Fairness in Federated Learning

Meriem Fouad
Heinz College
Carnegie Mellon University
mfouad@andrew.cmu.edu

Raj Shah
Heinz College
Carnegie Mellon University
rajshah@andrew.cmu.edu

Miguel Rivera-Lanas
Heinz College
Carnegie Mellon University
mriveral@andrew.cmu.edu

Abstract

Federated learning (FL) is a framework that allows distributed learning for neural networks and other machine learning models. Disparate servers with private data iteratively 1) share learned model parameters with a central server and 2) receive aggregated model parameters from the central server. This allows isolated servers to benefit from sharing data without actually revealing data with each other or with the central server. Our goal is to incorporate elements of fairness in Federated Learning, ensuring representation and fair outcomes are achieved, without sacrificing privacy and reducing accuracy.

In this paper, we use a feed forward Neural Network to classify recidivism cases from the COMPAS dataset, and evaluate the individual and group fairness scores given through different algorithmic techniques. We implement a novel approach of combining individual fairness and group fairness in our parameter aggregation methods to prioritize clients that balance both. Through our approach, we find that we can improve individual fairness without sacrificing accuracy, though there are effects on group fairness.

Introduction

Over the last few decades, the increasing promises and proven use cases for deep learning algorithms has been closely followed by another trend - the public's increasing awareness and anxiety about privacy risks. The two trends are connected through the unavoidable fact that training practical deep learning models requires vast and rich data sets. Privacy researchers have designed methods to more securely train deep learning models on sensitive personal data through methods that either avoid data sharing in the first place, such as through synthetic data generation [13] or methods that attempt to limit what can be learned about individuals from data collections by injecting statistical noise into key computations [14]. While these methods are promising, they still often require authorization of data sharing between data owners and external parties. This is a potential issue for two reasons. First, personal data necessary for machine learning applications are often linked to improper or insecure data sharing practices. Second, for some data owners, data sharing may simply not be legally feasible out of an abundance of caution.

In this paper, we focus on a methodology that specifically addresses this issue - Federated Learning (FL). The main idea is that a central server can build a machine learning model based on data sets that are distributed across multiple client devices while preventing communication of data between clients and the central server or between clients.

One challenge we hope shed light on is that related to evaluating and designing for **fairness** in the development of machine learning algorithms via FL. In centralized ML contexts, substantial advancement has been made in the field of fairness measures. These can related to group fairness - the parity of performance between identifiable groups - or individual fairness - the extend to which similar individuals are treated similarly [11]. While clients in the FL context can evaluate these metrics for their individual datasets, the central server's model does not have access to the data necessary to do the same.

The goal of this paper is to explore measures to train a fairness-aware Federated Learning feed forward Neural Network model. We build on an in-processing mechanism called *FairFed*. We first implement the client learning aggregation strategy proposed, using the Federated Learning framework Flower to replicate the paper's results. We then introduce and test extensions to the ideas proposed in this paper.

Literature Review

Federated learning

Advances in information technology and distributed systems over the last few decades has given rise to many applications characterized by "edge devices". That is, systems where data collection and data processing between services and clients occurs between client and edge devices rather than between clients and a centralized server.

Federated Learning applies this trend to the machine learning context. Federated Learning (FL), initially proposed by McMahan et al, refers to a technology that enables training machine learning models on data from different sources (edge devices) without the need to store the data at a central location [9]. This model of learning allows edge devices to benefit from training ML models based on their pooled data without revealing their data to either each other or to the central location. One current, high visibility implementation of FL is Google's use of FL to train next-word prediction models for smartphone keyboards [9].

Fairness In FL

Advances in group fairness in ML based on sensitive features cannot easily be applied to FL settings since each client only has access to their own datasets. Local fairness only considers the disparity of the model at each client whereas global fairness evaluates a model on the entire dataset.

Hammam and Dutta explore group fairness-accuracy trade-offs in the FL setting and the limits on the trade offs between global and local fairness metrics. They identify three sources of unfairness in federated learning: Unique Disparity, Redundant Disparity, and Masked Disparity [4]. The authors also introduce a convex optimization framework for quantifying accuracy-fairness trade-offs that systematically explores trade-offs between accuracy and both global and local fairness metrics [4].

Generally, methods to impose fairness criteria in FL learning contexts can occur via pre-processing, in-processing, or post-processing measures. Pre-processing measures rectify training data such that agnostic learning algorithms produce less unfair heuristics (local dibiasing ref). In-processing measures customize the ML algorithm to directly train for less unfair models. The FairFed algorithm is an in-processing measure. Finally, post-processing measures revise a model's resulting prediction scores to make predictions fairer (Menon et al.'18).

Why Healthcare Data Analysis Benefits from FL

By sharing only model updates rather than raw data, federated learning significantly reduces the risk of data breaches and ensures compliance with stringent privacy regulations like HIPAA. This decentralized learning architecture not only protects patient information but also enhances the quality of models by leveraging diverse data sources from multiple institutions. Consequently, it promotes patient trust and adherence to legal frameworks while simultaneously advancing the capabilities of machine learning in healthcare.

The Nature paper by Rieke et al. highlighted several crucial technical considerations for implementing Federated Learning in healthcare. One key challenge to keep in mind while using FL for healthcare is managing the data heterogeneity in terms of quality, format, and distribution. The heterogenous nature of healthcare data can have a significant impact on model performance. The paper also emphasized the need for efficient communication protocols to handle the substantial network overhead involved in transferring model updates between participants. Additionally, ensuring the integrity and security of the federated learning process is paramount since there is potential for malicious actors to attack the system and gain access to the private information of the patients.

Model and Architecture Description

Our ablations were performed on the following architectural setup, using the model description found in the table below.

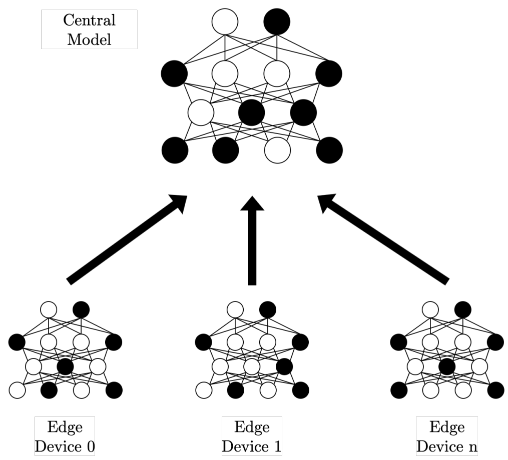

Figure 1: Federated Learning Setup - Clients train on their respective datasets, and then their weights are aggregated at a centralized node.

Attribute	Description
Type	Fully Connected Feedforward Neural Network.
Architecture	Three linear layers with $5 \rightarrow 16 \rightarrow 8 \rightarrow 1$ neurons.
Activation Functions	Leaky ReLU for the first two layers, no activation for the output layer.
Regularization	Batch Normalization layers after the first and second linear layers.
Loss Criterion	BCE with Logit Loss.
Optimizer	Adam

Table 1: Model Description - this model was initialized identically across each client.

The strategies used to aggregate client weights are what we tested in our experiment, alongside the hyperparameters that control those strategies.

Variable	Description
α	Measure of Data Heterogeneity - High values of alpha denote uniform distribution of the sensitive class across clients.
β	Measure of Fairness Budget - Higher values of Beta focus more on fairness and less on accuracy.
γ	Measure of Group vs Individual - Higher values of Gamma focus more on individual fairness, while lower ones focus on group fairness.

Table 2: Hyperparameter Description - these values were changed in our ablations to better understand the tradeoff between different models of fairness on different datasets.

Dataset

We used the COMPAS dataset as used by Ezzedine et al. COMPAS, which stands for *Correctional Offender Management Profiling for Alternative Sanctions*, is a dataset created by the ProPublica newsroom (Larson et al.) to investigate the COMPAS software - a legislative software used to identify which defendants were at risk for recidivism [7]. This dataset allowed ProPublica to report on the disenfranchisement of Black defendants, who were misclassified twice as much as White defendants for committing a repeated violent offense. The dataset contains non-anonymized name data, alongside race, marital status, and legal status. The race of the defendant is the sensitive attribute A.

This dataset is commonly used in fairness model evaluations [2], making it appropriate for our use case. To make it appropriate for our problem, we added a column identifying whether or not someone was Caucasian. This simulates the real world concern around ensuring a given model produces the same level of accuracy for minority defendants compared to defendants of a majority group.

A dirichlet partitioner was used to spread the minority class across different clients. The partitioner was able to take in an α value and use that to either evenly distribute the minority class across all clients, or produce a strong skew of classes among clients. This allowed us to simulate real world conditions, where members of the minority class may not be present on every training session across every device.

Evaluation Metrics

Our models were graded on 3 evaluation metrics:

- **Accuracy**: A measure of how often the correct recidivism outcome was assigned to each row of the given dataset.
- **Equal Opportunity Difference (EOD)**: Commonly used in centralized learning to measure group fairness. Equal opportunity assess whether a model's predictions perform equally well for different groups of our sensitive attribute A (Is Caucasian). A predictor $\hat{Y}$ is considered fair if the true positive rate is independent of the sensitive attribute A [3].

$$EOD = P(\hat{Y} = 1|A = 0) - P(\hat{Y} = 1|A = 1) \tag{1}$$

- **Individual Fairness Score (IFS)**: A score used to measure how much variance there is between members of the same prediction class. A predictor $\hat{Y}$ is considered fair if there is low variance in the outcome spaces of the two classes [11].

$$IFS = \frac{1}{2}\left(\mathbb{E}\left[(p-\mu_0)^2 \mid Y=0\right] + \mathbb{E}\left[(p-\mu_1)^2 \mid Y=1\right]\right) \tag{2}$$

These three metrics allow us to explore how fairness in different measures can be achieved without sacrificing model performance.

Loss Function

We used BCE Loss with a combined Sigmoid Layer to calculate our Loss.

$$l_n = [y_n \cdot log(\sigma(x_n)) + (1 - y_n) \cdot log(1 - \sigma(x_n))] \tag{3}$$

When the outcome for a given term is supposed to be $y_n = 0$, we only look at the second term, measuring how far $\sigma(x_n)$ is from 1. This was considered appropriate for our use case of predicting recidivism, as strongly confident and incorrect outcomes are given a strong penalty term.

Baseline Selection

The baseline selected is the FairFed algorithm proposed by Ezzeldine et al. [3]. FairFed is a fairness-aware aggregation algorithm to enhance group fairness in federated learning through a server-side approach. Traditional FL algorithms minimize the weighted average of the loss across all clients by weighting based on the size of each client's respective dataset. Instead, FairFed uses each client's fairness metric to update the weights aggregation. We explain the details of the weight aggregation technique in the next section.

FairFed is a state of the art model in fairness-aware FL settings. It has been shown to provide fairer models than other leading FL models. In particular, we selected the FairFed algorithm due to the following advantages:

- **Providing flexibility to clients to apply chosen local debiasing**: Fairfed is agnostic to the applied local debiasing method. Under FairFed, each client can apply a debiasing method of its choosing to its local dataset so clients can customize their debiasing method based on their datasets needs. This offers clients flexibility.
- **Enabling data decentralization and client privacy:** By performing debiasing locally, FairFed ensures that no sensitive information about a client's dataset is shared with the server. Local debiasing is generally safer than global debiasing (i.e debiasing at the server-side instead of client) and less likely to lead to data leakage.
- **Improving group fairness in data heterogeneous settings:** If all clients had homogeneous data, local debiasing would be enough to ensure group fairness. However, data homogeneity is highly unlikely. When faced with heterogeneous data distributions across clients, local debiasing is not enough to ensure fairness on the global data distribution. This highlights the need for FairFed's fairness-aware aggregation strategy which can improve performance in highly heterogeneous distribution settings [3].

Through integrating a client side (local debiasing) and a server side (fairness-aware aggregation strategy) approach, FairFed achieves group fairness while maintaining data de-centralization and client's privacy.

Baseline Model Description

The FairFed Aggregation Strategy

The FairFed algorithm proposed by Ezzeldine et al. [3] introduces one main novel idea to the standard federated learning set up: **a fairness-aware aggregation methodology**. In each round of training, the server aggregates the fairness metrics of all clients and adjusts the weights by favoring the clients whose fairness metric more closely matches the global fairness metric. This is in contrast to a standard FL set up where the weights are adjusted to favors clients with bigger datasets. We explain the FairFed aggregation strategy as proposed by Ezzeldine et al. below [3].

Let's define the following notation:
- EOD_{global} be the global EOD fairness metric aggregated by the server across all clients K;
- EOD_k be the local EOD fairness metric for client k;
- w_k^t be the weight for client k at time t
- n_k is the size of the dataset for client k
- β is a parameter to choose the fairness budget for each training round. Higher values of β will result in fairer models, but with a potential trade off on accuracy.
- θ^t is the global parameter as time t that minimizes the loss across all clients K.

Before the first round of training, we initialize the global parameter θ_0 and the weights such that:

$$w_k^0 = \frac{n_k}{\sum_{k=1}^{K} n_k} \tag{4}$$

For each round of training, the server computes global fairness metric EOD_{global} and sends it to the clients. Next, the client computes its metric gap Δ_k^t (equation (2)) and calculates its weight adjustment accordingly as shown in equation (3) below:

$$\Delta_k^t = |EOD_{global}^t - EOD_k^t| \tag{5}$$

$$\bar{w}_k^t = \bar{w}_k^{t-1} - \beta * (\Delta_k - \frac{1}{K}\sum_{k=1}^{K} \Delta_k) \tag{6}$$

The server can aggregate the weight updates and send the aggregated weight update $\sum_{k=1}^{K} \bar{w}_k^t$ back to clients to compute the new weights:

$$w_k^t = \frac{\bar{w}_k^t}{\sum_{k=1}^{K} \bar{w}_k^t} \tag{7}$$

Next, the server can calculate the new global parameter θ^{t+1} such that:

$$\theta^{t+1} = \frac{\sum_{k=1}^{K} \bar{w}_k^t \theta_k^t}{\sum_{k=1}^{K} \bar{w}_k^t} \tag{8}$$

The FairFed aggregation strategy prioritizes clients whose local fairness metric is closer to the global metric by assigning them higher weights and penalizes clients whose local fairness is significantly different from the global metric by reducing their weights.

Baseline Implementation

To set up a federated learning environment, we use Flower, an open source FL framework [12]. We customize the Flower implementation by changing the aggregation strategy from a standard FL aggregation method where weights are updated based on a client's dataset size, using the FedAvg algorithm prposed by McMahan et al., to the fairness-aware aggregation strategy (FairFed) explained in section 4.

Results

α Value	Mean Client Accuracies	Mean Client Fairness	Min Client Accuracies	Max Client Fairness	FairFed Accuracies	FairFed Fairness
0.2	0.6871	0.1443	0.6729	0.1744	0.6016	0.0821
0.5	0.6707	0.1836	0.6251	0.2375	0.6162	0.0958
10	0.6274	0.1404	0.4445	0.2009	0.6251	0.1005
5000	0.6740	0.1723	0.6518	0.2030	0.5927	0.0851

Table 3: Results of FairFed under different alpha values compared to non-Federated clients. Note that Fairness should be closer to 0, while Accuracy should be closer to 1.

Table 1 shows results of our replication of the FairFed. For different values of data heterogeneity, measured by alpha. Homogeneous data distributions have higher α values while heterogeneous data distributions lower α values.

The results align with the results from Ezzeldin et al., as applying the FairFed aggregation strategy results in higher fairness in heterogeneous settings.

Implemented Extensions and Experimentation

Individual Fairness

The contribution of the *FairFed* paper, which inspired this work, was in developing a group-fairness aware aggregation algorithm for federated learning and demonstrating that its utility was comparable to *FedAvg* while providing significant fairness improvements over local debiasing methods alone. This was shown to be especially true at high levels of data heterogeneity.

In extending this work, we proposed extendeding the evaluation of fairness criteria discussed in Ezzeldin et al., which only measured group fairness measures. Berk et. al discuss, when evaluating the fairness of machine learning models, notions of both individual and group fairness should be considered [11].

Individual fairness ensures that people who are 'similar' with respect to the classification task receive similar outcomes; whereas, group fairness measures average statistical disparities in how a model classifies groups. One important reason to consider both measures is the possibility of improving group fairness measures by simply achieving group parity by increasing false positive incidence (for example) in order to match that of another. To this end, we consider both group and individual fairness measures.

Formulation

Updating the notation:

- β: Controls adjustment of weights based on fairness deviations.
- γ: Denotes the weight for prioritizing **individual fairness (IFS)** over **group fairness (EOD)**. γ ranges between 0 (only group fairness) and 1 (only individual fairness).
- EOD: EOD fairness metric, tracked per client k and aggregated.
- IFS: Individual fairness metric, tracked per client k and aggregated.
- Δ_k^{EOD}: Deviation of local EOD from global EOD for client k.
- Δ_k^{IFS}: Deviation of local IFS from global IFS for client k.
- w_k^t: The weight assigned to client k at time t.
- n_k: The size of the dataset for client k
- θ^t: The global parameter as time t that minimizes the loss across all clients K.

Just as before, we initialize the global parameter θ_0 and the weights such that:

$$w_k^0 = \frac{n_k}{\sum_{k=1}^{K} n_k} \tag{9}$$

For each round of training, the server aggregates global fairness metric EOD_{global} and IFS_{global}, and the deviation is calculated for each client.

$$\Delta_k^{EOD} = |EOD_{global}^t - EOD_k^t| \tag{10}$$

$$\Delta_k^{IFS} = |IFS_{global}^t - IFS_k^t| \tag{11}$$

We then do a weighted combination of both deviations for each client:

$$\Delta_k^t = \gamma \cdot \Delta_k^{IFS} + (1 - \gamma) \cdot \Delta_k^{EOD} \tag{12}$$

And finally, do a weight adjustment:

$$\bar{w}_k^t = max(1 - \beta \cdot (\Delta_k^t - \frac{1}{K} \sum_{k=1}^{K} \Delta_k^t), 0) \tag{13}$$

The server can aggregate the weight updates and send the aggregated weight update $\sum_{k=1}^{K} \bar{w}_k^t$ back to clients to compute the new weights as discussed previously, to calculate a new global parameter.

$$w_k^t = \frac{\bar{w}_k^t}{\sum_{k=1}^{K} \bar{w}_k^t} \tag{14}$$

$$\theta^{t+1} = \frac{\sum_{k=1}^{K} \bar{w}_k^t \theta_k^t}{\sum_{k=1}^{K} \bar{w}_k^t} \tag{15}$$

This strategy allows us to leverage γ to prioritize EOD or IFS, and β to control how aggressively weights are adjusted based on fairness deviations.

Results

The table below shows how different values of α, β, and γ impacted our measures of accuracy, group fairness, and individual fairness.

α **Value**	β **Value**	γ **Value**	**Accuracy**	**EOD**	**IFS**
3	0.05	0	0.6283	0.1118	0.462
		0.25	0.6566	0.1875	0.1754
		0.5	0.6242	0.1661	0.1462
	0.1	0	0.6396	0.1801	0.4417
		0.25	0.6461	0.1265	0.42
		0.5	0.6178	0.1179	0.3844
	0.15	0	0.6493	0.0555	0.2809
		0.25	0.6793	0.175	0.2729
		0.5	0.6404	0.1342	0.1940
10	0.05	0	0.6761	0.1394	0.2653
		0.25	0.6445	0.1963	0.3075
		0.5	0.6558	0.1915	0.3251
	0.1	0	0.6323	0.157	0.3832
		0.25	0.6429	0.1889	0.3199
		0.5	0.6186	0.1537	0.18
	0.15	0	0.6153	0.1207	0.3215
		0.25	0.6461	0.1296	0.3746
		0.5	0.621	0.0518	0.2048
500	0.05	0	0.6485	0.1148	0.0865
		0.25	0.5943	0.0892	0.2161
		0.5	0.6599	0.1668	0.559
	0.1	0	0.6178	0.1781	0.2992
		0.25	0.6493	0.1381	0.2219
		0.5	0.6364	0.1102	0.35102
	0.15	0	0.6412	0.10492	0.3366
		0.25	0.62672	0.0872	0.20181
		0.5	0.63724	0.18345	0.2267

Table 4: Results showing our custom strategy, under different α, β, and γ values. Note that accuracy values should ideally be 1, while Equal Opportunity Difference (EOD) and Individual Fairness Score (IFS) should be closer to 0.

Discussion

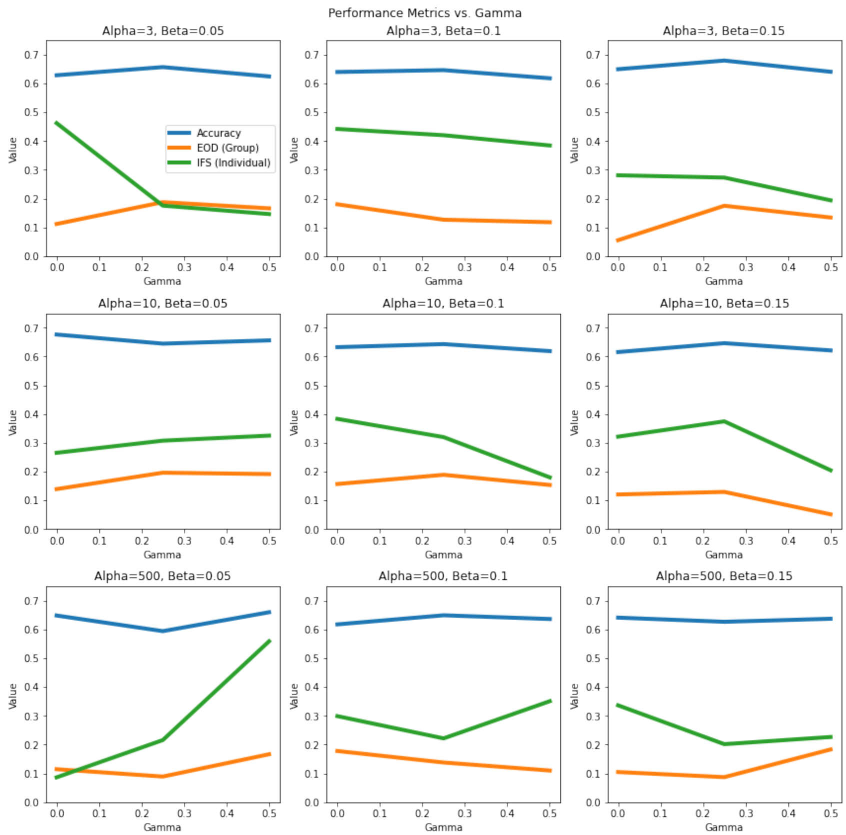

Figure 2: Graphic Visualization of our results - note that lower values of α indicate higher skew, and higher values of β imply more prioritization put towards Fairness. Changes in γ change weights allocated to Individual v.s. Group Fairness.

The results show that fairness scores are extremely sensitive to changes in all three values. Changes in α tended to change the difference between a model's performance in Individual Fairness versus Group Fairness when Individual Fairness was not prioritized - this can be seen in the left column of the graph, for γ values of 0. Surprisingly, changes in α also changed how the models reacted to changes in γ - this can be seen by how Individual Fairness scores decreased as γ increased when there was no skew in the data distribution.

Increases in β, at least for higher skew distributions, tended to promote better Individual Fairness scores and better Group Fairness scores. And these changes come with low variation in Accuracy scores.

Our key finding is that appending Individual Fairness into the existing FairFed strategy does boost a model's overall performance when it comes to Individual Fairness without a strong cost to Group Fairness or Accuracy, as long as a proper **fairness budget** is allocated for the model. Increases in γ did not come at a cost to decreases in accuracy nor penalties to Group Fairness - showing that FairFed can be modified to pay attention to multiple forms of Fairness.

The largest uncertainty in our results comes from the size of the dataset - COMPAS is considered one of the key datasets for evaluating fairness in Machine Learning, but it only has 6,000 rows. Additionally, the COMPAS dataset is not a perfect representation of the real world - it is a dataset created by a newsroom to investigate a specific software. This means that our results may not be generalizable to all datasets.

Future Work

Our results show that the FairFed algorithm can be extended to include individual fairness measures without a significant cost to accuracy or group fairness. However, our results are limited by the size and nature of the COMPAS dataset. Additionally, a larger range of α, β, and γ values should be tested to better understand the trade-offs between these values.

Future work should focus on testing our novel modification to the FairFed algorithm on a wider range of larger datasets, to better understand how the algorithm performs in different contexts. Additionally, this work should be extended to include a diverse set of models beyond a feed forward neural network. Ideally, a near continuous range of α, β, and γ values should be tested, providing us the ability to model the relationship between these values and the resulting accuracy-fairness trade-offs.

Conclusion

As Machine Learning becomes more prominent in societal use cases, it is important for methods, especially ones considered "black boxes" by the public, to be explicit about the fairness they promote and engage in.

In this work, we sought to extend the FairFed algorithm to include individual fairness measures. Our problem statement was to understand how this extension would impact the accuracy scores and fairness scores of a model.

Our results show that the FairFed algorithm can be extended to include individual fairness measures without a significant cost to accuracy or group fairness. This is significant in the field of fairness-aware Machine Learning - capturing metrics related to Individual Fairness and Group Fairness can allow policy makers, healthcare researchers, and other stakeholders to better understand how a model is performing, and understanding the interactions between α, β, and γ can allow them to pull more levers to serve their respective populations.

By working on appending Individual Fairness into existing Group Fairness metholdogies, we are pushing for more equitable outcomes across Machine Learning contexts.

Division of Labor

Miguel worked on researching fairness in Machine Learning, and how it can be applied to Federated Learning. He also worked on extending the FairFed algorithm to include Individual Fairness measures.

Meriem worked on the baseline implementation of the FairFed algorithm, and the ablation studies that were performed on the COMPAS dataset.

Raj worked on the data preprocessing and the evaluation metrics used in the ablation studies.

All authors contributed to the writing of the paper, and collaborated on the research and implementation of the work presented in this paper.

References

[1] Becker, Barry, and Ronny Kohavi. 1996. "UCI Machine Learning Repository." Archive.ics.uci.edu. April 30, 1996. https://archive.ics.uci.edu/dataset/2/adult.

[2] Ding, Frances, Moritz Hardt, Jon M Miller, and Ludwig Schmidt. 2021. "Retiring Adult: New Datasets for Fair Machine Learning." Advances in Neural Information Processing Systems 34 (August). https://doi.org/10.48550/arxiv.2108.04884.

[3] Ezzeldin, Yahya H., et al. "FairFed: Enabling Group Fairness in Federated Learning." Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 6, 26 June 2023, pp. 7494–7502, https://doi.org/10.1609/aaai.v37i6.25911. Accessed 4 Oct. 2024. https://neurips2021workshopfl.github.io/NFFL-2021/papers/2021/Ezzeldin2021.pdf

[4] Hamman, Faisal, and Sanghamitra Dutta. Demistifying Local and Global Fairness Trade-offs in Federated Learning Using Partial Information Decomposition. 4 Mar. 2024. https://arxiv.org/pdf/2307.11333

[5] "Federated Learning in Healthcare: Model Misconducts, Security, Challenges, Applications, and Future Research Directions-A Systematic Review.". https://arxiv.org/html/2405.13832v1.

[6] Larson, Jeff, Surya Mattu, Lauren Kirchner, and Julia Angwin. 2016. "How We Analyzed the COMPAS Recidivism Algorithm." ProPublica. May 23, 2016. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

[7] Rieke, Nicola, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, et al. "The Future of Digital Health with Federated Learning." Npj Digital Medicine 3, no. 1 (September 14, 2020): 1–7. https://doi.org/10.1038/s41746-020-00323-1.

[8] Ezzeldin, Yahya H, Yan Shen, Chaoyang He, Emilio Ferrara, and Salman Avestimehr. 2023. "FairFed: Enabling Group Fairness in Federated Learning." Proceedings of the ... AAAI Conference on Artificial Intelligence 37 (6): 7494–7502. https://doi.org/10.1609/aaai.v37i6.25911.

[9] Mansouri, Mohamad, Melek Önen, Wafa Ben Jaballah, and Mauro Conti. 2023. "SoK: Secure Aggregation Based on Cryptographic Schemes for Federated Learning." Proceedings on Privacy Enhancing Technologies 2023 (1): 140–57. https://doi.org/10.56553/popets-2023-0009.

[10] Narayanan, Arvind, and Vitaly Shmatikov. "Myths and Fallacies of "Personally Identifiable Information."" Communications of the ACM, vol. 53, no. 6, 1 June 2010, p. 24, https://doi.org/10.1145/1743546.1743558.

[11] Berk, Richard A, et al. "A Convex Framework for Fair Regression." ArXiv (Cornell University), 7 June 2017.

[12] The Flower. "Flower: A Friendly Federated Learning Framework." Flower.ai, flower.ai/.

[13] Beaulieu-Jones, Brett K et al. "Privacy-Preserving Generative Deep Neural Networks Support Clinical Data Sharing." Circulation. Cardiovascular quality and outcomes vol. 12,7 (2019): e005122. doi:10.1161/CIRCOUTCOMES.118.005122

[14] El Mestari, Soumia Zohra, et al. "Preserving Data Privacy in Machine Learning Systems." Computers & Security, vol. 137, 1 Feb. 2024, p. 103605, www.sciencedirect.com/science/article/pii/S0167404823005151, https://doi.org/10.1016/j.cose.2023.103605.

GitHub

`https://github.com/rivera-lanasm/flfair_idlf24/`