AGR	AGR-knowledge	MNIST	CIFAR-10
FedSGD	no	10.10	10.00
	yes	10.09	10.00
FLAIR	no	92.25	69.35
	yes	92.83	69.98

6 EVALUATION

6.1 Macro Experiments

In Table 2, we compare the test accuracy achieved by various aggregation techniques in benign and malicious conditions under the

Table 4: Fraction of malicious or benign clients allotted non-negligible weights ($> 10^{-4}$) averaged over 500 iterations. The weakness of FoolsGold and FLTrust stems in part from the fact that they assign negligible weight to a significant fraction of benign clients for high detection coverage.

Defense	Mal/ Ben	Benign	Full-trim	Full-krum
FoolsGold	n_{ben}	0.29	0.30	0.10
	n_{mal}	-	0.00	0.64
FLTrust	n_{ben}	0.48	0.45	0.49
	n_{mal}	-	0.52	0.63
FLAIR	n_{ben}	0.75	0.75	0.63
	n_{mal}	-	0.00	0.08

Fang attack. We do not claim to provide optimal model architectures that can achieve the best test accuracy, but we provide fair comparison of all the defenses on the same model with the same training parameters. For Shakespeare, which is an NLP dataset, we report the test loss; for all others, we report test accuracy, where the training and test datasets are disjoint. The final reported test loss value does not capture the training dynamics, which can be observed in Figure 3 for the more damaging Full-Krum attack. We see that FLAIR is the winner or 2nd place finisher in 7 of the 12 cells ((benign + two attacks) × 4 datasets). This on the surface appears to be not very promising, till one looks deeper. The baseline protocols that finish first in one configuration fare disastrously in other configurations, indicating that they are tailored to specific attacks or datasets (whether by conscious design or as an artifact of their design). For example, FoolsGold does creditably for the Full-Trim attack but is vulnerable against the Full-Krum attack.

Averaging across the configurations, it appears FABA is the closest competitor to FLAIR. We observe that FABA, although it performed well for $c/m = 0.2$, failed to defend when the number of attackers grew to $c/m = 0.3$, evident from the results in Figure 3(b). We have used F-MNIST, Ch-MNIST, and Breast cancer Wisconsin Dataset here to show the effect on diverse datasets. As the number of attackers increases, the mean starts to shift more toward them. One false positive detection by FABA can cause it to trim out a benign local update, which causes the mean to shift further toward the malicious updates iteratively, and fail. On the other hand, FLAIR is guaranteed to defend against the attack as long as $c \leq c_{max} < \frac{m}{2}$. We find empirically that FABA degrades fast, and much faster than FLAIR, when the fraction of malicious clients increases. Where FABA fails at $c/m = 0.3$, we show in Figure 6 that FLAIR remains stable until $c/m = 0.45$. We also show in Figures 7(a) and (b) that FLAIR remains robust across a wide range of non-IID bias, *i.e.*, 0.1 to 0.8. FLAIR and FABA require an estimate of the upper bound of number of malicious clients, c_{max} to be known. FoolsGold and FLTrust, on the other hand, make use of cosine similarity among clients, and with a trusted cross-check dataset at the server, respectively, to identify suspicious clients. We have found that both of these techniques unnecessarily penalize many benign clients and assign them a zero weight in order to conservatively defend against an attack, as can be seen in Table 4. This can have a significantly negative impact on a practical system where one wishes to learn from data that the different clients hold locally. On the other hand, FLAIR allows all clients to contribute to the global model update that do not have a large negative reputation score. Figure 4 shows the evolution of the average reputation score of malicious and benign clients. When benign and malicious clients start from the same point, the weights of malicious clients decrease with time tending to zero, while those of the benign clients increase.

Table 3 shows for the Shejwalkar attack, the test accuracy achieved in the presence and absence of FLAIR when the white-box and gray-box variants of the attack are used on the MNIST and CIFAR-10 datasets. The attack achieves higher error rate in the absence of a defense even when the aggregation algorithm is unknown, proving to be a stronger attack compared to the Fang attack. This is because it better optimizes the attacked gradient magnitudes. However, it also relies on gradient flips to achieve its target and the malicious behavior gets flagged by FLAIR that restores the training to benign standards.

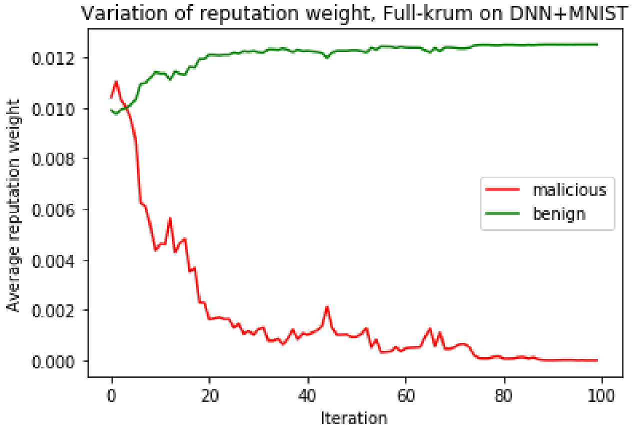

Figure 4: The figure demonstrates the typical training dynamics of clients' reputation weights against time with FLAIR, showing an increase in average reputation of benign clients and decrease in that of malicious clients as the training progresses.

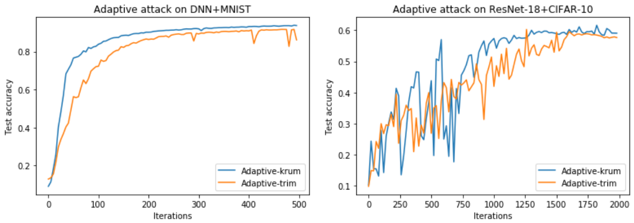

Figure 5: Performance of FLAIR against adaptive white-box attacks, specifically designed to attack FLAIR, evaluated on MNIST and CIFAR-10. We observe a significant improvement in test accuracy, compared to the base case impact of Full-Krum on Krum and Full-Trim on FedSGD, as reported in Table 2.

6.2 Adaptive attack

Having shown the performance of FLAIR against the above attacks, we proceed to analyze an adaptive attack scenario, *i.e.*, one where the attacker has full knowledge of FLAIR, including the dynamic value of the cutoff flip-scores. Thus, at iteration $t+1$, the attacker knows $FS_{low}(t)$ and $FS_{high}(t)$ beyond which the clients were penalized at iteration t. The Adaptive-Krum attack first computes the target malicious gradients at $t+1$ using Full-Krum algorithm, and if its flip-score goes above $FS_{high}(t)$, it reverses the attack on its less important parameters, *i.e.*, parameters that would have had low magnitude updates without attack. It does this by replacing 5% of the attacked parameters, at a time, with their benign values until the flip-score is brought down to ensure stealth. Since the stealthy attack will be less powerful than the original intended attack, the global model will only be partially poisoned, and the attackers are not expected to occupy the lower spectrum of the flip-score. This has been verified in our experiments as well. All the malicious clients send these attacked parameters with some added randomness in order to support one other. We observe a trade-off between stealth and attack impact in this case, as can be seen in Figure 5, that the attack loses its impact as it tries to evade the defense in a stealthy manner.

The Adaptive-Trim attack is a smarter and collaborative attack. Its target is to generate attacked gradients v_i, $i = 0, 1, \cdots c-1$. It first computes the Full-Trim target attack u_i for every malicious client i. Then, it initializes v_0 to u_0 and modifies it, until v_0 generates a flip-score that is less than $FS_{high}(t)$. This modification is done in a manner similar to that of Adaptive-Krum, as explained above. Client $i = 1$ then updates its target attack from u_1 to $v_1 = u_1 + (u_0 - v_0)$ in order to compensate for a sub-optimal attack created by client $i = 0$ because of the flip-score-evasion constraint. This is how the malicious clients collude among themselves. A malicious client, however may not necessarily find a solution as the target grows and the flip-score constraints may become too hard to solve for the updated target. The attacker hopes to attain $\sum_{i=0}^{c-1} u_i = \sum_{i=0}^{c-1} v_i$ in order to have the original intended attack impact.

The performance of FLAIR against this adaptive white-box attack is shown in Figure 5. We find that the adaptive attacks are not very effective against FLAIR, where for comparison, the benign accuracy for the two datasets are 92.45% and 71.17%. This happens due to multiple reasons: 1) the attack loses in strength while trying to gain in stealth, 2) the attackers need not be allotted equal reputation weights, so the weighted sum of the attacked gradients v do not match with the weighted sum of the target attack u, 3) the flip-score distribution is dynamic, as can be seen in Figure 2, and changes from time t to $t + 1$, and when it decreases in consecutive iterations by a significant amount, the attackers can still be blocked as the attack was crafted to evade the cutoff flip-score at time t that is accessible to the attackers, which could still be higher than the cutoff flip-score at time $t + 1$.

To counter the second point mentioned above, we have also created a more knowledgeable attacker, by modifying the above constraint with a weighted sum, weighted by the reputation scores. Here, the adversary even has the knowledge of its own reputation weight (W_R). It uses this information to come up with attacked gradients with better chances of a successful attack. We call this a “Weighted-Adaptive-Trim" attack. The modified constraint becomes

$$\sum_{i}^{c-1} W_{R,i} v_i = \sum_{i=0}^{c-1} W_{R,i} u_i$$

FLAIR successfully defends even against this attack. We show the experimental results on on MNIST in Figure 6. With FLAIR, the test accuracy reaches 90% while in the baseline case with FedSGD and the vanilla attack, it only reaches upto around 58% test accuracy. For context, without any attack, the model reaches an accuracy of 92.45%. The effectiveness of FLAIR against this attack can be attributed to reasons (1) and (3) given above.

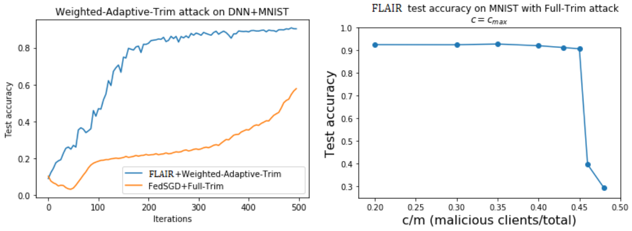

Figure 6: Figure (left) shows the test accuracy of FLAIR when evaluated on MNIST dataset under the default conditions with $m = 100, c = 20$ where the adversary launches the Weighted-Adaptive-Trim attack on the system, compared with the baseline performance of FedSGD against the Full-Trim attack. FLAIR successfully defends against this attack to achieve a 90% test accuracy. Figure (right) shows the performance of FLAIR on MNIST with increasing c. We see that FLAIR is stable across a large range and breaks only above $c = 0.45$ which is close to the theoretical limit of $c = 0.5^-$.

6.3 Robustness of FLAIR

Here, we provide additional evaluation of FLAIR in two specific situations. We stress-test it first by subjecting it to a higher number of malicious clients to find the breaking point of FLAIR, when trained on MNIST dataset in the presence of Full-trim attack. We assume that the number of compromised clients is still not greater than c_{max}, and to that end, we set $c = c_{max}$ to test FLAIR in the extreme condition. Since FLAIR requires $c_{max} < \frac{m}{2}$, we have swept c upto 49 where m was fixed at 100. We observe in Figure 6 that FLAIR is stable upto $c/m = 0.45$ whereas the rest of the defense techniques broke below $c/m = 0.30$ as can be seen in Table 2 and Figure 3 with $c = c_{max}$ set for all the defense techniques that require a knowledge of c_{max}. This shows that FLAIR remains robust across a wider range of malicious conditions as compared to any other previous defense techniques.

Figures 7(a) and (b) show the performance of FLAIR on MNIST dataset distributed among 100 clients with varying degrees of non-IIDness. We observe that, except for the extreme case of *bias* = 0.9, FLAIR remains exceptionally stable. This is because FLAIR does not discriminate against clients with unique data unless the gradients they send consistently oppose the benign direction, in which case, the global model would most likely be hurt by incorporating those gradients.

7 DISCUSSION AND CONCLUSION

We have presented FLAIR, a secure parameter server for federated learning, robust to any untargeted model poisoning attack. FLAIR uses a stateful algorithm to allocate reputation scores to the participating clients to lower the contribution of maliciously behaving clients. We define malicious behavior using a metric, flip-score, which when too high or too low, captures attacks that try to divert the global model away from convergence. This makes FLAIR a robust defense, no matter when it is instantiated, although we recommend enabling FLAIR right from the start. FLAIR can also be used to just filter out clients solely based on their flip-score, so that an aggregation of the user’s choice (other than weighted mean) can be used. We also add the fairness attribute to FLAIR by allowing clients to redeem themselves and contribute to the global model if they act benign. This is done using a user-defined *decay*

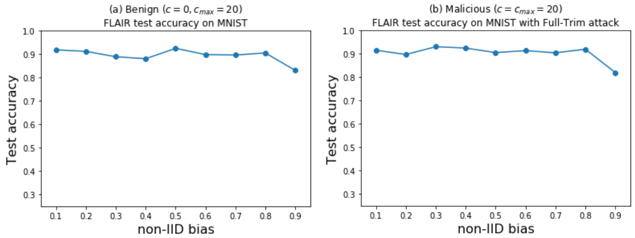

Figure 7: Figures (a) and (b) show the test accuracy of FLAIR on MNIST dataset distributed with varying non-IID bias across 100 clients in benign and malicious cases respectively. FLAIR can be seen to be robust enough for a wide range of bias, 0.1–0.8, with a small dip in test accuracy occurring at $bias = 0.9$.

parameter to control the importance of the past performance of a client and thereby its speed of redemption. We evaluate the benefits of FLAIR compared to the fundamental FL aggregation FedSGD and state-of-the-art defenses, namely Krum, Bulyan, FABA, FoolsGold, and FLTrust. We evaluate using full knowledge untargeted model poisoning attacks that have recently been found to be most damaging against FL. We find that different existing defenses shine under specific combinations of attacks and datasets/models. However, FLAIR provides transferable defense with accuracy competitive with respective winners under all configurations. Further, FLAIR holds up better than its closest competitor, FABA, when the fraction of malicious clients increases (beyond 20%). Finally, an adaptive white-box attacker with access to all internals of FLAIR, including dynamically determined threshold parameters, cannot bypass its defense.

All of our evaluation is limited to a synchronous setting with no gradient encryption, as FLAIR requires the server to be capable of accessing unencrypted client updates so that it can classify the update as benign or suspicious based on its flip-score. Gradient encryption is also less relevant in a cross-device scenario due to its high computational overhead [20]. However, these limitations present logical avenues for our future work.

ACKNOWLEDGMENTS

This material is based in part upon work supported by the National Science Foundation under Grant Numbers CNS-2038986 and CNS-2146449 (NSF CAREER award), and a contract from the Army Research Lab (ARL) W911NF-2020-221. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. We would also like to thank Dipesh Tamboli and Ashwinee Panda for their FLTrust re-implementation for its use as one of the baselines for FLAIR and for ResNet-18's implementation to train the CIFAR-10 dataset respectively.

REFERENCES

[1] Sana Awan, Bo Luo, and Fengjun Li. 2021. CONTRA: Defending against Poisoning Attacks in Federated Learning. In *European Symposium on Research in Computer Security*. Springer, 455–475.

[2] Peva Blanchard, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*. 119–129. Krum paper.

[3] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2019. LEAF: A Benchmark for Federated Settings. *arXiv preprint arXiv:1812.01097* (2019).

[4] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. 2021. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. *Network and Distributed System Security Symposium* (2021), 1–18.

[5] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *29th USENIX Security Symposium (USENIX Security 20)*. Boston, MA. https://www.usenix.org/conference/usenixsecurity20/presentation/fang

[6] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2020. The Limitations of Federated Learning in Sybil Settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. USENIX Association, San Sebastian, 301–316. https://www.usenix.org/conference/raid2020/presentation/fung

[7] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. 2019. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal* 6, 6 (2019), 10700–10714.

[8] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HJxNAnVtDS

[9] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*.

[10] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. 2018. The hidden vulnerability of distributed learning in byzantium. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. 3521–3530. Bulyan paper.

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

[12] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. *Internet Society* (2021), 18.

[13] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated multi-task learning. In *Advances in Neural Information Processing Systems*. 4424–4434.

[14] Sebastian U Stich. 2018. Local SGD converges fast and communicates little. *arXiv preprint arXiv:1805.09767* (2018).

[15] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. 2018. Sparsified SGD with memory. *Advances in Neural Information Processing Systems* 31 (2018).

[16] Qi Xia, Zeyi Tao, Zijiang Hao, and Qun Li. 2019. FABA: An Algorithm for Fast Aggregation against Byzantine Attacks in Distributed Neural Networks.. In *IJCAI*. 4824–4830.

[17] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

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

[19] Hao Yu, Sen Yang, and Shenghuo Zhu. 2019. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5693–5700.

[20] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20)*. 493–506.

[21] Yuchen Zhang, John C Duchi, and Martin J Wainwright. 2012. Communication-efficient algorithms for statistical optimization. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 6792–6792.

A APPENDIX

A.1 Proof for Theorem 1

Proof -

Let the k-th client hold n_k training data batches: $x_{k,1}, ...x_{k,n_k}$. The local objective function $F_k(\cdot)$ is given by

$$F_k(\mathbf{w}) = \frac{1}{n_k}\sum_{j=1}^{n_k} l(\mathbf{w}; x_{k,j}),$$

where $l(\cdot;\cdot)$ is the specified loss function for each client.

The global objective function is defined as

$$GM_{k,t}(\mathbf{w}) = \sum_{k=1}^{m} p_{k,t}F_k(\mathbf{w}).$$

The global model is updated as

$$\mathbf{w}_{t+1}^k = \mathbf{w}_t^k - \eta_t \sum_{k=1}^{m} p_{k,t}\nabla F_k(\mathbf{w}_t^k),$$

where $p_{k,t} = softmax(RS_{k,t})$ is the softmax of reputation score of client k at time t.

We update the weights by averaging the weights from selected clients $\bar{\mathbf{w}}_t = \sum_{k=1}^{m} p_{k,t}\mathbf{w}_t^k$. For convenience, we also define $g_t = \sum_{k=1}^{m} p_{k,t}\nabla F_k(\mathbf{w}_t^k, \xi_t^k)$, where ξ_t^k is the selected local data.

A.1.1 ***Analysis on consecutive steps.*** To bound the expectation of the global objective function at time T from its optimal value, we first consider to analyze the global weight from the optimal weights by calculating single step SGD:

$$\begin{aligned}\|\bar{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 &= \|\bar{\mathbf{w}}_t - \eta_t g_t - \mathbf{w}^* - \eta_t\bar{g}_t + \eta_t\bar{g}_t\|^2 \\ &= \|\bar{\mathbf{w}}_t - \mathbf{w}^* - \eta_t\bar{g}_t\|^2 + 2\eta_t\langle\bar{\mathbf{w}}_t - \mathbf{w}^* - \eta_t\bar{g}_t, \bar{g}_t - g_t\rangle \\ &\quad + \eta_t^2\|\bar{g}_t - g_t\|^2.\end{aligned} \tag{3}$$

The first term of Equation. 3 can be expressed as

$$\|\bar{\mathbf{w}}_t - \mathbf{w}^* - \eta_t\bar{g}_t\|^2 = \|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 - 2\eta_t\langle\bar{\mathbf{w}}_t - \mathbf{w}^*, \bar{g}_t\rangle + \eta_t^2\|\bar{g}_t\|^2. \tag{4}$$

The second term of Equation. 4 can be expressed as

$$\begin{aligned}-2\eta_t\langle\bar{\mathbf{w}}_t - \mathbf{w}^*, \bar{g}_t\rangle &= -2\eta_t\sum_{k=1}^{m} p_{k,t}\langle\bar{\mathbf{w}}_t - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k)\rangle \\ &= -2\eta_t\sum_{k=1}^{m} p_{k,t}\langle\bar{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k)\rangle \\ &\quad - 2\eta_t\sum_{k=1}^{m} p_{k,t}\langle\mathbf{w}_t^k - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k)\rangle.\end{aligned} \tag{5}$$

By Cauchy-Schwarz inequality and AM-GM inequality, we have

$$-2\langle\bar{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k)\rangle \leq \frac{1}{\eta_t}\|\bar{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \eta_t\|\nabla F_k(\mathbf{w}_t^k)\|^2. \tag{6}$$

By the μ-strong convexity of $F_k(\cdot)$, with $v = \mathbf{w}^*$ and $w = \mathbf{w}_t^k$, we have

$$-\langle\mathbf{w}_t^k - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k)\rangle \leq -(F_k(\mathbf{w}_t^k) - F_k(\mathbf{w}^*)) - \frac{\mu}{2}\|\mathbf{w}_t^k - \mathbf{w}^*\|^2. \tag{7}$$

By the convexity of $\|\cdot\|$ and the L-smoothness of $F_k(\cdot)$, we can express third term of Equation. 4 as

$$\eta_t^2\|\bar{g}_t\|^2 \leq \eta_t^2\sum_{k=1}^{m} p_{k,t}\|\nabla F_k(\mathbf{w}_t^k)\|^2 \leq 2L\eta_t^2\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - F_k^*). \tag{8}$$

Combining Equations. 4 − 8, we have

$$\begin{aligned}\|\bar{\mathbf{w}}_t - \mathbf{w}^* - \eta_t\bar{g}_t\|^2 &\leq \|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 \\ &\quad + \eta_t\sum_{k=1}^{m} p_{k,t}(\frac{1}{\eta_t}\|\bar{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \eta_t\|\nabla F_k(w_t^k)\|^2) \\ &\quad - 2\eta_t\sum_{k=1}^{m} p_{k,t}((F_k(\mathbf{w}_t^k) - F_k(\mathbf{w}^*)) + \frac{\mu}{2}\|\mathbf{w}_t^k - \mathbf{w}^*\|^2) \\ &\quad + 2L\eta_t^2\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - F_k^*) \\ &= (1 - \mu\eta_t)\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \sum_{k=1}^{m} p_{k,t}\|\bar{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \\ &\quad + 2L\eta_t^2\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - F_k^*) \\ &\quad + \eta_t^2\sum_{k=1}^{m} p_{k,t}\|\nabla F_k(\mathbf{w}_t^k)\|^2 - 2\eta_t\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - F_k(\mathbf{w}^*)) \\ &\leq (1 - \mu\eta_t)\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \sum_{k=1}^{m} p_{k,t}\|\bar{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \\ &\quad + 4L\eta_t^2\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - F_k^*) - 2\eta_t\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - F_k(\mathbf{w}^*)),\end{aligned} \tag{9}$$

where we use the L-smoothness of $F_k(\cdot)$ in the last inequality, and Jensen inequality on $\mathbf{w}_t^k$ with $\phi(x) = \|x - w^*\|^2$ in the 2nd inequality.

$$\begin{aligned}&4L\eta_t^2\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - F_k^*) - 2\eta_t\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - F_k(\mathbf{w}^*)) \\ &= -\gamma_t\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - GM^*) - \gamma_t\sum_{k=1}^{m} p_{k,t}(GM^* - F_k^*) \\ &\quad + 2\eta_t\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}^*) - F_k^*) \\ &= -\gamma_t\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - GM^*) - \gamma_t\sum_{k=1}^{m} p_{k,t}(GM^* - F_k^*) \\ &\quad + 2\eta_t\sum_{k=1}^{m} p_{k,t}(GM^* - F_k^*) \\ &= -\gamma_t\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - GM^*) + (2\eta_t - \gamma_t)\sum_{k=1}^{m} p_{k,t}(GM^* - F_k^*) \\ &= -\gamma_t\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - GM^*) + 4L\eta_t^2\Gamma,\end{aligned} \tag{10}$$

where $\Gamma = \sum_{k=1}^{m} p_{k,t}(GM^* - F_k^*) = GM^* - \sum_{k=1}^{m} p_{k,t}F_k^*$, and $\gamma_t = 2\eta_t(1 - 2L\eta_t)$.

The first term of Equation. 10

$$\begin{aligned}
\sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - GM^*) &= \sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - F_k(\bar{\mathbf{w}}_t)) \\
&\quad + \sum_{k=1}^{m} p_{k,t}(F_k(\bar{\mathbf{w}}_t) - GM^*) \\
&\geq \sum_{k=1}^{m} p_{k,t}\langle \nabla F_k(\bar{\mathbf{w}}_t), \mathbf{w}_t^k - \bar{\mathbf{w}}_t) \rangle + \sum_{k=1}^{m} p_{k,t}(F_k(\bar{\mathbf{w}}_t) - GM^*) \\
&= \sum_{k=1}^{m} p_{k,t}\langle \nabla F_k(\bar{\mathbf{w}}_t), \mathbf{w}_t^k - \bar{\mathbf{w}}_t) \rangle + GM(\bar{\mathbf{w}}_t) - GM^* \\
&\geq -\frac{1}{2}\sum_{k=1}^{m} p_{k,t}(\eta_t \|F_k(\bar{\mathbf{w}}_t)\|^2 + \frac{1}{\eta_t}\|\mathbf{w}_t^k - \bar{\mathbf{w}}_t\|^2) \\
&\quad + GM(\bar{\mathbf{w}}_t) - GM^* \\
&\geq -\sum_{k=1}^{m} p_{k,t}(\eta_t L(F_k(\bar{\mathbf{w}}_t) - F_k^*) + \frac{1}{2\eta_t}\|\mathbf{w}_t^k - \bar{\mathbf{w}}_t\|^2) \\
&\quad + GM(\bar{\mathbf{w}}_t) - GM^*,
\end{aligned} \tag{11}$$

where the first inequality results from the convexity of $F_k(\cdot)$, the second inequality from AM-GM inequality and the third inequality from L-smoothness of $F_k(\cdot)$.

Therefore, Equation. 10 becomes

$$\begin{aligned}
&-\gamma_t \sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - GM^*) + 4L\eta_t^2\Gamma \\
\leq& \gamma_t(\sum_{k=1}^{m} p_{k,t}(\eta_t L(F_k(\bar{\mathbf{w}}_t) - F_k^*) + \frac{1}{2\eta_t}\|\mathbf{w}_t^k - \bar{\mathbf{w}}_t\|^2)) \\
&-\gamma_t(GM(\bar{\mathbf{w}}_t) - GM^*) + 4L\eta_t^2\Gamma \\
=& \gamma_t(\sum_{k=1}^{m} p_{k,t}(\eta_t L(F_k(\bar{\mathbf{w}}_t) - GM^*) + \frac{1}{2\eta_t}\|\mathbf{w}_t^k - \bar{\mathbf{w}}_t\|^2)) \\
&+ \gamma_t\eta_t L\Gamma - \gamma_t(GM(\bar{\mathbf{w}}_t) - GM^*) + 4L\eta_t^2\Gamma \\
=& \gamma_t(\eta_t L - 1)\sum_{k=1}^{m} p_{k,t}(F_k(\bar{\mathbf{w}}_t) - GM^*) \\
&+ \frac{\gamma_t}{2\eta_t}\sum_{k=1}^{m} p_{k,t}\|\mathbf{w}_t^k - \bar{\mathbf{w}}_t\|^2 + (4L\eta_t^2 + \gamma_t\eta_t L)\Gamma,
\end{aligned} \tag{12}$$

With $GM(\bar{\mathbf{w}}_t) - GM^* > 0$ and $\eta_t L - 1 < 0$, we have

$$\gamma_t(\eta_t L - 1)\sum_{k=1}^{m} p_{k,t}(F_k(\bar{\mathbf{w}}_t) - GM^*) \leq 0, \tag{13}$$

and recall $\gamma_t = 2\eta_t(1 - 2L\eta_t)$, so $\frac{\gamma_t}{2\eta_t} \leq 1$ and $4L\eta_t^2 + \gamma_t\eta_t L \leq 6L\eta_t^2$.

Therefore,

$$\begin{aligned}
-\gamma_t \sum_{k=1}^{m} p_{k,t}(F_k(\mathbf{w}_t^k) - GM^*) + 4L\eta_t^2\Gamma \\
\leq \sum_{k=1}^{m} p_{k,t}\|\mathbf{w}_t^k - \bar{\mathbf{w}}_t\|^2 + 6L\eta_t^2\Gamma.
\end{aligned} \tag{14}$$

Thus, Equation. 9 becomes

$$\begin{aligned}
\|\bar{\mathbf{w}}_t - \mathbf{w}^* - \eta_t\bar{g}_t\|^2 \leq& (1 - \mu\eta_t)\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 2\sum_{k=1}^{m} p_{k,t}\|\mathbf{w}_t^k - \bar{\mathbf{w}}_t\|^2 \\
&+ 6L\eta_t^2\Gamma.
\end{aligned} \tag{15}$$

A.1.2 *Bound for variance of gradients.*

Next, to bound the gradient, using assumption 3, we have

$$\begin{aligned}
\mathbb{E}\|g_t - \bar{g}_t\|^2 =& \mathbb{E}\|\sum_{k=1}^{m} p_{k,t}(\nabla F_k(\mathbf{w}_t^k, \xi_t^k) - \nabla F_k(\mathbf{w}_t^k))\|^2 \\
=& \sum_{k=1}^{m} p_{k,t}^2 \mathbb{E}\|\nabla F_k(\mathbf{w}_t^k, \xi_t^k) - \nabla F_k(\mathbf{w}_t^k)\|^2 \\
\leq& \sum_{k=1}^{m} p_{k,t}^2 \sigma_k^2.
\end{aligned} \tag{16}$$

A.1.3 ***Bound for divergence of weights.***

Based on Assumption 5, for malicious clients $k = 1, 2, \ldots, c$, we have

$$\begin{aligned}
p_{k,t} =& softmax(RS_{km}^t) = \frac{e^{RS_{km}^t}}{\sum_{i=1}^{m} RS_i^t} \\
=& \frac{e^{RS_{km}^{t-M} - \delta_m}}{\sum_{i=1}^{c} e^{RS_i^{t-M} - \delta_m} + \sum_{i=c+1}^{m} e^{RS_i^{t-M} + \delta_b}} \\
=& \frac{e^{RS_{km}^{t-M}}}{\sum_{i=1}^{c} e^{RS_i^{t-M}} + \sum_{i=c+1}^{m} e^{RS_i^{t-M} + \delta_b + \delta_m}} \\
\leq& \frac{e^{RS_{km}^{t-M}}}{\sum_{i=1}^{c} e^{RS_i^{t-M}} + \sum_{i=c+1}^{m} e^{RS_i^{t-M}}} \\
=& p_{k,t-M}.
\end{aligned} \tag{17}$$

To bound the weights, we assume within E communication steps, there exists $t_0 < t$, such that $t - t_0 \leq E - 1$ and $\mathbf{w}_{t_0}^k = \bar{\mathbf{w}}_{t_0}$ for all $k = 1, 2, \ldots, m$. And we know η_t is non-increasing and $\eta_{t_0} \leq 2\eta_t$.

With the fact $\mathbb{E}\|X-\mathbb{E}X\|^2 \leq \mathbb{E}\|X\|^2$ and Jensen inequality, we have

$$\begin{aligned}
\mathbb{E}\sum_{k=1}^{m} p_{k,t}\|\bar{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \leq& \mathbb{E}\sum_{k=1}^{m} p_{k,t}\|\bar{\mathbf{w}}_{t_0} - \mathbf{w}_t^k\|^2 \\
\leq& \sum_{k=1}^{m} p_{k,t}\mathbb{E}\sum_{t_0}^{t-1}(E-1)\eta_t^2\|F_k(\mathbf{w}_t^k,\xi_t^k)\|^2 \\
\leq& \sum_{k=1}^{m} p_{k,t}\mathbb{E}\sum_{t_0}^{t-1}(E-1)\eta_{t_0}^2 G^2 \\
\leq& \sum_{k=1}^{m} p_{k,t}\mathbb{E}(E-1)^2\eta_{t_0}^2 G^2 \\
=& \sum_{k=1}^{c} p_{k,t}\mathbb{E}(E-1)^2\eta_{t_0}^2 G^2 + \sum_{k=c+1}^{m} p_{k,t}\mathbb{E}(E-1)^2\eta_{t_0}^2 G^2 \\
\leq& \sum_{k=1}^{c} p_{k,t}\mathbb{E}(E-1)^2\eta_{t_0}^2 G^2 + 4\eta_t^2(E-1)^2G^2 \\
\leq& 4\sum_{k=1}^{c} p_{k,t}\eta_t^2(E-1)^2G^2 + 4\eta_t^2(E-1)^2G^2 \\
\leq& 4\sum_{k=1}^{c} p_{k,0}\eta_t^2(E-1)^2G^2 + 4\eta_t^2(E-1)^2G^2,
\end{aligned} \tag{18}$$

where $p_{k,0}$ is the initial weight assigned to the kth client.

A.1.4 ***Convergence bound.*** Combining Equations.(3), (15), (16), and (18), we have

$$\begin{aligned}
\|\bar{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 =& \|\bar{\mathbf{w}}_t - \mathbf{w}^* - \eta_t\bar{g}_t\|^2 \\
&+2\eta_t\langle\bar{\mathbf{w}}_t - \mathbf{w}^* - \eta_t\bar{g}_t, \bar{g}_t - g_t\rangle + \eta_t^2\|\bar{g}_t - g_t\|^2 \\
\leq& (1-\mu\eta_t)\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 2\sum_{k=1}^{m} p_{k,t}\|\mathbf{w}_t^k - \bar{\mathbf{w}}_t\|^2 + 6L\eta_t^2\Gamma \\
&+ 2\eta_t\langle\bar{\mathbf{w}}_t - \mathbf{w}^* - \eta_t\bar{g}_t, \bar{g}_t - g_t\rangle + \eta_t^2\|\bar{g}_t - g_t\|^2.
\end{aligned} \tag{19}$$

Since $\mathbb{E}[g_t] = \bar{g}_t$, Therefore,

$$\begin{aligned}
\mathbb{E}\|\bar{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq& (1-\mu\eta_t)\mathbb{E}\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 2\mathbb{E}\sum_{k=1}^{m} p_{k,t}\|\mathbf{w}_t^k - \bar{\mathbf{w}}_t\|^2 \\
&+ 6L\eta_t^2\Gamma + 2\eta_t\mathbb{E}\langle\bar{\mathbf{w}}_t - \mathbf{w}^* - \eta_t\bar{g}_t, \bar{g}_t - g_t\rangle \\
&+ \mathbb{E}\eta_t^2\|\bar{g}_t - g_t\|^2 \\
\leq& (1-\mu\eta_t)\mathbb{E}\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 8\sum_{k=1}^{c} p_{k,0}\eta_t^2(E-1)^2G^2 \\
&+ 8\eta_t^2(E-1)^2G^2 + 6L\eta_t^2\Gamma + \eta_t^2\sum_{k=1}^{m} p_{k,t}^2\sigma_k^2 \\
=& (1-\mu\eta_t)\mathbb{E}\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 \\
&+ \eta_t^2[8\sum_{k=1}^{c} p_{k,0}(E-1)^2G^2 + 8(E-1)^2G^2 \\
&+ 6L\Gamma + \sum_{k=1}^{m} p_{k,t}^2\sigma_k^2]
\end{aligned} \tag{20}$$

We set $\eta_t = \frac{\beta}{t+\gamma}$ for some $\beta > \frac{1}{\mu}$ and $\gamma > 0$, such that $\eta_1 \leq min\{\frac{1}{\mu}, \frac{1}{4L}\} = \frac{1}{4L}$ and $\eta_t \leq 2\eta_{t+E}$. We want to prove $\mathbb{E}\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 \leq \frac{v}{\gamma+t}$, where $v = max\{\frac{\beta^2 B}{\beta\mu-1}, (\gamma+1)\mathbb{E}\|\bar{\mathbf{w}}_1 - \mathbf{w}^*\|^2\}$ and $B = 8\sum_{k=1}^{c} p_{k,0}(E-1)^2G^2 + 8(E-1)^2G^2 + 6L\Gamma + \sum_{k=1}^{m} p_{k,t}^2\sigma_k^2$.

Firstly, the definition of v ensures that $\mathbb{E}\|\bar{\mathbf{w}}_1 - \mathbf{w}^*\|^2 \leq \frac{v}{\gamma+1}$. Assume the conclusion holds for some t, we have

$$\begin{aligned}
\mathbb{E}\|\bar{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq& (1-\mu\eta_t)\mathbb{E}\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \eta_t^2 B \\
\leq& (1 - \frac{\beta\mu}{t+\gamma})\frac{v}{t+\gamma} + \frac{\beta^2 B}{(t+\gamma)^2} \\
=& \frac{t+\gamma-1}{(t+\gamma)^2}v + [\frac{\beta^2 B}{(t+\gamma)^2} - \frac{\beta\mu-1}{(t+\gamma)^2}v] \\
\leq& \frac{v}{t+\gamma+1}.
\end{aligned} \tag{21}$$

By the L-smoothness of $GM(\cdot)$, $\mathbb{E}[GM(\bar{\mathbf{w}}_t)] - GM^* \leq \frac{L}{2}\mathbb{E}\|\bar{\mathbf{w}}_t - \mathbf{w}^*\|^2 \leq \frac{L}{2}\frac{v}{\gamma+t}$.

Thus we have

$$\begin{aligned}
&\mathbb{E}[GM(\mathbf{w}_T)] - GM^* \\
\leq& \frac{2}{\mu^2}\cdot\frac{L}{\gamma+T}(\sum_{k=1}^{m} p_k^2\sigma_k^2 + 6L\Gamma + 8G^2 + 8G^2\sum_{k=1}^{c} p_{k,0} \\
&+ \frac{\mu^2}{4}\|w_0 - w^*\|^2).
\end{aligned}$$

A.2 Label Flipping

The attack that we target, the directed-deviation attack, has been shown to be the most powerful attack in federated learning [5], and specifically claims to be more effective than state-of-the-art untargeted data poisoning attacks for multi-class classifiers, that is, label flipping attack, Gaussian attack, and back-gradient optimization based attacks [11]. They show that the existing data poisoning attacks are insufficient and cannot produce a high testing error rate, not higher than 0.2 (at least on MNIST) in the presence of

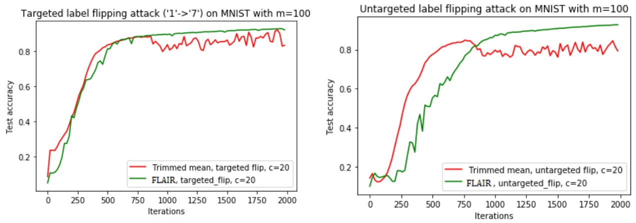

Figure 8: FLAIR's Performance against the targeted and untargeted label flipping attacks on the MNIST dataset. We observe that the attacks have some damage on the model, but FLAIR is able to remedy this for both attacks.

byzantine-robust aggregation techniques (Krum, trimmed mean, and median).

We have validated in our experiments that both state-of-the-art targeted and untargeted label flipping attacks (that fall under the class of data poisoning attacks) are not powerful enough on the CIFAR-10 and FEMNIST datasets and have neglible damage that is not significant enough to be observed in the test accuracy plot. The attacks, however, do have some damaging impact on the MNIST dataset, as observed in Figure 8 but when FLAIR is used, the damage is completely mitigated, as seen in Figure 8. Thus, we verify the claims from [5] and show that FLAIR's intuition is general enough to counteract both the more powerful directed deviation attacks and the weaker state-of-the-art data poisoning attacks.