Enforcing fairness in private federated learning via the modified method of differential multipliers

Borja Rodríguez-Gálvez*
KTH
borjarg@kth.se

Filip Granqvist, Rogier van Dalen, and Matt Seigel
Apple
{fgranqvist,rogier_vandalen,mseigel}@apple.com

Abstract

Federated learning with differential privacy, or private federated learning, provides a strategy to train machine learning models while respecting users' privacy. However, differential privacy can disproportionately degrade the performance of the models on under-represented groups, as these parts of the distribution are difficult to learn in the presence of noise. Existing approaches for enforcing fairness in machine learning models have considered the centralized setting, in which the algorithm has access to the users' data. This paper introduces an algorithm to enforce group fairness in private federated learning, where users' data does not leave their devices. First, the paper extends the modified method of differential multipliers to empirical risk minimization with fairness constraints, thus providing an algorithm to enforce fairness in the central setting. Then, this algorithm is extended to the private federated learning setting. The proposed algorithm, FPFL, is tested on a federated version of the Adult dataset and an "unfair" version of the FEMNIST dataset. The experiments on these datasets show how private federated learning accentuates unfairness in the trained models, and how FPFL is able to mitigate such unfairness.

1 Introduction

Federated learning (FL) [1] is a machine learning setting where multiple entities (or users) collaborate in training a machine learning model under the coordination of a central server or a service provider. In this setting, only some statistics relevant to the model's training are shared with the central server, and the user's raw data is always stored on their device and never leaves it. In many applications of interest, the data is distributed across many users, is large compared to the model's size, and is privacy-sensitive. Therefore, this decentralized setting is attractive since (i) it is amenable to parallelizing the computation across multiple devices (which can easily be accommodated on such devices with modern, fast, processors) and (ii) it is not dependent on a central dataset, which could be susceptible to privacy leaks via re-identification attacks [2].

The statistics transmitted to the central server contain less information than the raw data by the data processing inequality, thus reducing the risk of privacy leaks. However, there exist determined enough adversaries that can extract delicate information from these updates or the model itself, see e.g. [3, 4]. Therefore, in order to provide guarantees of privacy to users, it is customary to ensure that the training of the model is differentially private, see e.g. [5, 6, 7, 8, 9]. Differential privacy (DP) [10, 11] is a strong privacy standard for algorithms that operate on data that limits the amount of information that any attacker (regardless of their compute power or access to auxiliary information) may obtain about any user's identity after observing the model.

*Work done during an internship at Apple.
The full version can be found in this file, after the references, and will soon be on arXiv with the same title.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

Neural networks have seen great success in a wide range of applications such as image classification and natural language processing [12, 13]. Federated learning [1] has been used to train neural networks for language modeling and speaker verification [9]. However, these models are susceptible to perpetuating societal biases existing in the data [14] or to discriminate against certain groups even when the data is balanced [15]. Moreover, when the training is differentially private [16], degradation in the performance of these models disproportionately impacts under-represented groups [17].

In the realm of federated learning, there has been some research studying how to achieve individual fairness, i.e. that the performance of the model is similar among users [18, 19, 20, 21]. However, this notion of fairness falls short in terms of protecting users from under-represented groups falling foul of disproportionate treatment. Conversely, there is little work proposing solutions to enforce group fairness, i.e. that the performance is similar among users from different groups. The current work in this area [22] is limited to a specific measure of fairness (demographic parity) when using logistic regression models. Moreover, all such prior work focuses on non-private federated learning, and do not consider the adverse affects that differential privacy has on the models trained with PFL.

On the other hand, work studying the trade-offs between privacy and group fairness proposes solutions that are either limited to simple models such as linear logistic regression [23], require an oracle [24], scale poorly for large hypothesis classes [25], or only offer privacy protection for the variable determining the group [26, 27]. Furthermore, these papers consider only the central learning paradigm, and the techniques are not directly adaptable to federated learning.

For all the above reasons, in this paper, we propose an algorithm to train deep learning models with private federated learning while enforcing group fairness. We pose the problem as an optimization problem with fairness constraints and extend the modified method of differential multipliers (MMDM) [28] to solve such a problem with FL and DP. Hence, the resulting algorithm (i) applies to any model that can be learned using stochastic gradient descent (SGD) or any of its variants, (ii) can be tailored to enforce a majority of the group fairness metrics, (iii) can consider any number of attributes determining the groups, and (iv) can consider both classification and regression tasks.

2 An Algorithm for Fair and Private Federated Learning (FPFL)

Setting We consider a dataset $d = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n)$ of n instances, where each instance $\boldsymbol{z}_i$ belongs to a group $a_i \in \mathcal{A}$. We also consider a supervised learning setting where $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i, a_i)$, the output of the model $\hat{y}_i$ is an approximation of y_i, and the model is parametrized by the parameters $\boldsymbol{w} \in \mathbb{R}^p$. Moreover, we consider a federated scenario where the dataset d is distributed across K users such that each user maintains a local dataset d^k with n^k samples. Finally, we consider we have no information about the distribution of the variables $(\boldsymbol{X}, Y, A)$ aside from the available samples.

Objective We concern ourselves with the task of finding the model's parameters $\boldsymbol{w}^\star$ that minimize a loss function ℓ across the data samples while enforcing a measure of fairness on the model. Many fairness metrics can be written as the (dis)similarity of the expected value of a function of interest f of the model evaluated on the general population d and on the population of each group $d_a = \{(\boldsymbol{x}_i, y_i, a_i) \in d : a_i = a\}$ [29, 30]. Therefore, our objective is to solve

$$\boldsymbol{w}^\star = \begin{cases} \arg\min\limits_{\boldsymbol{w} \in \mathbb{R}^p} & \dfrac{L(d, \boldsymbol{w})}{n} \\ \text{s.t.} & \left| \dfrac{F(d', \boldsymbol{w})}{n'} - \dfrac{F(d'_a, \boldsymbol{w})}{n'_a} \right| \leq \alpha, \text{ for all } a \in \mathcal{A} \end{cases}, \tag{P}$$

where $F(d', \boldsymbol{w}) := \sum_{\boldsymbol{z} \in d'} f(\boldsymbol{z}, \boldsymbol{w})$, d' is a subset of d that varies among different fairness metrics, n' is the number of samples in d, d'_a is the set of samples in d' such that $a_i = a$, n'_a is the number of samples in d_a, and α is a tolerance threshold. The subset d' is chosen based on on the form of the fairness metric: if the function of interest does not involve any conditional expectation (e.g. accuracy, where $f(\boldsymbol{X}, Y, \hat{Y}) = \mathbb{1}(\hat{Y} = Y)$), then $d' = d$; if, on the other hand, the subset involves a conditional expectation, then the d' is the subset of d where that condition holds, e.g. when the function is the false negative rate (FNR), where $f(\boldsymbol{X}, Y, \hat{Y}) = \mathbb{E}[\mathbb{1}(\hat{Y} = 0) | Y = 1]$, then $d' = \{(\boldsymbol{x}, y, a) \in d : y = 1\}$. Note that the constraints are not strict, meaning that there is a tolerance α for how much the function f can vary between certain groups and the overall population. The reason for this choice is twofold: (i) it facilitates the training since the solution subspace is larger. Moreover, (ii) it is known that some

fairness metrics, such as FNR parity, are incompatible with DP and non-trivial accuracy. However, if the fairness metric is relaxed, fairness, accuracy, and privacy can coexist [25, 24].

A solution The constrained optimization (P) can be solved using the MMDM algorithm [28]. This is related to the method in [26], that essentially uses the basic method of differential multipliers (BMDM), which attempts to solve the objective (P) without a tolerance, i.e. with $\alpha = 0$, and discards the damping parameter which will be introduced below, thereby making it less suited for non-convex optimization [28]. The algorithm consists of solving the following set of differential equations resulting from the Lagrangian of (P) with an additional quadratic penalty on the fairness constraints using gradient descent/ascent. This results in the following iterative update algorithm:

$$\begin{cases} \boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \gamma \boldsymbol{g}(d, \boldsymbol{w}) \\ \boldsymbol{w} \leftarrow \boldsymbol{w} - \eta\big(1/n \nabla_{\boldsymbol{w}} L(d, \boldsymbol{w}) + \boldsymbol{\lambda}^T \nabla_{\boldsymbol{w}} \boldsymbol{g}(d, \boldsymbol{w}) + c\boldsymbol{g}(d, \boldsymbol{w})^T \nabla_{\boldsymbol{w}} \boldsymbol{g}(d, \boldsymbol{w})\big) \end{cases}, \tag{1}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^r$ is a Lagrange (or dual) multiplier, $c \in \mathbb{R}_+$ is a damping parameter, and η and γ are the parameters' and Lagrange multiplier's learning rates. Following the original formulation [28], the function $\boldsymbol{g}(d, \boldsymbol{w})$ is a concatenation of r functions $g_j(d, \boldsymbol{w})$ defining the constraints $g_j(d, \boldsymbol{w}) = 0$. In our setting, we can accommodate the fairness constraints to this framework by letting $r = |\mathcal{A}|$,

$$g_a(\boldsymbol{w}) = \begin{cases} h_a(\boldsymbol{w}) & \text{if } h_a(\boldsymbol{w}) \geq 0 \\ 0 & \text{otherwise} \end{cases}, \text{ and } h_a(\boldsymbol{w}) := \left| \frac{F(d', \boldsymbol{w})}{n'} - \frac{F(d'_a, \boldsymbol{w})}{n'_a} \right| - \alpha.$$

Now, the fairness-enforcing problem (P) can be solved with gradient descent/ascent or mini-batch stochastic gradient descent/ascent (and/or their differentially private counterparts [16]), where instead of the full dataset d, d', and d'_a, one considers batches b, b', and b'_a (or subsets) of that dataset.

Extending the solution to PFL To achieve this goal, we might first combine the ideas from `FederatedSGD` [1] and the previous section to extend the developed adaptation of MMDM to FL. In order to perform the model updates dictated by (1), the central server requires the following statistics:

$$\nabla_{\boldsymbol{w}} L(d, \boldsymbol{w}); \;\; F(d', \boldsymbol{w}); \;\; \big[F(d'_a, \boldsymbol{w})\big]_{a \in \mathcal{A}}; \;\; \nabla_{\boldsymbol{w}} F(d', \boldsymbol{w}); \;\; \big[\nabla_{\boldsymbol{w}} F(d'_a, \boldsymbol{w})\big]_{a \in \mathcal{A}}; \;\; n'; \;\; \big[n'_a\big]_{a \in \mathcal{A}}. \tag{2}$$

However, some of these statistics can be obtained from the others, namely $F(d', \boldsymbol{w}) = \sum_{a \in \mathcal{A}} F(d'_a, \boldsymbol{w})$, $\nabla_{\boldsymbol{w}} F(d', \boldsymbol{w}) = \sum_{a \in \mathcal{A}} \nabla_{\boldsymbol{w}} F(d'_a, \boldsymbol{w})$, and $n' = \sum_{a \in \mathcal{A}} n'_a$. Moreover, as mentioned in the previous section, one might use a sufficiently large batch b of the data instead of the full dataset for each update. Therefore, we consider an iterative algorithm where, at each iteration, the central server samples m users S that report a vector with the sufficient statistics for the update, that is

$$\boldsymbol{v}^k = \Big[\nabla_{\boldsymbol{w}} L(d^k, \boldsymbol{w}), \big[F(d^{k\prime}{}_a, \boldsymbol{w})\big]_{a \in \mathcal{A}}, \big[\nabla_{\boldsymbol{w}} F(d^{k\prime}{}_a, \boldsymbol{w})\big]_{a \in \mathcal{A}}, \big[n^{k\prime}{}_a\big]_{a \in \mathcal{A}}\Big]. \tag{3}$$

This way, if we define the batch b as the sum of the m users' local datasets $b := \sum_{k \in S} d^k$ and the batch b' analogously, then the aggregation of each user's vectors results in

$$\boldsymbol{v} = \sum_{k \in S} \boldsymbol{v}^k = \Big[\nabla_{\boldsymbol{w}} L(b, \boldsymbol{w}), \big[F(b'_a, \boldsymbol{w})\big]_{a \in \mathcal{A}}, \big[\nabla_{\boldsymbol{w}} F(b'_a, \boldsymbol{w})\big]_{a \in \mathcal{A}}, \big[|b'_a|\big]_{a \in \mathcal{A}}\Big], \tag{4}$$

which contains all the sufficient statistics for the parameters' update.

Finally, the resulting algorithm, termed Fair PFL or `FPFL` and described in Algorithm 1, inspired by the ideas from [7, 8, 9, 31] guarantees the users' privacy as follows: (i) it makes sure that the aggregation of the users' sent vectors is done securely by a trusted third party, e.g., with secure aggregation [31]. Moreover, (ii) it clips the vectors $\boldsymbol{v}^k$ with a clipping bound C to restrict their ℓ_2 sensitivity and ensures central (ϵ, δ)-DP by adding Gaussian noise with variance $C^2\sigma^2$ to the clipped vector. The parameter σ is calculated according to the refined moments accountant privacy analysis from [32], taking into account the number of iterations (or communication rounds) T the algorithm will be run, the number of users (or cohort size) m that are used per iteration, the total number of users (or population size) K, and the privacy parameters ϵ and δ.

3 Experimental Results

A sample of our experiments is given in Table 1. For the full set of experiments with more fairness metrics and datasets, and the experimental details, please refer to the full version of the paper below.

Algorithm 1: FPFL. The K users are indexed by k, m is the cohort size, T is the number of iterations, C is the clipping bound, and (ε, δ) are the DP parameters.

Server Executes:

$\boldsymbol{w}_0, \boldsymbol{\lambda}_0 \leftarrow$ InitializeParameters()
$\sigma \leftarrow$ CalculateNoiseScale$(K, m, \varepsilon, \delta, T)$
for each iteration $t = 1, 2, \ldots, T$ **do**
  $S_t \leftarrow$ (random set of m users)
  $\boldsymbol{v}_t \leftarrow$ SecureAggregation$(\boldsymbol{w}_{t-1}, S_t, C, \sigma)$
  $\boldsymbol{\lambda}_t \leftarrow$ UpdateMultiplier$(\boldsymbol{v}_t, \boldsymbol{w}_{t-1})$
  $\boldsymbol{w}_t \leftarrow$ UpdateParameters$(\boldsymbol{v}_t, \boldsymbol{\lambda}_t)$
end

SecureAggregation$(\boldsymbol{w}, S, C, \sigma)$**:** `/* run by a trusted third party */`
for each user k in S **do**
  $\boldsymbol{v}^k \leftarrow$ UserStatistics$(k, \boldsymbol{w})$
  $\boldsymbol{v}^k \leftarrow \boldsymbol{v}^k \cdot \min\left\{1, {}^{C}/_{\|\boldsymbol{v}^k\|_2}\right\}$
end
$\boldsymbol{v} \leftarrow \sum_{k \in S} \boldsymbol{v}^k + \mathcal{N}\left(0, C^2 \cdot \sigma^2\right)$
return $\boldsymbol{v}$ to server

UserStatistics$(k, \boldsymbol{w})$**:** `/* run in the user's device */`
$\boldsymbol{v} \leftarrow \left[\nabla_{\boldsymbol{w}} L(\boldsymbol{w}, d^k), \left[F(\boldsymbol{w}, {d'}_a^k)\right]_{a \in \mathcal{A}}, \left[\nabla_{\boldsymbol{w}} F(\boldsymbol{w}, {d'}_a^k)\right]_{a \in \mathcal{A}}, \left[{n'}_a^k\right]_{a \in \mathcal{A}}\right]$
return $\boldsymbol{v}$ to trusted third party

In this experiment, we train a shallow neural network on the Adult dataset [33], where the samples are distributed across users according to a Poisson distribution with mean 2. This dataset consists of 32,561 training and 16,281 testing samples of demographic data from the US census. Each datapoint contains various demographic attributes. Though the particular task, predicting individuals' salary ranges, is not itself of interest, this dataset serves as a proxy for tasks with inequalities in the data. A reason this dataset is often used in the literature on ML fairness is that the fraction of individuals in the higher salary range is 30% for the men and only 10% for the women. The experiments in this paper aim to stop this imbalance from entering into the model by balancing the false negative rate (FNR) [34]. The highest difference between the FNR of any of the groups and the general population is the FNR gap, and we aim to ensure that it is below $\alpha = 0.02$ using the FPFL algorithm.

Table 1: Network performance on Adult. Privacy parameters: $\epsilon = 2$, $\delta = 1/n_{\text{users}}$.

Algorithm	Accuracy	FNR gap
FL	0.851	0.121
FFL	**0.855**	**0.036**
FL + Clip	0.844	0.169
FFL + Clip	**0.853**	**0.018**
PFL	0.844	0.167
FPFL	**0.840**	**0.001**

Our initial experiments are in line with the observations from [17], where we observe that the clipping and noise addition from DP training disproportionately deteriorates the performance of under-represented groups. We observe this phenomenon by the increase in FNR gap.

Conversely, the FPFL algorithm is not largely affected by the clipping nor noise addition. Moreover, it maintains the accuracy of the non-fairly trained models while achieving lower (by an order of magnitude) fairness gaps. To select a model, FPFL keeps the one with the highest accuracy among those that are fair. The model's accuracy and fairness are estimated with the statistics from a cohort of users during training, and thus sometimes it believes a model is fairer than it is, e.g. the example with neither noise nor clipping from Table 1. Hence, a large cohort size m is key for this algorithm.

Further experiments on harder tasks with larger models reveal how the accuracy of FPFL is more sensitive to DP noise than standard PFL. The reason is that the group information protection in the sent statistics increases the noise variance. Hence, ensuring that the population (or number of users n_{users}) is large, in order to exploit the sub-sampling privacy amplification from [32], is also crucial.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2017, pp. 1273–1282.

[2] L. Sweeney, "Simple demographics often identify people uniquely," *Health (San Francisco)*, vol. 671, no. 2000, pp. 1–34, 2000.

[3] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[4] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[5] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," *arXiv preprint arXiv:1812.00984*, 2018.

[6] H. B. McMahan, G. Andrew, U. Erlingsson, S. Chien, I. Mironov, N. Papernot, and P. Kairouz, "A general approach to adding differential privacy to iterative training procedures," *arXiv preprint arXiv:1812.06210*, 2018.

[7] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *International Conference on Learning Representations (ICLR)*, 2018.

[8] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 2019, pp. 1–11.

[9] F. Granqvist, M. Seigel, R. van Dalen, Á. Cahill, S. Shum, and M. Paulik, "Improving on-device speaker verification using federated learning with privacy," *Interspeech*, pp. 4328–4332, 2020.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[11] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[13] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mT5: A massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 483–498.

[14] A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

[15] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on fairness, accountability and transparency*. PMLR, 2018, pp. 77–91.

[16] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (CCS)*, 2016, pp. 308–318.

[17] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, pp. 15 479–15 488, 2019.

[18] Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu, "FedMGDA+: Federated learning meets multi-objective optimization," *arXiv preprint arXiv:2006.11489*, 2020.

[19] W. Huang, T. Li, D. Wang, S. Du, and J. Zhang, "Fairness and accuracy in federated learning," *arXiv preprint arXiv:2012.10069*, 2020.

[20] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," *arXiv preprint arXiv:1905.10497*, 2019.

[21] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 6357–6368.

[22] W. Du, D. Xu, X. Wu, and H. Tong, "Fairness-aware agnostic federated learning," in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 2021, pp. 181–189.

[23] J. Ding, X. Zhang, X. Li, J. Wang, R. Yu, and M. Pan, "Differentially private and fair classification via calibrated functional mechanism," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 622–629.

[24] R. Cummings, V. Gupta, D. Kimpara, and J. Morgenstern, "On the compatibility of privacy and fairness," in *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, 2019, pp. 309–315.

[25] M. Jagielski, M. Kearns, J. Mao, A. Oprea, A. Roth, S. Sharifi-Malvajerdi, and J. Ullman, "Differentially private fair learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 3000–3008.

[26] C. Tran, F. Fioretto, and P. Van Hentenryck, "Differentially private and fair deep learning: A Lagrangian dual approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 9932–9939.

[27] B. Rodríguez-Gálvez, R. Thobaben, and M. Skoglund, "A variational approach to privacy and fairness," in *2nd Privacy Preserving Artificial Intelligence Workshop (PPAI) of the AAAI Conference on Artificial Intelligence*, 2021. [Online]. Available: https://arxiv.org/abs/2006.06332

[28] J. C. Platt and A. H. Barr, "Constrained differential optimization," in *Proceedings of the 1987 International Conference on Neural Information Processing Systems*, 1987, pp. 612–621.

[29] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach, "A reductions approach to fair classification," in *International Conference on Machine Learning*. PMLR, 2018, pp. 60–69.

[30] F. Fioretto, P. Van Hentenryck, T. W. Mak, C. Tran, F. Baldo, and M. Lombardi, "Lagrangian duality for constrained deep learning," *arXiv preprint arXiv:2001.09394*, 2020.

[31] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[32] Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled Rényi differential privacy and analytical moments accountant," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2019, pp. 1226–1235.

[33] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[34] A. Castelnovo, R. Crupi, G. Greco, and D. Regoli, "The zoo of fairness metrics in machine learning," *arXiv preprint arXiv:2106.00467*, 2021.

ENFORCING FAIRNESS IN PRIVATE FEDERATED LEARNING VIA THE MODIFIED METHOD OF DIFFERENTIAL MULTIPLIERS

Borja Rodríguez-Gálvez*, Filip Granqvist†, Rogier van Dalen‡, Matt Seigel§

Abstract

Federated learning with differential privacy, or private federated learning, provides a strategy to train machine learning models while respecting users' privacy. However, differential privacy can disproportionately degrade the performance of the models on under-represented groups, as these parts of the distribution are difficult to learn in the presence of noise. Existing approaches for enforcing fairness in machine learning models have considered the centralized setting, in which the algorithm has access to the users' data. This paper introduces an algorithm to enforce group fairness in private federated learning, where users' data does not leave their devices. First, the paper extends the modified method of differential multipliers to empirical risk minimization with fairness constraints, thus providing an algorithm to enforce fairness in the central setting. Then, this algorithm is extended to the private federated learning setting. The proposed algorithm, `FPFL`, is tested on a federated version of the Adult dataset and an "unfair" version of the FEMNIST dataset. The experiments on these datasets show how private federated learning accentuates unfairness in the trained models, and how `FPFL` is able to mitigate such unfairness.

1 Introduction

Federated learning (FL) [McMahan et al., 2017] is a machine learning paradigm where multiple entities (or users) collaborate in training a machine learning model under the coordination of a central server or a service provider. In this setting, only some statistics relevant to the model's training are shared with the central server, and the user's raw data is always stored on their device and never leaves it.

In many applications of interest, the data is distributed across many users, is large compared to the model's size, and is privacy-sensitive. Therefore, this decentralized setting is attractive since (i) it is amenable to parallelizing the computation across multiple devices (which can easily be accommodated on such devices with modern, fast, processors) and (ii) it is not dependent on a central dataset, which could be susceptible to privacy leaks via re-identification attacks [Sweeney, 2000].

The statistics transmitted to the central server contain less information than the raw data by the data processing inequality, and therefore reduce the risk of privacy leaks. However, there exist determined enough adversaries that can extract delicate information from these updates or the model itself. For example, even in the extreme case where an adversary only has access to queries of the model (or "black-box" access), they can discover the identity of users present in the training data for a generic task [Shokri et al., 2017] or even reconstruct the faces used to train a face recognition system [Fredrikson et al., 2015].

*KTH Royal Institute of Technology – Information Science and Engineering (ISE): `borjarg@kth.se`
Work done during an internship at Apple.
†Apple: `fgranqvist@apple.com`
‡Apple: `rogier_vandalen@apple.com`
§Apple: `mseigel@apple.com`

Therefore, in order to provide guarantees of privacy to users, it is customary to ensure that the training of the model is differentially private, see e.g. [Bhowmick et al., 2018; McMahan et al., 2018a,b; Truex et al., 2019; Granqvist et al., 2020]. Differential privacy (DP) [Dwork et al., 2006, 2014] is a strong privacy standard for algorithms that operate on data. More precisely, DP guarantees that the probability of obtaining a model using all the users' data is close to the probability of obtaining that same model if any one of the users does not participate in the training. Hence, DP limits the amount of information that any attacker (regardless of their compute power or access to auxiliary information) may obtain about any user's identity after observing the model.

Deep learning models have seen great success in a wide range of applications such as image classification, speech recognition, or natural language processing [He et al., 2015; Amodei et al., 2016; Xue et al., 2021]. For this reason, they are usually the model of choice for FL [McMahan et al., 2017]. In fact, DL models have been successfully trained with differentially private FL (or PFL) for tasks like next word prediction or speaker verification [McMahan et al., 2017; Granqvist et al., 2020]. Nonetheless, these models are susceptible to perpetuate societal biases existing in the data [Caliskan et al., 2017] or to discriminate against certain groups even when the data is balanced [Buolamwini and Gebru, 2018]. Moreover, when the training is differentially private [Abadi et al., 2016], degradation in the performance of these models disproportionately impacts under-represented groups [Bagdasaryan et al., 2019]. More specifically, the accuracy of the model on minority groups is deteriorated to a larger extent than the accuracy for the majority groups.

In the realm of federated learning, there has been some research studying how to achieve individual fairness, i.e. that the performance of the model is similar among devices [Hu et al., 2020; Huang et al., 2020; Li et al., 2019, 2021]. However, this notion of fairness falls short in terms of protecting users from under-represented groups falling foul of disproportionate treatment. For example, [Castelnovo et al., 2021, Section 3.6] shows that a model that performs well on the majority of the users will have a good score on individual fairness metrics, even if all the users suffering from bad performance belong to the same group. Conversely, there is little work proposing solutions to enforce group fairness, i.e. that the performance is similar among users from different groups. The current work in this area [Du et al., 2021] is devoted to a specific measure of fairness (demographic parity) when using logistic regression models. Moreover, all such prior work focuses on non-private federated learning, and therefore do not consider the adverse affects that differential privacy has on the models trained with PFL.

On the other hand, work studying the trade-offs between privacy and group fairness proposes solutions that are either limited to simple models such as linear logistic regression [Ding et al., 2020], require an oracle [Cummings et al., 2019], scale poorly for large hypothesis classes [Jagielski et al., 2019], or only offer privacy protection for the variable determining the group [Tran et al., 2021; Rodríguez-Gálvez et al., 2021]. Furthermore, in the aforementioned work, the central learning paradigm is the only one considered, and the techniques are not directly adaptable to federated learning.

For all the above reasons, in this paper, we propose an algorithm to train deep learning models with private federated learning while enforcing group fairness. We pose the problem as an optimization problem with fairness constraints and extend the modified method of differential multipliers (MMDM) [Platt and Barr, 1987] to solve such a problem with FL and DP. Hence, the resulting algorithm (i) is applicable to any model that can be learned using stochastic gradient descent (SGD) or any of its variants, (ii) can be tailored to enforce the majority of the group fairness metrics, (iii) can consider any number of attributes determining the groups, and (iv) can consider both classification and regression tasks.

The paper is structured as follows: in Section 2 we review the background on differential privacy, private federated learning, the MMDM algorithm, and group fairness; in Section 3 we present out approach for fair private federated learning; in Section 4 we describe our experimental results; and in Section 5 we conclude with a summary and interpretation of our findings.

2 Background

In this section we review several aspects on the theory of differential privacy, private federated learning, the MMDM algorithm, and group fairness that will be necessary to develop and understand the proposed algorithm.

2.1 Differential Privacy

In this subsection, we start describing the definition of privacy that we will adopt, differential privacy (DP) [Dwork et al., 2006, 2014]. Then, we describe some of the details and properties of this standard that are useful to understand the proposed algorithm.

Differential privacy formalizes the maximum amount of information that an "almighty" adversary can obtain from the release of private data. This private data can be anything, from an obfuscated version of the users' data samples themselves to a noisy function of that data. Formally, the definition of DP establishes a bound on how different the distribution of a randomized function of the data can be if a user's contribution is included or not in such data.

Definition 1 ((ε, δ)-Differential Privacy)**.** *A randomized function $f : \mathcal{D} \to \mathcal{R}$ satisfies (ε, δ)-DP if for any two adjacent datasets $d, d' \in \mathcal{D}$ and for any subset of outputs $R \in \mathcal{R}$ it holds that*

$$\mathbb{P}\big[f(d) \in R\big] \leq e^{\varepsilon}\mathbb{P}\big[f(d') \in R\big] + \delta. \tag{1}$$

Two datasets d and d' are said to be adjacent if dataset d' can be formed by adding or removing all data associated with a user from d.

An alternative definition to differential privacy is given in [Mironov, 2017], where the dissimilarity of the distributions is measured using the Rényi divergence [Rényi, 1961]. This definition is attractive since it enjoys similar properties to the original definition, and gives a more efficient privacy analysis of certain iterative algorithms such as differentially private SGD (DP-SGD) [Abadi et al., 2016; Wang et al., 2019], even for the same (ϵ, δ)-DP budget.

Definition 2 ((α, ε)-RDP)**.** *A randomized function $f : \mathcal{D} \to \mathcal{R}$ satisfies ε-Rényi differential privacy of order α, or (α, ε)-RDP, if for any two adjacent datasets $d, d' \in \mathcal{D}$ it holds that*

$$D_{\alpha}\Big(\mathbb{P}\big[f(d)\big] \big\| \mathbb{P}\big[f(d')\big]\Big) \leq \varepsilon, \tag{2}$$

where $D_{\alpha}(\cdot \,||\, \cdot)$ is the Rényi divergence of order α.

A common way to privatize some statistics $\phi(d)$ of the data d is to obfuscate them with Gaussian noise. Namely, instead of directly releasing $\phi(d)$ one instead releases $f(d)$, where

$$f(d) = \phi(d) + \mathcal{N}\big(0, \Delta_{\phi}^2\sigma^2\big) \tag{3}$$

and $\Delta_{\phi} = \max_{d,d' \in \mathcal{D}} \|\phi(d) - \phi(d')\|_2$ is the ℓ_2 sensitivity of the statistic. This is known as the Gaussian mechanism [Dwork et al., 2014] and it is $(\alpha, \alpha/(2\Delta_{\phi}^2\sigma^2))$-RDP for any $\alpha > 1$ [Mironov, 2017].

Differential privacy, in either of the two mentioned forms, enjoys several desirable properties for a privacy definition. Among those, we highlight three (in terms of RDP) that are important for the privacy guarantees of the presented algorithm:

- *Post-processing:* Consider two functions $f : \mathcal{D} \to \mathcal{R}$ and $g : \mathcal{R} \to \mathcal{R}'$. If f is (α, ε)-RDP, then $g \circ f$ is also (α, ε)-RDP. That is, no amount of post-processing can reduce the privacy guarantees provided by the function f [Mironov, 2017].

- *Adaptive Composition:* Consider two functions $f : \mathcal{D} \to \mathcal{R}$ and $g : \mathcal{D} \times \mathcal{R} \to \mathcal{R}'$. Consider also a function defined as $h(d) = g(d, f(d))$, where $d \in \mathcal{D}$. If f is (α, ε_f)-RDP, and for all r, $g(\cdot, r)$ is (α, ε_g)-RDP, then h is $(\alpha, \varepsilon_f + \varepsilon_g)$-RDP. That is, the privacy guarantees of a function f degrade gracefully when its realization is used by another function g [Mironov, 2017].

- *Sub-sampling privacy amplification:* Consider an (α, ε_f)-RDP function $f : \mathcal{D} \to \mathcal{R}$ where $\alpha \geq 2$. Now consider the function $g(d)$ defined as applying f to a random sub-sample (without replacement) of fixed length of $d \in \mathcal{D}$. Then g is (α, ε_g) with $\varepsilon_g \leq \varepsilon_f$. That is, the privacy guarantees of a function f are amplified by applying that function to a random sub-sample of the dataset [Wang et al., 2019].

2.2 Private Federated Learning

The federated learning setting focuses on learning a model that minimizes the expected value of a loss function ℓ using a dataset $d = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n) \in \mathcal{D}$ of n samples distributed across K users. This paper will consider models parametrized by a fixed number of parameters $\boldsymbol{w} \in \mathbb{R}^p$ and differentiable loss functions ℓ, which includes neural networks.

Since the distribution of the data samples Z is unknown and each user k is the only owner of their private dataset d^k, the goal of FL is to find the parameters $\boldsymbol{w}^\star$ that minimize the loss function across the samples of the users. That is, to solve

$$\boldsymbol{w}^\star = \arg\min_{\boldsymbol{w} \in \mathbb{R}^p} \frac{1}{n} \sum_{k=1}^{K} L(d^k, \boldsymbol{w}), \tag{4}$$

where $L(d^k, \boldsymbol{w}) := \sum_{\boldsymbol{z} \in d^k} \ell(\boldsymbol{z}, \boldsymbol{w})$.

This way, the parameters $\boldsymbol{w}$ can be learned with an approximation of SGD. McMahan et al. [2017] suggests that the central server iteratively samples m users; they compute and send back to the server an approximation of $\eta \nabla_{\boldsymbol{w}} L(d^k, \boldsymbol{w})$, where η is the learning rate; and finally the server updates the parameters as dictated by gradient descent $\boldsymbol{w} \leftarrow \boldsymbol{w} - \frac{1}{n} \sum_k \eta \nabla_{\boldsymbol{w}} L(d^k, \boldsymbol{w})$. If the users send back the exact gradient the algorithm is known as `FederatedSGD`. A generalisation of this, `FederatedAveraging`, involves users running several local epochs of mini-batch SGD and sending up the difference. However, the method in this paper relies on an additional term in the loss function, so it will extend `FederatedSGD`.

The above algorithm can be modified so it becomes diferentially private by following the structure of the differentially private SGD algorithm [Abadi et al., 2016; Wang et al., 2019]. This modification consists of clipping (i.e. restricting the sensitivity to a pre-defined clipping bound C) and applying the Gaussian mechanism to each user's gradient approximation before they are used to update the parameters [McMahan et al., 2018a,b]. Then, the adaptive composition and sub-sampling privacy amplification properties of DP ensure that the whole algorithm is differentially private.

In this architecture, where clients contribute statistics to a server which updates a model, differential privacy can be used in its local or central forms. In local DP [Dwork et al., 2014], the statistics are obfuscated before leaving the device. However, models trained with local DP often suffer from low utility [Granqvist et al., 2020]. Instead, this paper will assume (ϵ, δ)-central DP. This can involve trust in the server that updates the model, or a trusted third party separated from the model, which is how differential privacy was originally formulated. In some cases, like `FederatedSGD`, all the server needs to know is a sum over client contributions. In this case, the trusted third party can be replaced by multi-party computation [Goryczka and Xiong, 2015], such as secure aggregation [Bonawitz et al., 2017].

2.3 The Modified Method of Differential Multipliers

Ultimately, our objective is to find a parametric model that minimizes a differentiable loss function while respecting some fairness constraints. Therefore, in this section we review an algorithm for constrained

differential optimization, the modified method of differential multipliers (MMDM) [Platt and Barr, 1987].

This algorithm tries to find a solution to the following constrained optimization problem:

$$\boldsymbol{w}^\star = \begin{cases} \arg\min\limits_{\boldsymbol{w}\in\mathbb{R}^p} & \phi(\boldsymbol{w}) \\ \text{s.t.} & \boldsymbol{g}(\boldsymbol{w}) = \boldsymbol{0} \end{cases}, \tag{P1}$$

where $\phi : \mathbb{R}^p \to \mathbb{R}$ is the function to minimize, $\boldsymbol{g}(\boldsymbol{w}) = (g_1(\boldsymbol{w}), g_2(\boldsymbol{w}), \dots, g_r(\boldsymbol{w}))$ is the concatenation of r constraint functions, and $\{\boldsymbol{w} \in \mathbb{R}^p : \boldsymbol{g}(\boldsymbol{w}) = 0\}$ is the solution subspace.

The algorithm consists of solving the following set of differential equations resulting from the Lagrangian of (P1) with an additional quadratic penalty $c\boldsymbol{g}(w)^2$ using gradient descent/ascent. This results in the following iterative update algorithm

$$\begin{cases} \boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \gamma \boldsymbol{g}(\boldsymbol{w}) \\ \boldsymbol{w} \leftarrow \boldsymbol{w} - \eta\Big(\nabla_{\boldsymbol{w}}\phi(\boldsymbol{w}) + \boldsymbol{\lambda}^T\nabla_{\boldsymbol{w}}\boldsymbol{g}(\boldsymbol{w}) + c\boldsymbol{g}(\boldsymbol{w})^T\nabla_{\boldsymbol{w}}\boldsymbol{g}(\boldsymbol{w})\Big) \end{cases}, \tag{5}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^r$ is a Lagrange (or dual) multiplier, $c \in \mathbb{R}_+$ is a damping parameter, η is the learning rate of the model parameters and γ is the learning rate of the Lagrange multiplier.

Intuitively, these sets of updates gradually fulfill (P1). The parameter updates against $\nabla_{\boldsymbol{w}}\phi(\boldsymbol{w})$ enforce the function minimization and the parameter updates against $\nabla_{\boldsymbol{w}}\boldsymbol{g}(\boldsymbol{w})$ enforce the constraints' satisfaction. Then, the multiplier $\boldsymbol{\lambda}$ and the multiplicative factor $c\boldsymbol{g}(\boldsymbol{w})$ control how strongly the constraints' violations are penalized in the parameters' update.

A desirable property of MMDM is that for small enough learning rates η, γ and large enough damping parameter c, there is a region comprising the surroundings of each constrained minimum such that if the parameters' initialization is in that region and the parameters remain bounded, then the algorithm converges to a constrained minimum [Platt and Barr, 1987]. Intuitively, this condition comes from the fact that the term $c\boldsymbol{g}(\boldsymbol{w})\nabla_{\boldsymbol{w}}\boldsymbol{g}(\boldsymbol{w})$ enforces a quadratic shape on the optimization search space on the neighbourhood of the solution subspace, and this local behavior is stronger the larger the damping parameter c. Therefore, if the parameters' initialization is in this locally quadratic region, then the algorithm is guaranteed to converge to the minimum.

2.4 Group Fairness

In this subsection, we mathematically formalize what *group fairness* means. To simplify the exposition, we describe this notion in the central setting, i.e. where the data is not distributed across users.

Let us consider a dataset $d = (\boldsymbol{z}_1, \boldsymbol{z}_2, \dots, \boldsymbol{z}_n)$ of n instances, where each instance $\boldsymbol{z}_i$ belongs to a group $a_i \in \mathcal{A}$. Group fairness considers how differently a model treats the instances belonging to each group. Many fairness metrics can be written in terms of the similarity of the expected value of a function of interest f of the model evaluated on the general population d with that on the population of each group $d_a = \{(\boldsymbol{x}_i, y_i, a_i) \in d : a_i = a\}$ [Agarwal et al., 2018; Fioretto et al., 2020]. That is, if we consider a supervised learning problem, where $\boldsymbol{z}_i = (\boldsymbol{x}_i, \boldsymbol{y}_i, a_i)$ and where the output of the model is an approximation $\hat{\boldsymbol{y}}_i$ of $\boldsymbol{y}_i$, we say a model is fair if

$$\mathbb{E}[f(\boldsymbol{X}, \boldsymbol{Y}, \hat{\boldsymbol{Y}}) \mid A = a] = \mathbb{E}[f(\boldsymbol{X}, \boldsymbol{Y}, \hat{\boldsymbol{Y}})] \tag{6}$$

for all $a \in \mathcal{A}$.

Most of the group fairness literature focuses on the case of binary classifiers, i.e. $\hat{y}_i \in \{0, 1\}$, and on the binary group case, i.e. $\mathcal{A} = \{0, 1\}$. However, many of the fairness metrics can be extended to general output spaces $\mathcal{Y}$ and categorical groups $\mathcal{A}$. It is common that the function f is the indicator function $\mathbb{1}$ of

some logical relationship between the random variables, thus turning (6) to an equality between probabilities. As an example, we describe two common fairness metrics that will be used later in the paper. For a comprehensive survey of different fairness metrics and their inter-relationships, please refer to [Verma and Rubin, 2018; Castelnovo et al., 2021].

1. *False negative rate (FNR) parity (or equal opportunity) [Hardt et al., 2016]*: This fairness metric is designed for binary classification and binary groups. It was originally defined as equal true positive rate between the groups, which is equivalent to an equal FNR between each group and the overall population. That is, if we let $f(\boldsymbol{X}, Y, \hat{Y}) = \mathbb{E}[\mathbb{1}(\hat{Y} = 0) \mid Y = 1]$ then (6) reduces to

$$\mathbb{P}[\hat{Y} = 0 \mid Y = 1, A = a] = \mathbb{P}[\hat{Y} = 0 \mid Y = 1] \tag{7}$$

   for all $a \in \mathcal{A}$.

   This is usually a good metric when the target variable Y is something positive such as being granted a loan or being hired for a job, since we want to minimize the group disparity of misclassification among the individuals that deserved such a loan or such a job [Hardt et al., 2016; Castelnovo et al., 2021].

2. *Accuracy parity (or overall misclassification rate) [Zafar et al., 2017]*: This fairness metric is also designed for binary classification and binary groups. Nonetheless, it applies well to general tasks and categorical groups. Similarly to the previous metric, if we let $f(\boldsymbol{X}, Y, \hat{Y}) = \mathbb{1}(\hat{Y} = Y)$ then (6) reduces to

$$\mathbb{P}[\hat{Y} = Y \mid A = a] = \mathbb{P}[\hat{Y} = Y] \tag{8}$$

   for all $a \in \mathcal{A}$.

   This is usually a good metric when there is not a clear positive or negative semantic meaning to the target variable, and also when this variable is not binary.

3 An Algorithm for Fair and Private Federated Learning (FPFL)

In this section we describe the proposed algorithm to enforce fairness in PFL of parametric models learned with SGD. First, we describe an adaptation of the MMDM algorithm to enforce fairness in standard central learning. Then, we extend that algorithm to PFL.

3.1 Adapting the MMDM Algorithm to Enforce Fairness

Consider a dataset $d = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n)$ of n instances, where each instance $\boldsymbol{z}_i$ belongs to a group $a_i \in \mathcal{A}$. Consider also a supervised learning setting where $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i, a_i)$, the output of the model $\hat{y}_i$ is an approximation of y_i, and the model is parametrized by the parameters $\boldsymbol{w} \in \mathbb{R}^p$. Finally, we consider we have no information about the distribution of the variables $(\boldsymbol{X}, Y, A)$ aside from the available samples.

We concern ourselves with the task of finding the model's parameters $\boldsymbol{w}^\star$ that minimize a loss function ℓ across the data samples while enforcing a measure of fairness on the model. That is, to solve

$$\boldsymbol{w}^\star = \begin{cases} \arg\min\limits_{\boldsymbol{w} \in \mathbb{R}^p} & \dfrac{L(d, \boldsymbol{w})}{n} \\ \text{s.t.} & \left| \dfrac{F(d', \boldsymbol{w})}{n'} - \dfrac{F(d'_a, \boldsymbol{w})}{n'_a} \right| \leq \alpha, \text{ for all } a \in \mathcal{A} \end{cases}, \tag{P2}$$

where $F(d', \boldsymbol{w}) := \sum_{\boldsymbol{z} \in d'} f(\boldsymbol{z}, \boldsymbol{w})$, d' is a subset of d that varies among different fairness metrics, n' is the number of samples in d', d'_a is the set of samples in d' such that $a_i = a$, n'_a is the number of samples in

d'_a, f is the function employed for the fairness metric definition, and α is a tolerance threshold. The subset d' is chosen based on the function f used for the fairness metric: if the fairness function does not involve any conditional expectation (e.g. accuracy parity), then $d' = d$; if, on the other hand, the subset involves a conditional expectation, then the d' is the subset of d where that condition holds, e.g. when the fairness metric is FNR parity $d' = \{(\boldsymbol{x}, y, a) \in d : y = 1\}$.

Note how the expected values of the fairness constraints in (6) are substituted with empirical averages in (P2). Also, note that the constraints are not strict, meaning that there is a tolerance α for how much the function f can vary between certain groups and the overall population. The reason for this choice is twofold:

1. It facilitates the training since the solution subspace is larger.

2. It is known that some fairness metrics, such as FNR parity, are incompatible with DP and non-trivial accuracy. However, if the fairness metric is relaxed, fairness, accuracy, and privacy can coexist [Jagielski et al., 2019; Cummings et al., 2019].

This way, we may re-write (P2) in the form of (P1) to solve the problem with MMDM. To do so we let $\phi(\boldsymbol{w}) = L(d, \boldsymbol{w})/n$ and $\boldsymbol{g}(\boldsymbol{w}) = (g_0(\boldsymbol{w}), g_1(\boldsymbol{w}), \dots, g_{|\mathcal{A}|-1}(\boldsymbol{w}))$, where

$$g_a(\boldsymbol{w}) = \begin{cases} h_a(\boldsymbol{w}) & \text{if } h_a(\boldsymbol{w}) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and $h_a(\boldsymbol{w}) := \left| \frac{F(d', \boldsymbol{w})}{n'} - \frac{F(d'_a, \boldsymbol{w})}{n'_a} \right| - \alpha$. Therefore, the parameters are updated according to

$$\begin{cases} \boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \gamma \boldsymbol{g}(\boldsymbol{w}) \\ \boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \left(1/n \nabla_{\boldsymbol{w}} L(d, \boldsymbol{w}) + \boldsymbol{\lambda}^T \nabla_{\boldsymbol{w}} \boldsymbol{g}(\boldsymbol{w}) + c \boldsymbol{g}(\boldsymbol{w})^T \nabla_{\boldsymbol{w}} \boldsymbol{g}(\boldsymbol{w}) \right) \end{cases}, \tag{10}$$

where we note that $\nabla_{\boldsymbol{w}} \boldsymbol{g}(\boldsymbol{w}) = (\nabla_{\boldsymbol{w}} g_0(\boldsymbol{w}), \nabla_{\boldsymbol{w}} g_1(\boldsymbol{w}), \dots, \nabla_{\boldsymbol{w}} g_{|\mathcal{A}|-1}(\boldsymbol{w}))$ and

$$\nabla_{\boldsymbol{w}} g_a(\boldsymbol{w}) = \begin{cases} \text{sign}\left(\frac{F(d', \boldsymbol{w})}{n'} - \frac{F(d'_a, \boldsymbol{w})}{n'_a} \right) \left(\frac{\nabla_{\boldsymbol{w}} F(d', \boldsymbol{w})}{n'} - \frac{\nabla_{\boldsymbol{w}} F(d'_a, \boldsymbol{w})}{n'_a} \right) & \text{if } h_a(\boldsymbol{w}) \geq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

Now, the fairness-enforcing problem (P2) can be solved with gradient descent/ascent or mini-batch stochastic gradient descent/ascent, where instead of the full dataset d, d', and d'_a, one considers batches b, b', and b'_a (or subsets) of that dataset. Moreover, it can be learned with DP adapting the DP-SGD algorithm from [Abadi et al., 2016], where a clipping bound and Gaussian noise is included in both the network parameters' and multipliers' individual updates. Nonetheless, there are a series of caveats of doing so:

- The batch size $|b'|$ should be large enough to, on average, have enough samples of each group $a \in \mathcal{A}$ so that the difference
$$\frac{F(b', \boldsymbol{w})}{|b'|} - \frac{F(b'_a, \boldsymbol{w})}{|b'_a|} \tag{12}$$
is well estimated.

- In many situations of interest such as when we want to enforce FNR parity or accuracy parity, the function f employed for the fairness metric is not differentiable and thus $\nabla_{\boldsymbol{w}} F(d', \boldsymbol{w})$ does not exist. To solve this issue, we resort to estimate the gradient $\nabla_{\boldsymbol{w}} F(d', \boldsymbol{w})$ using a differentiable estimation of the function aggregate $F(d', \boldsymbol{w})$. For instance:

– Enforcing FNR parity on a neural network $\psi_{\boldsymbol{w}}$ (with a Sigmoid output activation function) in a binary classification task. We note that given an input $\boldsymbol{x}_i$, the raw output of the network $\psi_{\boldsymbol{w}}(\boldsymbol{x}_i)$ is an estimate of the probability that $\hat{y}_i = 1$. Hence, the function aggregate can be estimated as

$$\frac{1}{n'}F(d', \boldsymbol{w}) \approx \frac{1}{n'} \sum_{\boldsymbol{x}_i \in d'} (1 - \psi_{\boldsymbol{w}}(\boldsymbol{x}_i)) \approx \mathbb{P}[\hat{Y} = 0 | Y = 1]. \tag{13}$$

– Enforcing accuracy parity on a neural network $\psi_{\boldsymbol{w}}$ (with softmax output activation function) in a multi-class classification task. We note that given an input $\boldsymbol{x}_i$, the raw j_{th} output of the network $\psi_{\boldsymbol{w}}(\boldsymbol{x}_i)_j$ is an estimate of the probability that $\hat{y}_i = j$. Hence, the function aggregate can be estimated as

$$\frac{1}{n'}F(d', \boldsymbol{w}) \approx \frac{1}{n'} \sum_{\boldsymbol{x}_i \in d'} \boldsymbol{\psi}_{\boldsymbol{w}}(\boldsymbol{x}_i)^T \boldsymbol{y}_i \approx \mathbb{P}[\hat{Y} = Y], \tag{14}$$

where $\boldsymbol{y}_i$ the one-hot encoding vector of y_i.

Finally, we conclude this subsection noting the similarities and differences of this work and [Tran et al., 2021]. Even though their algorithm is derived from first principles on Lagrangian duality, their resulting algorithm is equivalent to an application of the basic method of differential multipliers (BMDM) to solve a problem equivalent to (P2) when $\alpha = 0$. Nonetheless, the two algorithms differ in three main aspects:

1. The difference between BMDM and MMDM: BMDM is equivalent to MMDM when $c = 0$, that is, when the effect of making the neighbourhood of the solution subspace quadratic is not present. Moreover, the guarantee of achieving a local minimum that respects the constraints does not hold for $c = 0$ unless the problem is simple (e.g. quadratic programming).

2. How they deal with impossibilities in the goal of achieving perfect fairness together with privacy and accuracy. In [Tran et al., 2021], the authors include a limit $\boldsymbol{\lambda}^{\max}$ to the Lagrange multiplier to avoid floating point errors and reaching trivial solutions, which in our case is taken care by the tolerance α, which, in contrast to $\boldsymbol{\lambda}^{\max}$, is an interpretable parameter.

3. In this paper we consider the exact expression for the gradient $\nabla_{\boldsymbol{w}} \left| \frac{F(d', \boldsymbol{w})}{n'} - \frac{F(d'_a, \boldsymbol{w})}{n'_a} \right|$, see (11), while in [Tran et al., 2021] the authors employ the following approximation

$$\left| \frac{\nabla_{\boldsymbol{w}} F(d', \boldsymbol{w})}{n'} - \frac{\nabla_{\boldsymbol{w}} F(d'_a, \boldsymbol{w})}{n'_a} \right|, \tag{15}$$

where the sign of the difference is ignored.

In the next subsection, we extend the algorithm to PFL, which introduces two new differences with [Tran et al., 2021]. Firstly, the privacy guarantees will be provided for the individuals, and not only to the group to which they belong; and secondly, the algorithm will be tailored to federated learning.

3.2 Extending the Algorithm to Private Federated Learning

In the federated learning setting we now consider that the dataset d is distributed across K users such that each user maintains a local dataset d^k with n^k samples. Nonetheless, as in the central setting, the task is to find the model's parameters $\boldsymbol{w}^\star$ that minimize the loss function across the data samples of the users while enforcing a measure of fairness to the model. That is, to solve (P2).

To achieve this goal, we might first combine the ideas from `FederatedSGD` [McMahan et al., 2017] and the previous section to extend the developed adaptation of MMDM to FL. In order to perform the model updates dictated by (10), the central server requires the following statistics:

$$\nabla_{\boldsymbol{w}} L(d, \boldsymbol{w});\;\; F(d', \boldsymbol{w});\;\; \left[F(d'_a, \boldsymbol{w})\right]_{a\in\mathcal{A}};\;\; \nabla_{\boldsymbol{w}} F(d', \boldsymbol{w});\;\; \left[\nabla_{\boldsymbol{w}} F(d'_a, \boldsymbol{w})\right]_{a\in\mathcal{A}};\;\; n';\;\; \left[n'_a\right]_{a\in\mathcal{A}}. \tag{16}$$

However, some of these statistics can be obtained from the others, namely $F(d', \boldsymbol{w}) = \sum_{a\in\mathcal{A}} F(d'_a, \boldsymbol{w})$, $\nabla_{\boldsymbol{w}} F(d', \boldsymbol{w}) = \sum_{a\in\mathcal{A}} \nabla_{\boldsymbol{w}} F(d'_a, \boldsymbol{w})$, and $n' = \sum_{a\in\mathcal{A}} n'_a$. Moreover, as mentioned in the previous section, one might use a sufficiently large batch b of the data instead of the full dataset for each update. Therefore, we consider an iterative algorithm where, at each iteration, the central server samples m users S that report a vector with the sufficient statistics for the update, that is

$$\boldsymbol{v}^k = \left[\nabla_{\boldsymbol{w}} L(d^k, \boldsymbol{w}), \left[F(d^{k\prime}{}_a, \boldsymbol{w})\right]_{a\in\mathcal{A}}, \left[\nabla_{\boldsymbol{w}} F(d^{k\prime}{}_a, \boldsymbol{w})\right]_{a\in\mathcal{A}}, \left[n^{k\prime}{}_a\right]_{a\in\mathcal{A}}\right]. \tag{17}$$

This way, if we define the batch b as the sum of the m users' local datasets $b := \sum_{k\in S} d^k$ and the batch b' analogously, then the aggregation of each user's vectors results in

$$\boldsymbol{v} = \sum_{k\in S} \boldsymbol{v}^k = \left[\nabla_{\boldsymbol{w}} L(b, \boldsymbol{w}), \left[F(b'_a, \boldsymbol{w})\right]_{a\in\mathcal{A}}, \left[\nabla_{\boldsymbol{w}} F(b'_a, \boldsymbol{w})\right]_{a\in\mathcal{A}}, \left[|b'_a|\right]_{a\in\mathcal{A}}\right], \tag{18}$$

which contains all the sufficient statistics for the parameters' update.

Finally, the resulting algorithm, termed Fair PFL or `FPFL` and described in Algorithm 1, inspired by the ideas from [McMahan et al., 2018b; Truex et al., 2019; Granqvist et al., 2020; Bonawitz et al., 2017] guarantees the users' privacy as follows:

1. It makes sure that the aggregation of the users' sent vectors is done securely by a trusted third party, e.g. with secure aggregation [Bonawitz et al., 2017].

2. It clips the vectors $\boldsymbol{v}^k$ with a clipping bound C to restrict their ℓ_2 sensitivity, i.e. replace $\boldsymbol{v}^k$ by $\boldsymbol{v}^k \cdot \min\{1, C/\|\boldsymbol{v}^k\|_2\}$. Then, it ensures (ϵ, δ)-DP by adding Gaussian noise with variance $C^2\sigma^2$ to the clipped vector. The parameter σ is calculated according to the refined moments accountant privacy analysis from [Wang et al., 2019], taking into account the number of iterations (or communication rounds) T the algorithm will be run, the number of users (or cohort size) m that are used per iteration, the total number of users (or population size) K, and the privacy parameters ϵ and δ.

   Note that even if it later performs several post processing computations to extract the relevant information from the vector $\boldsymbol{v}$ and update the Lagrangian $\boldsymbol{\lambda}$ and the parameters $\boldsymbol{w}$, the privacy guarantees do not change thanks to the post-processing property of DP.

3.2.1 Local updates and batch size

The proposed algorithm extends the MMDM algorithm from Section 3.1 to FL adapting `FederatedSGD`, where each user uses all their data, computes the necessary statistics for a model update, and sends them to the third-party.

A natural question could be why is not `FederatedAveraging` adapted instead. That is, to perform several stochastic gradient descent/ascent updates of the model's parameters $\boldsymbol{w}$ and the Lagrange multipliers $\boldsymbol{\lambda}$, and send the difference of the updated and the original version, i.e. to send the vector $\boldsymbol{v}^k := [\boldsymbol{w}^k - \boldsymbol{w}, \boldsymbol{\lambda}^k - \boldsymbol{\lambda}]$. This way, the size of the communicated vector would be reduced from $(|\mathcal{A}|+1)p + 2|\mathcal{A}|$ to just $p + |\mathcal{A}|$ and a larger part of the computation would be done locally, increasing the convergence speed. Moreover, the clipping bound could be reduced, thus decreasing the noise necessary for the DP guarantees.

Algorithm 1: `FPFL`. The K users are indexed by k, m is the cohort size, T is the number of iterations, C is the clipping bound, and (ε, δ) are the DP parameters.

Server Executes:

$\boldsymbol{w}_0, \boldsymbol{\lambda}_0 \leftarrow$ InitializeParameters()
$\sigma \leftarrow$ CalculateNoiseScale$(K, m, \varepsilon, \delta, T)$
for each iteration $t = 1, 2, \ldots, T$ **do**
  $S_t \leftarrow$ (random set of m users)
  $\boldsymbol{v}_t \leftarrow$ SecureAggregation$(\boldsymbol{w}_{t-1}, S_t, C, \sigma)$
  $\boldsymbol{\lambda}_t \leftarrow$ UpdateMultiplier$(\boldsymbol{v}_t, \boldsymbol{w}_{t-1})$
  $\boldsymbol{w}_t \leftarrow$ UpdateParameters$(\boldsymbol{v}_t, \boldsymbol{\lambda}_t)$
end

SecureAggregation$(\boldsymbol{w}, S, C, \sigma)$**:** `/* run by a trusted third party */`
for each user k in S **do**
  $\boldsymbol{v}^k \leftarrow$ UserStatistics$(k, \boldsymbol{w})$
  $\boldsymbol{v}^k \leftarrow \boldsymbol{v}^k \cdot \min\left\{1, \frac{C}{\|\boldsymbol{v}^k\|_2}\right\}$
end
$\boldsymbol{v} \leftarrow \sum_{k \in S} \boldsymbol{v}^k + \mathcal{N}\left(0, C^2 \cdot \sigma^2\right)$
return $\boldsymbol{v}$ to server

UserStatistics$(k, \boldsymbol{w})$**:** `/* run on the user's device */`
$\boldsymbol{v} \leftarrow \left[\nabla_{\boldsymbol{w}} L(\boldsymbol{w}, d^k), \left[F(\boldsymbol{w}, d'^k_a)\right]_{a \in \mathcal{A}}, \left[\nabla_{\boldsymbol{w}} F(\boldsymbol{w}, d'^k_a)\right]_{a \in \mathcal{A}}, \left[n'^k_a\right]_{a \in \mathcal{A}}\right]$
return $\boldsymbol{v}$ to trusted third party

Unfortunately, for the proposed MMDM algorithm, this option could lead to catastrophic effects. Imagine for instance a situation where each user only has data points belonging to one group, say $a = 0$ or $a = 1$. Then in their local dataset the general population is equivalent to the population of their group and thus $F(d^{k\prime}, \boldsymbol{w}) = F(d^{k\prime}_a, \boldsymbol{w})$, implying that locally $\boldsymbol{g}(\boldsymbol{w}) = 0$. Therefore, the Lagrange multipliers will never be locally updated and the weights updates will be equivalent to those updates without considering the fairness constraints. That is, using this approach one would (i) recover the same algorithm than standard PFL and (ii) communicate a vector of size $p + |\mathcal{A}|$ instead of size p.

Another question is if one could use the algorithm as described in Section 3.2 but using only a fraction of the users' data in each update, i.e. using a batch b^k of their local dataset d^k. The answer to this is that this is possible. Nonetheless, (i) it is convenient to delegate as much computation to the user as possible and (ii) it is desirable to use as much users' data as possible to have a good approximation of the performance metrics $F(d'_a, \boldsymbol{w})$, which are needed to enforce fairness.

4 Experimental Results

We study the performance of the algorithm in two different classification tasks. The first task is a binary classification based on some demographic data from the publicly available Adult dataset [Dua and Graff, 2017]. The fairness metric considered for this task is FNR parity. The second task is a multi-class classification where there are three different attributes. This task uses a modification of the publicly available FEMNIST dataset [Caldas et al., 2018] and the fairness metric considered is accuracy parity.

For the first task we first compare the performance of the MMDM algorithm to vanilla SGD centrally. After that, for both tasks, we confirm how `FederatedSGD` deteriorates the performance of the model for the under-represented classes when clipping and noise (DP) are introduced. Finally, we demonstrate

how FPFL can, under the appropriate circumstances, level the performance of the model across the groups without largely decreasing the overall performance of the model. In all our experiments, the fairness metrics are defined as the maximum difference between the value of a measure of performance on the general testing data and the value of that performance measure for each of the groups described by the sensitive attribute.

4.1 Results on the Adult dataset

Adult dataset, from the UCI Machine Learning Repository [Dua and Graff, 2017]. This dataset consists of 32,561 training and 16,281 testing samples of demographic data from the US census. Each datapoint contains various demographic attributes. Though the particular task, predicting individuals' salary ranges, is not itself of interest, this dataset serves as a proxy for tasks with inequalities in the data. A reason this dataset is often used in the literature on ML fairness is that the fraction of individuals in the higher salary range is 30% for the men and only 10% for the women. The experiments in this paper will aim to stop this imbalance from entering into the model by balancing the false negative rate [Castelnovo et al., 2021].

Federated Adult dataset. To generate a version of the Adult dataset suitable for federated learning, it must be partitioned into individual contributions. In the experiments, differential privacy will be guaranteed per contribution. In this paper, the number of datapoints per contribution is Poisson-distributed with mean of 2.

Privacy and fairness parameters. For all our experiments, we considered the privacy parameters $\epsilon = 2$ and $\delta = 1/K \approx 5 \cdot 10^{-5}$ and the fairness tolerance $\alpha = 0.02$.

Data pre-processing. The 7 categorical variables where one-hot encoded and the 6 numerical variables where normalized with the training mean and variance. There is an underlying assumption that these means and variances can be learned at a low privacy cost. Hence, to be precise, the private models are $(\epsilon_0 + 2, 5 \cdot 10^{-5})$-DP, where ϵ_0 is a small constant representing the privacy budget for learning said parameters for the normalization.

Models considered. We experimented with two different fully connected networks. The first network, from now on the shallow network, has one hidden layer with 10 hidden units and a ReLU activation function. The second network, henceforth the deep network, has three hidden layers with 16, 8, and 8 hidden units respectively and all with ReLU activation functions. Both networks ended with a fully connected layer to a final output unit with a Sigmoid activation function.

Hyperparameters. For all the experiments, the learning rate was $\eta = 0.1$ for the network parameters and $\gamma = 0.01$ for the Lagrange multipliers. The damping parameter was $c = 2$. The batch size for the experiments learned centrally was $n_{\text{batch}} = 400$ and the cohort sized studied for the federated experiments were $m = 200$ and $m = 1000$. Finally, the clipping bounds for the shallow and the deep networks were, respectively and depending if the training algorithm was PFL or FPFL, $C = 1.3$ or $C = 2$ and $C = 2$ or $C = 2.4$. These hyper-parameters where not selected with a private hyper-parameter search and were just set as an exemplary configuration. If one desires to find the best hyper-parameters, one can do so at an additional privacy budget cost following e.g. [Abadi et al., 2016, Appendix D].

We start our experiments comparing the performance and fairness of the models trained with vanilla SGD and the MMDM adaptation. We trained the shallow and deep networks with these algorithms, where we tried to enforce FNR parity with a tolerance $\alpha = 0.02$. The results after 1,000 iterations are displayed in

Table 1, where we also consider the gap in other common measures of fairness such as the equalized odds, the demographic parity, or the predictive parity, see e.g. [Castelnovo et al., 2021].

The MMDM algorithm reduces the FNR gap from 7% to the targeted 2% for both the shallow and deep networks, while reducing the accuracy of the model by less than 0.5%, thus succeeding in its objective. Similarly, the gap in the equalized odds, which is a stronger fairness notion than the FNR parity, also decreases from around 7% to 3%. Moreover, the demographic parity gap, which considers the probability of predicting one or the other target class, also improves. In terms of predictive parity, which uses the precision as the performance function, the MMDM algorithm did not improve the parity among groups.

Table 1: Performance of a deep and a shallow network on the Adult dataset when trained with SGD and the MMDM algorithm. The fairness metric considered for MMDM is FNR parity and the tolerance is $\alpha = 0.02$.

Model	**Algorithm**	**Accuracy**	**FNR gap**	**EO gap**	**DemP gap**	**PP gap**
Deep	SGD	0.858	0.070	0.070	0.117	0.006
Shallow	SGD	0.857	0.071	0.071	0.113	0.016
Deep	MMDM	0.854	0.020	0.026	0.087	0.074
Shallow	MMDM	0.855	0.019	0.029	0.092	0.044

The second experiment is to study how clipping and DP deprecates the performance for the under-represented groups, thus increasing the fairness gap on the different fairness metrics. For that, we trained the same shallow and deep networks with `FederatedSGD` and versions of this algorithm where only clipping was performed and where both clipping and DP where included. They were trained with a cohort size of $m = 200$ for $T = 1,000$ iterations and the model with best training cohort accuracy was selected. The results are displayed in Table 2. First, we note how the performance and fairness of the deep network does not change much when going from the central to the federated setting. The shallow network, on the other hand, becomes less fair under all the metrics considered. The introduction of clipping largely increases the unfairness of the models reaching more than a 16% gap in FNR. The addition of DP does not have a larger effect on the unfairness of the models. These observations are in line with [Bagdasaryan et al., 2019], where they note that the under-represented groups usually have the higher loss gradients and thus clipping affects them more than the majority groups.

After that, we repeated the above experiment with `FPFL`. However, we noted that `FPFL` converged to a solution faster than `FederatedSGD`, and thus the models were trained for only $T = 250$ iterations. Here the model with the best training cohort accuracy that respected the fairness condition on the training cohort data was selected. Note that the fairness condition is evaluated with the noisy statistics of the cohort users recovered from the aggregation done by the third party, so a model may be deemed as fair while slightly violating the desired constraints. The results are also included in Table 2 to aid the comparison. We note how, similarly to the central case, models trained with `FPFL` achieve to enforce the fairness constraints while keeping a similar accuracy. Then, clipping does not seem to affect largely the performance of `FPFL` since it compensates the gradient loss clipping with the fairness enforcement. Finally, the addition of noise to guarantee DP is not a concern to the shallow network but it deteriorates the performance of the deep network. This is largely due to the fact that the noise is large enough so that sign of the constraints' gradient, see (11), is sometimes mistaken.

Finally, we repeat the experiments with PFL and `FPFL` with a larger cohort size, $m = 1,000$, to see if a smaller relative noise would aid the training with PFL or with `FPFL`. The results with PFL were almost identical, with similar levels of accuracy and unfairness. On the other hand, the larger signal-to-DP noise ratio helped the models trained with `FPFL` to keep models with the desired levels of FNR gap and lower unfairness measured with any other metric. Moreover, the accuracy of the models, that now work better for the under-represented group, is in fact slightly higher than for the models trained with PFL.

Table 2: Performance of a deep and a shallow network on the Adult dataset when trained with different algorithms: `FederatedSGD` without privacy, with norm clipping, and with DP, denoted as FL, FL + Clip, and PFL respectively; and `FPFL` without privacy nor norm clipping, with norm clipping only, and with DP, denoted as FFL, FFL + Clip, and FPFL respectively. The fairness metric considered for `FPFL` is FNR parity and the tolerance is $\alpha = 0.02$.

Model	Algorithm	Accuracy	FNR gap	EO gap	DemP gap	PP gap
			$m = 200$			
Deep	FL	0.853	0.078	0.078	0.125	0.015
Shallow	FL	0.851	0.121	0.121	0.122	0.036
Deep	FFL	0.854	0.001	0.030	0.093	0.048
Shallow	FFL	0.855	0.036	0.039	0.108	0.033
Deep	FL + Clip	0.848	0.160	0.160	0.131	0.056
Shallow	FL + Clip	0.844	0.169	0.169	0.129	0.051
Deep	FFL + Clip	0.852	0.008	0.023	0.081	0.031
Shallow	FFL + Clip	0.853	0.018	0.029	0.090	0.016
Deep	PFL	0.849	0.123	0.123	0.126	0.057
Shallow	PFL	0.844	0.167	0.167	0.129	0.530
Deep	FPFL	0.804	0.079	0.080	0.045	0.092
Shallow	FPFL	0.840	0.001	0.028	0.080	0.020
			$m = 1000$			
Deep	PFL	0.847	0.167	0.167	0.132	0.043
Shallow	PFL	0.847	0.148	0.148	0.126	0.041
Deep	FPFL	0.848	0.027	0.027	0.080	0.026
Shallow	FPFL	0.851	0.001	0.027	0.087	0.002

4.2 Results on the modified FEMNIST dataset

FEMNIST dataset [Caldas et al., 2018]. This dataset is an adaptation of the Extended MNIST dataset [Cohen et al., 2017], which collects more than 800,000 samples of digits and letters distributed across 3,550 users. The task considered is to predict which of the 10 digits or 26 letters (upper or lower case) is depicted in the image, so it is a multi-class classification with 62 possible classes.

Unfair FEMNIST dataset. We considered the FEMNIST dataset with only the digit samples. This restriction consists of 3,383 users spanning 343,099 training and 39,606 testing samples. The task now is to predict which of the 10 digits is depicted in the image, so it is a multi-class classification with 10 possible classes. Since the dataset does not contain clear sensitive groups, we artificially create three classes (see fig. 1):

- Users that write with a black pen in a white sheet. These users represent the first (lexicographical) 45% of the users, i.e. $\lfloor 0.45 \cdot 3,383 \rfloor = 1,522$ users. These users contain 146,554 (42.7%) training and 16,689 (42.1%) testing samples.

  The images belonging to this group are unchanged.

- Users that write with a blue pen in a white sheet. These users represent the second (lexicographical) 45% of the users, i.e. 1,522 users as well. These users contain 159,902 (46.6%) training and 18,672

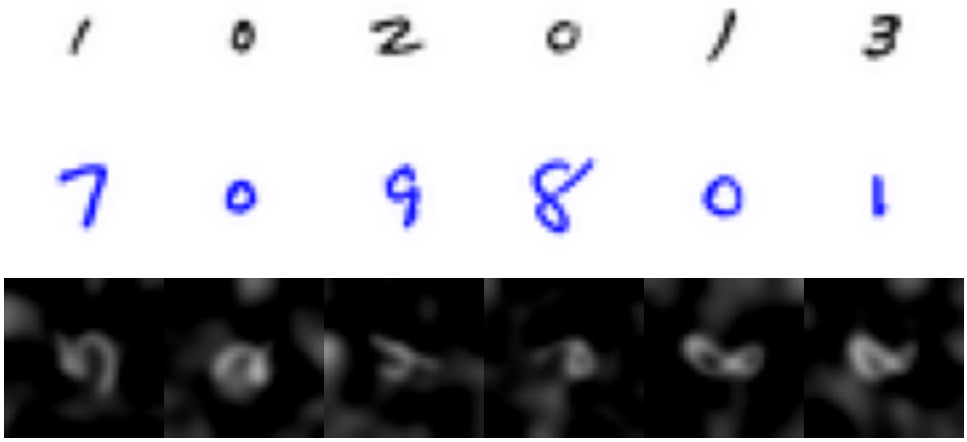

Figure 1: Example of samples from the Unfair FEMNIST dataset. Top row: samples of black pen on a white sheet digits. Middle row: samples of the blue pen on a white sheet digits. Bottom row: samples of the white chalk on a blackboard digits.

(47.1%) testing samples.

The images belonging to this group are modified making sure that the digit strokes are blue instead of black.

- Users that write with white chalk in a blackboard. These users represent the last remaining 10% of the users, i.e. 339 users. These users contain 36,643 (10.7%) training and 4,245 (10.7%) testing samples.

  The images belonging to this group are modified making sure that the digit strokes are white and the background is black. Moreover, to make the task more unfair, we simulated the blurry effect that chalk leaves in a blackboard. With this purpose, we added Gaussian blurred noise to the image, and then we blended them with further Gaussian blur. To be precise, if x is the image normalized to $[0, 1]$, the blackboard effect is the following.

$$x \leftarrow (x + \xi \circledast \kappa_2) \circledast \kappa_1, \tag{19}$$

  where $\xi_1 \sim \mathcal{N}(0, I)$ is Gaussian noise of the size of the image, κ_1 and κ_2 are Gaussian kernels[1] with standard deviation 1 and 2, respectively, and $\circledast$ represents the convolution operation. Moreover, the images are rotated 90 degrees, simulating how the pictures were taken with the device in horizontal mode due to the usual shape of the blackboards.

Privacy and fairness parameters. For all our experiments, we considered the privacy parameters $\epsilon = 2$ and $\delta = 1/K \approx 2.5 \cdot 10^{-4}$. However, for the last experiment, we consider the hypothetical scenario where we had a larger number of users $K \leftarrow 1000K$ and $K \leftarrow 1,000K$, thus decreasing the privacy parameter to $\delta \leftarrow \delta/100$ and $\delta \leftarrow \delta/1,000$ and reducing the added noise in the analysis from [Wang et al., 2019].

Model considered. We experimented with a network with 2 convolution layers with kernel of size 5×5, stride of 2, ReLU activation function, and 32 and 64 filters respectively. These layers are followed by a fully connected layer with 100 hidden units and a ReLU activation function, and a fully connected output layer

[1] We used the Gaussian filter implementation from SciPy [Virtanen et al., 2020].

Table 3: Performance of a deep and a shallow network on the Adult dataset when trained with different algorithms: `FederatedSGD` without privacy, with norm clipping, and with DP, denoted as FL, FL + Clip, and PFL respectively; and `FPFL` without privacy nor norm clipping, with norm clipping only, and with DP, denoted as FFL, FFL + Clip, and FPFL respectively. The fairness metric considered for `FPFL` is accuracy parity and the tolerance is $\alpha = 0.04$.

Algorithm	**Population**	**Accuracy**	**Accuracy gap**
$m = 100$			
FL	K	0.960	0.134
FFL	K	0.950	0.047
FL + Clip	K	0.946	0.166
FFL + Clip	K	0.954	0.053
PFL	K	0.807	0.409
FPFL	K	0.093	0.015
$m = 2,000$			
PFL	$100K$	0.951	0.157
FPFL	$100K$	0.903	0.074
PFL	$1,000K$	0.951	0.153
FPFL	$1,000K$	0.927	0.073

with 10 hidden units and a Softmax activation function. From now on, this model will be referred as the convolutional network.

Hyperparameters. For all the experiments the learning rate was $\eta = 0.1$ for the network parameters and $\gamma = 0.05$ for the Lagrange multipliers. The damping parameter was $c = 20$. Note that we set a larger damping parameter to increase the strength to which we want to enforce the constraints, given that the task is harder than before. The cohort sizes considered were $m = 100$ and $m = 2,000$. Finally, the clipping bound for the convolutional network was $C = 250$ if the training algorithm was PFL and $C = 350$ if it was `FPFL`. As previously, these hyper-parameters where not selected with a private hyper-parameter search and were just set as an exemplary configuration. If one desires to find the best hyper-parameters, one can do so at an additional privacy budget cost following e.g. [Abadi et al., 2016, Appendix D].

We start our experiments confirming again the hypothesis and findings from [Bagdasaryan et al., 2019] stating that clipping and DP disproportionately affect under-represented groups. For that, we trained a convolutional network with `FederatedSGD` and versions of this algorithm where only clipping was performed and where both clipping and DP where included. They were trained with a cohort size of $m = 100$ for $T = 2,000$ iterations and the model with best training cohort accuracy was selected. The results are displayed in Table 3. Similarly to before, we see how clipping increases the accuracy gap from 13% to almost 17%. In this case, since the number of users K is small, the necessary DP noise standard deviation is large compared to the users' sent statistics norm, and thus both the accuracy and the accuracy gap are severely affected by the addition of DP. Namely, the accuracy drops from more than 94% with clipping to 80.7% when DP is also included, and the accuracy gap increases until more than 40%.

The second experiment tests if `FPFL` can remedy the unfairness without deteriorating too much the accuracy. We trained the same convolutional network for also $T = 2,000$ iterations and the model with the best training cohort accuracy that respected the fairness condition on the training cohort data was selected.

We see how, when DP noise is not included, FPFL manages to reduce the accuracy gap with respect to FederatedSGD by around 9% while keeping the accuracy within 1%. We note how, as before, clipping does not affect largely the ability of FPFL to enforce fairness. However, note that since the data is more non-iid than before (i.e. there are more differences between the distribution of each user and the general data distribution) the models that are deemed fair in the training cohort may not be as fair in the general population, and now we see a larger gap between the desired tolerance $\alpha = 0.04$ and the obtained accuracy gap from FPFL without noise (0.047 and 0.053 without and with clipping).

When DP is included, the noise is too big for FPFL to function properly and many times the sign of the constraints' gradient, see (11), is flipped. Note that in the estimation of the performance function, i.e. $F(d_a, \boldsymbol{w})/n_a$, both the numerator and denominator are obtained from a noisy vector, thus increasing the variance of the estimation and being more sensitive to noise than the estimators for FederatedSGD.

For this reason, we considered the scenario where the number of users was 100 and 1,000 times larger, i.e. $K = 338,300$ and $K = 3,383,000$, which is a conservative assumption for federated learning settings [Differential Privacy Team, 2017]. Then, we repeated the experiment with DP FederatedSGD and FPFL where the DP noise was calculated assuming this larger number of users and where we increased the cohort size to $m = 2,000$. In this scenario, DP FederatedSGD maintained an accuracy gap of more than 15% while FPFL reduced this gap to less than a half in both cases. Nonetheless, the accuracy was slightly more deteriorated than before, with a reduction of around 5% and 2% with respect to DP FederatedSGD when the hypothetical population was increased by 100 and 1,000, respectively.

5 Conclusions

In this paper, we studied and proposed a solution for the often overlooked problem of group fairness in private federated learning. For this purpose, we adapt the modified method of multipliers (MMDM) [Platt and Barr, 1987] to empirical loss minimization with fairness constraints, which in itself serves as an algorithm for enforcing fairness in central learning. Then, we extend this algorithm to private federated learning.

Through experiments in the Adult [Dua and Graff, 2017] and a modified version of the FEMNIST [Caldas et al., 2018] datasets, we first corroborate previous knowledge that DP disproportionately affects the performance to under-represented groups [Bagdasaryan et al., 2019], with the further observation that this is true for many different fairness metrics, and not only for accuracy parity. Moreover, we demonstrate how the proposed FPFL algorithm is able to remedy this unfairness even in the presence of DP.

Limitations. The FPFL algorithm is more sensitive to DP noise than other algorithms for PFL. In our experiments, this usually requires either to increase the cohort size or to ensure that enough users take part on the model's training. Nonetheless, for the present experiments, the number of users required is still lower (more than an order of magnitude) than the usual amount of users available in professional federated learning settings [Differential Privacy Team, 2017].

6 Acknowledgments

The authors would like to thank Kamal Benkiran and Áine Cahill for their helpful discussions.

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (CCS)*, pages 308–318, 2016.

A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep speech 2: End-to-end speech recognition in English and Mandarin. In *International conference on machine learning (ICML)*, pages 173–182. PMLR, 2016.

E. Bagdasaryan, O. Poursaeed, and V. Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in Neural Information Processing Systems (NeurIPS)*, 32:15479–15488, 2019.

A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018. URL https://arxiv.org/abs/1812.01097.

A. Caliskan, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

A. Castelnovo, R. Crupi, G. Greco, and D. Regoli. The zoo of fairness metrics in machine learning. *arXiv preprint arXiv:2106.00467*, 2021.

G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

R. Cummings, V. Gupta, D. Kimpara, and J. Morgenstern. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pages 309–315, 2019.

Differential Privacy Team. Learning with privacy at scale. Technical report, Apple, 2017.

J. Ding, X. Zhang, X. Li, J. Wang, R. Yu, and M. Pan. Differentially private and fair classification via calibrated functional mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 622–629, 2020.

W. Du, D. Xu, X. Wu, and H. Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 181–189. SIAM, 2021.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

F. Fioretto, P. Van Hentenryck, T. W. Mak, C. Tran, F. Baldo, and M. Lombardi. Lagrangian duality for constrained deep learning. *arXiv preprint arXiv:2001.09394*, 2020.

M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

S. Goryczka and L. Xiong. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 2015. doi: 10.1109/TDSC.2015.2484326.

F. Granqvist, M. Seigel, R. van Dalen, Á. Cahill, S. Shum, and M. Paulik. Improving on-device speaker verification using federated learning with privacy. *Interspeech*, pages 4328–4332, 2020.

M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu. FedMGDA+: Federated learning meets multi-objective optimization. *arXiv preprint arXiv:2006.11489*, 2020.

W. Huang, T. Li, D. Wang, S. Du, and J. Zhang. Fairness and accuracy in federated learning. *arXiv preprint arXiv:2012.10069*, 2020.

M. Jagielski, M. Kearns, J. Mao, A. Oprea, A. Roth, S. Sharifi-Malvajerdi, and J. Ullman. Differentially private fair learning. In *International Conference on Machine Learning (ICML)*, pages 3000–3008. PMLR, 2019.

T. Li, M. Sanjabi, A. Beirami, and V. Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

T. Li, S. Hu, A. Beirami, and V. Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 6357–6368. PMLR, 2021.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282. PMLR, 2017.

H. B. McMahan, G. Andrew, U. Erlingsson, S. Chien, I. Mironov, N. Papernot, and P. Kairouz. A general approach to adding differential privacy to iterative training procedures. *arXiv preprint arXiv:1812.06210*, 2018a.

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations (ICLR)*, 2018b.

I. Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

J. C. Platt and A. H. Barr. Constrained differential optimization. In *Proceedings of the 1987 International Conference on Neural Information Processing Systems*, pages 612–621, 1987.

A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561. University of California Press, 1961.

B. Rodríguez-Gálvez, R. Thobaben, and M. Skoglund. A variational approach to privacy and fairness. In *2nd Privacy Preserving Artificial Intelligence Workshop (PPAI) of the AAAI Conference on Artificial Intelligence*, 2021. URL https://arxiv.org/abs/2006.06332.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

L. Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, 671(2000):1–34, 2000.

C. Tran, F. Fioretto, and P. Van Hentenryck. Differentially private and fair deep learning: A Lagrangian dual approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9932–9939, 2021.

S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019.

S. Verma and J. Rubin. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pages 1–7. IEEE, 2018.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Y.-X. Wang, B. Balle, and S. P. Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1226–1235. PMLR, 2019.

L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, 2021.

M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017.