Chapter 8

Fairness in Federated Learning

Xiaoqiang Lin[a], Xinyi Xu[a], Zhaoxuan Wu[a], Rachel Hwee Ling Sim[a], See-Kiong Ng[a], Chuan-Sheng Foo[a], Patrick Jaillet[b], Trong Nghia Hoang[c], and Bryan Kian Hsiang Low[a]

[a]*National University of Singapore, Singapore,* [b]*Massachusetts Institute of Technology, MA, USA,* [c]*Washington State University, WA, USA*

ABSTRACT

Federated Learning (FL) enables a form of collaboration among multiple clients in jointly learning a machine learning (ML) model without centralizing their local datasets. Like in any collaboration, it is imperative to guarantee *fairness* so that the clients are willing to participate. For instance, it is unfair if one client benefits significantly more than others, or if some client benefits disproportionately to its contribution in the collaboration. Additionally, it is also unfair if the ML model makes biased predictions against certain groups of clients. This chapter discusses three specific notions of fairness by highlighting their motivations from real-world use-cases, examining several specific definitions for each notion and lastly describing the corresponding algorithms proposed to achieve each notion of fairness. At the end, this chapter will also summarize the identified gaps in current research efforts into open problems.

KEYWORDS

Fairness, Federated Learning

1.1 INTRODUCTION

Federated Learning (FL) [15] allows multiple clients to collaborate in training a model with better performance (than before collaboration) without centralizing their local datasets [5, 30]. However, most existing FL systems [5, 15, 27, 30, 31] do not explicitly consider the willingness and simply assume that all clients want to collaborate [15]. This assumption can be problematic when the clients are self-interested and not obliged to participate. For example, clients might not participate if they are treated *unfairly* (e.g., receiving no or less reward while contributing resources) by the FL system. Therefore, it is imperative to guarantee *fairness* to encourage such collaborations in FL. This chapter discusses three fairness notions, *equitable fairness*, *collaborative fairness* and *algorithmic fairness*, by (1) motivating them in Section 1.1; (2) providing the formal definitions in Section 1.2; (3) describing the respective algorithms for achieving them in Section 1.3; and (4) shedding light on some open problems in Section 1.4.

FL systems can be viewed as a form of cost-sharing or resource-allocation collaboration. The clients share the costs of collecting data by tapping into the information from others' local datasets to train a *global model* (i.e., model trained on the server). The global model can be viewed as a "medium" of some resources to be allocated to the clients. Each client receives the same global model to make predictions on their local datasets.

However, due to the difference in their local datasets, the predictive performance (or loss/objective function) of a single global model differs across clients. In essence, the predictive performance is the resource that is allocated, through the medium that is the global model. Because the clients do not necessarily have an identical objective (i.e., heterogeneous [13]), it is important the training in FL is *fair* w.r.t. these different objectives (i.e., training losses of their respective local datasets). Specifically, it is unfair to allocate all the resources to a single client (i.e., training the global model to exclusively optimize the performance on that client's local dataset) because it can result in poor performance on other clients' local datasets. Instead, the global model resource should be allocated fairly (among all the clients' objectives) so that the model has small performance disparities on all local datasets, formalized as *equitable fairness*.

In some other practical scenarios, the clients compete with each other (e.g., companies providing similar services/products) [27, 31]. These clients are *self-interested*, namely they focus on maximizing their own utility (e.g., the trained model's performance on their local dataset). In contrast to the aforementioned scenario, these self-interested clients are only willing to contribute to help others if doing so (strictly) improves their own utility. Moreover, it can appear exploitative if a client i that contributes more than client j receives a reward lower than that received by client j, and can discourage collaboration. It thus motivates a different notion of fairness. Formally, the *contribution* of a client characterizes how much a client shares (directly or indirectly by training on its local dataset) with other clients [22, 25] and the *reward* of a client specifies how much a client gains from the collaboration [21, 29]. The so-called *collaborative fairness* [14] stipulates that the rewards should be commensurate with the contributions, so that the clients are rewarded more if they contribute more, and vice versa.

Lastly, dealing with bias in data in machine learning (ML) is a known challenge [16, 33], which can be exacerbated by the multi-client setting of FL. Note that this perspective differs from the two previous settings in that it explicitly considers *how* the trained global model predicts on certain data. From the perspective of each client, the goal of removing bias is to ensure the trained model does not discriminate against certain protected groups (e.g., data whose sensitive features such as gender, race having certain values) [2]. In practice, the clients do not necessarily have aligned goals due to different local datasets [1]. Consequently, trying to eliminate/reduce discrimination among groups for

one client can deteriorate that for another. Therefore, ideally, the trained model should not be biased w.r.t. any client's local dataset, called *algorithmic fairness*. In other words, suppose a data point is in the protected group (e.g., a record of a person), then this data point should *not* be discriminated against regardless of which local dataset it belongs to.

1.2 NOTIONS OF FAIRNESS

TABLE 1.1 Algorithms analyzed in this chapter to achieve different notions of fairness.

Category	Fairness notion	Algorithm
Equitable fairness	Def. 1	Algo. 1
	Def. 2	Algo. 2
	Def. 3	FAFL[2]
Collaborative fairness	Def. 4	FGFL [29]
		GoG [18]
	Def. 5	FGFL [29]
		GoG [18]
	Def. 6	FLI [32]
Algorithmic fairness	Def. 7	FAFL [2]
	Def. 8	FAFL [2]

TABLE 1.2 A summary of current works on fairness in FL.

	Algorithmic fairness	Equitable fairness	Collaborative fairness
FairFL [34]	✓		
FairFed [4]	✓		
FCFL [1]	✓	✓	
FAFL [2]	✓	✓	
AFL [17]		✓	
q-FFL [12]		✓	
Ditto [11]		✓	
CFFL [14]			✓
RFFL [28]			✓
FGFL [29]			✓
FLI [32]			✓

1.2.1 Equitable Fairness

The equitable fairness aims to equalize the performance of the global model on all clients. As enforcing strict equality is not always desirable depending on the applications [1], three types of equitable fairness are defined: good-intent equitable fairness [17], performance equitable fairness [12], and Pareto-optimal equitable fairness [1].

In FL, clients normally have heterogeneous local data distributions and local objectives. However, the objective of FedAvg [15] algorithm is to minimize the weighted average loss: $g(\mathbf{w}) = (1/\sum_{i=1}^{n} |D_i|) \sum_{m=1}^{n} |D_m| g_m(\mathbf{w})$, which can not guarantee equitable losses across all clients in the resultant model. Note that $g_m(\mathbf{w})$ is the loss of model $\mathbf{w}$ on the local dataset of client m. For example, the client m with less data points than others will have a lower weight $|D_m|/\sum_{i=1}^{n} |D_i|$, meaning that during training "less optimization resource" is allocated to client m. Consequently, the client might suffer from a larger loss

than others w.r.t. the trained model $\mathbf{w}$. Additionally, if a client m has local data distribution significantly different from others', namely heterogeneous, it might suffer from bad performance on its local data distribution with the trained model. Intuitively, the data from other clients are not so helpful in improving the performance of the model on the client m's data due to the heterogeneity. Consequently, the client m might have higher losses than others.

Therefore, without extra equality guarantees from the system, clients with fewer data points or with more heterogeneous data distribution will not participate in the collaboration due to the low performance of the trained model on their local datasets. In that case, the inclusiveness of the system will decrease. For example, there might exist a monopoly market in which one client (i.e., company) has majority of the data and the rest of the clients each have very little data. The clients with less data can be treated unfairly with low performance on their local datasets and might leave the collaboration which will result in an undesirable single-participant collaboration. Additionally, the utility of the collaboration will be reduced (i.e., the performance of the trained model on a jointly agreed test dataset) due to having less data in the collaboration. Based on this intuition, a fairness notion that seeks to maximize the performance of the worst-performing client is introduced:

Definition 1 (Good-intent equitable fairness [17])**.** For trained models $\mathbf{w}$ and $\tilde{\mathbf{w}}$, the model $\mathbf{w}$ achieves better good-intent equitable fairness than $\tilde{\mathbf{w}}$ if $\max_{m\in\{1,\dots,n\}} g_m(\mathbf{w}) < \max_{m\in\{1,\dots,n\}} g_m(\tilde{\mathbf{w}})$.

The good-intent equitable fairness states that a model from FL training is fairer if the maximum loss across all clients is lower. Therefore, a fair model will not underfit to a particular local dataset (i.e., having a very high loss on some client whilst having low losses on others) and can generalize better to all local datasets. In that case, the clients with less data or more heterogeneous data distribution can obtain a better performance which will result in a lower performance disparity among all clients. The good-intent equitable fairness does not enforce a strong equitable performance across all the clients since it only optimizes the performance of the worst-performing client without considering other clients and thus high performance disparity might still be observed. In contrast, another notion of equitable fairness considers the equitable performance across all the clients directly, via a formal equality measure over the variation of the performances. The standard deviation of model performances across the local datasets of all clients is used to characterize how much the performances differ from each other. It leads to the definition:

Definition 2 (Performance equitable fairness [12])**.** For a set of models $\mathbf{W} = \{\mathbf{w}' : |g(\mathbf{w}') - \min_{\mathbf{w}} g(\mathbf{w})| \leq \epsilon\}$, model $\mathbf{w}_1 \in \mathbf{W}$ achieves better ϵ performance equitable fairness than model $\mathbf{w}_2 \in \mathbf{W}$ if $\text{std}((a_m(\mathbf{w_1}))_{m=1}^n) < \text{std}((a_m(\mathbf{w_2}))_{m=1}^n)$. ϵ is the tolerance of the degradation in performance, the $a_m(\mathbf{w})$ is the prediction

accuracy on the local dataset of client m with model $\mathbf{w}$ and $\text{std}((a_m(\mathbf{w}))_{m=1}^n)$ is the standard deviation of $(a_m(\mathbf{w}))_{m=1}^n$.

The performance equitable fairness aims to find a model that has the most equitable performances (quantified by standard deviation) among all models that have the same (i.e., $\epsilon = 0$) or similar performance w.r.t. the FedAvg objective. To interpret, the model that achieves performance equitable fairness does not sacrifice too much on the overall model performance with a tolerance of at most ϵ. The performance equitable fairness is similar to the egalitarian's perspective which also favors equal treatment (i.e., equal in performances in this case).

In some cases, Definition 2 can be unfair. For example, the dataset in client m can have naturally higher *irreducible error* [7] than that of client m'. Enforcing the loss on client m' to be the same as client m will be unfair to client m' since it might be possible for a model to achieve a lower loss on client m' without making the loss on client m higher. Therefore, it is preferable to have a better overall performance that does not trade off the performances of other clients. A Pareto-optimal outcome is where no one in the collaboration can be better off without making someone else worse off. Building on the good-intent equitable fairness, a notion that additionally considers the Pareto-optimality is introduced:

Definition 3 (Pareto-optimal equitable fairness [1])**.** Among the models $\mathbf{W} = \{\mathbf{w}^* : \mathbf{w}^* = \arg\min_{\mathbf{w}} \max_{m \in N} g_m(\mathbf{w})\}$, a model $\mathbf{w}^p \in \mathbf{W}$ achieves Pareto-optimal equitable fairness if:

$$\nexists \mathbf{w} \in \mathbf{W}, s.t.\ \forall m \in N : g_m(\mathbf{w}) \leq g_m(\mathbf{w}^p) \text{ and } \exists m' \in N : g_{m'}(\mathbf{w}) < g_{m'}(\mathbf{w}^p)\ .$$

The Pareto-optimal equitable fairness considers the improvement of performance not only on the worst-performing client as in Definition 1 but also on other clients. Specifically, if there exist multiple models that can achieve the min-max losses (i.e., $|\mathbf{W}| > 1$), the Pareto-optimal equitable fairness favors a model that achieves maximum performances on the clients in which any of their performance cannot be improved without decreasing some others'.

The models that achieve Pareto-optimal equitable fairness also achieve good-intent equitable fairness while the same does not hold reversely. However, the Pareto-optimal equitable fairness can conflict with performance equitable fairness sometimes. For example, assume that there exists a model whose local loss on each client's local dataset is equivalent to the irreducible error of the corresponding client's dataset. Additionally, assume the irreducible errors are different among the clients. Consequently, among all the models that achieve the lowest loss on the worst-performing client (i.e., $\mathbf{W}$ in Definition 3), there exists a model that can achieve better performances on some clients without hurting the performance of others. It is fair in this case for these high-performing clients to keep the better than worst-performing client's performances (i.e., Pareto-optimal

equitable fairness) instead of eliminating the excess performances completely (i.e., performance equitable fairness).

To conclude, the notion of good-intent equitable fairness and performance equitable fairness aim to achieve equal model performances across different clients which are more suitable for application scenarios where clients have similar data distribution. In contrast, when clients have highly heterogeneous local data distributions (i.e., companies with user data from different geographic populations), improving the worst-performing client or forcing other clients to have similar performances to the worst-performing ones would degrade some clients' model performances on their local data distribution dramatically. In that case, Pareto-optimal equitable fairness would be better since it allows improvements in the model performances of some clients without hurting others.

1.2.2 Collaborative Fairness

In the case of self-interested clients (e.g., companies that compete with each other), it will be unfair to the clients with data of higher quantity/quality (i.e., higher contribution clients) to receive the same models/rewards as the clients with data of lower quantity/quality (i.e., lower contribution clients) in the FL system. Otherwise, higher contribution clients may lose their competitive edge and thus be discouraged from the collaboration. Therefore, to encourage the clients with high-quality data to join the collaboration, the rewards given to all the clients should be commensurate with their contributions. Pearson collaborative fairness is introduced as a general idea of designing rewards that are commensurate with the contributions of the clients [28]. The Shapley fairness [21, 29], incorporating the Shapley value from cooperative game theory, provides some desirable properties. Finally, a notion of regret-minimized collaborative fairness [32] is defined to additionally consider the costs for the resources of the clients.

To define collaborative fairness, a contribution estimation method and a reward mechanism based on the contribution estimates are needed. The contribution estimates are the values assigned to each client to represent their contributions in training the global model. For example, an intuitive contribution estimate for a client can be defined by how much the performance of the global model (e.g., test accuracy on a test dataset) is due to the participation of the client. Interested readers can refer to the latter chapter on data valuation in federated learning for a more detailed discussion on contribution estimates defined on data quality. Some other works consider more specific contributions. [8] proposes to evaluate the contribution based on the clients' reputations and the amount of resources they spend on computing/communicating the gradients obtained from their data. [24] proposes to evaluate the contributions by a voting mechanism and design rewards based on the voting results. [32] considers the long-term profit sharing setting in which the waiting time of rewards is accounted for in designing the re-

wards. Rewards can be classified into monetary reward [19] and non-monetary reward [18, 28, 29]. The non-monetary reward is normally considered when the monetary reward is unavailable [21]. An example of non-monetary reward is the model reward which gives clients models with different performances based on their contributions. Interested readers can find a more detailed discussion on model reward in a latter chapter on incentive for federated learning. In general, the reward for each client should be commensurate with its contribution.

Pearson correlation coefficient can be used to quantitatively evaluate the commensurate relationship between rewards and contributions. Denote the rewards for all clients as $(r_m)_{m\in N}$ and the contributions of all clients as $(c_m)_{m\in N}$. It leads to the following definition:

Definition 4 (Pearson collaborative fairness [28])**.** A federated learning (FL) system achieves Pearson collaborative fairness if $\rho((c_m)_{m\in N}, (r_m)_{m\in N}) > 0$ where $\rho((c_m)_{m\in N}, (r_m)_{m\in N})$ denotes Pearson correlation coefficient between $(c_m)_{m\in N}$ and $(r_m)_{m\in N}$.

The Pearson collaborative fairness provides a simple method to certify if an FL system achieves the core idea of collaborative fairness (i.e., higher rewards to higher contribution clients). It does not specify how the contributions of clients are defined or which form of rewards should be given to the clients. The Pearson collaborative fairness only requires the reward r_m to be positively correlated to the contribution c_m.

However, it is not sufficient in some cases where a more careful and detailed design of rewards based on the contributions is needed. For example, assume that we have a contribution estimate of $\{0, 1, 1.1, 3\}$ for a collaboration involving 4 clients with the corresponding reward values $\{1, 2, 2.1, 4\}$. This will result in a Pearson correlation coefficient of 1 but since the reward is non-zero for the client with zero contribution, it can attract free riders. In addition, take another example where the corresponding reward values are set to be $\{0, 1.1, 1, 3\}$, which also results in a high Pearson correlation coefficient (i.e., 0.998) with respect to the contributions $\{0, 1, 1.1, 3\}$. In this case, the second client contributes less than the third client but it receives a better reward. This is unfair for the third client despite the fact that the rewards are positively correlated to the contributions in general. Thus, a more detailed reward mechanism design is needed. To define it properly, more detailed contribution estimates should be considered.

The existing literature [18, 22, 23, 29] commonly adopts the Shapley value [20] to define the contributions of clients in FL. To determine the contribution of a client, Shapley value computes the average marginal contribution of the client to its predecessor coalitions over all possible sequential orders of participation. Therefore, Shapley value is independent of the order of participation which is favorable in FL since for a fixed iteration of FL training, the clients are selected

without enforcing a particular order [5, 15, 27]. Besides, Shapley value uniquely satisfies several properties (e.g., linearity, symmetry, null player etc.) [20] which are desirable for designing the rewards. Denote $N = \{1, \ldots, n\}$ as the grand coalition formed by all the clients, and any $C \subseteq N$ as coalitions of clients, and $\nu(C)$ is the utility function that computes the utility of the model (e.g., test accuracy or negative log-likelihood on a test dataset) trained on the dataset $\{D_m\}_{m \in C}$. The Shapley value [20] for client $m \in N$ is

$$\phi_m(\nu) \quad = \quad \frac{1}{N} \sum_{C \in N \setminus \{m\}} \frac{1}{\binom{N-1}{|C|}} \Big[\nu(C \cup \{m\}) - \nu(C) \Big] .$$

Based on the Shapley value definition of contribution estimates, another notion of collaborative fairness is defined:

Definition 5 (Shapley collaborative fairness [21, 22])**.** Given a utility function ν, an FL system achieves Shapley collaborative fairness if $r_m = f(\phi_m(\nu)), m \in N$, where $f(\cdot)$ is a strictly increasing function with $f(0) = 0$. The reward defined has the following properties:

- **Uselessness.** If client m has zero marginal contribution to all coalitions, $r_m = 0$.
- **Symmetry.** If client m has the same marginal contributions to all coalitions as another client m', $r_m = r_{m'}$.
- **Strict Desirability.** If client m makes a strictly better marginal contribution to a specific coalition than client m' and the same marginal contributions to any other coalitions as m', $r_m > r_{m'}$.
- **Strict Monotonicity.** If the client m makes a strictly better contribution to a specific coalition, ceteris paribus, it will receive a strictly better reward.

In contrast to Pearson collaborative fairness, Shapley collaborative fairness provides specific details of how the reward values should be decided. Moreover, it inherits several desirable properties from Shapley value (i.e., uselessness, symmetry, strict desirability). Specifically, Shapley collaborative fairness explicitly rewards more to the clients with higher marginal contributions than clients with lower marginal contributions (i.e., Strict Desirability). For two clients with exactly the same marginal contributions, it assigns the same reward to them (i.e., Symmetry). It also discourages free-riders from participating in the FL system since they will get a zero reward (i.e., Uselessness) but have to bear their own communication/computation costs. Beyond these properties, Shapley collaborative fairness provides an extra property of Strict Monotonicity. It ensures that if a client contributes more, *ceteris paribus*, this client will be better rewarded. Hence, it incentivizes clients which have the potential to contribute more (e.g., having the ability to collect more data) to do so and receive a better reward. Some existing works adopt this Shapley collaborative fairness. For instance, in ρ-Shapley fairness [21], $r_m = k\phi_m^\rho$ where k and ρ are adjustable parameters and

in FGFL [29], $r_m \propto \lfloor \tanh(\beta\phi_m)/\max_{i\in N} \tanh(\beta\phi_i) \rfloor$ where β is an adjustable parameter and the function tanh() can be replaced with another monotonic function while preserving the theoretical properties.

The collaborative fairness defined above does not consider the costs of resources within each client (e.g., costs of collecting data or computational resources). However, the costs can also be a vital measure for clients to decide whether to participate or not, especially if clients are organizations that make decisions based on the net profit (i.e., reward minus cost). If a client receives a reward lower than the costs of providing the corresponding resources, it will regret participating in the collaboration due to the negative net profit and will not participate next time. Therefore, it is desirable to minimize the overall regrets of the clients. Denote the cost of resources of the clients as $(s_m)_{m\in N}$, and the regret of clients as $k_m = \max\big((s_m - r_m), 0\big), m \in N$, it leads to the following notion of fairness which is a simplified version of [32]:

Definition 6 (Regret-minimized collaborative fairness)**.** A federated learning (FL) system achieves regret-minimized collaborative fairness if $(r_m^*)_{m\in N} = \arg\min_{(r_m)_{m\in N}} \sum_{m=1}^{n} (k_m)^2 - \alpha \sum_{m=1}^{n} r_m c_m, s.t. \sum_{m=1}^{n} r_m \leq b$ where b is the total reward budget and α is an adjustable parameter.

In the objective of the optimization problem in Definition 6, the term $\sum_{m=1}^{n} (k_m)^2$ is the sum of square of the clients' regrets. Thus, minimizing it will reduce overall regrets. The quadratic expression k_m^2 naturally avoids the case that a few clients have very high regrets while others have zero regrets. The term $\sum_{m=1}^{n} r_m c_m$ will be large if the rewards r_m's are high for clients with high contributions c_m's. Therefore, the regret-minimized collaborative fairness tries to divide the total budget b as rewards to minimize the overall regret while simultaneously ensuring the clients with larger contributions receive higher rewards. The adjustable parameter $\alpha > 0$ balances the importance of these two objectives.

1.2.3 Algorithmic Fairness

Informally, a model is said to achieve algorithmic fairness (specifically, group fairness) in ML if it does not discriminate against certain groups (i.e., data with sensitive attributes having certain values). Put differently, the trained model should have similar performances across different groups, namely low performance disparities across groups [16]. The algorithmic fairness in FL is closely related to that in ML, so a notion of global algorithmic fairness is introduced as an extension of that in ML. Another notion of multi-client algorithmic fairness is introduced and specific to the FL setting.

Without loss of generality, consider the case of binary classification where the label $y \in \{0, 1\}$, and assume that there exists a global sensitive attribute (e.g., gender) $s \in \{0, 1\}$. Denote the trained federated model as a prediction function

$f(\mathbf{w}, \cdot)$ parameterized by the learned parameters $\mathbf{w}$, and denote the predicted label for input x as $\hat{y}_{\mathbf{w}} = f(\mathbf{w}, x)$. Denote $D_m^{(ij)} = \{(x, y, s) : (x, y, s) \in D_m, \hat{y}_{\mathbf{w}} = i, s = j\}$ and $D_m^{(\cdot j)} = \{(x, y, s) : (x, y, s) \in D_m, \hat{y}_{\mathbf{w}} \in \{0, 1\}, s = j\}$. There are various definitions using different measures to quantify the disparity of the model performances among all groups (e.g., demographic parity [3], equalized odds [6] and equal opportunity [6]). We focus on demographic parity to describe the following definitions which can be easily extended to other disparity measures. A definition of algorithmic fairness based on this measure is:

Definition 7 (Global algorithmic fairness [2, 4])**.** In a federated learning (FL) system, a model $f(\mathbf{w}, \cdot)$ achieves ϵ global algorithmic fairness if:

$$\Delta\mathrm{DP}_g(\mathbf{w}) \quad = \quad \left| \frac{\sum_{m=1}^{n} |D_m^{(11)}|}{\sum_{m=1}^{n} |D_m^{(\cdot 1)}|} - \frac{\sum_{m=1}^{n} |D_m^{(10)}|}{\sum_{m=1}^{n} |D_m^{(\cdot 0)}|} \right| \quad \leq \quad \epsilon \,. \tag{1.1}$$

Here, $\sum_{m=1}^{n} |D_m^{(11)}| / \sum_{m=1}^{n} |D_m^{(\cdot 1)}|$ is the probability of the group with $s = 1$ being predicted as positive, and $\sum_{m=1}^{n} |D_m^{(10)}| / \sum_{m=1}^{n} |D_m^{(\cdot 0)}|$ is that for the group with $s = 0$. Therefore, $\Delta\mathrm{DP}_g(\mathbf{w})$ is the absolute difference of the probability of data been predicted as positive with model $\mathbf{w}$ between two groups on the aggregated dataset $\{D_m\}_{m \in N}$. A model achieves global algorithmic fairness if the difference of probability of predicting positive between two groups is less than ϵ. Though the global algorithmic fairness is straightforward, enforcing it can sometimes have limited usefulness, especially when the clients have heterogeneous local data distributions. Then, it is possible that the federated model achieves algorithmic fairness in a specific client but not in some others.

For example, in the task of income level prediction (i.e., Adult dataset [9]), a model is trained to predict if a person has high or low income (i.e., binary classification) based on 14 features characterizing personal information (e.g., age, sex, education, occupation etc.). We define the age within 30 - 50 as high-income age since most people with high income lie within this age interval. Assume that sex is the sensitive attribute to protect. Additionally, assume that in client m the percentage of populations with high-income age is similar between males and females while in client m' the percentage of populations with high-income age is higher in males than that of females. A simple model that makes predictions based solely on the attribute age will probably achieve fairness in client m (i.e., predicting 1 with the same probability between male group than female group). However, the model can hardly achieve fairness in client m' since it will predict 1 with a higher probability on the male group due to its higher percentage of high-income age populations. Another problem is that the protected attribute can vary across clients. For example, the sensitive attribute in the dataset of client m is sex while it is race in that of client m'. It is not clear how enforcing global fairness w.r.t. some unified sensitive attributes can be useful to

each client in these scenarios. Therefore, a more refined algorithmic fairness notion with performance disparity defined w.r.t. each client is needed. Define a unique sensitive attribute for each client m: $c_m \in \{0, 1\}$ which can be the same across different clients. Denote $D_m^{(ij)} = \{(x, y, c_m) : (x, y, c_m) \in D_m, \hat{y}_{\mathbf{w}} = i, c_m = j\}$ and $D_m^{(\cdot j)} = \{(x, y, c_m) : (x, y, c_m) \in D_m, \hat{y}_{\mathbf{w}} \in \{0, 1\}, c_m = j\}$. It leads to the definition:

Definition 8 (Multi-client algorithmic fairness [1])**.** In a federated learning (FL) system, a model $f(\mathbf{w}, \cdot)$ achieves $\{\epsilon_m, m \in \{1, \ldots, n\}\}$ multi-client algorithmic fairness if:

$$\Delta\text{DP}_m(\mathbf{w}) \quad = \quad \left| \frac{|D_m^{(11)}|}{|D_m^{(\cdot 1)}|} - \frac{|D_m^{(10)}|}{|D_m^{(\cdot 0)}|} \right| \quad \leq \quad \epsilon_m \ \ \forall m \in \{1, \ldots, n\} \ . \tag{1.2}$$

$\Delta\text{DP}_m(\mathbf{w})$ is the absolute difference of the probability of samples been predicted as positive with model $\mathbf{w}$ between two groups on the local dataset of client m.

The multi-client algorithmic fairness requires low performance disparity with respect to each client m with a possibly unique sensitive attribute c_m and an individual budget ϵ_m for the performance disparity in each client. As illustrated in the income level prediction example, achieving fairness in the local dataset of different clients can sometimes conflict with each other. Therefore, it is more challenging to achieve multi-client algorithmic fairness than global algorithmic fairness even if $\sum_{m=1}^{n} \epsilon_m = \epsilon$.

1.3 ALGORITHMS TO ACHIEVE FAIRNESS IN FL

1.3.1 Algorithms to achieve equitable fairness

We will focus on discussing the details of the AFL [17] and q-FFL [12] algorithms to achieve good-intent equitable fairness and performance equitable fairness correspondingly here and refer readers to [2] for an in-depth discussion on the algorithm to achieve Pareto-optimal equitable fairness.

A common approach to equitable fairness is to modify the global objective of the training to achieve similar model performances on the clients' local data. AFL [17] proposes a min-max objective. Define Δ^n as a $n-1$ dimension probability simplex and $\lambda \in \Delta^n$, the objective function is defined as $g(\mathbf{w}, \lambda) = \sum_{m=1}^{n} \lambda_m g_m(\mathbf{w})$ where $g_m(\mathbf{w})$ is the local training loss of model parameterized by $\mathbf{w} \in \mathcal{W}$ on client m. The objective of the AFL is defined as:

$$\min_{\mathbf{w} \in \mathcal{W}} \max_{\lambda \in \Delta^n} g(\mathbf{w}, \lambda) \ . \tag{1.3}$$

The objective in Eq. (1.3) can be viewed as a two-player game, where the player A wants to find λ such that the weighted loss can be maximized, and the player B

wants to find the parameter $\mathbf{w}$ such that the weighted loss can be minimized. Since $g(\mathbf{w}, \lambda)$ is linear in λ, the optimal λ would be in $\{\lambda : \exists m \in N : \lambda_m = 1, \forall m' \in N \setminus \{m\} : \lambda_{m'} = 0\}$. Therefore, the solution $\mathbf{w}^*$ of Eq. (1.3) is also the solution of $\min_{\mathbf{w}\in\mathcal{W}} \max_{m\in\{1,\dots,n\}} g_m(\mathbf{w})$. Consequently, solving Eq. (1.3) will get a model that achieves good-intent fairness according to the definition. To solve the optimization problem in Eq. (1.3), gradient estimators for $\nabla_\lambda g(\mathbf{w}, \lambda)$ and $\nabla_\mathbf{w} g(\mathbf{w}, \lambda)$ can be used. Denote the gradient estimators as $\delta_\lambda g(\mathbf{w}, \lambda)$ and $\delta_\mathbf{w} g(\mathbf{w}, \lambda)$ accordingly. Denote [n] as the uniform distribution on $\{1, \dots, n\}$. In AFL, $\delta_\lambda g(\mathbf{w}, \lambda)$ is computed as follows: sample $m \sim [n]$, and then sample $i \sim [|D_m|]$. Then set $[\delta_\lambda g(\mathbf{w}, \lambda)]_m = n\ell(\mathbf{w}; x_{m,i}, y_{m,i})$ and $[\delta_\lambda g(\mathbf{w}, \lambda)]_k = 0, \forall k \in N \setminus \{m\}$ where $\ell(\mathbf{w}; x_{m,i}, y_{m,i})$ is the loss function of model $\mathbf{w}$ on the i-th data point in client m. Similarly, to compute $\delta_\mathbf{w} g(\mathbf{w}, \lambda)$, firstly sample $i_m \sim [|D_m|], \forall m \in N$ with uniform distribution accordingly, then $\delta_\mathbf{w} g(\mathbf{w}, \lambda) = \sum_{m=1}^n \lambda_m \nabla_\mathbf{w} \ell(\mathbf{w}; x_{m,i_m}, y_{m,i_m})$. The pseudo-code for STOCHASTIC-AFL which is the algorithm to solve AFL objective is presented in Algorithm 1. The PROJECT$(\tilde{\mathbf{w}}, \mathcal{W})$ in Algorithm 1 is the projecting function that finds $\mathbf{w}_p = \arg\min_{w\in\mathcal{W}} \|\mathbf{w} - \tilde{\mathbf{w}}\|^2$ and can be efficiently solved in near-linear time [26]. Its convergence guarantee is established in [17].

Algorithm 1 STOCHASTIC-AFL

Input: Step size for gradient update $\gamma_\mathbf{w}$ and γ_λ, number of gradient update step T.
Initialization: $\mathbf{w}_0$ and λ_0
  for $t = 1$ to T **do**.
    Compute the stochastic gradient estimators: $\delta_\mathbf{w} g(\mathbf{w}, \lambda)$ and $\delta_\lambda g(\mathbf{w}, \lambda)$.
    $\tilde{\mathbf{w}}_t = \mathbf{w}_{t-1} - \gamma_\mathbf{w} \delta_\mathbf{w} g(\mathbf{w}_{t-1}, \lambda_{t-1})$.
    $\tilde{\lambda}_t = \lambda_{t-1} - \gamma_\lambda \delta_\lambda g(\mathbf{w}_{t-1}, \lambda_{t-1})$.
    $\mathbf{w}_t = \text{PROJECT}(\tilde{\mathbf{w}}_t, \mathcal{W})$.
    $\lambda_t = \text{PROJECT}(\tilde{\lambda}_t, \Delta^n)$.
  end for
Output:
  $\mathbf{w}^T = 1/T \sum_{t=1}^T \mathbf{w}_t$
  and $\lambda^T = 1/T \sum_{t=1}^T \lambda_t$.

Algorithm 2 q-FedSGD

Input: The number of clients selected every iteration K, the total training iteration T, constant L.
Initialization: $\mathbf{w}_0$.
  for $t = 0$ to T **do**
    M clients are selected which form a set S_M, each client m is chosen with probability p_k.
    for $k \in S_M$ **do**
      $\delta_k^t = g_k^q(\mathbf{w}_t) \nabla g_k(\mathbf{w}_t)$.
      $h_k^t = q g_k^{q-1}(\mathbf{w}_t) \|\nabla g_k(\mathbf{w}_t)\|^2 + L g_k^q(\mathbf{w}_t)$.
    end for
    $w_{t+1} = w_t - \frac{\sum_{k\in S_M} \delta_k^t}{\sum_{k\in S_M} h_k^t}$.
  end for
Output: w_T.

For performance equitable fairness, q-FFL [12] is proposed. Intuitively, if a local objective has a higher weight, the global objective prioritizes the minimization of this local objective. Therefore, q-FFL proposes to assign higher weights for the local objectives with higher loss values and thus make the losses distributed equitably among clients. The reweighting process is done during

training dynamically since it is difficult to do *a priori*. The objective of q-FFL is

$$\min_{\mathbf{w}} f_q(\mathbf{w}) \quad = \quad \sum_{m=1}^{n} \frac{p_m}{q+1} g_m^{q+1}(\mathbf{w}) \tag{1.4}$$

where $p_m = |D_m| / \sum_{m=1}^{n} |D_m|$ and q is an adjustable parameter. Borrowing the idea from α-fairness [10], q-FFL can adjust q to satisfy different levels of equality for the clients' performances. A larger q means that the objective emphasizes the loss of the lower performing clients and thus enforcing better equality of performance across all clients. In contrast, a lower q makes the objective more similar to that in FedAvg which does not consider equality, in particular, $q = 0$ recovers the FedAvg objective. The pseudo-code for q-FedSGD, the algorithm to solve the q-FFL objective, is shown in Algorithm 2. To make the algorithm converge, the step size of the gradient update is chosen according to different values of q. q-FedSGD proposes to use h_k^t to control the step size so that no manual tuning on step size is needed for different q to ensure convergence.

Both STOCHASTIC-AFL and q-FedSGD make changes to the training objective to achieve their corresponding targeted fairness notions. Additionally, both algorithms are shown to converge under certain assumptions [12, 17]. q-FedSGD provides a more flexible control over the trade-off between fairness and utility than STOCHASTIC-AFL due to the adjustable parameter q in Eq. (1.4). Surprisingly, though q-FedSGD does not explicitly strive to get a better performance on the worst-performing client, it achieves better good-intent equitable fairness than STOCHASTIC-AFL on several datasets [12] while achieving better performance equitable fairness simultaneously. From Tab. 1.3, STOCHASTIC-AFL is more costly than q-FedSGD in both communication and running time. Both of them take extra communication costs than FedAvg.

1.3.2 Algorithms to achieve collaborative fairness

For different notions of collaborative fairness, FGFL [29] and GoG [18] are proposed to achieve Pearson collaborative fairness and Shapley collaborative fairness, and FLI [32] is proposed to achieve regret-minimized collaborative fairness. We will specifically outline FGFL and GoG algorithms in detail and refer readers to [32] for FLI due to the space limits. Since a contribution estimation and an incentive mechanism based on the contribution estimates are two vital components in designing algorithms to achieve collaborative fairness, we will focus on discussing the differences between FGFL and GoG in designing these two components correspondingly.

To compute the contribution estimates, FGFL uses the gradient information from each client and calculates their similarities to the aggregated gradients to estimate their contribution to the training. Intuitively, the aggregated gradient is the

direction in which the global loss will decrease. If a client has a gradient (vector) that is (directionally) similar/aligned to the aggregated gradient (vector), it means that the client's gradient is highly effective in reducing global loss and thus has a high contribution to the training. Using this intuition of vector alignment between gradient vectors, a cosine similarity-based Shapley value contribution is defined (i.e., cosine similarity as ν). With the contribution estimates, FGFL gives clients models with different performances commensurate with their contributions. To differentiate the performance, FGFL gives clients gradient updates with different proportions of values masked by zero. Intuitively, a higher proportion of masked values means less information about gradient update is given to the clients, which will lead to lower model performance. The gradients are computed as follows:

$$r_m^{(t)} = \left\lfloor \mathbf{D}_{\mathbf{w}} \tanh\left(\beta c_m^{(t)}\right) / \max_{i \in \mathcal{N}} \tanh\left(\beta c_i^{(t)}\right) \right\rfloor \text{ and } \mathbf{v}_m^{(t)} = \text{mask}\left(\mathbf{u}_{\mathcal{N}}^{(t)}, r_m^{(t)}\right) \quad (1.5)$$

where $c_m^{(t)}$ and $r_m^{(t)}$ are contribution and reward of client m up to iteration t, $\mathbf{u}_{\mathcal{N}}^{(t)}$ is the aggregated gradients in iteration t, $\mathbf{D}_{\mathbf{w}}$ is the number of dimension of parameters for model $\mathbf{w}$ and the function $\text{mask}(\mathbf{u}_{\mathcal{N}}^{(t)}, r_m^{(t)})$ is to retain the largest $\max(r_m^{(t)}, 0)$ number of values in terms of magnitude and assign zero to all other values in the aggregated gradient $\mathbf{u}_{\mathcal{N}}^{(t)}$. The sparsified gradient $\mathbf{v}_m^{(t)}$ is distributed to the client m as the reward. Therefore, a high contributing client will have a *less* sparsified gradient update and thus better model performance. The β is to control the degree of altruism. When $\beta \to \infty$ the framework returns to vanilla FL in terms of the clients receiving the same unmasked/unspasified gradient (i.e., $\mathbf{u}_{\mathcal{N}}^{(t)}$) .

GoG uses an additional validation dataset D_v to compute the contribution estimates for each client with Shapley value. As an interpretation, the more a client's model update improves to the model performance on the validation dataset, the higher contribution estimate is for this client [18, 22]. For the reward mechanism, GoG gives different clients different chances to be selected and synchronize their local model with the most up-to-date global model thus achieving different model performances for each client. To elaborate, in iteration t the model will only select $k < n$ clients to be updated and the probabilities of the clients being selected are commensurate with their contribution estimates up to iteration t: $c_m^{(t)}$. Thus, a lower contributing client will get a low probability of being selected and its local model will stall for a longer period which will result in relatively lower model performance, and vice versa.

Both FGFL and GoG adopt Shapley value to compute the contribution estimates and use the model rewards (i.e., non-monetary reward). Therefore, both algorithms achieve the Shapley collaborative fairness and also the Pearson collaborative fairness. GoG requires a jointly agreed validation dataset to compute

the contribution estimates which can be unavailable in some cases. Additional performance evaluations on the validation dataset for models trained on different coalitions also bring higher computational complexity in GoG. In contrast, FGFL removes the need for the validation dataset and uses the gradient similarity as a proxy to compute the utility of coalitions which reduces the computational complexity. From Tab. 1.3, GoG always has $O(rM|D_v|)$ times higher running time complexity than FGFL, but lower communication costs than FGFL due to the allowance of partial clientz selection. However, both FGFL and GoG lack convergence guarantees for the models [18, 29].

1.3.3 Algorithms to achieve algorithmic fairness

For algorithmic fairness, we will focus on discussing the FAFL [2] algorithm. FAFL can be used to achieve both global algorithmic fairness and multi-client algorithmic fairness separately with minor modifications.

Intuitively, we can formulate optimization problems to find models that achieve the global algorithmic fairness as $\min_{\mathbf{w}\in\mathcal{W}} g(\mathbf{w})$ s.t. $h(\mathbf{w}) \leq \epsilon$ where $h(\mathbf{w}) = \Delta\text{DP}_g(\mathbf{w})$ (defined in Definition 7). Similarly, for the multi-client fairness, the optimization problem is defined by changing the constraint to $h_k(\mathbf{w}) \leq \epsilon_k, \forall k \in \{1, \ldots, n\}$ where $h_k(\mathbf{w}) = \Delta\text{DP}_k(\mathbf{w})$ (defined in Definition 8). Both optimization problems minimize the average loss under the constraint that the disparity measures do not exceed their respective budgets. However, the disparity measures are not differentiable w.r.t. the model parameters $\mathbf{w}$. To address this, FAFL [2] proposes an alternative constraint that is differentiable w.r.t. $\mathbf{w}$. Intuitively, if the distance of data points to the decision boundary is similar across different groups, the model performance on these groups can be similar. Following this intuition, the constraint can be replaced by: $h'(\mathbf{w}) = 1/n \sum_{m=1}^{n} \sum_{i=1}^{|D_m|} (s_m^{(i)} - \bar{s}) d(\mathbf{w}, x_m^{(i)}) \leq \epsilon'$ where the $s_m^{(i)}$ is the value of the attribute for i-th data point in client m. Here, $d(\mathbf{w}, x_m^{(i)})$ is the distance of data point $x_m^{(i)}$ to the decision boundary defined by model parameter $\mathbf{w}$. It is tractable in the case of linear model (e.g., logistic regression model). $\bar{s}$ is the average attribute value defined as $\bar{s} = \sum_{m=1}^{n} \sum_{i=1}^{|D_m|} s_m^{(i)} / \sum_{m=1}^{n} |D_m|$. Consequently, FAFL uses the following objective:

$$J(\mathbf{w}) \quad = \quad \frac{1}{n} \sum_{m=1}^{n} g_m(\mathbf{w}) \; + \; \lambda \left(\frac{1}{n} \sum_{m=1}^{n} \sum_{i=1}^{|D_m|} \left(s_m^{(i)} - \bar{s} \right) d\left(\mathbf{w}, x_m^{(i)} \right) \right)^2 \tag{1.6}$$

where λ balances the importance of fairness constraint and utility (i.e., model performance). By simply replacing the objective function in Eq. (1.6) with the multi-client case, the FAFL can achieve multi-client algorithmic fairness. From Tab. 1.3, FAFL has no extra communication costs and no extra running time for the fairness mechanism compared to FedAvg. However, it is only applicable to linear models due to the computation of distances between data points and the

decision boundary.

There are other algorithms to achieve global algorithmic fairness. FairFed [4] proposes to reweight the objective of FL dynamically during the training to achieve global algorithmic fairness. FairFL [34] proposes to apply multi-agent reinforcement learning to achieve global algorithmic fairness. We leave the reader to refer to [4, 34] for more details.

TABLE 1.3 Communication cost and running time for the fairness mechanism for different algorithms. n_g is the number of dimensions of model gradients, r is the fraction of clients selected in each iteration, M is the number of Monte Carlo simulations in GoG and $|D_v|$ is the number of data points in validation dataset for GoG.

Algorithm	Communication costs	Running time
FedAvg	$2n_g r\tau n$	–
STOCHASTIC-AFL [a]	$2n_g r\tau n + r\tau n^2$	$O((n_g \log(n_g) + n\log(n))\tau)$
q-FedSGD	$2n_g r\tau n + r\tau n$	$O(n_g \tau)$
FGFL	$2n_g \tau n$	$O(n_g n\tau)$
GoG	$2n_g r\tau n$	$O(rMnn_g \lvert D_v \rvert \tau)$
FAFL [a]	$2n_g r\tau n$	–

[a] We compute the communication costs for STOCHASTIC-AFL and FAFL by modifying the algorithms to select rn number of clients in each iteration for fair comparison.

1.4 OPEN PROBLEMS AND CONCLUSION

Apart from the notions and algorithms discussed here, there are still unsolved open problems in fairness FL. For collaborative fairness, contribution estimation remains a challenging problem (e.g., it is time-consuming to perform contribution estimation in FL, demonstrated in Table 1.3). Additionally, there is relatively little work on gauging the quality of contribution estimates (i.e., how well do these contribution estimates in FL reflect the clients' true contributions).

For the reward mechanism, developing algorithms with fair model rewards while guaranteeing the convergence of the model is worth exploring since most current works (e.g., GoG and FGFL) do not provide convergence guarantees which can be crucial to make the mechanism applicable in real applications. For algorithmic fairness and equitable fairness, the discussed algorithms provide convergence guarantees without incurring significant increases in communication costs/running time compared to FedAvg. Therefore, these notions are relatively well studied when considered separately. However, for systems to satisfy multiple fairness notions simultaneously, some open challenges remain. For example, to increase the inclusiveness of the FL system, we might want to incentivize low-contribution clients to participate (e.g., clients with less data) with the equitable

fairness guarantee while at the same time incentivizing high-contribution clients to participate with the collaborative fairness guarantee. It is still unclear how to design algorithms to achieve both equitable and collaborative fairness.

In conclusion, creating a fair environment is an emerging and important research area, especially to the real-world applications of FL. In this chapter, we provide a summary of the existing notions of fairness in FL motivated by different application scenarios. We also provide a comparative analysis on various algorithms to achieve the respective fairness notions with respect to the assumptions, target applications, communication costs and running time complexity. We discuss some open problems in improving certain fairness algorithms and point out some remaining research gaps in application scenarios where multiple fairness notions may need to be satisfied altogether.

1.5 ACKNOWLEDGEMENT

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-018).

BIBLIOGRAPHY

[1] Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang. Addressing algorithmic disparity and performance inconsistency in federated learning. In *Proc. NeurIPS*, volume 34, 2021.

[2] Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *Proc. SDM*, pages 181–189. SIAM, 2021.

[3] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proc. ITCS*, pages 214–226, 2012.

[4] Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. *arXiv preprint arXiv:2110.00857*, 2021.

[5] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction, 2018.

[6] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Proc. NeurIPS*, 29, 2016.

[7] Gareth James and Trevor Hastie. Generalizations of the bias/variance decomposition for prediction error. *Dept. Statistics, Stanford Univ., Stanford, CA, Tech. Rep*, 1997.

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

[9] Ron Kohavi et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Proc. KDD*, volume 96, pages 202–207, 1996.

[10] Tian Lan, David Kao, Mung Chiang, and Ashutosh Sabharwal. An axiomatic theory of fairness in network resource allocation. In *Proc. IEEE INFOCOM*, pages 1–9, 2010.

[11] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated

learning through personalization. In *Proc. ICML*, pages 6357–6368. PMLR, 2021.

[12] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *Proc. ICLR*, 2019.

[13] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *Proc. ICLR*, 2020.

[14] Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. Collaborative fairness in federated learning. In *Federated Learning*, pages 189–204. Springer, 2020.

[15] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. AISTATS*, 2017.

[16] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[17] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *Proc. ICML*, volume 97, pages 4615–4625, 2019.

[18] Lokesh Nagalapatti and Ramasuri Narayanam. Game of gradients: Mitigating irrelevant clients in federated learning. In *Proc. AAAI*, volume 35, pages 9046–9054, 2021.

[19] Olga Ohrimenko, Shruti Tople, and Sebastian Tschiatschek. Collaborative machine learning markets with data-replication-robust payments. *arXiv preprint arXiv:1911.09052*, 2019.

[20] L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games (AM-28), Volume II*, pages 307–318. Princeton University Press, Princeton, 1953.

[21] Rachael Hwee Ling Sim, Yehong Zhang, Mun Choon Chan, and Bryan Kian Hsiang Low. Collaborative machine learning with incentive-aware model rewards. In *Proc. ICML*, pages 8927–8936, 2020.

[22] Tianshu Song, Yongxin Tong, and Shuyue Wei. Profit allocation for federated learning. In *Proc. IEEE Big Data*, pages 2577–2586, 2019.

[23] Sebastian Shenghong Tay, Xinyi Xu, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Incentivizing collaboration in machine learning via synthetic data rewards. In *Proc. AAAI*, 2022.

[24] Kentaroh Toyoda and Allan N Zhang. Mechanism design for an incentive-aware blockchain-enabled federated learning platform. In *Proc. IEEE Big Data*, pages 395–403, 2019.

[25] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A principled approach to data valuation for federated learning. In *Federated Learning*, pages 153–167. Springer, 2020.

[26] Weiran Wang and Miguel A Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.

[27] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021.

[28] Xinyi Xu and Lingjuan Lyu. A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in federated learning. In *International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML 2021 (FL-ICML'21)*, 2021.

[29] Xinyi Xu, Lingjuan Lyu, Xingjun Ma, Chenglin Miao, Chuan Sheng Foo, and Bryan Kian Hsiang Low. Gradient driven rewards to guarantee fairness in collaborative machine learning. *Proc. NeurIPS*, 34, 2021.

[30] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*,

10(2):1–19, 2019.

[31] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu. FFD: A federated learning based method for credit card fraud detection. In *International conference on big data*, pages 18–32, 2019.

[32] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A fairness-aware incentive scheme for federated learning. In *Proc. AIES*, pages 393–399, 2020.

[33] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proc. WWW*, pages 1171–1180, 2017.

[34] Daniel Yue Zhang, Ziyi Kou, and Dong Wang. Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In *Proc. IEEE Big Data*, pages 1051–1060, 2020.