Can Private Machine Learning Be Fair?

Joseph Rance, Filip Svoboda

Department of Computer Science & Technology
University of Cambridge
Cambridge, United Kingdom
{jr879, fs437}@cam.ac.uk

Abstract

We show that current SOTA methods for privately and fairly training models are unreliable in many practical scenarios. Specifically, we **(1)** introduce a new type of adversarial attack that seeks to introduce unfairness into private model training, and **(2)** demonstrate that the use of methods for training on private data that are robust to adversarial attacks often leads to unfair models, regardless of the use of fairness-enhancing training methods. This leads to a dilemma when attempting to train fair models on private data: either **(A)** we use a robust training method which may introduce unfairness to the model itself, or **(B)** we train models which are vulnerable to adversarial attacks that introduce unfairness. This paper highlights flaws in robust learning methods when training fair models, yielding a new perspective for the design of robust and private learning systems.

Code — https://github.com/Joseph-Rance/unfair-fl

1 Introduction

Ethically training Machine Learning systems on user data has long been a source of interest to the Machine Learning community (Arachchige et al. 2020; Cho, Wang, and Joshi 2020; Cao et al. 2020). We often wish to train a model that has uniform accuracy between different groups of participating users (fairness) without requiring private data to leave a user's local device; for example, this is important when training models on hospital patient data (Soltan et al. 2024; Nicola et al. 2020). We show that methods of privately training fair models can be unreliable in practical scenarios.

Previous work has demonstrated effective methods for fair and private training (Mohri, Sivek, and Suresh 2019; Li et al. 2020, 2021a), however these works assume no users are malicious. There are similarly many techniques for ensuring robustness to attacks from participating users during private training (Blanchard et al. 2017; Nguyen et al. 2023; Sun et al. 2019), yet few have examined the problem of ensuring privacy, fairness, and robustness *simultaneously*. We show that attempting to achieve all three attributes by combining these separate methods serves to reverse their effects.

Furthermore, we show that it is possible for adversarial attacks to seek to introduce unfairness into the training process. Therefore, methods that are not robust to these attacks cannot be assumed to be fair, leading to a dilemma when attempting to train a fair model on private data:

A. To defend against attacks on fairness, we introduce a robust training method which, as we claim in this paper, introduces unfairness itself.

B. We do not defend against attacks on fairness, leaving the model vulnerable to unfairness from these attacks.

Thus, due to **(1)** the threat of attacks on fairness and **(2)** our inability to ensure robustness without introducing unfairness, **we cannot be confident a private training procedure will result in a fair model**. For example, consider a model trained to predict interactions between pairs of molecules using private data produced by a group of participating companies (Mittone et al. 2023). One company may employ an attack on fairness to reduce accuracy on drugs produced by competitors. Either we risk vulnerability to this attack, or we risk discarding data on rare drug interactions.

Although, in *some* tasks, a robust learning method can be configured to introduce negligible unfairness while sufficiently protecting against most attacks, to our knowledge, there is no learning method that will have such a configuration on *any* given task. We make the following contributions:

- We propose a new type of training-time attack that introduces unfairness into the model. We show that attacks on fairness are well-founded and that they can be a threat in a wide range of practical learning scenarios (section 3).
- We explore the shortcomings of current strategies for ensuring robustness, fairness, and privacy and show that three methods for robust model training can introduce unfairness (section 4).
- We design a framework for testing robust learning methods that includes attacks on fairness.

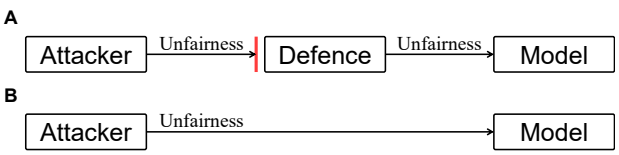

Figure 1: Two possible scenarios under an attack on fairness, both leading to unfairness in the global model.

2 Training Trustworthy Models on Private Data

Private machine learning. In private machine learning, we wish to train a model on user data without these data leaving the users' devices. In Federated Learning (FL) (McMahan et al. 2023) with DP-FedAvg (McMahan et al. 2018), clients locally train separate models on their data, that are then aggregated by a central server into a single, global model, which is then sent back to the clients to begin the next round of local training. Differential Privacy (DP) (Dwork 2006) can be applied to FL to allow the central server to obtain an effective global model without compromising user privacy. We focus on FL as a platform for ensuring privacy, but our claims do not necessarily only apply to the FL case.

Robust federated learning. FL is vulnerable to attacks from malicious clients which construct local models that introduce out-of-distribution behaviour to the global model (Bagdasaryan et al. 2019). The most effective methods for preventing these attacks eliminate local models that they determine lie outside benign model clusters. For example Krum discards models that are far from their closest neighbours in Euclidean space (Blanchard et al. 2017). The trimmed mean aggregation function instead discards the the n highest and lowest client weight values (Yin et al. 2021).

Alternatively, Sun et al. (2019) show that a weaker version of DP-FedAvg can improve robustness without directly rejecting specific clients. However, it has been shown that DP can have a disproportionate impact on clients that are further from the common model distribution (Bagdasaryan and Shmatikov 2019; Ganev, Oprisanu, and Cristofaro 2022; Farrand et al. 2020), so this defence may indirectly have very similar effects to methods like Krum and trimmed mean.

Fair federated learning. In this paper, we define a model to be fair if it has an even accuracy distribution between different subpopulations of the dataset. For example, a phone manufacturer is likely to require that face recognition accuracy is equal between different ethnic groups. Fair aggregation functions typically attempt to increase the weights assigned to clients that hold less common types of data to increase the impact of these subsets of data. For example, in agnostic FL (Mohri, Sivek, and Suresh 2019), the aggregator returns the loss produced by the worst-case weighting of client loss values, while q-FFL (Li et al. 2020) weights clients by their loss, raised to the power of a parameter q. Previous work has considered alternative notions of fairness (Mehrabi et al. 2021; Zhang et al. 2022), however, these are often due to distribution shift between our data and the task we aim to learn, which we do not investigate in this paper.

Combining fair and robust FL methods. When attempting to construct a training procedure with both fair and robust methods, we encounter a contradiction: robust aggregation eliminates local models that lie outside the distribution of common models, while fair aggregation attempts to increase the effect of these models. After combining these techniques, we therefore do not expect the full benefits of each method to persist. In this paper, we show that common robust aggregation methods can introduce unfairness during model training, and discuss how this affects our ability to prevent attacks on fairness. This is a fundamental problem with anomaly detection algorithms, which has so far not been directly addressed.

Related work. A similar idea has been previously presented by Wang et al. (2020), however their claims focus specifically on the implications for their proposed backdoor attack, while ours are broader. There has been previous work on constructing fair and robust training methods, however these methods either attempt to balance the effects of methods to those described above (Hu et al. 2022; Jin et al. 2023; Bowen et al. 2024), which we argue cannot always be an effective strategy, or solve the problem in non-standard setups, such as with personalisation (Li et al. 2021b), which are not applicable to the general case.

To our knowledge, no previous work has proposed attacks that target fairness in Federated Learning.

3 Attacks on Fairness in Federated Learning

In this section, we outline how fairness can be attacked in federated learning, and show that attacks on fairness are a threat in a wide range of realistic scenarios.

Threat model for attacks on fairness. We assume a similar threat model to Bagdasaryan et al. (2019): the attacker has access to the initial model sent to clients on the current round, G_t, and may control a small subset of clients to submit arbitrary parameters for aggregation. **The attacker cannot see the models submitted by benign clients.** The server may view the submitted parameters, but does not know which are submitted by the attacker and which are from benign clients. Such a threat could reasonably exist in systems where clients are untrusted (i.e. the attacker joins as a new client), or can be compromised (e.g. by malware).

In an attack on fairness, the attacker aims to train the global model, G_{t+1}, to have high accuracy on some target dataset, $D_T \subseteq D$, and low accuracy on the other data, $D_N = D \backslash D_T$. For example, they may want high accuracy on only classes 0 and 1. The attacker can therefore obtain a set of target parameters, $\mathbf{x}$, which they wish to substitute into G_{t+1}, by fine-tuning G_t using data in $\mathcal{D}_T$.

The attacker then uses $\mathbf{x}$ to compute a set of local, malicious parameters, $\mathbf{c}_0$, such that, after aggregation with the other clients' models, the resulting global model, G_{t+1}, is approximately equal to $\mathbf{x}$.

Model replacement attacks. If we directly submit $\mathbf{x}$ to the aggregator (i.e. $\mathbf{c}_0 = \mathbf{x}$), our parameters are unlikely to have a significant effect on the global model after aggregation with a much larger volume of benign parameters. Bagdasaryan et al. (2019) propose a more powerful strategy, *model replacement*, that allows the attacker to substitute the arbitrary target parameters, $\mathbf{x}$, directly into the global model. Bagdasaryan et al. (2019) train $\mathbf{x}$ to learn a *backdoor*, instead of introducing unfairness. Under the FedAvg (McMahan et al. 2023) aggregator, the attack by Bagdasaryan et al. (2019) is able to influence the global model to be $G_{t+1} = \mathbf{x} + \sum_{i=1}^{m-1} \frac{n_i}{n} (\mathbf{c}_i - G_t)$, where n_i is the size of client i's dataset, and $n = \sum_{i=0}^{m-1} n_i$. Bagdasaryan et al.

(2019) implicitly assume the model converges (and therefore $\mathbf{c}_i - G_t \approx 0$; see eqn. 3 in Bagdasaryan et al. (2019)), to obtain $G_{t+1} = \mathbf{x}$. However, unlike in the backdoor case, attacks on fairness prevent convergence.

The update prediction attack. The model replacement attack requires this convergence assumption because we do not know the parameters produced by other clients. For our attack on fairness, we solve this problem by letting the attacker *predict* the parameters submitted by other clients. If we assume that the difference between the mean client parameters, $\sum_{i=0}^{m} \frac{n_i}{n}(\mathbf{c}_i)$, and some other set of parameters, $\mathbf{w}$, that have been trained on data that is i.i.d. to the union of the clients' datasets is normally distributed with 0 mean, and variance a decreasing function of the amount of data seen during training, tending to 0 in the limit (**Theorem 1**), an attacker may be able to accurately predict the value $\sum_{i=0}^{m} \frac{n_i}{n}(\mathbf{c}_i)$. This forms the basis for the update prediction attack.

Instead of subtracting the global model, G_t, from our $\mathbf{x}$ to yield $\mathbf{c}_0$, as is the case in the model replacement attack, the attacker can now subtract their model predictions, $\mathbf{w}$, thus (approximately) eliminating *all* other terms in the FedAvg update rule.[1] We set the attacker's parameters to $\mathbf{c}_0 = \frac{n_0 - n}{n_0}\mathbf{w} + \frac{n}{n_0}\mathbf{x}$, so the FedAvg update becomes

$$
\begin{aligned}
G_{t+1} &= G_t + \sum_{i=0}^{m-1} \frac{n_i}{n}(\mathbf{c}_i - G_t) \\
&= G_t + \frac{n_0}{n}\left(\frac{n_0 - n}{n_0}\mathbf{w} + \frac{n}{n_0}\mathbf{x} - G_t\right) + \sum_{i=1}^{m} \frac{n_i}{n}(\mathbf{c}_i - G_t) \\
&= G_t + \frac{n_0 - n}{n}\mathbf{w} + \mathbf{x} + \sum_{i=1}^{m} \frac{n_i}{n}(\mathbf{c}_i) - \sum_{i=0}^{m} \frac{n_i}{n}(G_t) \\
&= \mathbf{x} + \sum_{i=1}^{m} \frac{n_i}{n}(\mathbf{c}_i) - \frac{n - n_0}{n}\mathbf{w} \\
&\approx \mathbf{x}
\end{aligned}
$$

Here we directly get $G_{t+1} = \mathbf{x}$ without the convergence assumption, allowing us to use any set of parameters for $\mathbf{x}$ (see fig. 2).

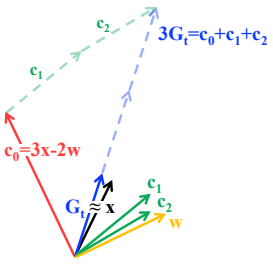

Figure 2: Visual representation of how the proposed attack. In practice, the angles between $\mathbf{w}$, and $\mathbf{x}$ tend to be small, so the length of $\mathbf{c}_0$ is not as extreme as this diagram suggests.

[1] In this case of a single attacker, the model replacement attack is a special case of our update prediction attack where we assume all clients submit 0-length updates.

Following our threat model, **no knowledge of parameters submitted by other clients is required by this attack.** The attacker only requires an approximate estimate for the amount of data, $n - n_0$, contributed by other clients. Such an estimate need not be exact and could be iteratively increased each round until an effective value is found.

Attacks on fairness are well-founded. Our update prediction attack on fairness assumes that, when a federated learning model trains with a large amount of private data, the variance in benign client parameters introduced by unknown training data is low. We now prove this for strongly convex functions.

We begin by considering the following general optimisation problem for client i:

$$\min_{\mathbf{c}_i \in \mathbb{R}^d} f(\mathbf{c}_i) = \min_{\mathbf{c}_i \in \mathbb{R}^d} \frac{1}{n_i} \sum_{j=0}^{n_i - 1} f_j(\mathbf{c}_i) \tag{1}$$

where each $f_i \in \mathbb{R}^d \to \mathbb{R}$ is a continuously differentiable function. We want to use minibatch gradient descent to solve this problem:

$$\mathbf{c}_i^{(k)} = \mathbf{c}_i^{(k-1)} - \frac{\alpha_k}{b} \sum_{j=0}^{b-1} \nabla f_{s_{k,j}}(\mathbf{c}_i^{(k-1)}) \tag{2a}$$

$$= \mathbf{c}_i^{(k-1)} - \alpha_k[\nabla f(\mathbf{c}_i^{(k-1)}) + \xi_k] \tag{2b}$$

where α_k is the learning rate, each $s_{k,j}$ is uniformly sampled at random from $\{0, ..., n_i - 1\}$, and ξ_k represents the noise introduced by sampling the training distribution on round k. This problem description and the following assumptions follow that of Li, Xiao, and Yang (2023), whose proof of SGD convergence to normally distributed models in the central case provides a basis for the following proofs. For simplicity, we set the learning rate to $\alpha_k = \alpha_1 k^{-1/2}$, which satisfies the assumptions of Li, Xiao, and Yang (2023).

We make the following assumptions.

(A1) Mean and covariance of ξ_k. $\forall \varepsilon > 0$. $\exists$ a symmetric, positive definite matrix, Σ, such that

$$\mathbb{E}[\xi_k | \mathcal{F}_{k-1}] = 0 = \lim_{n \to \infty} P(||\mathbb{E}[\xi_k \xi_k^T | \mathcal{F}_{k-1}] - \Sigma|| \geq \varepsilon)$$

where $\mathcal{F}_k = \sigma(x_0, \xi_1, \xi_2, \ldots, \xi_k)$ is the σ-algebra generated from the initialisation and noise terms up to round k.

(A2) L-smoothness of f. $\exists L$ such that

$$\forall x, y \in \mathbb{R}^d. \, ||\nabla f(x) - \nabla f(y)|| \leq L||x - y||$$

(A3) μ-strong convexity of f. $\exists \mu$ such that

$$\forall x, y \in \mathbb{R}^d. \, f(x) \geq f(y) + \nabla f(y)^T (x - y) + \frac{\mu}{2}||x - y||^2$$

(A4) Further smoothness condition for f. $\exists p_0, r_0, K_d > 0$ such that, for any $||x - x^*|| < r_0$,

$$||\nabla f(x) - \nabla^2 f(x^*)(x - x^*)|| \leq K_d ||x - x^*||^{1 - p_0}$$

(A5) Dataset size heterogeneity. If client i has a dataset of size n_i, modelled as a random variable, and $n = \sum_{c=0}^{m-1} n_i$ for m clients, then

$$\exists h \geq 1 \in \mathbb{R}. \, \forall i \in [0, 1, ..., m - 1]. \, m n_i \leq n h$$

With this definition, if $h = 1$, all client datasets must have the same amount of data, while as $h \to \infty$, the client data distribution constraints disappear.

Lemma 1 (variance for a single client)**.** Under assumptions (A1)-(A4), if $\frac{1}{\alpha_1} < 2\mu$, there exists some matrix, W^*, such that

$$k^{1/4}(\mathbf{c}_i^{(k)} - \mathbf{c}_i^*) \Rightarrow^k N(0, \alpha_1 W^*) \tag{3}$$

where $\Rightarrow^k$ denotes convergence in probability, $\mathbf{c}^*$ is the unique minimum of f, $k \in \mathbb{N}$ is large, and $W^*_{k,i,j} \in O\left(\frac{1}{b^2}\right)$.

Proof. This extends the result from Li, Xiao, and Yang (2023). Under the above assumptions, Li, Xiao, and Yang (2023) show that for $b = 1$ we have eq. (3) for some matrix W^*, where $AW^* + W^*A^T - d_0W^* = \Sigma$ and A is independent of all ξ_i.

Now consider the variance of ξ_k as b increases. We assume that the summed gradients are independent and have finite first 2 moments. Thus, for large b, by the classical central limit theorem, the gradient estimate, $\nabla\hat{f}(\mathbf{c}_i^{(k-1)})$, is unbiased and normally distributed:

$$\begin{aligned}\nabla\hat{f}(\mathbf{c}_i^{(k-1)}) &= \frac{1}{b}\sum_{j=0}^{b-1}\nabla f_{s_{k,j}}(\mathbf{c}_i^{(k-1)}) \\ &\sim N\left(\mathbb{E}_{f_j}[\nabla f_j(\mathbf{c}_i^{(k-1)})], \frac{\mathbb{V}_{f_j}[f_j(\mathbf{c}_i^{(k-1)})]}{b}\right)\end{aligned}$$

This yields a noise term, $\xi_k \sim N\left(0, \mathbb{V}[f_i(\mathbf{c}_i^{(k-1)})]/b\right)$, with variance inversely proportional to batch size.

The maximum element in the covariance matrix for $\text{vec}(\xi_k\xi_k^T)$ (where vec is a function that flattens a matrix into a vector) must be the variance of $(\xi_k)_i^2$ for some i. Since $(\xi_k)_i$ is normally distributed with variance $\frac{c}{b}$ for some constant c, we know that each element of this covariance matrix must be bounded by $\mathbb{V}_{\xi_k}[(\xi_k)_i^2] = \frac{2c^2}{b^2}$.

We have established that $AW^* + W^*A^T - d_0W^* = \Sigma$ and that the elements of the covariance matrix for $\xi_k\xi_k^T$ (and therefore also those of Σ) are in $O\left(\frac{1}{b^2}\right)$, so the elements of W^* must also be in $O\left(\frac{1}{b^2}\right)$. □

Now consider the FedAvg aggregation function (McMahan et al. 2023) to compute the global model, G, from the model $\mathbf{c}_i^{(u)}$ produced by each client i after u batches using the above SGD setup for the current training round:

$$G = \frac{1}{n}\sum_{i=0}^{m-1} n_i\mathbf{c}_i^u \tag{4}$$

where $n = \sum_{i=0}^{m} n_i$.

Theorem 1 (Variance of FedAvg)**.** Under assumptions (A1)-(A5), if $\frac{1}{\alpha_1} < 2\mu$ for each client, the global model, G, must be normally distributed with covariance matrix M_g such that $M_{g,p,q} \in O\left(1/\sqrt[4]{enm^3b^7}\right)$ for a large epoch index, e, and batch size, b.

Proof. From **Lemma 1**, we know that each $\mathbf{c}_i^{(e-1)}$ is an independent, normally distributed random variable with covariance matrix M_i, where $M_{i,p,q} \in O\left(1/\sqrt[4]{ub^8}\right) = O\left(1/\sqrt[4]{en_ib^7}\right)$, for large e and b. After applying FedAvg, we get

$$G \sim N\left(\sum_{i=0}^{m-1}\frac{n_i}{n}\mathbf{c}_i^*, \sum_{i=0}^{m-1}\frac{n_i^2}{n^2}M_i\right) \tag{5}$$

Since, by (A5), $\max_i \frac{n_i}{n} \in O\left(\frac{1}{m}\right)$ for all clients, i, the covariance matrix $M_g = \sum_{i=0}^{m-1}\frac{n_i^2}{n^2}M_i$ must have $M_{g,p,q} \in O\left(1/\sqrt[4]{en_im^4b^7}\right) = O\left(1/\sqrt[4]{enm^3b^7}\right)$. □

Therefore, by Chebyshev's inequality, **the probability our model prediction error is greater than γ is in** $O\left(1/\sqrt[4]{enm^3b^7\gamma^8}\right)$.

The above proof can similarly be adapted for SGD with momentum, using the relevant results from Li, Xiao, and Yang (2023). This proof does not extend to all non-convex models, however, as with model convergence in general, it is reasonable to assume that for a sufficiently smooth loss function and large enough batch size, because the attacker knows the model's initial parameters, the non-convex case is locally similar to the strongly convex case above.

The update prediction attack also assumes that $\mathbf{w}$ is an unbiased estimator of $\sum_{i=0}^{m}\frac{n_i}{n}(\mathbf{c}_i)$. This is false when there is heterogeneity between clients (FedAvg introduces some unfairness itself). The attacker could construct an unbiased estimator by directly locally simulating the entire FL training process, however we find that predicting $\mathbf{w}$ in a centralised manner is effective in practice.

Experimental results. We test the fairness attack for the datasets described in table 2 (Becker and Kohavi 1996; Krizhevsky 2009; Pushshift 2017). These datasets were selected to cover a range of tasks and to provide clear comparison with previous work (Bagdasaryan et al. 2019; Bhagoji et al. 2019; Wang et al. 2020; Nguyen et al. 2023; McMahan et al. 2023). Here, we substitute a single client's model with our malicious set of parameters, using the original client's model as the prediction $\mathbf{w}$ (i.e. we predict $\mathbf{w}$ using the same amount of data that would be held on a single client). For simplicity, we include the attack in every round. We additionally test its performance against the Krum, trimmed mean, and weak differential privacy defences. We select hyperparameters by performing a grid search over all reasonable combinations at multiple levels of granularity and present the median result across three trials in table 1. All experiments were performed on 2 NVIDIA RTX 2080 GPUs.

We record the change in fairness after the attack is introduced for each dataset-defence combination. The attack is effective at introducing unfairness into all three tasks. In practice, it may be preferable to perform a more subtle version of this attack. Although the size of the dataset needed to train $\mathbf{w}$ depends on the task, the attack remains effective even with the small local datasets available in the Reddit task. Table 2 shows that the Krum and trimmed mean defences are effective at preventing the attack on fairness, however we find that the weak-DP defence was not successful under any of the hyperparameter configurations we tested.

Dataset	Attack	No defence		Krum		Trimmed mean		Weak-DP	
		Acc.	Δ fairness	Acc.	Δ fairness	Acc.	Δ fairness	Acc.	Δ fairness
Census	None	84.81	0.00	84.82	0.00	84.83	0.00	75.66	0.00
	Fairness	81.57	53.44	84.64	-1.72	84.76	1.82	76.38	2.23
CIFAR-10	None	92.70	0.00	91.59	0.00	91.98	0.00	93.60	0.00
	Fairness	17.90	85.24	63.78	6.26	63.11	6.28	17.62	85.29
Reddit	None	18.08	0.00	17.82	0.00	18.08	0.00	08.55	0.00
	Fairness	4.52	118.94	17.97	-0.16	17.90	0.19	04.52	108.39

Table 1: Accuracy (%) achieved by different robust aggregation schemes for each dataset. The attack on fairness attempts to increase the accuracy on one subset of data while reducing the accuracy on another, so 'Δ fairness' indicates the increase in accuracy disparity between these two sets (lower is better).

Dataset	#Records	Model	#Clients/Per round
UCI Census	49k	3-layer FC	10 / 10
CIFAR-10	60k	ResNet-50	10 / 10
Reddit	2.3M	LSTM	10,000 / 100

Table 2: We train clients for 10, 2, and 5 epochs on i.i.d. data, for a total of 40, 120, and 100 rounds for the Census, CIFAR, and Reddit datasets respectively. We use the same augmentation scheme as Zagoruyko and Komodakis (2017) for the CIFAR-10 dataset and the *albert-base-v2* tokeniser (Lan et al. 2020) for the Reddit task. The attack reduces the accuracy of entries labelled as female, increases accuracy on classes 0 and 1, and reduces accuracy following the word 'I' for the Census, CIFAR, and Reddit tasks respectively.

Previous work has shown similar results for the weak-DP defence (Wang et al. 2020), although it is difficult to justify specifically why the defence is ineffective against this attack-task combination. However, recent works have also shown that weak differential privacy can introduce unfairness into a model (Bagdasaryan and Shmatikov 2019; Ganev, Oprisanu, and Cristofaro 2022; Farrand et al. 2020).

Attacks on momentum-based aggregators. Momentum-based aggregatiors (Reddi et al. 2021) are common in practice, which could lead to the attack on fairness becoming ineffective. We repeat our experiments for the Census task in table 1 for three different aggregators (table 3), finding that attacks on fairness remain effective against these aggregators without modification.

Aggregator	Attack	Overall	Δ fairness
FedAdaGrad	Baseline	84.58	-2.84
	Fairness	75.79	36.47
FedYogi	Baseline	78.72	43.76
	Fairness	80.78	51.87
FedAdam	Baseline	84.50	1.64
	Fairness	80.14	45.91

Table 3: Attacks on different aggregators for the Census task. All hyperparameters were identical to table 1, which accounts for the drop in initial accuracy and fairness. Δ fairness is calculated with respect to the baseline in table 1.

Attacks on fairness under data heterogeneity. We have shown attacks on fairness function on i.i.d. data distributions. In practice, this is not always the case (Yang et al. 2018; Hard et al. 2019; Huba et al. 2022). Furthermore, due to **(A5)**, heterogeneity should reduce the attack's stability.

To show that attacks on fairness can be a threat in settings where there is high heterogeneity, we repeat the baseline experiment from table 1, with the CIFAR-10 dataset distributed between clients using a log-normal label distribution across the clients parameterised by $\mu = 0$ and $\sigma \in \{0, 1, 2\}$. Figure 3 shows that heterogeneity reduces the attack's effectiveness. However, even at high σ values, it remains effective. Here, we do not include a defence, although we expect that increased heterogeneity would make detection more difficult (Ozdayi and Kantarcioglu 2021).

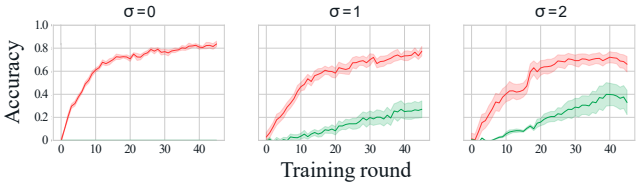

Figure 3: Accuracy per training round for the attack on fairness under different heterogeneity values (σ) for the CIFAR-10 task. The red (upper) line represents the accuracy on data the attack seeks to increase, while the green (lower) line represents the accuracy on other data. As heterogeneity increases, the attack's strength decreases. Without the fairness attack, the accuracy on both datasets is approximately equal.

4 Robust Aggregation Introduces Unfairness

In the previous section we have shown that attacks on fairness are a realistic threat against FL. This motivates the need for a training process that is robust to these attacks, without introducing unfairness itself. In this section we seek to answer the question **can we prevent adversarial attacks on federated learning without introducing unfairness?**

We focus this analysis on synthetic datasets, to help test whether these problems are likely to generalise to other defences that follow a similar construction, rather than repeating previous work to show unfairness can be introduced by specific defences in the wild (Wang et al. 2020).

Fairness impact of Krum and trimmed mean. Both the Krum and trimmed mean robust aggregation methods remove client models that lie far from a mode of the distribution. This could lead to unfairness because clients holding certain types of uncommon data are more likely to produce models that are far from clusters of common models (see fig. 4). Figure 5 shows that some clients hold meaningfully less common datasets even when there is low data heterogeneity between clients (some clients are consistently ranked less trustworthy than others when data is randomly distributed). Furthermore, removal of a small number of these clients can have a disproportionately high impact due to the uncommon nature of the data they hold.

We now show that, this can lead to the reduction - or, in this extreme case, elimination - of critical functionality from a model that is only present in a minority of clients.

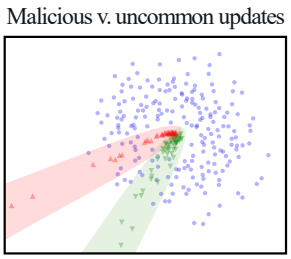

Figure 4: 2D projection of models produced by clients on the MNIST dataset (LeCun, Cortes, and Burges 1998). Points coloured green (inverted triangles) are produced by clients that only hold data with classes 0 and 1, while points coloured red (upright triangles) are produced by clients running a backdoor attack. Although a low-dimensional representation exaggerates the problem, there is little difference between malicious models and models trained on specific subpopulations of the dataset from a clustering perspective.

Figure 5: Client trustworthiness ranking on each round according to the Krum defence for the CIFAR-10 task. The attacker (red) is consistently ranked least trustworthy, however, we also observe some clients (e.g. the lower, blue line) are lower ranked than others (e.g. the upper, green line).

We construct the dataset shown in fig. 6, in which five clients (group A) do not have any data where the input begins with a 0, and one client (group B) does not have data where the input ends with a 0. This construction leads to the

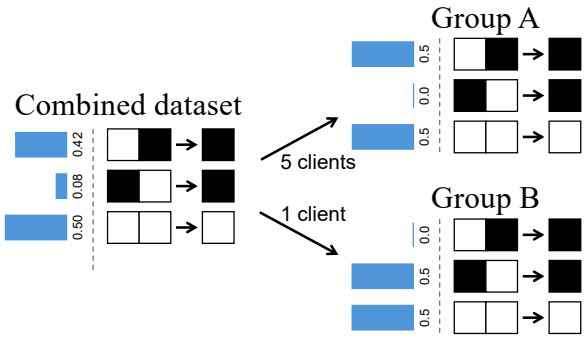

Figure 6: This dataset represents the OR function, where black squares indicate the value 1, and white the value 0. We split it so that the input $[0, 1]$ only occurs in 5/6 clients, while the input $[1, 0]$ only occurs in the remaining client.

client from group B producing different models compared to group A when training on a simple, fully-connected model.

As shown in table 4, both defence methods incorrectly determine that the group B client is malicious across multiple tests, leading to the uncommon functionality that is unique to this client ($[0, 1] \to 1$) being lost in the aggregated model. As this data is lost, fair aggregation methods that reweight uncommon models would be ineffective in this scenario. Although we eliminate more models (1) than there are attackers (0) on each round, this is realistic because we will not know how many attackers there are in practice.

Defence	Combined	Group A	Group B
No defence	100	100	100
Trimmed mean	92	100	50
Krum	92	100	50

Table 4: Accuracy (%) for each dataset under different defences.

Furthermore, the aggregator does not know how each local model is obtained (due to our privacy constraints) so there cannot exist *any* unsupervised anomaly-detection-based defence that would be able to identify group B as benign, while still rejecting all attacks. This is true because, if such a defence existed, it would continue to accept models from group B when the task is redefined as returning the value of the second input (i.e. $[0, 1] \to 1; [0, 0] \to 0; \ldots$). In this scenario, group B introduces behaviour that directly opposes the training goal, so it should be classed as malicious.

More generally, the local training function that produces models from local data by SGD is non-injective and thus has no left inverse, so overlap between the tail of the benign distribution and the set of malicious models is possible (see fig. 7). Shumailov et al. (2021) show that malicious parameters can be learnt by manipulating the order of clean data, which implies that this overlap exists in realistic scenarios. **Therefore, even without DP constraints, for some datasets it is *impossible* to detect all attacks without misidentifying some benign models as malicious**.

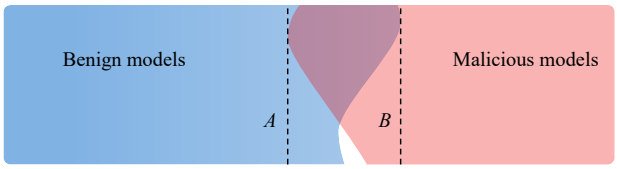

Figure 7: There may be overlap between the distribution of models produced by legitimate clients (blue, left) and the set of models produced by malicious processes (red, right). Methods based on anomaly detection must reject the tail of the benign distribution (e.g. by accepting only models to the left of boundary A, leading to unfairness due to the omission of uncommon data) or accept some malicious models (e.g. by accepting only models to the left of boundary B).

In our testing shown in table 1, we also find that accuracy disparity between common and uncommon data in the Census task[2] increased under all three defences, with the disparity growing by a median of 1.95, 0.53, and 51.72 for the trimmed mean, Krum, and weak DP defences respectively. Similar results to these have previously been observed in realistic setups, often showing a more significant reduction in fairness compared to these tasks (for example, Wang et al. (2020) 2020), although analysis of this issue has never been extended to the general problem that we investigate here. Thus, even without the use of techniques to evade detection by robust aggregators (Bagdasaryan et al. 2019), which can increase the difficulty of separating uncommon from malicious models, unfairness is introduced due to this overlap.

While robust aggregation methods that attempt to retain fairness have been presented, they are forced to make significant compromises compared to the methods studied here. For example, Fed-MGDA+ (Hu et al. 2022) employs a gradient clipping strategy, which is clearly weaker than the weak differential privacy defence described above.

Unfair-update detection: testing a new defence for fairness attacks. While Wang et al. (2020) have shown that verifying that a model does not contain any backdoors is computationally intractable, it is relatively simple to verify that a model is fair across a set of predetermined attributes with high confidence. This suggests a simpler defence for attacks that only attempt to introduce unfairness may be to measure the fairness impact of each client's model and assume clients that significantly reduce fairness are malicious.

When repeating the experiments for the CIFAR-10 task with this defence, the change in accuracy disparity (Δ fairness) is only 0.34 under the fairness attack. Additionally, the unfair-update detection algorithm initially appears to solve the problem of overlapping malicious and benign model distributions by accepting models based on their impact on the global model rather than based on how we believe they have been trained. However, a client which submits a model trained on new data that may be necessary to achieve a more fair *final* model (after convergence) is likely to have reduced accuracy/fairness in the short term, leading to its rejection in some scenarios. The problem arises from the attack's greedy setup: we select the clients that produce the most fair model *on the next round*, not those that result in a more fair *final* model, when it may be necessary to temporarily reduce fairness in order to achieve a higher final value.

[2]In our other two tasks, there is no clear dataset split that yields such uneven subpopulations

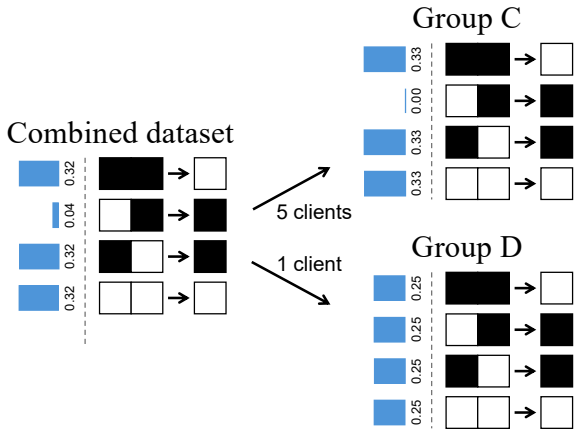

Figure 8: This dataset represents the XOR function. We split it so that the input $[0, 1]$ only occurs in 1 out of 6 clients.

Defence	Combined	Group C	Group D
No defence	100	100	100
Unfair-update det.	96	100	75

Table 5: Accuracy (%) per dataset under different defences.

5 Conclusion

We have shown that common defence methods can introduce unfairness and that attacks on fairness are a real threat to the federated learning training process. Furthermore, to our knowledge, there does not exist any defence that can ensure the robustness of the global model without using a method based on those analysed in this paper. Thus, assuming such a defence does not currently exist, **in the presence of untrusted clients, we cannot be confident that training on private data will result in a fair model**.

Even if robust aggregation may introduce a negligible amount of unfairness for *some* datasets, it is difficult to predict which datasets will present this problem and how to manage the tradeoff between fairness and robustness. This presents a risk, particularly for systems that are expensive to retrain. Future work would therefore benefit from approaching the problem of robustness in FL from a new, fairness-aware perspective.

Ethical Statement

We introduce a new attack that could be used against real systems. We provide an implementation of this attack, because we believe the benefits of improving the accessibility of tools to test robustness outweigh the disadvantages of making these attacks more accessible.

References

Arachchige, P. C. M.; Bertók, P.; Khalil, I.; Liu, D.; Çamtepe, S. A.; and Atiquzzaman, M. 2020. A Trustworthy Privacy Preserving Framework for Machine Learning in Industrial IoT Systems. *IEEE Transactions on Industrial Informatics*, 16: 6092–6102.

Bagdasaryan, E.; and Shmatikov, V. 2019. Differential Privacy Has Disparate Impact on Model Accuracy. arXiv:1905.12101.

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2019. How To Backdoor Federated Learning. arXiv:1807.00459.

Becker, B.; and Kohavi, R. 1996. Adult. UCI Machine Learning Repository.

Bhagoji, A. N.; Chakraborty, S.; Mittal, P.; and Calo, S. 2019. Analyzing Federated Learning through an Adversarial Lens. arXiv:1811.12470.

Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bowen, D.; Haiquan, W.; Yuxuan, L.; Zhao, J.; Ma, Y.; and Runhe, H. 2024. Fair and Robust Federated Learning via Decentralized and Adaptive Aggregation based on Blockchain. *ACM Trans. Sen. Netw.* Just Accepted.

Cao, X.; Fang, M.; Liu, J.; and Gong, N. Z. 2020. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. *ArXiv*, abs/2012.13995.

Cho, Y. J.; Wang, J.; and Joshi, G. 2020. Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies. arXiv:2010.01243.

Dwork, C. 2006. Differential Privacy. In Bugliesi, M.; Preneel, B.; Sassone, V.; and Wegener, I., eds., *Automata, Languages and Programming*, 1–12. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-35908-1.

Farrand, T.; Mireshghallah, F.; Singh, S.; and Trask, A. 2020. Neither Private Nor Fair: Impact of Data Imbalance on Utility and Fairness in Differential Privacy. arXiv:2009.06389.

Ganev, G.; Oprisanu, B.; and Cristofaro, E. D. 2022. Robin Hood and Matthew Effects: Differential Privacy Has Disparate Impact on Synthetic Data. arXiv:2109.11429.

Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; and Ramage, D. 2019. Federated Learning for Mobile Keyboard Prediction. arXiv:1811.03604.

Hu, Z.; Shaloudegi, K.; Zhang, G.; and Yu, Y. 2022. Federated Learning Meets Multi-Objective Optimization. *IEEE Transactions on Network Science and Engineering*, 9(4): 2039–2051.

Huba, D.; Nguyen, J.; Malik, K.; Zhu, R.; Rabbat, M.; Yousefpour, A.; Wu, C.-J.; Zhan, H.; Ustinov, P.; Srinivas, H.; Wang, K.; Shoumikhin, A.; Min, J.; and Malek, M. 2022. Papaya: Practical, Private, and Scalable Federated Learning. arXiv:2111.04877.

Jin, S.; Li, Y.; Chen, X.; Li, R.; and Shen, Z. 2023. Blockchain-Based Fairness-Enhanced Federated Learning Scheme Against Data Poisoning Attack. In *Smart Computing and Communication: 7th International Conference, SmartCom 2022, New York City, NY, USA, November 18–20, 2022, Proceedings*, 329–339. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-28123-5.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv:1909.11942.

LeCun, Y.; Cortes, C.; and Burges, C. J. 1998. The mnist database of handwritten digits.

Li, T.; Beirami, A.; Sanjabi, M.; and Smith, V. 2021a. Tilted Empirical Risk Minimization. arXiv:2007.01162.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021b. Ditto: Fair and Robust Federated Learning Through Personalization. arXiv:2012.04221.

Li, T.; Sanjabi, M.; Beirami, A.; and Smith, V. 2020. Fair Resource Allocation in Federated Learning. arXiv:1905.10497.

Li, T.; Xiao, T.; and Yang, G. 2023. Revisiting the central limit theorems for the SGD-type methods. arXiv:2207.11755.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2023. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629.

McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2018. Learning Differentially Private Recurrent Language Models. arXiv:1710.06963.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6): 1–35.

Mittone, G.; Svoboda, F.; Aldinucci, M.; Lane, N.; and Lió, P. 2023. A Federated Learning Benchmark for Drug-Target Interaction. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, 1177–1181. New York, NY, USA: Association for Computing Machinery. ISBN 9781450394192.

Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic Federated Learning. arXiv:1902.00146.

Nguyen, T. D.; Rieger, P.; Chen, H.; Yalame, H.; Möllering, H.; Fereidooni, H.; Marchal, S.; Miettinen, M.; Mirhoseini, A.; Zeitouni, S.; Koushanfar, F.; Sadeghi, A.-R.; and Schneider, T. 2023. FLAME: Taming Backdoors in Federated Learning (Extended Version 1). arXiv:2101.02281.

Nicola, R.; Jonny, H.; Wenqi, L.; Fausto, M.; Holger, R.; Shadi, A.; Spyridon, B.; Mathieu, G.; Bennett, L.; Klaus, M.; andS. Micah, O. S.; Ronald, S.; Andrew, T.; Daguang, X.; Maximilian, B.; and Jorge, C. 2020. The future of digital health with federated learning.

Ozdayi, M. S.; and Kantarcioglu, M. 2021. The Impact of Data Distribution on Fairness and Robustness in Federated Learning. In *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 191–196.

Pushshift. 2017. Reddit Comment Data. https://files.pushshift.io/reddit/comments/.

Reddi, S.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečný, J.; Kumar, S.; and McMahan, H. B. 2021. Adaptive Federated Optimization. arXiv:2003.00295.

Shumailov, I.; Shumaylov, Z.; Kazhdan, D.; Zhao, Y.; Papernot, N.; Erdogdu, M. A.; and Anderson, R. 2021. Manipulating SGD with Data Ordering Attacks. arXiv:2104.09667.

Soltan, A. A. S.; Thakur, A.; Yang, J.; Chauhan, A.; D'Cruz, L. G.; Dickson, P.; Soltan, M. A.; Thickett, D. R.; Eyre, D. W.; Zhu, T.; and Clifton, D. A. 2024. A scalable federated learning solution for secondary care using low-cost microcomputing: privacy-preserving development and evaluation of a COVID-19 screening test in UK hospitals.

Sun, Z.; Kairouz, P.; Suresh, A. T.; and McMahan, H. B. 2019. Can You Really Backdoor Federated Learning? arXiv:1911.07963.

Wang, H.; Sreenivasan, K.; Rajput, S.; Vishwakarma, H.; Agarwal, S.; yong Sohn, J.; Lee, K.; and Papailiopoulos, D. 2020. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. arXiv:2007.05084.

Yang, T.; Andrew, G.; Eichner, H.; Sun, H.; Li, W.; Kong, N.; Ramage, D.; and Beaufays, F. 2018. Applied Federated Learning: Improving Google Keyboard Query Suggestions. arXiv:1812.02903.

Yin, D.; Chen, Y.; Ramchandran, K.; and Bartlett, P. 2021. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. arXiv:1803.01498.

Zagoruyko, S.; and Komodakis, N. 2017. Wide Residual Networks. arXiv:1605.07146.

Zhang, F.; Kuang, K.; Liu, Y.; Chen, L.; Lu, J.; Wu, F.; Wu, C.; Xiao, J.; et al. 2022. Towards multi-level fairness and robustness on federated learning. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*.