Privacy-preserving and Fairness-aware Secure Aggregation in Federated Learning

Tonia El Bacha, Aftab Akram[0009−0003−2402−4058], Ibtissam Harrouche[0009−0002−8249−3672], Melek Önen[0000−0003−0269−9495]

Department of Digital Security, EURECOM, 06410 Biot, France
{tonia.el-bacha, aftab.akram, ibtissam.harrouche, melek.onen}@eurecom.fr

Abstract. Federated Learning (FL) faces a fundamental tension: correcting demographic bias requires collecting statistics, while protecting user privacy forbids exposing them. A recent solution resolves this through Secure Multi-Party Computation (MPC) executed on external servers, which adds deployment complexity and cost beyond what a standard FL system carries. We introduce **SaFer-FL**, a privacy-preserving federated learning framework that achieves group fairness using secure aggregation layer that protects both model updates and per-subgroup counts, from which the server derives label-aware reweighting coefficients without ever observing any individual client's contribution. To additionally guarantee Differential Privacy (ϵ-DP) against gradient-leakage and membership-inference attacks, each client injects a fractional share of the Symmetric Geometric Discrete Laplace (SGDL) mechanism whose per-client contributions sum to a single calibrated ϵ-DP perturbation at the aggregate. Evaluated on the COMPAS and MovieLens-1M datasets with gender as the protected attribute, SaFer-FL reduces Equalized Odds and Equal Opportunity gaps relative to standard FedAvg while preserving competitive accuracy, with a controllable privacy-fairness-utility trade-off.

Keywords: Federated Learning· Secure Aggregation · Fairness · Differential Privacy.

1 Introduction

The rapid proliferation of distributed systems and the increasing sensitivity of user data have driven the machine learning community toward privacy-preserving collaborative training paradigms. In this context, Federated Learning (FL) [10] has emerged as one of the most widely adopted frameworks, enabling multiple clients to jointly train a global model without ever sharing their raw data. By keeping data locally and exchanging only model updates, FL reduces privacy exposure compared to centralized training. However, the model updates shared during training are derived from sensitive local data and can still leak private information [12]. This vulnerability has led to extensive research on secure aggregation (SA) techniques that aim to protect individual client updates through

cryptographic mechanisms [9]. Although SA solves the model privacy issues in FL, models trained across heterogeneous and distributed clients during FL exhibit biased behavior. Indeed, standard FedAvg [10] which consists of computing an average of the updates of FL clients, implicitly favors clients with larger or more representative datasets and disadvantages others. Therefore, fairness has become an important focus in FL, just like privacy protection. Addressing such bias typically requires access to global statistics such as the distribution of sensitive attributes across clients [13] (e.g., gender-label combinations). However, these statistics themselves are sensitive and should not be exposed, creating a fundamental tension between fairness and privacy in FL.

Previous work *PrivFairFL-Pre* [11] attempts to resolve this tension with the computation of a fairness-based average of model updates combined with multi-party computation (MPC). Unfortunately, *PrivFairFL-Pre* suffers from some limitations in terms of fairness, real-world deployment, and performance. In this paper, we study this solution more in depth and identify that both the fairness-weight calculation and the underlying privacy-enhancing technology can be improved. Our new solution *SaFer-FL* integrates secure aggregation with an improved fairness weight calculation. To further protect the aggregated model against global membership inference attacks, *SaFer-FL* enables clients to apply a calibrated Signed-Gamma Discrete Laplace (SGDL) mechanism. Each client adds noise to its local updates before securing them with SA. As a result, the server only accesses a noisy aggregated global model, providing stronger privacy guarantees. Our experimental study shows that *SaFer-FL* improves fairness while keeping both the updates and the statistical information private.

Section 2 discusses the related work and section 3 mainly discusses the background. Section 4 identifies the limitations in the current solution and presents our approach to overcome them. In Section 5, the main proposed solution *SaFer-FL* is described which is further studied in terms of performance in Section 6. Section 7 provides conclusive remarks along with future directions.

2 Related Work

Fairness has received growing attention in FL as a distinct and equally important concern as privacy protection. Standard FedAvg implicitly favors clients with larger or more representative datasets, leading to pronounced accuracy disparities across devices in heterogeneous networks. [8] addresses this by proposing q-FFL, an optimization objective inspired by α-fair resource allocation in wireless networks, which assigns higher aggregation weights to clients with larger loss values so as to reduce the variance of per-device accuracy. [7] extends this direction with Ditto, a personalized FL framework that jointly addresses fairness and robustness against poisoning attacks through a global-regularized local objective. These approaches establish fairness-aware aggregation weighting as a necessary component of any realistic FL deployment. Crucially, however, they operate on plaintext updates and provide no cryptographic protection of the aggregation process itself. A further complication arises from the interaction be-

tween DP and fairness: [2] demonstrate that DP-SGD exacerbates pre-existing accuracy disparities, as the noise introduced by gradient clipping falls disproportionately on under-represented subgroups. [5] conducts a systematic analysis of when DP and fairness objectives are aligned or antagonistic and provides catalog available mitigation strategies. To the best of our knowledge, no existing protocol simultaneously addresses secure aggregation, differential privacy, and fairness within a single unified framework, which is the gap our work aims to fill.

3 Background

3.1 Federated Learning

FL is a machine learning approach designed for training models across many devices or organizations without collecting their raw data in one place. Instead of sending data to a central server, each FL client trains the model locally on its own data and only shares model updates. The FL server then combines these updates to improve a global model. However, even without sharing raw data, FL can leak information through the exchanged updates. A key privacy risk is the Membership Inference Attack (MIA) [12], where an attacker attempts to determine whether a specific data point was included in a client's training data.

3.2 Secure Aggregation

SA is a protocol involving multiple clients and an aggregator, where each client holds a private input and the aggregator only learns their sum without accessing the individual values. It is widely used in federated learning [4, 9] to privately aggregate client updates at each round. A SA protocol consists of three phases: **SA.Setup**, **SA.Protect**, and **SA.Agg**:

- **SA.Setup:** Users and the aggregator obtain public parameters and keys, generated either by a trusted third party or a distributed method. Each user $u \in \mathcal{U}$ receives a unique secret key sk_u, while the aggregator holds the aggregation key sk_0.
- **SA.Protect:** Each user locally protects its input $x_{u,\tau}$ with sk_u, for time period τ and sends the protected value to the aggregator.
- **SA.Agg:** After collecting all protected inputs, the aggregator computes the sum of user inputs for time period τ, using sk_0.

3.3 Fairness in Machine Learning

Machine-learning models learn from the past and when that past is unequal, they tend to reproduce, and sometimes amplify, the same inequalities at scale. In high-stakes domains such as credit, hiring, and criminal justice, this means decisions can systematically disadvantage demographic groups defined by sensitive

attributes such as gender, race, or age. *Group fairness* formalises this concern by requiring the model's predictions and error rates to be similar across these groups. Let $\hat{y} \in \{0,1\}$ be a classifier's prediction, $y \in \{0,1\}$ the ground-truth label, and $g \in \mathcal{G}$ a protected attribute. We use two standard notions of group fairness. **Equalized Odds** requires the prediction to be independent of g conditional on the true label, so both groups should have the same true-positive rate *and* the same false-positive rate:

$$\Pr(\hat{y} = 1 \mid y = 1, g_i) = \Pr(\hat{y} = 1 \mid y = 1, g_j), i \neq j$$
$$\Pr(\hat{y} = 1 \mid y = 0, g_i) = \Pr(\hat{y} = 1 \mid y = 0, g_j), i \neq j.$$

This is the relevant criterion when both kinds of mistakes: false positives and false negatives carry meaningful costs, and neither should fall disproportionately on one group (e.g. in recidivism prediction, where wrongly flagging a low-risk defendant and wrongly releasing a high-risk one both cause harm).
Equal Opportunity relaxes the requirement to the positive class only: both groups should have the same true-positive rate, regardless of how the false-positive rates behave:

$$\Pr(\hat{y} = 1 \mid y = 1, g_i) = \Pr(\hat{y} = 1 \mid y = 1, g_j), i \neq j.$$

This is the relevant criterion when failing to identify that a positive example carries the larger cost (e.g. missing a qualified candidate from a protected group).

4 Privacy-Preserving Fairness Aware Federated Learning: Limitations and Our Approach

In this section, we revisit *PrivFairFL-Pre* [11], a privacy-preserving solution for fairness-aware federated learning. We then discuss its key limitations and finally provide an overview of our proposed approach, explaining how it addresses each of these limitations.

4.1 PrivFairFL-Pre

PrivFairFL-Pre is a privacy-preserving bias mitigation technique designed for FL environments. The method aims to reduce demographic bias before model training by reweighing training samples according to the distribution of sensitive attributes and class labels across all participating clients. Unlike traditional centralized fairness methods, *PrivFairFL-Pre* achieves this without requiring clients to reveal their raw data or sensitive attributes to a central server.

The framework combines FL and MPC. Each client locally computes counts of samples belonging to different sensitive groups and label categories, then secret shares these statistics before sending them to MPC servers. The servers collaboratively aggregate the counts and compute fairness-aware sample weights while never accessing the underlying sensitive information in plaintext. Formally, let each client u hold dataset $D_u = \{(x_i, y_i, s_i)\}$, where x_i denotes the feature

vector of sample i, $y_i \in \{0,1\}$ is the corresponding class label (binary classification) and $s_i \in \{p, un\}$ (protected vs unprotected) represents the sensitive attribute of that sample. Each client computes local joint counts: $LC_u(s,y) = \sum_{(x_i,y_i,s_i)\in D_u} 1[s_i = s, y_i = y]$. These local statistics are secret-shared and sent to MPC servers. The servers securely aggregate them to obtain global counts: $C(s,y) = \sum_{u=1}^{\mathcal{U}} LC_u(s,y)$. The total global count is computed as: $N^{'} = \sum_{s\in\{p,un\}} \sum_{y\in\{0,1\}} C(s,y)$. Finally, fairness-aware weights are defined as: $W(s,y) = \frac{N^{'}}{4\cdot\tilde{C}(s,y)}$. These weights are broadcast to all clients and used during federated training to reweight the loss function: $\mathcal{L}_u = \sum_{(x_i,y_i,s_i)\in D_u} W(s_i,y_i)\,\ell(f(x_i),y_i)$.

4.2 Limitations

Imbalanced Weight Normalization: The *PrivFairFL-Pre* weighting mechanism computes the normalization factor as $N^{'} = \sum_{s\in\{p,u\}} \sum_{y\in\{0,1\}} \tilde{C}(s,y)$ which aggregates all samples without considering the relative size of each sensitive group. This assumes that protected and unprotected groups are equally represented across the data distribution. However, in practical FL settings, sensitive groups are often less represented, with one group significantly dominating the overall data distribution across clients. Consequently, the normalization term is dominated by the majority group, while minority groups receive disproportionately large weights $W(s,y) = \frac{N^{'}}{4\cdot\tilde{C}(s,y)}$. Although minority groups obtain higher weights, the optimization process can still remain biased toward the majority group due to its overwhelming presence in training and client datasets. Therefore, the reweighting strategy does not fully account for cross-group population imbalance, which degrades fairness performance.

MPC-Based Solution: *PrivFairFL-Pre* relies on MPC for aggregating local counts, where clients secret-share their values and a set of servers jointly reconstruct global counts. While this preserves privacy, it introduces a strong dependency on a multi-server MPC setup and assumes correct and honest execution of the aggregation protocol. In practice, this creates a deployment bottleneck, since the system requires reliable coordination among multiple, non-colluding servers. From a scalability perspective, MPC introduces non-negligible communication overhead because each client must distribute secret shares to multiple servers, and secure reconstruction requires multiple rounds of interaction. This becomes increasingly expensive in large-scale FL settings with thousands of clients, making the aggregation step a major performance bottleneck.

4.3 Our Approach

In this section, we present the design goals of our proposed framework, *SaFer-FL*, which is developed to address the limitations identified in the previous section. The detailed architecture and methodology of *SaFer-FL* are described in Section 4.

Cross-group Normalization Rule: We improve the normalization by making sure that each group's sample count is adjusted using the *client representation* of the other group. This helps prevent the larger group from dominating the final value. Let $N_p = C(p,0)+C(p,1)$ and $N_{un} = C(un,0)+C(un,1)$ denote the global sample counts of the privileged and unprivileged groups, and let C_p and C_{un} denote the global counts of privileged and unprivileged *clients*. We define the client fractions as $\alpha = \frac{C_p}{C_p+C_{un}}$ and $\beta = \frac{C_{un}}{C_p+C_{un}}$. The improved normalization term is computed as $N^{'}_{\text{new}} = \alpha\, N_{un} + \beta\, N_p$. This way, both groups are fairly balanced when computing the final weights, so that no single group dominates the training process and fairness is improved.

SA for Fairness: SA computes fairness-weights without exposing any client's sensitive data. Each client locally computes local counts and sends them to the server through a secure aggregation protocol. The server only sees the combined global statistics, not individual contributions. These aggregated values are then used to estimate fairness metrics and derive reweighting factors that reduce bias during training. This ensures fairness computation is done at a global level, while fully preserving client-level privacy.

SA for Privacy-preserving FL: In FL, clients do not send raw data to the server, but they still transmit model updates or gradients. Even these gradients can leak sensitive information about the training data through attacks such as membership inference attacks (MIA). SA addresses this issue by ensuring that the server never sees individual client updates at all. Instead, each client masks or encrypts its gradient, and only the aggregated sum of all client updates is revealed. This prevents the server (or an adversary observing updates) from isolating any single client's contribution. As a result, SA significantly reduces the risk of information leakage from gradients, which is a known vulnerability in standard FL.

Symmetric Geometric Discrete Laplace (SGDL) Mechanism: Standard DP mechanisms such as the Laplace and Gaussian mechanisms are not well-suited for SA based on homomorphic encryption [9]. Although SA reveals only the aggregate value, an honest-but-curious server can still track how these aggregated value changes across rounds when the participating set varies, enabling MIA. A common mitigation is to add local DP noise before aggregation. However, this becomes problematic in SA because classical DP mechanisms generate real-valued noise in $\mathbb{R}$, while mostly SA schemes [9] operate over integers in $\mathbb{Z}_q$ with modular addition. As a result, Laplace or Gaussian noise cannot be directly embedded into the cryptographic pipeline. Quantizing real-valued noise to fit $\mathbb{Z}_q$ introduces deterministic rounding effects that break the original DP guarantees, while floating-point representations are not preserved under modular arithmetic. Consequently, inserting standard Laplace or Gaussian noise inside SA either breaks correct decryption or invalidates the DP proof, making these mechanisms unsuitable for iterative SA. In order to solve this problem, and also to add

another layer of privacy protection to protect against global mem- bership inference attacks, we propose to integrate the Symmetric Geometric Discrete Laplace (SGDL) distribution, the natural integer-valued analogue of Laplace. SGDL(p) places mass $\Pr[X = k] = \frac{1-p}{1+p}p^{|k|}$ on each integer $k \in \mathbb{Z}$, with $p = \exp(-\epsilon/\Delta)$ calibrated to the privacy budget ϵ and the ℓ_1-sensitivity Δ. For a query with bounded sensitivity, adding a single SGDL sample to the integer-valued output yields ϵ-DP by the same calibration argument as the continuous Laplace, but in the discrete domain so the privacy proof transfers without modification, and no fixed-point approximation enters the analysis. SGDL is also infinitely divisible, a sample can be split into k independent fractional shares, each distributed as $\boldsymbol{\eta}_i \sim \mathrm{NB}(1/k, p) - \mathrm{NB}(1/k, p)$, whose sum reconstructs it. This lets each participating client draw its own share locally and add it to its quantized contribution, so the aggregate carries an exact SGDL noise with no trusted curator.

5 SaFer-FL

In this section, we describe *SaFer-FL*, which leverages SA to compute fairness-weights for each client and to train the global federated model in a privacy-preserving manner. We consider a threat model in which the FL server is honest-but-curious, following the protocol correctly but potentially attempting to infer sensitive information, and all FL clients are assumed to behave honestly throughout the training process. We also assume secure communication channels between clients and the server, ensuring that all exchanged messages are encrypted and protected from external adversaries. *SaFer-FL* ensures the privacy of clients' inputs against passive adversaries and its security proofs follow the framework outlined in [1] .*SaFer-FL* consists of three phases: (1) *Setup Phase* whereby FL clients are registered and each party receives its keying material (2) *Fairness-weight calculation Phase* whereby the fairness-weights are calculated in a privacy-preserving way and (3) *Training phase* whereby model parameters are aggregated and updated in several *FL Rounds*.

5.1 Setup Phase

The *Setup Phase* comprises two steps: *Registration* and *Key Setup*. Each of these steps is executed only once. At the *Registration* step, on request by client $u \in \mathcal{U}$, a trusted dealer (TD) calls $\mathsf{SA.Setup}$ and generates the secret keys sk_u for each client and aggregation key sk_0 for the server along with relevant public parameters pp required by the clients and server. We assume that an authenticated channel is defined for each pair of clients and between each client and the server.

5.2 Fairness-weight Calculation.

During *Fairness-weight Calculation* phase, each FL client u first locally counts how many of its training samples fall into each of the four buckets defined by the sensitive attribute $s \in \{un, p\}$ and class label $y \in \{0, 1\}$ i.e., binary

SaFer-FL

Setup Phase:

Registration: Security parameter λ and clients in $\mathcal{U}$ such that $|\mathcal{U}| = n$.
Trusted Dealer (TD):
1) Run $(pp, \{ek_u\}_{u\in\mathcal{U}}, sk_0) \leftarrow \mathbf{SA.Setup}(1^\lambda, n)$.
2) Send (pp, ek_u) to each client $u \in \mathcal{U}$, and (pp, sk_0) to the server.
Client $u \in \mathcal{U}$**:**
1) Receive (pp, ek_u) from TD.
Server:
1) Receive $(pp, ak_u$ from TD.
2) Broadcast to all clients the set of clients $\mathcal{U}$.
Key Setup:
Client $u \in \mathcal{U}$**:**
1) Collect $\mathcal{U}$ the set of all registered clients.
2) Establish a virtual secret channel with each other client in $\mathcal{U}$.

Fairness-weight Calculation:
Clients ($\forall u \in \mathcal{U}$)**:**
1) Compute local count vector $LC_u = [LC(un, 0)_u,\ LC(un, 1)_u,\ LC(p, 0)_u,\ LC(p, 1)_u,\ \mathbb{I}[T(u) = 0],\ \mathbb{I}[T(u) = 1]]$.
2) $\mathbf{LC_u} \leftarrow \mathbf{SA.Protect}(pp, sk_u, \tau_0, LC_u)$: Protect each Local count vector and send $\mathbf{LC_u}$ to server.
Server:
1) Collect all $\mathbf{LC_u}$ from clients.
2) $C \leftarrow \mathbf{SA.Agg}(pp, sk_0, \tau_0, \{\mathbf{LC_u}\}_{u\in\mathcal{U}})$: computes the global count as $C = \sum_{u\in U} \mathbf{LC}_u = [C(un, 0),\ C(un, 1),\ C(p, 0),\ C(p, 1),\ C_{un},\ C_p]$
3) Computes the privileged and unprivileged client fractions as $\alpha = \frac{C_p}{C_{\text{un}}+C_p}$ and $\beta = \frac{C_{\text{un}}}{C_{\text{un}}+C_p}$, where C_p and C_{un} are the global counts of privileged and unprivileged clients.
4) Computes the normalization term $N'_{\text{new}} = \alpha \cdot N_{\text{un}} + \beta \cdot N_p$, with $N_p = C(p, 0) + C(p, 1)$ and $N_{\text{un}} = C(\text{un}, 0) + C(\text{un}, 1)$ the global sample counts of the privileged and unprivileged groups.
5) Computes the fairness weights as: $W(s, y) = \frac{N'_{\text{new}}}{4\cdot C(s,y)}$ and broadcasts these weights to all clients in $\mathcal{U}$.
Clients ($\forall u \in \mathcal{U}$)**:**
1) Store $W(s, y)$ as per-sample loss reweighting co-efficients for local training.

Training Round:
Clients ($\forall u \in \mathcal{U}$):
1) Load global model: $\theta_{u,\tau} \leftarrow \theta^{\tau-1}$.
2) For $e = 1, \ldots, E$ local epochs, for each batch b:

$$\mathcal{L}_u = \frac{\sum_i W(s_i, y_i) \cdot \ell(f(x_i), y_i)}{\sum_i W(s_i, y_i)}, \quad \theta_{u,\tau} \leftarrow \theta_{u,\tau} - \eta\, \nabla_{\theta_u} \mathcal{L}_u.$$

3) $\tilde{\theta}_{u,\tau} \leftarrow \theta_{u,\tau} + \text{SGDL}(n = |\mathcal{U}|,\ \epsilon,\ \Delta_f)$, where SGDL is the distributed Laplace mechanism as described in section 4.3.
4) $Y_{u,\tau} \leftarrow \mathbf{SA.Protect}(pp, ek_u, \tau, \tilde{\theta}_{u,\tau})$.
5) Send $Y_{u,\tau}$ to the server.
Server:
1) Collect $Y_{u,\tau}$ from all $u \in \mathcal{U}$.
5)Aggregate: $\theta_\tau \leftarrow \mathbf{TJL.Agg}(pp, sk_0, \tau,\ \{Y_{u,\tau}\}_{u\in\mathcal{U}})$.

Algorithm 1: Detailed description of SaFer-FL algorithm.

classification. Specifically, each client computes a six-element vector: $LC_u = [LC(un,0)_u,\ LC(un,1)_u,\ LC(p,0)_u,$
$LC(p,1)_u,\ \mathbb{I}[T(u)=0],\ \mathbb{I}[T(u)=1]]$ where $LC(un,0)_u$ and $LC(un,1)_u$ are the counts of unprivileged samples with labels 0 and 1 respectively. Similarly, $LC(p,0)_u$ and $LC(p,1)_u$ are the counts of privileged samples with labels 0 and 1 respectively. The value $T(u) \in \{0,1\}$ indicates group membership, where $T(u) = 1$ denotes the privileged group and $T(u) = 0$ denotes the unprivileged group. Each FL client u then protects LC_u with the **SA** secret key sk_u (generated in the previous *Key Setup* step) using **SA.Protect** and sends them to the FL server. The server performs their aggregation using **SA.Agg** and computes the global sums as $C = \sum_{u \in U} \mathbf{LC}_u = [C(un,0),\ C(un,1),\ C(p,0),\ C(p,1),\ C_{un},\ C_p]$ where each capital letter denotes the sum across all clients. Here, $C_{un} = \sum_i \mathbb{I}[T(i) = 0]$ represents the total number of unprivileged clients and $C_p = \sum_i \mathbb{I}[T(i) = 1]$ represents the total number of privileged clients. From these aggregated global counts, the server computes fairness-weights that assign higher importance to underrepresented groups. Specifically, the server calculates the total group sample counts as $N_p = C(p,0) + C(p,1)$ for the privileged group and $N_{un} = C(un,0) + C(un,1)$ for the unprivileged group. The client fractions are then computed as $\alpha = \frac{C_p}{C_p + C_{un}}$ and $\beta = \frac{C_{un}}{C_p + C_{un}}$. The server next computes an improved normalization term $N'_{\text{new}} = \alpha \cdot N_{un} + \beta \cdot N_p$, which balances the two group sizes by weighting each by the proportion of the other. Finally, the server computes the fairness-weights as: $W(s,y) = \frac{N'_{\text{new}}}{4 \cdot C(s,y)}$ and broadcasts these weights back to all clients for use in their local training loss functions. This phase, like the *Setup Phase*, is executed only once, before the actual training of the ML model.

5.3 Training Phase

During the *Training Phase* each client loads the global model $\theta_{u,\tau} \leftarrow \theta^{\tau-1}$ and performs local training using fairness-aware weighting. Specifically, it computes the weighted loss $\mathcal{L}_u = \frac{\sum_i W(s_i,y_i) \cdot \ell(f(x_i),y_i)}{\sum_i W(s_i,y_i)}$, where unprivileged samples receive higher weights and privileged samples receive lower weights. The model is then updated as $\theta_{u,\tau} \leftarrow \theta_{u,\tau} - \eta \nabla_{\theta_u} \mathcal{L}_u$. This scheme amplifies the influence of unprivileged samples while reducing that of privileged ones, thereby promoting fairness. After computing the model update, the client adds DP noise calibrated to the desired privacy budget (ε, δ) using the SGDL mechanism as defined in section 4.3. The noisy updates are then protected via **SA.Protect** using the secret key sk_u and forwarded to the server. The server aggregates all protected updates using **SA.Agg** with sk_0. This ensures that the server can only access the aggregated values, which already include the added noise.

6 Experiments

6.1 Experimental Setting

We evaluate SaFer-FL on two tabular datasets, MovieLens-1M (movie-rating prediction) [6] and COMPAS (recidivism prediction) [3], with *gender* being considered as the protected attribute in both. Every client trains a logistic-regression classifier. Unless stated otherwise we use a learning rate of 0.03, 300 global rounds, $E = 2$ local epochs per round, and full client participation (frac = 1.0, i.e. $|\mathcal{U}^\tau| = n$ in every round). The number of FL clients is set to $n = 75$ in MovieLens-1M with a local batch size of 75 and $n = 20$ in the COMPAS dataset with a local batch size of 20. All other hyperparameters are kept identical across the two datasets.

6.2 Pre-processing of MovieLens-1M and COMPAS

MovieLens-1M. We construct the dataset from the raw MovieLens-1M files (`ratings.dat`, `users.dat`, `movies.dat`), where each example is a single (user, movie) rating. The feature vector $\mathbf{x}_i \in \mathbb{R}^m$ concatenates user attributes — sex, a categorical age band, and a one-hot encoding of occupation — with movie attributes, the number of years between rating and release, and a multi-hot encoding of genres. The label y_i is binary: a rating strictly greater than 3 is positive (1), otherwise negative (0). MovieLens is *naturally federated*: each user is a client u holding only their own ratings $\mathcal{D}_u$, so no artificial cross-client partition is required. We subsample $n = 75$ users with approximately 32% female representation and a positive-label rate matched to the global rate (≈ 0.58), so that the federation preserves the demographic and label skew of the full dataset.

COMPAS. We construct the dataset from the full COMPAS release published by ProPublica (`compas-scores.csv`), where each example corresponds to a single defendant. The feature vector $\mathbf{x}_i \in \mathbb{R}^{10}$ includes sex together with demographic, criminal-history, case-timing, and COMPAS-score features. The binary label y_i is `is_recid` (1 if recidivism is observed during the follow-up window, 0 otherwise); we use the full COMPAS file rather than the two-year subset to retain more samples per client, which better reflects a realistic federated deployment. Unlike MovieLens, COMPAS is not naturally federated: each row is a defendant rather than a user holding a personal history. We therefore partition the corpus across $n = 20$ synthetic clients using a label-wise Dirichlet split with concentration $\alpha = 1.0$, yielding moderate label heterogeneity and unequal shared sizes. We use $\alpha = 1.0$ rather than a near-IID setting ($\alpha = 10$) because SaFeR-FL's reweighting operates at the sample level, so increasing or decreasing the Dirichlet concentration produces comparable fairness behavior, and the lower α corresponds to the more realistic federated scenario.

In both datasets the protected attribute is *sex*, encoded as Female= 0 and Male= 1, which induces the four label–group cells $(F, 0), (F, 1), (M, 0), (M, 1)$ used by both the SaFer-FL bucket weights and the fairness metrics defined below.

After the dataset-specific construction described next, each client's dataset $\mathcal{D}_u$ is split 80%/20% into local train and test sets with a fixed seed.

6.3 Evaluation Metrics

We measure utility with global test-set accuracy: after each round, the server evaluates the current global model on the union of all clients' 20% held-out test partitions. Group fairness is reported with two standard metrics computed across the gender groups:

$$\Delta\text{EOP} = |\text{TPR}_F - \text{TPR}_M|,$$
$$\Delta\text{EODD} = \tfrac{1}{2}\big(|\text{TPR}_F - \text{TPR}_M| + |\text{FPR}_F - \text{FPR}_M|\big).$$

Lower values of ΔEOP and ΔEODD indicate fairer behavior across groups.

6.4 Results

In order to conduct a comparative experimental study, we have implemented the standard federated-averaging protocol (denoted as **FL**), the prior solution proposed in [11] (**PrivFairFL-Pre**), and our solution **SaFer-FL**. Table 1 reports the model accuracy and the two fairness metrics on MovieLens-1M and COMPAS; we could not reproduce **PrivFairFL-Pre** on COMPAS because the authors did not study it; hence, we put the dashes in that row. **SaFer-FL** is evaluated without DP and with distributed SGDL noise at $\epsilon \in \{1, 5, 10\}$. We observe that without DP, SaFer-FL nearly matches the non-private FL baseline in accuracy on both datasets while outperforming PrivFairFL-Pre on MovieLens-1M and substantially improving both fairness gaps. Adding DP noise yields the expected privacy–utility trade-off on accuracy: a smaller ϵ injects more noise and lowers accuracy, while accuracy recovers toward the no-DP level as ϵ grows. The relationship between ϵ and fairness is less regular than the one between ϵ and accuracy: both fairness gaps move non-monotonically with the privacy budget. Several factors may contribute to this behavior, including the variance introduced by the noise at small budgets and the interaction between the noise and the reweighting mechanism. Overall, the no-DP run yields the best fairness on MovieLens-1M and remains the strongest or tied on COMPAS, suggesting that the cleanest fairness gains arise when the reweighting operates on noise-free updates.

When compared with the solution in [1], **SaFer-FL** incurs the same complexity. For communication and storage, the client requires $O(n + m)$ and the server requires $O(n^2 + nm)$. For computation, the client requires $O(n^2 + m)$, while the server requires $O(n^2 + nm)$. Here, n denotes the number of clients and m denotes the number of model parameters.

7 Conclusion and Future Work

We have presented SaFer-FL whereby the cryptographic building block is already protecting model updates in FL and it also carries the demographic statistics

Table 1: Utility and fairness on MovieLens-1M and COMPAS for the federated baselines and our method (SaFer-FL), with and without differential privacy. Accuracy is in percent; ΔEODD and ΔEOP are fairness gaps (lower is fairer).

	MovieLens-1M			**COMPAS**		
Method	Acc.	ΔEODD	ΔEOP	Acc.	ΔEODD	ΔEOP
FL	62.39%	0.141	0.091	70.79%	0.076	0.121
PrivFairFL-Pre	58.52%	0.051	0.042	—	—	—
SaFer-FL (without DP)	61.97%	0.014	0.0003	70.38%	0.018	0.025
SaFer-FL (DP, $\epsilon = 1$)	58.84%	0.042	0.001	67.98%	0.043	0.070
SaFer-FL (DP, $\epsilon = 5$)	59.84%	0.038	0.036	68.84%	0.018	0.029
SaFer-FL (DP, $\epsilon = 10$)	61.18%	0.032	0.020	69.38%	0.023	0.035

required to make those updates fair. The same fault-tolerant secure-aggregation layer privately collects (per-group, label) counts, the server derives label-aware reweighting coefficients from the resulting global totals, and a distributed signed-gamma discrete-Laplace mechanism gives the entire pipeline a formal ϵ-DP guarantee — all without external MPC servers, trusted curators, or any infrastructure beyond what FL already deploys. On COMPAS and MovieLens-1M, this design narrows both equalized-odds and equal-opportunity gaps relative to standard FedAvg while maintaining competitive accuracy, and by varying the privacy budget we map the controllable trade-off space between privacy, fairness, and utility. In FL, privacy alone can hide unfairness in the model without fixing it. Fairness alone can risk revealing private information about clients. *SaFer-FL* solves both problems together by using SA to compute a fair global model in federated learning while still protecting privacy. For future work, we plan to investigate stronger threat models while guaranteeing both privacy and correctness of the aggregated output in the presence of malicious users and/or aggregators.

References

1. Akram, A., Pham, H.N.H., Önen, M., Gritti, C.: SAAFL: Secure aggregation for label-aware federated learning. In: ICT Systems Security and Privacy Protection — 40th IFIP TC 11 International Conference (IFIP SEC 2025). pp. 32–46. IFIP Advances in Information and Communication Technology, Springer, Cham (2025). https://doi.org/10.1007/978-3-031-92882-6_3
2. Bagdasaryan, E., Poursaeed, O., Shmatikov, V.: Differential privacy has disparate impact on model accuracy. In: Advances in Neural Information Processing Systems 32 (NeurIPS 2019). pp. 15479–15488. Curran Associates (2019), https://proceedings.neurips.cc/paper/2019/hash/fc0de4e0396fff257ea362983c2dda5a-Abstract.html
3. Barenstein, M.: Propublica's compas data revisited. arXiv preprint arXiv:1906.04711 (2019)
4. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical Secure Aggregation for Privacy-Preserving Machine Learning. In: ACM SIGSAC Conference on Computer and Communications Security (CCS) (2017)

5. Fioretto, F., Tran, C., Van Hentenryck, P., Zhu, K.: Differential privacy and fairness in decisions and learning tasks: A survey. arXiv preprint arXiv:2202.08187 (2022), https://arxiv.org/abs/2202.08187
6. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis) **5**(4), 1–19 (2015)
7. Li, T., Hu, S., Beirami, A., Smith, V.: Ditto: Fair and robust federated learning through personalization. In: Proceedings of the 38th International Conference on Machine Learning (ICML 2021). Proceedings of Machine Learning Research, vol. 139, pp. 6357–6368. PMLR (2021), https://proceedings.mlr.press/v139/li21h.html
8. Li, T., Sanjabi, M., Beirami, A., Smith, V.: Fair resource allocation in federated learning. In: 8th International Conference on Learning Representations (ICLR 2020) (2020), https://openreview.net/forum?id=ByexElSYDr
9. Mansouri, M., Önen, M., Ben Jaballah, W.: Learning from Failures: Secure and Fault-Tolerant Aggregation for Federated Learning. In: 38th Annual Computer Security Applications Conference (ACSAC) (2022)
10. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Agüera y Arcas, B.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017). Proceedings of Machine Learning Research, vol. 54, pp. 1273–1282. PMLR (2017), https://proceedings.mlr.press/v54/mcmahan17a.html
11. Pentyala, S., Neophytou, N., Nascimento, A., De Cock, M., Farnadi, G.: Privfairfl: Privacy-preserving group fairness in federated learning. arXiv preprint arXiv:2205.11584 (2022)
12. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership Inference Attacks Against Machine Learning Models. In: Proc. of the IEEE Symposium on Security and Privacy (S&P) (2017)
13. Zhou, Z., Chu, L., Liu, C., Wang, L., Pei, J., Zhang, Y.: Towards fair federated learning. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 4100–4101 (2021)