Are You Left Out? An Efficient and Fair Federated Learning for Personalized Profiles on Wearable Devices of Inferior Networking Conditions

PENGYUAN ZHOU*, University of Science and Technology of China, China
HENGWEI XU, University of Science and Technology of China, China
LIK HANG LEE, KAIST, South Korea
PEI FANG, Tongji University, China
PAN HUI, Hong Kong University of Science and Technology, Hong Kong

Wearable computers engage in percutaneous interactions with human users and revolutionize the way of learning human activities. Due to rising privacy concerns, federated learning has been recently proposed to train wearable data with privacy preservation collaboratively. However, under the state-of-the-art (SOTA) schemes, user profiles on wearable devices of inferior networking conditions are regarded as 'left out'. Such schemes suffer from three fundamental limitations: (1) the widely adopted network-capacity-based client selection leads to biased training; (2) the aggregation has low communication efficiency; (3) users lack convenient channels for providing feedback on wearable devices.

Therefore, this paper proposes a Fair and Communication-efficient Federated Learning scheme, namely FCFL. FCFL is a full-stack learning system specifically designed for wearable computers, improving the SOTA performance in terms of communication efficiency, fairness, personalization, and user experience. To this end, we design a technique named ThrowRightAway (TRA) to loose the network capacity constraints. Clients with poor networks are allowed to be selected as participators to improve the representation and guarantee the model's fairness. Remarkably, we propose Movement Aware Federated Learning (MAFL) to aggregate only the model updates with top contributions to the global model for the sake of communication efficiency. Accordingly, we implemented an FCFL-supported prototype as a sports application on smartwatches. Our comprehensive evaluation demonstrated that FCFL is a communication efficient scheme significantly reducing uploaded data by up to 29.77%, with a prominent feature of guaranteeing enhanced fairness up to 65.07%. Also, FCFL achieves robust personalization performance (i.e., 20% improvements of global model accuracy) in the face of packet loss below a certain fraction (10%–30%). A follow-up user survey shows that our FCFL-supported prototypical system on wearable devices significantly reduces users' workload.

ACM Reference Format:
Pengyuan Zhou, Hengwei Xu, Lik Hang Lee, Pei Fang, and Pan Hui. 2022. Are You Left Out? An Efficient and Fair Federated Learning for Personalized Profiles on Wearable Devices of Inferior Networking Conditions. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 0, 0, Article 0 (2022), 26 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

With the popularization of mobile and wearable devices, intelligent activity learning applications have been prominently used by consumers and generate more user data. Despite the potential to act as effective data sources for machine learning tasks, the training of machine learning models for mobile and wearable applications usually demands data far more than each device collects. Currently, aggregating user data in the cloud for extensive data analysis is the *de facto* solution. However, privacy concerns have spawned a series of policies that limit data collection and storage only to consumer-consented and necessary usage [32]. For example, most data collected from mobiles and wearables are subject to data protection regulations such as European Commission's General

*Corresponding author.

Authors' addresses: Pengyuan Zhou, pyzhou@ustc.edu.cn, University of Science and Technology of China, China; Hengwei Xu, xuhw@mail.ustc.edu.cn, University of Science and Technology of China, China; Lik Hang Lee, likhang.lee@kaist.ac.kr, KAIST, South Korea; Pei Fang, greilfang@gmail.com, Tongji University, China; Pan Hui, panhui@ust.hk, Hong Kong University of Science and Technology, Hong Kong.

Data Protection Regulation (GDPR) [12] and Consumer Privacy Act (CCPA) in USA [9]. Such regulations make it harder to aggregate user data for the sake of large-scale data analysis.

In the face of the above privacy-preserving challenges, federated learning rises as a new distributed paradigm where multiple clients collaboratively train a model without revealing private data while naturally complying with the GDPR. Based on whether the clients are different organizations or a large number of mobile devices, federated learning is divided into cross-silo and cross-device. Mobile and wearable devices fit the cross-device federated learning structure and encounter several unresolved issues.

First, communication is seen as a major bottleneck as cross-device federated learning systems rely on unstable wireless communication networks, which is even more severe for wearables due to lower communication bandwidth and device capacity than most other mobile devices. As a result, most related approaches propose to select clients based on network capacities [6, 38, 45], leading to a significant portion of user devices (24%) being '*left out*' or, equivalently, never-represented (detail explanation in Section 3). However, such proposals inevitably cause data shifts during client selection. Until very recently, researchers have proposed fairness schemes [31, 37] focusing on the data shift after client selection and during model updates aggregation. Unfortunately, *the data shift occurring at the beginning of client selection has been overlooked*. Consequently, the fairness and personalization performance of federated learning is impacted.

Second, the selected clients do not necessarily provide considerable contributions to the global model convergence. For instance, some clients may have very limited weight changes (e.g., 0) and thus waste the uploading quota for aggregation. There are a few related approaches. For example, as proposed by [41], the contribution of each local update is relevant with its movement[1], which can be used as a reference to select valuable updates. Based on movement, we define a new term *update relevance* and a lightweight algorithm to improve the communication and aggregation efficiency.

Third, a few existing federated learning solutions for wearables [8, 10] have largely overlooked the user experience perspective. For instance, how to reduce the demand for user operations and allow users to give feedback on wrong inference results conveniently. In the end, user experience is the most straightforward factor for the successful promotion of such techniques, and, eventually, demonstrates a crucial role in the system design of wearable computing.

In this paper, we propose **F**air and **C**ommunication-efficient **F**ederated **L**earning (FCFL) to collaboratively train models over wearable devices. Concretely, we make the following contributions in this work:

(1) *Re-examine 'never-represented' devices.* We conducted a trace-driven analysis and learned that the network limit challenge might be overstated in some aspects. Meanwhile, we identify an overlooked bias caused by network-capacity-based client selection. We further analyze its impact on the performances of the state-of-the-art (SOTA) algorithms in the fields of accuracy, fairness, and personalization (Section 3).

(2) *Communication efficiency (uploaded data) and fairness.* We explore the fair and communication-efficient federated learning (FCFL) by using ThrowRightAway (TRA). TRA ignores and replaces some lost data with light-weight recovery to avoid straggling retransmissions. Meanwhile, TRA lifts the network capacity threshold, thus enabling fully fair client selection regardless of networking conditions (Section 4.2). As a result, the '*never-presented*' clients and their contributions are sufficiently addressed by FCFL. We further propose Movement Aware Federated Learning (MAFL), an algorithm in FCFL, to spot the most important updates out of the participators, thus further improving the communication and aggregation efficiency (Section 4.3).

(3) *Performance.* The empirical evaluation results show that, compared with the SOTA work, i.e., Oort [29] and CMFL [43], FCFL improves the communication efficiency (i.e., the uploaded data) by up to 29.77% and 27.93% in lossy networks, respectively. Meanwhile, FCFL outperforms Oort in fairness by up to 65.07%.

[1]Movement refers to how fast a weight is moving away from 0.

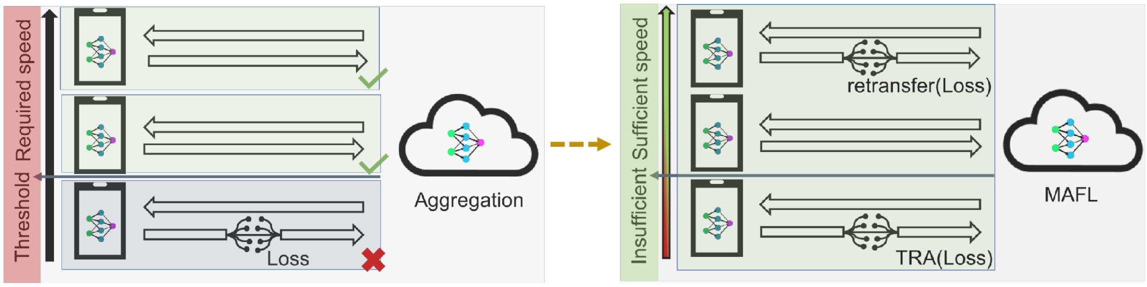


Fig. 1. Network-capacity based schemes (left) select clients with better network conditions to avoid packet loss and stragglers during aggregation. However, biased training is caused due to certain clients with poor networking conditions are '*never-represented*' (Details available in Section 3). Our proposal (right) allows clients participate in the aggregation regardless of network conditions.

FCFL improves the fairness and personalization performance by up to 45.07% and 20%, compared with q-FedAvg [31] and pFedMe [14], respectively (Section 5). We also design and implement a prototypical sports-monitoring system following the architecture shown in Figure 5, consisting of smartwatches, smartphones, and Linux server(s). The activity recognition model on the smartwatch trains with the prototypical system, resulting in > 97% accuracy. Our user evaluation shows that users with the FCFL-supported prototype significantly reflect reduced physical workload and efforts and become less frustrated (Appendix A).

2 BACKGROUND AND MOTIVATION

In this section, we describe the background and aforementioned drawbacks in current solutions, and state the motivations of our work.

2.1 Fair Client Selection

As noted by Bonawitz et al. [6], the FedAvg [35] model aggregation protocol's assumption about equitable participation of all devices is not the case in practice. Consequently, fairness [4, 18, 34] is impacted and results in bias. For instance, for better communication efficiency, cross-device federated learning systems commonly use transmission speed as a criterion for mobile client selection to avoid packet error and client drop (Figure 1 left). In such cases, the clients with more packet errors and drops are unlikely taken into model aggregation. Even worse, *users consistently having worsened networking conditions may never be represented in the model aggregation (being 'left out'), resulting in a biased model.* We do note that stochastic delay and network congestion during peak hours could generate temporary bad-network users following non-biased distributions. However, users paying for worse network service due to financial constraints also play an important role in different network conditions and result in biased client selection.

Mimicking common issues in model training, i.e. over-fitting and under-fitting, we summarize common factors for bias in federated learning as: (1) *over-represented*, (2) *under-represented*, (3) ***never-represented***. They refer to the clients that are (1) selected too frequently, (2) selected too infrequently, (3) never/barely selected. Although recent approaches [31, 37] partly solve (1) and (2) by mitigating bias during the training procedure, they can not solve bias caused by unfair client selection in (3), as also noted by the authors of in [37]. As a **result**, users whose patterns share less similarity with the good-networking users (who get selected the most in network-capacity-based client selection) experience lower model accuracy due to biased learning. Consequently, their personalization performance also suffers.

2.2 Aggregation Efficiency

Capacity-driven client selection, be it network-capacity, computation-capacity, or any others, does not consider the contribution of each client's updates to the global model convergence. For instance, some clients selected more than the others have similar models with the global model, and thus their updates provide only limited contributions. Although the selected participators can fulfill the configuration requirements, e.g., local training delay and update uploading delay, it is hard to guarantee that their updates make a meaningful contribution to the global model convergence. When meaningless updates consume the aggregation quota, it is unavoidably that the communication efficiency gets negatively impacted, and user devices consume more-than-necessary networking resource and energy for the training. Thus, we have to search for an efficient scheme of aggregation and model updating that represents all the clients.

2.3 User-centered Systems and Inspirations

Until recently, most activity monitoring apps on commercial wearable platforms require users to select the activity type before starting manually. A few exceptions, such as Apple Watch and Samsung Galaxy Watch, provide automatic workout detection functions but require a few minutes for the warm-up stage of automatic detection [3, 40].

More importantly, none of the existing wearable learning solutions (including both federated learning and traditional cloud computing) provide a real-time user **feedback** mechanism to correct the wrong detection result for better learning performance. Consequently, each client's model has its performance left to the mercy of the global training with limited personalization potential. We pinpoint the below issues that hindered the owner of wearable devices from personalized user experiences and further describe the latest solutions for such issues.

Lossy aggregation. A variety of techniques attempt to ease the gap between demanded and actual network capacities by intentionally "sacrificing" some information and hence achieving low latency and communication efficiency. For instance, some related works have proposed to use lossy compression to reduce the transferred data volume. The authors in [15, 25] perform lossy compression on the model updates using both structured and sketched updates. The main idea is to learn from a restricted space or upload a compressed model. Authors in [7] focus on the server-to-client communication and similarly applies a lossy compression scheme with less frequent updates. The authors in [44] tapped into the loss tolerance potential in distributed machine learning, which shows its bounded loss tolerance via evaluations.

Movement relevance. Recently, researchers have proposed to use "movement" to assess the importance of a weight update for model fine-tuning [41]. The authors in [43] propose to use the same-sign parameters in the update to select the local updates which have the most significant effects on the model aggregation. We think the two schemes, with proper adaption, can be integrated as a movement-based algorithm to select important updates during aggregation in federated learning.

User feedback. After reviewing a number of commercial wearable activity monitoring apps, we discover that they commonly lack of a crucial feature, i.e., real-time user feedback on activity recognition results. Due to different body shapes and movement habits, activity recognition may never become perfectly tailored for every individual, regarded as the grand challenge of achieving highly personalized services (i.e., hyper-personalization). Even after a long training period over a huge amount of data, such apps sometimes generate incorrect recognition results. Therefore, user feedback mechanisms for model-tuning are crucial for improving user experience. The most current apps can provide is to allow users to select the correct activities afterwards manually. Although the current apps, generally speaking, offer users to adjust the (mis-)recognized activity afterward, many users may forget to make corrections or skip such manual corrections due to burdensome tap-and-swipe operations on client UIs.

These works inspire us to explore the communication efficiency and loss tolerance of cross-device federated learning. The differences between our work and the works mentioned above are **fourfold**:

(1) We propose a loss-tolerant scheme (TRA) to address communication efficiency and guarantee fairness during client selection.
(2) We propose a new definition of "update relevance" and a lightweight algorithm (MAFL) to select the most important local updates. As such, FCFL can further improve communication and aggregation efficiency.
(3) As a standalone solution, FCFL improves communication efficiency and loss tolerance. Meanwhile, FCFL also guarantees fairness and personalization. Remarkably, FCFL can be easily integrated with SOTA algorithms for performance improvements.
(4) FCFL enables users to conveniently operate with smart wearables and provide feedback for improved user experience.

Note: The network threshold for selection can be bandwidth, transmission speed, packet loss, or hybrids. In this work, we convey different network constraints to packet loss.

3 PROBLEM STUDY

In this section, we analyze the problems mentioned in Section 2.2 in detail. First, we learn the disparate networking conditions by analyzing a real-world dataset and discover its biased impact on client selection. Then we show how the SOTA approaches regarding fairness and personalization for federated learning suffer from the data shift due to the biased selection.

3.1 Users Being 'Left Out' ('*never-represented*') due to Mobile Network Conditions

Transmission speed is an important metric during client selection and has been adopted by both industrial and academic works [38, 39]. For instance, Openmined [39] sets 2 Mbps as the default upload speed threshold for client selection. Therefore, it is worth looking at user network capacity in real life. We use a mobile broadband dataset provided by FCC [11] to study the mobile network conditions in reality. We select data from the "Download speed and upload speed" category in the 2019 Q1 & Q2 collection. The data is measured via Android and iOS applications and contains uploading traces from thousands of volunteered participants, recording the average received packets, lost packets, and throughput. After processing the trace according to unique identifiers, the cumulative distributions of the average packet loss ratio and upload speed are shown in Figure 2. It shows that the majority of the users have sufficient network capacities required by common federated learning systems (> 2 Mbps). However, the upload speeds vary tremendously across users. For instance, 24% of the users have upload speed < 2 Mbps while 51% of the users have upload speed > 8 Mbps. According to current common standard (e.g., > 2 Mbps according to [39]), 24% of the users fail to meet the network threshold thus would be *never-represented* in the model aggregation. Consequently, users who are *never-represented* and share fewer data similarities with the mainstream would experience lower model accuracies. They would also encounter worsened personalization performance since the aggregated model needs more fine-tuning to learn their datasets.

Takeaway: The trace-driven analysis shows that the network conditions of most mobile clients are not so "limited" and "challenging" as most related works assumed. However, the tremendously varied upload speeds may indeed cause biased client selection in network-capacity-based settings.

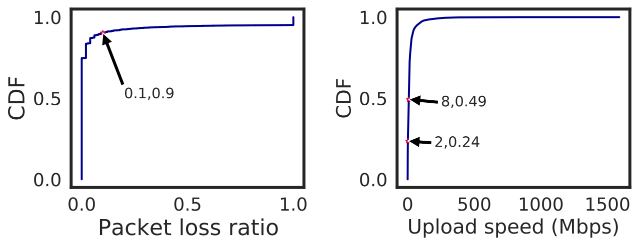


Fig. 2. Network conditions analysis result. 10% of the users experienced > 10% packet loss ratio. 24% of the users experienced < 2 Mbps upload speed, regarded as '*never-represented*'.

3.2 Impacts

Following the takeaway in Section 3.1, we investigate the impact of biased selection caused by network-capacity-based settings. We define the essential terms as follows.

Definition 1 (*Eligible client*). An eligible client is one that meets the required network threshold to participate in federated learning aggregation.

Definition 2 (*Eligible ratio*). Eligible ratio is the proportion of the eligible clients out of all the clients.

Only the eligible clients within the eligible ratio may be selected for aggregation in network-capacity-based settings. As some users have lower network capacities than the threshold (Figure 2), the system only can choose eligible clients for aggregation and generate bias and result in models with discrimination. For the completeness of the work, we adjust the eligible ratios between 100%, 90%, 80%, and 70% in the evaluation of the paper. More specifically, we investigate the impacts on accuracy, fairness, and personalization, respectively. We use the same datasets[2] for both bottleneck analysis and evaluation for consistency.

Accuracy. First, we examine the impact of biased selection on accuracy. We target at the prevailing and common FedAvg, which evenly averages the selected clients' models. As Figure 3 shows, smaller eligible ratios have higher impacts on the model performance. The final model accuracy of FedAvg with eligible ratios of 100%, 90%, 80%, and 70%, are 83.52%, 75.60%, 64.10%, and 62.60%. For the users in Figure 2, the model accuracy would **decrease** around 10% if using 2 Mbps as the selection threshold.

Fairness. As noted in Section 2.1, existing schemes improve fairness for *over-represented* and *under-represented* clients, but fail to serve the *never-represented* clients. To validate this argument, we reproduce the evaluations of q-FedAvg with a 70% eligible ratio to get the bottleneck performance. We adjust the distribution of training sample data on each device from i.i.d (independent and identically distributed [21]) to non-i.i.d to comprehensively test the degradation of both accuracy and fairness performance caused by biased client selection. Table 1 shows that the performances of q-FedAvg are impacted due to biased selection with both i.i.d and non-i.i.d data distributions. Non.i.i.d data presents larger performance degradation than i.i.d data in terms of both accuracy and fairness.

Personalization. Some of the existing approaches train a new deep neural network (transfer learning) [10], with loss function measuring the heterogeneity for local and global models, other than the one for the task. In

[2]In the rest of the paper, we use the synthetic **datasets** generated following the process described in the experiment detail of q-FedAvg [31], where α and β allow the precise manipulation of the degree of heterogeneity. Increasing the values of α and β result in higher statistical heterogeneity.

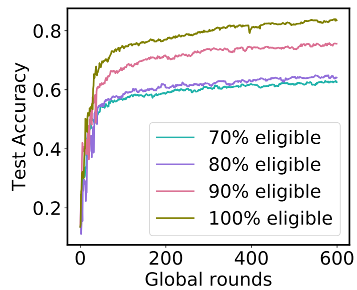


Fig. 3. Impact of biased client selection on the accuracy performance of the prevailing FedAvg with a Synthetic(0.5,0.5) dataset (Footnote 2).

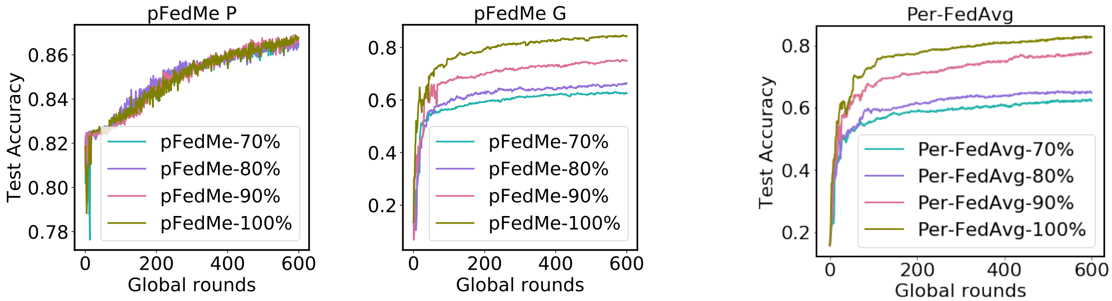
Table 1. Impact of biased client selection on the fairness performance of q-FedAvg [31]. Threshold (TH) indicates whether considering the 70% eligible ratio (see Definition 2) during client selection. The 4th column of Best/Worst 10% indicates the top 10% best/worst accuracies.

Dataset	TH	Average	Best/Worst 10%	Variance
Synthetic (i.i.d)		72.47%	91.85% / 43.19%	179
	✓	68.67%	94.25% / 36.30%	245
Synthetic (0.5,0.5)		66.21%	98.30% / 22.51%	536
	✓	52.81%	99.79% / 0	1350
Synthetic (1,1)		64.17%	100% / 7.67%	937
	✓	55.24%	100% / 0	1439
Synthetic (2,2)		75%	100% / 20.24%	651
	✓	62%	100% / 0	1584

resource-intensive cases, transfer learning reduces the model size so that a device can simultaneously hold two transferable models. Still, its advantage over a single larger model requires further exploration. Per-FedAvg [17] looks for an initial shared model that clients can quickly adapt via a few gradient descents concerning their data. pFedMe [14] adds constraints into the loss function of global training and shows the outperformance of Per-FedAvg. Therefore, we use pFedMe as the target to examine the impact of biased selection on personalization performance.

As shown in Figure 4, pFedMe offers resilient performance in its personalized model. However, the performance of the global model presents considerable degradation in lower eligible ratios. We note that pFedMe achieves robustness on personalized model performance via more computation and power cost. Unlike most approaches selecting clients before local training, pFedMe lets all clients do local training and then select some to upload. As such, its performance of personalized model is less depending on the convergence of the global model, while costing more computation and power of the client devices as a tradeoff. For example, applying an eligible ratio to Per-FedAvg gets degraded performance as shown in Figure 4b.

Takeaway: Network-capacity-based solutions cause biased client selection, which severely deteriorates the performance of accuracy, fairness, and personalization. Therefore, an alternative communication efficient scheme allowing fair participation is demanded.



(a) pFedMe

(b) Per-Fedavg

Fig. 4. The impact of biased client selection on personalized and global performance of pFedMe (a) and Per-Fedavg (b). Label **p** refers to the average local accuracy after personalization while **G** refers to the global accuracy. The dataset is Synthetic(0.5,0.5) (Footnote 2). We use the fine-tuned hyperparameters of Table. 1 in the paper of pFedMe [14].

4 FAIR AND COMMUNICATION-EFFICIENT FEDERATED LEARNING

In this section, we propose a system architecture and an alternative solution to network-capacity based client selection, named Fair and Communication-efficient Federated Learning (FCFL), to tackle the performance degradation caused by biased client selection and packet loss. FCFL is lightweight and can be easily integrated into different kinds of federated learning algorithms to augment their performances.

4.1 System Architecture

We design FCFL with a typical three-layer architecture as shown in Figure 5. Wearables function as data collectors and run inference during user activities. Periodically, wearables send collected data to paired smartphones that run local training and participate in federated learning. After the global model updates, the smartphones send back the new model to the paired wearables and thus complete a cycle. The key **differences** between FCFL and other federated learning wearable systems are:

(1) FCFL employs TRA to remove the network-capacity threshold during client selection, thus achieving fair training.
(2) FCFL employs MAFL to select the most important contributors from the participators, thus improving communication and aggregation efficiency.
(3) FCFL allows users to operate conveniently and feedback inference errors in real-time for better user experience.

The core of FCFL is ThrowRightAway (TRA) and Movement Aware Federated Learning (MAFL), as summarized in Algorithm 1. Next, we explain the details.

4.2 ThrowRightAway

The authors in [44] have recently demonstrated that contrary to common sense, data loss to an extent is not necessarily harmful in distributed learning systems. Through empirical evaluations, they discover that machine learning algorithms tolerate bounded data loss (10%–35% in their tests). Inspired by the work, we propose to explore the loss tolerance in cross-device federated learning systems. We propose TRA scheme to allow the server to accept any client as an eligible participant even if it has worse network capacities than the requirement and undesired packet loss ratio during updates uploading.

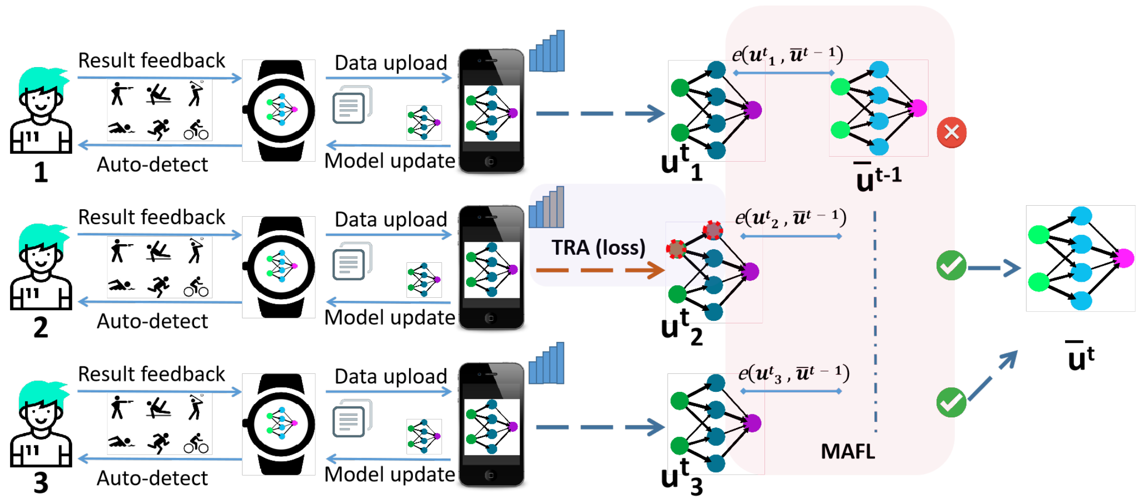


Fig. 5. The architecture of FCFL: user 2 is experiencing bad network signal while user 1 and user 3 have good network connections. Unlike common selection scheme which would drop user 2, TRA allows user 2 to join the federated learning by replacing the data loss with recalculation (Section 4.2). Then, FCFL selects the most important contributors using MAFL (Section 4.3). As seen, at time t, the local updates of user 2 and user 3 are chosen for model aggregation. Once converged, a new global model is sent back to all clients and their wearable devices.

At the beginning of the selection, each client compares its network condition with preset standards and sends a sufficiency investigation report to the server. The report contains only critical information, e.g., 0 or 1, indicating insufficient or sufficient, thus adding negligible network load[3]. After collecting the sufficiency reports of all willing-to-participate clients, the server classified the candidate clients into *sufficient* and *insufficient*. Then the server randomly selects some clients regardless of the belonging groups and sends the global model. The clients send back updates after local training. Upon detecting loss, the server sends retransmission notification if the client belongs to the *sufficient* group or conducts light-weight "recovery", as follows.

$$W^t_{agg} = \frac{1}{m+n}\left(\sum_{i=1}^{n} W^t_i + \sum_{j=1}^{m} \hat{W}^t_j\right) \tag{1}$$

$$\hat{W}^t_{jk} = \begin{cases} W(global)_k{}^{t-1} & if\ \hat{W}^t_{jk}\ loss \\ \hat{W}^t_{jk} & else \end{cases} \quad \forall \hat{W}^t_{jk} \in \hat{W}^t_j \tag{2}$$

W^t_i and $\hat{W}^t_j$ are respectively model weights in n users with sufficient and m users with insufficient network capacities at t round. r indicates the package drop rate. Hence each weight w in $\hat{W}$ has probability r to be dropped. If $\hat{W}^t_{jk}$ ($\forall \hat{W}^t_{jk} \in \hat{W}^t_j$) had been dropped, we replace $\hat{W}^t_{jk}$ with the corresponding parameter $W(global)^{t-1}_k$ from the previous round of the global model.

[3]For example, the report per client can be carried by one TCP packet. Even assuming the standard TCP MTU size as the upper bound of additional networking load, it adds only 0.0008% of the model update data volume in the tests of Section 5.2, which is negligible.

4.3 Movement Aware Federated Learning

As mentioned in the end of Section 2.1, some local updates may provide very limited contributions. Therefore, to further improve the communication and aggregation efficiency, we explore the relevance of local updates to the global model convergence. We propose MAFL to spot the local updates with top contributions to global model convergence. MAFL leverages the concept of "movement pruning" [41], i.e., selecting weights that are moving the most away from 0. The movement $\boldsymbol{mov}\left(\frac{\partial L}{\partial W_{i,j}}\right)$, i.e., the gradient of loss L with respect to weight $W_{i,j}$, is given by $\boldsymbol{mov}\left(\frac{\partial L}{\partial W_{i,j}}\right) = \frac{\partial L}{\partial W_{i,j}} W_{i,j}$. Referring to $\frac{\partial L}{\partial W_{i,j}}$ as $u_{i,j}$ (update), the movement of a client model update with respect to the model W at t is

$$\boldsymbol{mov}\left(\boldsymbol{u}^t\right) = \begin{pmatrix} \mathrm{mov}\left(u_{11}^t\right) & \cdots & \mathrm{mov}\left(u_{1n}^t\right) \\ \vdots & \ddots & \vdots \\ \mathrm{mov}\left(u_{n1}^t\right) & \cdots & \mathrm{mov}\left(u_{nn}^t\right) \end{pmatrix} = \begin{pmatrix} \left(u_{11}^t W_{11}^t\right) & \cdots & \left(u_{1n}^t W_{1n}^t\right) \\ \vdots & \ddots & \vdots \\ \left(u_{n1}^t W_{n1}^t\right) & \cdots & \left(u_{nn}^t W_{nn}^t\right) \end{pmatrix} \tag{3}$$

For simplicity, we only shows the movement of a single layer and assume it is a n-length square in Eq. (3).

Definition 3 (*Update relevance*). For a M-layer client model update $\mathbf{u}^t$ and the global model update $\overline{\mathbf{u}^{\mathbf{t}}}$, we informally say $\mathbf{u}^t$'s relevance to $\overline{\mathbf{u}^{\mathbf{t}}}$ positively correlates to their cosine similarity:

$$e(\mathbf{u}^t, \overline{\mathbf{u}^{\mathbf{t}}}) = \frac{1}{M} \sum_{m=1}^{M} \frac{\boldsymbol{mov}(\mathbf{u}_m^t) \bullet \boldsymbol{mov}(\overline{\mathbf{u}_m^t})}{\|\boldsymbol{mov}(\mathbf{u}_m^t)\| \|\boldsymbol{mov}(\overline{\mathbf{u}_{\mathbf{m}}{}^t})\|} \tag{4}$$

The goal of MAFL is to select the most irrelevant updates. The rationale is that the less similar a local update is with the collaborative convergence trend, the more changes it would make toward the new global model. Because MAFL runs before client selection, it requires the global model update $\overline{\mathbf{u}^{\mathbf{t}}}$ in advance. Therefore we use the last round global model update instead. Then the relevance of client c becomes

$$e(\mathbf{u}^t, \overline{\mathbf{u}^{\mathbf{t}}}) \approx \frac{1}{M} \sum_{m=1}^{M} \frac{\boldsymbol{mov}(\mathbf{u}_m^t) \bullet \boldsymbol{mov}(\overline{\mathbf{u}_m^{t-1}})}{\|\boldsymbol{mov}(\mathbf{u}_m^t)\| \|\boldsymbol{mov}(\overline{\mathbf{u}_{\mathbf{m}}{}^{t-1}})\|} \tag{5}$$

Each client calculates its update relevance with the last round global model and reports it to the parameter server during aggregation. The server selects the top-K contributors (i.e., bottom-K update relevant updates) to upload updates. The performance of MAFL is validated in Section 5.3.1.

The complexity of MAFL is determined by its major process, i.e., the calculation of update relevance. For a model update $\mathbf{u}$, the complexity of calculating relevance is $O(\mathbf{u})$, which is similar to the complexity of a forward propagation. Since each client calculates its own relevance, the complexity of this process of all clients equals to that of one client. Thus MAFL is a lightweight algorithm that adds only negligible delay. Please refer to Section 5.2 for the numerical results.

Takeaway: TRA and MAFL are two logical procedures of FCFL. In the concrete realization, they share some processes such as the local training, improving the learning performance from different perspectives.

- TRA guarantees communication efficiency by safely avoiding retransmissions while providing fully fair client selection.
- MAFL further improves the communication and aggregation efficiency by selecting the most important contributors.

Algorithm 1: Fair and Communication-efficient Federated Learning (FCFL)

Procedure *Server*:
    Input: Server weight w_0, users $\mathbf{C} = \langle c_1, c_2, ...c_D \rangle$, local update step E
    for $t = 1$ *to* T **do**
        Collect(sufficiencyReport)
        Categorize(sufficiencyGroup)
        Select a number of users $\mathbf{C}^t_{initial} = \langle C^t_1, ...C^t_n \rangle$
        $\mathbf{C}^t_{final} \leftarrow MAFL(\mathbf{C}^t_{initial}, u^{t-1}) = \langle C^t_1, ...C^t_m \rangle$
        $\mathbf{w}^{t+1} \leftarrow TRA(\mathrm{C}^t_{final})$
        Get global update $\mathbf{u}^{t+1} \leftarrow \mathrm{w}^{t+1} - \mathbf{w}^t$

Procedure *MAFL*:
    Input: $\mathbf{C}^t_{initial} = \langle C^t_1, ...C^t_n \rangle$, global Update $\mathbf{u}^{t-1}$
    for *each user* $c \in \mathbf{C}^t_{initial}$ **do**
        $\mathbf{u}^t_c \leftarrow LocalUpdate(E, \eta, \mathbf{w}^{t-1}_c)$// train with learning rate η for E steps
        Return relevance $e(\boldsymbol{mov}(\mathbf{u}^t_c), \boldsymbol{mov}(\mathbf{u}^{t-1}))$
    Get the Top-K contributors (i.e., bottom-K update relevant updates) $\mathbf{C}^t_{final}$ based on Definition 3
    Return $\mathbf{C}^t_{final}$

Procedure *TRA*:
    for *each user* $c \in \mathbf{C}^t_{final}$ **do**
        upload($\mathbf{u}^t_c$))
        if *loss* **then**
            if *sufficient* **then**
                retransmit(loss)
            else
                replace(loss) according to Eq. (1)
    Return $\mathbf{w}^{t+1}$

5 EVALUATION

In this section, we evaluate FCFL on the performance of communication efficiency, recovery efficiency, fairness, and personalization. Since there has not been a solution targeting all the metrics mentioned above, we compare the performances with different baselines separately. A recently published work, Oort [29], has proposed a client selection mechanism targeting similar metrics. Hence we include Oort as one of the baselines. Because Oort is implemented with its own framework, FedScale [28], and dataset setup, we constructed the comparison following its setup for objectivity. We found via tests that other baselines perform differently using FedScale's setup from their original papers. Therefore we construct the comparisons with other baselines following their initial setup.

5.1 Experimental Setting

First, we describe the details of the experiment setup. As mentioned, Oort has its own testbed and dataset setup, and therefore, we provide its experimental information separately.

Oort setting. We used the testbed FedScale [28] in Oort to compare its performance with FCFL. FedScale emulates heterogeneous device runtimes of different models, network throughput, and connectivity, using AI Benchmark [1] and Network Measurements [2] on mobiles. We picked two representative datasets in FedScale with different scales and tasks: (1) Image Classification: the small-scale FEMNIST dataset with 810k images across 3600 clients. (2) Speech Recognition: the large-scale Google Speech dataset with 105K speech commands over 2600 clients. We followed the original data distribution method provided by the authors to split the data across the clients. We trained ShuffleNet-V2 for image classification and ResNet-18 for speech recognition. For both datasets, we set both the minibatch size of each participant and the number of local steps to 20. In addition, the initial learning rates are 1e-3 and 0.05 for FEMNIST dataset and Google Speech dataset. We set the bandwidth threshold dynamically to control the packet loss ratio. When the client's bandwidth is less than the threshold, the client loses packets to a degree less than the threshold.

Other baselines. We used in FCFL and the baselines the same learning rate, batch size, and number of iterations. We only considered nonconvex settings with a two-layer deep neural network (DNN) using ReLU activation and a softmax layer for realistic concern. The synthetic dataset is split randomly with 90% and 10% for training and testing, respectively. All experiments were conducted using PyTorch version 1.7.1.

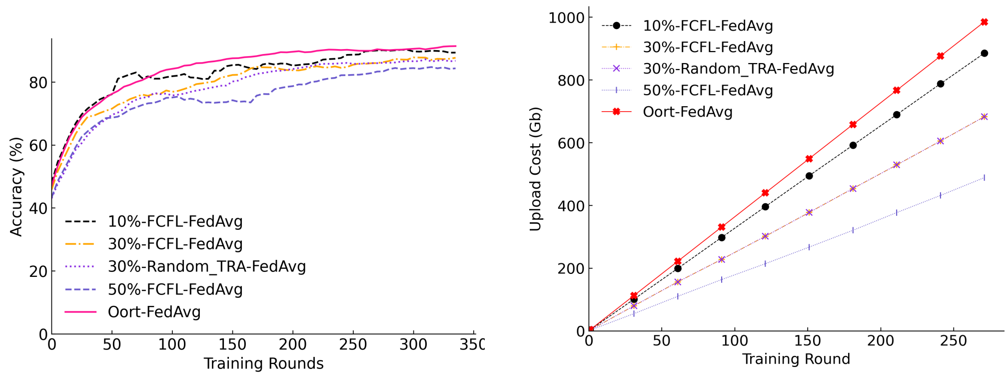
5.2 Comparison with Oort

Model performance and cost. As shown in Figure 6, Figure 7, and Table 2, When using FedAvg for model aggregation, FCFL outperforms Oort in fairness by up to 65.07% and 60.00%, in networking cost up to 29.77% and 27.06% with only minor accuracy differences at packet loss raio of 30% (-3.42% and -2.94%). We also test the "top-K" method used in MAFL compared with random selection, both of which used TRA in the face of packet loss, to further assess the performance of FCFL. Naturally, random selection performs better in fairness (Table 2). However its convergence is not as stable as using MAFL and the accuracy is a bit lower. Though, overall, it still considerably performs better than Oort in fairness and networking cost with little sacrifice of accuracy. In Table 2, less than 2 Mb refers to the ratio of the selected clients with less than 2 Mb (not the packet loss threshold), which is similar for 8 Mb. Cov represents the correlation coefficient between the selected times of each client and its bandwidth. Var represents the variance of the selected times of each client. As shown, the clients selected by Oort are strongly related to bandwidth, and the numbers of times the clients are selected are not balanced.

Table 2. Client selection variances of different algorithms on FEMNIST/GoogleSpeech datasets. The variance of rounds reports how fairness is enforced in terms of the number of participating rounds across clients. A smaller variance implies better fairness.

Loss ratio	Algorithm	<2Mb	>8Mb	Cov	Var (Rounds)
0%	*Oort+FedAvg*	0.029/0.097	0.573/0.505	0.209/0.151	6.076/28.774
30%	*Random_TRA+FedAvg*	0.088/0.043	0.480/0.523	0.097/0.203	1.317/11.440
10%	*FCFL+FedAvg*	0.081/0.033	0.496/0.543	0.132/0.199	2.337/14/253
30%	*FCFL+FedAvg*	0.094/0.046	0.489/0.527	0.116/0.223	2.212/11.509
50%	*FCFL+FedAvg*	0.100/0.051	0.463/0.504	0.079/0.177	1.682/9.120

Recovery efficiency of the proposal can be assessed by the amount of retransmitted data and model performance. As shown, FCFL avoided lots of retransmissions thus has much lower uploading cost. Yet the discarded lost packets had only minimum impact on FCFL's model performance, which proves that FCFL efficiently recovered the lost weights. We choose the Euclidean distance of recovered and lost weight matrices as a complementary measurement metric to quantify the recovery efficiency. Note that each existing distance metric has its pros and



(a) Round to accuracy

(b) Round to uploading cost.

Fig. 6. Training accuracy and upload cost with different packet loss ratios on FEMNIST dataset. Random indicates randomly selecting clients with TRA algorithm.

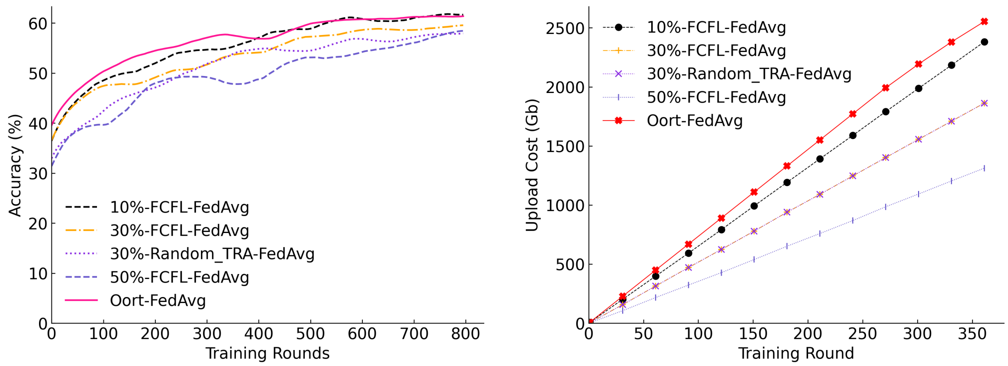
cons and there is not yet a standard one to accurately measure the difference between weight matrices, thus it only functions as an estimation. As shown in Figure 8, as the model converges, the average Euclidean distances between recovered and lost weight matrices became smaller, which is reasonable since the gradients became closer to zeros. We observe that the efficiency difference is much smaller compared with the packet loss ratio difference, thus proving the robustness of the recovery method in the face of different packet loss ratios to some extent.

Lightweight. As mentioned in the end of Section 4.3, MAFL of FCFL is a lightweight algorithm. To valid this argument, we measured the additional processing delay brought by MAFL on both datasets. On FEMNIST dataset, for ShuffleNet-V2 model, the average training time per epoch is 1.1758 second, while the processing delay of MAFL is 0.1223 second. On Google Speech dataset, for ResNet18 model, the average training time per epoch is 4.4653 seconds while the processing delay of MAFL is 0.0738 second. As such, the delay brought by MAFL is indeed negligible.

5.3 Other Baselines

5.3.1 Communication Efficiency. We select CMFL [43] and vanilla FedAvg as other baselines for communication efficiency (please refer to the beginning of Section 5). During aggregation, CMFL also uploads the clients' model based on the similarity of local and global models. The fundamental differences between FCFL and CMFL are **twofold**: (1) The definition of "relevance": FCFL selects the clients' weight to update based on the cosine similarity of the model[4] weight's movement while CMFL is based on the percentage of same-sign parameters in the updates. (2) Scope of comparison: FCFL compares relevance only among selected participators while CMFL compares among all clients. To select **top-K** contributors (**K** is automatically adjusted according to the movement similarity), we assign a pre-defined threshold, $TH = th/\sqrt{t}$ [43], to both FCFL and CMFL. That is, among the selected participators, only the model updates with a *relevance* lower than the threshold are required to upload for aggregation. We conduct two sets of evaluations, i.e., in the ideal network without packet loss and lossy networks.

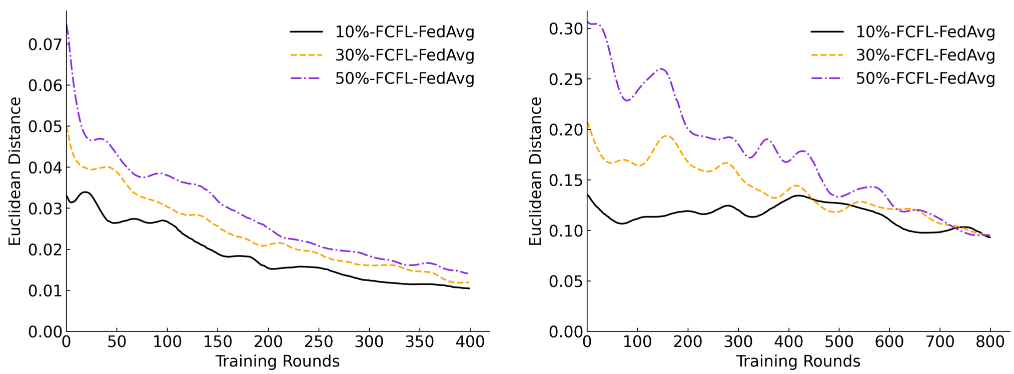
[4]In the evaluation, we used FedAvg as the basics of the aggregation algorithm. Thus each local update was essentially a local model. Therefore, we use *update* and *model* interchangeably in this context.



(a) Round to accuracy

(b) Round to uploading cost

Fig. 7. Training accuracy and upload cost with different packet loss ratios on Google Speech dataset. Random indicates randomly selecting clients with TRA algorithm.



(a) FEMNIST dataset

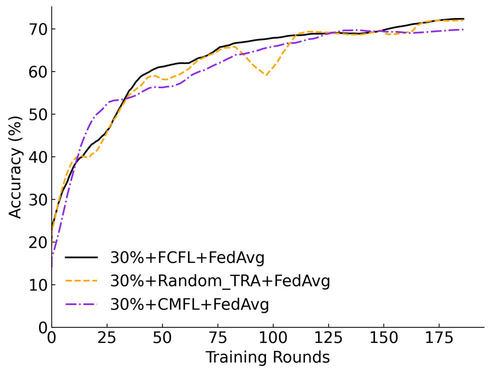
(b) Google Speech dataset

Fig. 8. Average Euclidean distance between the recovered and lost weight matrices during training on FEMNIST and Google Speech datasets.

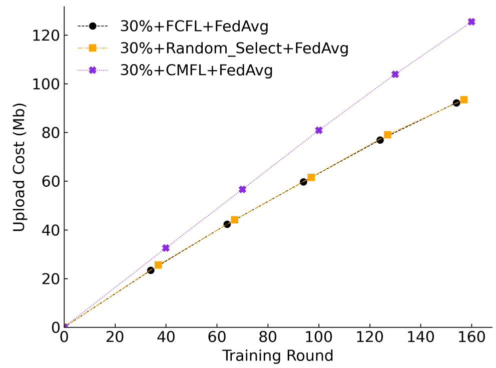
We use Synthetic(1,1) dataset as described in Section 3.2. The goal is to show the algorithms' communication efficiencies in different network conditions and their robustness in the face of packet loss.

Ideal network. First, we evaluate the algorithms in an ideal network condition without packet loss. As shown in Table 3, FCFL converges faster than the baselines to a similar accuracy. Meanwhile, FCFL decreases communication cost by 26.27% and 27.09% compared with CMFL and vanilla FedAvg. As seen, FCFL provides better communication efficiency than the baselines in all accuracy-achieving points in ideal network conditions.

Lossy network. Next, we evaluate the loss tolerances of the algorithms. Characterizing the client transmission delays with a lognormal distribution, we select three delay thresholds to practically function as the packet loss controllers. That is, when a delay larger than the threshold occurs, the algorithm processes or discards the loss with its own mechanism, e.g., TRA (FCFL) or retransmission (others). More specifically, we select 60, 115 and 280 as the thresholds, indicating 10%, 30% and 50% packet loss ratios. As shown in Figure 9, FCFL is more robust and



(a) Round to accuracy.



(b) Round to upload cost.

Fig. 9. Performance with 30% packet loss ratio on Synthetic dataset using different client selection methods.Random indicates randomly selecting clients with TRA algorithm.

Table 3. Communication cost on Synthetic dataset in ideal network condition without packet loss. x% *acc* and $y(z)$ mean achieving model accuracy of x% with y Mb uploading cost and z training rounds.

Algorithm	50% acc	60% acc	70% acc
FedAvg	12.726 (11)	31.845 (27)	90.395 (76)
CMFL	12.547 (11)	30.411 (26)	89.380 (76)
FCFL	9.201 (9)	26.348 (25)	65.899 (69)

converges to a higher accuracy in lossy network conditions than CMFL. Table 4 shows that FCFL decreases the communication cost compared with the baseline by more than 35.76% in all cases. The reasons are below listed.

(1) Cosine similarity of the movements (employed by FCFL's MAFL) characterizes the relevance of local updates with global update more accurately than the percentage of same-sign parameters (employed by CMFL)
(2) FCFL is more computation efficient than CMFL (require fewer comparisons).
(3) When a client's local model meets packet loss on some of its weight and replaced by *TRA*, movement-similarity-based *MAFL* better captures the noise led by such replacement thus uploading local models with similar local dataset distribution more frequently.

The performance comparison between "top-K" and random selection, both based on TRA, is similar with the result in Oort setup (Section 5.2). That is, random selection algorithm converges less stably to a lower accuracy than MAFL, while performing better than CMFL in accuracy and networking cost.

5.3.2 Fairness and Personalization. FCFL is highly integrable with relevant algorithms to improve fairness and personalization performances. For verification, we redo the evaluations conducted in Section 3.2. We compare the performance of the algorithms (FedAvg, q-FedAvg, pFedMe) limited by the network-capacity based selection with the integrated algorithms. For realistic concern, we only consider nonconvex settings. Similarly with Section 3.2, we consider three eligible ratios (Definition 2), i.e., 70%, 80%, and 90% which cause different degrees of bias in client selection in network-capacity based settings. For each eligible ratio, we consider a variety of packet loss ratios, i.e, 10%, 30%, and 50%, for the *insufficient* clients (defined in Section 4.2). Since data heterogeneity has important

Table 4. Communication cost on Synthetic dataset with different packet loss ratios, i.e., 10%, 30% and 50%. $x\%acc$ and $y(z)$ mean achieving model accuracy of $x\%$ with y Mb and z training rounds.

Loss ratio	Algorithm	60% acc	65% acc	70% acc
10%	*CMFL*	44.618 (42)	63.554 (60)	98.249 (94)
	FCFL	35.833 (40)	42.277 (48)	63.108 (74)
30%	*CMFL*	53.885 (67)	75.689 (94)	144.379 (188)
	FCFL	30.774 (45)	49.557 (75)	91.835 (153)
50%	*CMFL*	80.824 (140)	97.322 (171)	157.396 (288)
	FCFL	32.618 (69)	53.652 (122)	97.145 (235)

effects on fairness and personalization, we use both Synthetic(1,1) and Synthetic(2,2) datasets (Footnote 2) to get better understanding of the performances under the bias.

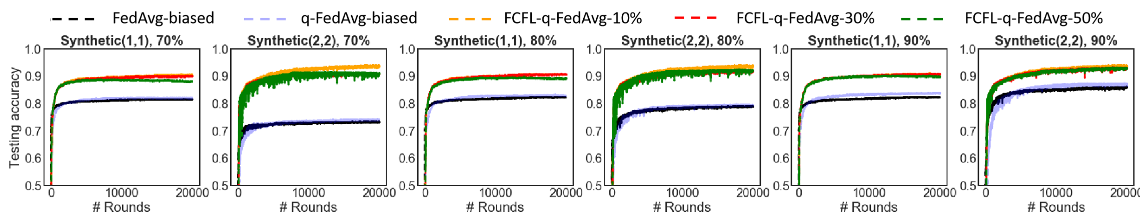


Fig. 10. Sample based accuracy performance of FedAvg and q-FedAvg using the biased network-capacity based selection, and FCFL-q-FedAvg on Synthetic(1,1) and Synthetic(2,2) datasets (Footnote 2) with 70%, 80%, and 90% eligible ratios (Definition 2). FCFL-a-FedAvg-X% indicates the packet loss ratios %(10%, 30%, 50%).

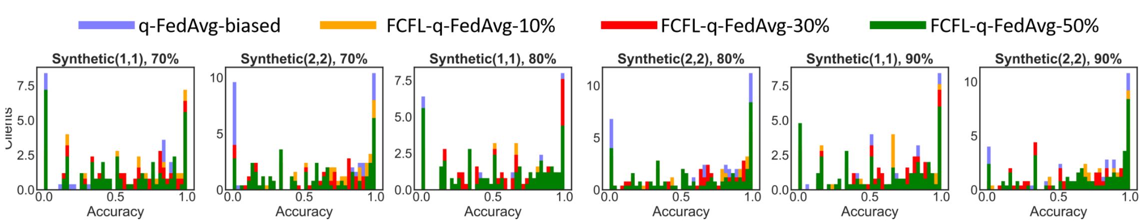


Fig. 11. Fairness performance distribution of q-FedAvg using the biased network-capacity based selection and FCFL-q-FedAvg on Synthetic(1,1) and Synthetic(2,2) datasets (Footnote 2) with 70%, 80%, and 90% eligible ratios (Definition 2). FCFL-a-FedAvg-X% indicates the packet loss ratios (10%, 30%, 50%).

Accuracy. The integration of FCFL and q-FedAvg presents the best accuracy performance in the face of packet loss. As shown in Figure 10, FCFL-q-FedAvg outperforms biased-FedAvg and biased-q-FedAvg in all scenarios. With slightly longer convergence periods, FCFL-q-FedAvg (10% loss ratio) improves the model accuracy on Synthetic(1,1) by 10.35%/6.69%, 8.44%/3.48%, and 9.31%/-0.79%, compared to biased-FedAvg and biased-q-FedAvg in 70%, 80%, and 90% eligible ratio scenarios, respectively. On Synthetic(2,2), the corresponding improvements are 9.88%/7.39%, 3.62%/1.62%, and 2.75%/-1.4%. In a word, when more than 10% clients have worse network than

Table 5. Client based fairness performance of q-FedAvg with biased network-capacity based client selection Vs FCFL-q-FedAvg, with 70%, 80%, and 90% eligible ratios (Definition 2). Best/Worst 10% indicate the top 10% best/worst accuracies. The gray color highlights the best performance algorithms.

70%	Synthetic(1,1)	Average	Best/Worst 10%	Variance	Synth(2,2)	Average	Best/Worst 10%	Variance
	q-FedAvg-biased	55.00%	100% / 0	1439		62.34%	100% / 0	1584
	FCFL-q-FedAvg-10%	61.63%	100% / 6.01%	1031		69.72%	100% / 9.81%	870
	FCFL-q-FedAvg-30%	59.44%	100% / 4.11%	1021		55.38%	99.69% / 0	1109
	FCFL-q-FedAvg-50%	50.99%	99.97% / 0	1220		55.00%	99.98% / 2.81%	1125
80%	**Synthetic(1,1)**	**Average**	**Best/Worst 10%**	**Variance**	**Synth(2,2)**	**Average**	**Best/Worst 10%**	**Variance**
	q-FedAvg-biased	58.90%	100.00%/0	1286		67.14%	100.00%/0	1379
	FCFL-q-FedAvg-10%	62.38%	100.00%/4.11%	1020		68.76%	100.00%/8.45%	916
	FCFL-q-FedAvg-30%	62.79%	100.00%/8.10%	926		61.59%	100.00%/1.36%	1073
	FCFL-q-FedAvg-50%	54.45%	99.83%/0	1194		60.80%	100.00%/0	1195
90%	**Synthetic(1,1)**	**Average**	**Best/Worst 10%**	**Variance**	**Synth(2,2)**	**Average**	**Best/Worst 10%**	**Variance**
	q-FedAvg-biased	64.04%	100.00%/5.39%	1009		70.60%	100.00%/3.43	918
	FCFL-q-FedAvg-10%	63.25%	100.00%/2.92%	1030		67.74%	99.64%/15.01%	759
	FCFL-q-FedAvg-30%	63.53%	100.00%/4.35%	985		65.07%	99.85%/11.78%	876
	FCFL-q-FedAvg-50%	57.42%	100.00%/0	1162		67.33%	100.00%/5.27%	1012

standard, FCFL-q-FedAvg would considerably improve aggregated model accuracy over FedAvg and q-FedAvg with network-capacity based selection. We reason the performance is because (1) FCFL allows a wider selection of participants thus increasing the learning space with the cost of some data integrity. (2) q-FedAvg employs the idea of α-fairness [36] to give higher relative weights to the clients with higher losses. As such, q-FedAvg compensates for the effect of the packet loss due to FCFL.

Fairness. We utilize FCFL-q-FedAvg to tackle the fairness degradation caused by biased client selection in Table 1. As shown in Figure 11, FCFL-q-FedAvg outperforms biased-q-FedAvg in most scenarios and the superiority increases as the data heterogeneity increases and the eligible ratio decreases. Table 5 summarizes the accuracy and variance results and highlights the best-performed algorithms in different scenarios. Note that the accuracies presented in Table 5 are on the granularity of per-client to depict inter-client fairness better. In contrast, the accuracies in Figure 10 are sample-based for higher granularity. As seen, FCFL improves the fairness performance in all cases and at the most by 45.07%.

Personalization. We integrate FCFL with pFedMe to tackle the personalization performance degradation caused by biased client selection as shown in Figure 4. As shown in Figure 12, FCFL-pFedMe demonstrates comparable mean accuracy to pFedMe in the local personalized model. Although FCFL-pFedMe is sightly less accurate than pFedMe by 1% in the local personalized model, FCFL-pFedMe outperforms pFedMe in the global model significantly by 20% at the most.

Takeaway:. FCFL increases the communication efficiency compared with the baselines by achieving similar accuracies with fewer uploading updates. FCFL also shows better loss tolerance that the model accuracy is more robust in the face of packet loss. Integrating FCFL with q-FedAvg enables learning from the entire sample space while mitigating the effect of packet loss by adaptively recalculation. As a result, it improves both accuracy and fairness performances. FCFL considerable improves the global performance of pFedMe compared to in network-capacity based settings with a relevantly negligible cost of local model accuracy.

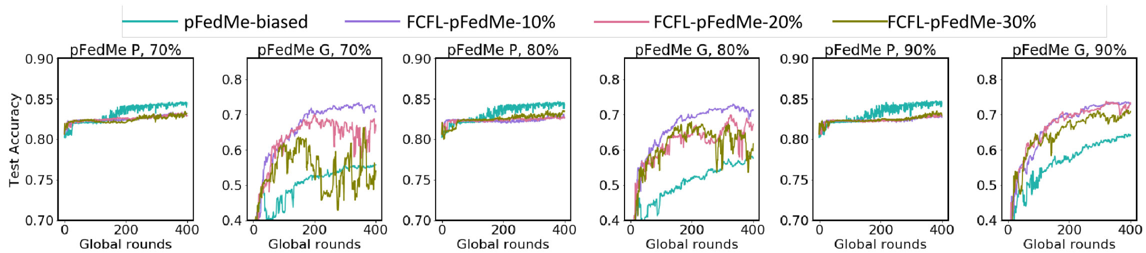


Fig. 12. Personalization performance of pFedMe using the biased network-capacity based selection and FCFL-pFedMe with 70%, 80%, and 90% eligible ratios (Definition 2). Label **p** refers to the average local accuracy after personalization while **G** refers to the global accuracy. FCFL-pFedMe-X% indicates the packet loss ratios (10%, 20%, 30%). We adapted the tested loss ratios according to the observed performance boundary.

6 RELATED WORK

6.1 Federated Learning Over Wearables

As mentioned in Section 1, federated learning can fundamentally improve privacy in the context of large-scale wearable data learning. In general, federated learning can be divided into cross-silo and cross-device federated learning systems based on different kinds of clients. The cross-silo federated learning system meets relatively fewer failures caused by clients because each client can be specifically accessed thanks to a clear and unique identity and is available for local model updates or parameter updates almost at any time [22]. In contrast, the cross-device federated learning system faces challenges from stateless and unreliable clients due to dynamic client participation and communication bottlenecks. Compared with other mobile devices, wearables have lower computation, storage, and networking capacity thus face more severe challenges.

All the problems above elicit the demand for a federated learning system fully utilizing the capacity and data of wearables while avoiding draining the batteries. In the context of other systems, DeepWear [46] lets a wearable device adaptively offload the partition of training task to a paired handheld device, based on the resource status of both devices. We leverage federated learning to improve the training efficiency using distributed users' datasets. Further, we notice another systematic flaw overlooked by existing works. Like all other machine learning systems, Federated learning systems should allow human oversight to monitor and adjust its performance for better QoE. Therefore, the cross-device federated learning should incorporate a component in the system design that allows user corrections or feedback to the model performance.

6.2 Fairness

Machine learning models can often exhibit unfair behaviors not on purpose. For example, we may categorize the model as "biased" when undesirable effects happen on some users who share similar characteristics with the others, or different outcomes occur for certain sensitive groups [4, 16]. The criterion of counterfactual fairness requires that a user receive the same treatment regardless of the belonging group [27].

Relatedly, cross-device federated learning does not have access to sensitive attributes for most cases. For instance, wearable activity monitoring applications require only sensor data and do not need the information of the age and gender of the users. As a result, device characteristics (e.g., computation capacity) and conditions (e.g., battery status) become the key factor of fairness instead of sensitive user attributes (e.g., gender, race, age). As mentioned in Section 2, we summarize common factors for bias in federated learning as: (1) *over-represented*, (2) *under-represented*, (3) *never-represented*.

(1) and (2) can be solved with some approaches targeting training procedure bias such as AFL [37] and q-FedAvg [31]. AFL minimizes the maximum loss incurred on the worst-performing devices as a classical minimax problem. q-FedAvg generalizes AFL by allowing for a flexible tradeoff between fairness and accuracy. These approaches focus on enforcing accuracy equity by mitigating the training procedure bias. However, they can not solve (3) caused by training data bias, as also noted by the authors of AFL. On the other hand, aggregation approaches with only model weights taken into account have also been proved unable to tackle this challenge [23, 33]. Therefore, a scheme tackling this challenge from the client selection phase is demanded.

6.3 Personalization

Due to the different user behaviors and heterogeneous devices, it is safe to assume wearables generate non-i.i.d datasets. Such a situation necessitates the personalized models customized by local data for different clients, as they may outperform the best possible global model. The tension between the fairness/uniformity and the average accuracy [31] further stresses the necessity of personalization while improving global model accuracy and fairness. Recent works have proposed varied personalization schemes for federated learning [26], e.g., featurization, transfer learning, multi-task learning, and meta-learning [10, 14, 17, 19, 20, 24, 42] etc.

To the best of our knowledge, all schemes mentioned above still (at least partly) rely on the convergence of the global model. More specifically, existing schemes use different methodologies to convey the information of personalized model into the global model as a reference, to balance the convergences of both models. When some users are *never-represented* in the aggregation, the global model does not incorporate the knowledge of such users, thus generating a biased model. As a result, the personalization performance on *never-represented* users is inevitably impacted. **On the other hand**, conveniently allowing user feedback on the wrong recognition is essential for personalized model-tuning. To the best of our knowledge, our work serves as the first effort to enable user feedback anytime during or after activities and record such feedback in the next-round training dataset.

7 DISCUSSION

Benefits of including data from under- and never-represented users. FCFL serves as a groundwork for fairness-aware distributed machine learning [5], which considers the participators under various constraints and hence achieves comprehensive representation of the users [13]. Such algorithmic fairness could prevent biased services or decision-making processes that disadvantaged the '*under-represented*' and '*never-represented*' users. Because the never-represented user group can be the one who demands the service the most, involving their data in training can potentially bring considerable benefits in numerous cases. For instance, many healthcare products require users to wear wearable devices periodically or even daily to effectively collect enough data for model training. The users who became never-represented due to various reasons, e.g., networking, computational resources, battery life, or any other constraints, may happen to be the ones who demand the most care, e.g., patients with the most severe illnesses. Using techniques like FCFL to allow their data to participate in federated learning is crucial to improving their experience of the services. Furthermore, the aging individuals may own limited budgets on their mobile service plans and have received unsatisfied network capability for distributed machine learning. It is worthwhile to mention that the aging population could serve as a valuable yet indispensable data source to improve the generalizability of such machine learning models especially designed for monitoring and tracking of health, sleeping patterns, and sports activity.

Potential Application. Besides sports monitoring applications such as our prototype, FCFL can be further applied to diversified applications. Relatedly, distributed machine learning is expected to be employed in the era of advanced network communication (e.g., 6G network), which is primarily designed to serve robotics, unmanned vehicles, surveillance camera, and IoT devices. For example, as one of the emerging unmanned vehicles appeared in the commercial market, automotive requires a comprehensive understanding of not only the surrounding items but also the dynamic behaviors of other vehicles so that the vehicle can predict the near future and be precautious of potential dangers. As each driver's safety gets improved as the training data gets more completed, every vehicle's data into the training matters. In the end, the vehicles that cause accidents have always been a minor group (e.g., careless drivers nowadays), and they can very well happen to be the never-represented users. In this case, FCFL can help to include more users' data with fewer networking constraints.

Limitation. We developed TRA as a lightweight algorithm to lower the computational complexity. It can efficiently mitigate the effect of packet loss ($< 30\%$) by adaptive recalculation. However, when a packet loss $> 30\%$ occurs, TRA is not sufficient to compensate for the lost data and impact the model training. We reason this is due to the simplicity of the recalculation (Eq. 1), which has a limited capability of loss recovery. On the other hand, although MAFL effectively selects the most critical updates from the participators, thus improving communication and aggregation efficiency, it is possible to bring bias. The reason is some local updates may make less contributions, but they do represent some users' data distributions. In such cases, a bias towards them can happen by excluding their updates. Although we have not noticed the impact of this point in the numerous evaluations, a comprehensive theoretical analysis could be helpful.

Future directions. Through empirical evaluations, we find that the lightweight FCFL works well in lots of scenarios. However, we also note that the FCFL performance is sensitive to the hyperparameters occasionally. Therefore, we plan to conduct a theoretical analysis of the algorithm and explore its potential with comprehensive optimization problem formulation. The next research milestone attempts to generalize the algorithms to make the system performance robust in the face of varied hyperparameters. Additionally, we will conduct follow-up studies to examine the effect of network-driven algorithmic bias on the user satisfaction of mobile services supported by distributed machine learning.

8 CONCLUSION

In this work, we investigate FCFL, a fair and communication-efficient federated learning system for wearables. The trace-driven analysis finds that the commonly assumed limit network challenge is overstated but can cause biased client selection. We show through evaluations that the induced bias has severe impacts on the performance of federated learning, i.e., model accuracy, fairness, and personalization. FCFL can avoid the bias by allowing all clients, regardless of networking constraints, to participate in the training with loss tolerance (up to 30%) and thus improve fairness during client selection. Further, FCFL selects the most critical updates from participators based on *update relevance* (Definition 3) and thus improves the communication efficiency during model aggregation. FCFL is easily integrable with SOTA federated learning algorithms. Through numerous tests, the FCFL-integrated algorithms present superior performances on the accuracy, fairness, and personalization in most scenarios. Last but not least, we implemented a full-stack prototype system and developed a sports app with a convenient user feedback mechanism for a better personalized model-tuning experience. We demonstrate the system's training performance over HAM datasets, and a follow-up user study shows the FCFL-supported prototype significantly reduces physical workload, user efforts, and frustration.

ACKNOWLEDGEMENT

The work was supported by the Academy of Finland 5GEAR project (Grant Number 319669), FIT project (Grant Number 325570), and National Key R&D Program (Grant Number 2021YFC3300500).

REFERENCES

[1] 2021. AI Benchmark: All About Deep Learning on Smartphones. http://ai-benchmark.com/ranking_deeplearning_detailed.html.
[2] 2021. MobiPerf. https://www.measurementlab.net/tests/mobiperf/.
[3] Apple. 2020. *Use the Workout app on your Apple Watch.* https://support.apple.com/en-us/HT204523.

[4] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2017. Fairness in machine learning. *NIPS Tutorial* 1 (2017).
[5] Sarah Bird, K. Kenthapadi, Emre Kıcıman, and Margaret Mitchell. 2019. Fairness-Aware Machine Learning: Practical Challenges and Lessons Learned. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (2019).
[6] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. *MLSys* (2019).
[7] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210* (2018).
[8] Yekta Said Can and Cem Ersoy. 2021. Privacy-preserving Federated Deep Learning for Wearable IoT-based Biomedical Monitoring. *ACM Transactions on Internet Technology (TOIT)* 21, 1 (2021), 1–17.
[9] CCPA. 2021. California Consumer Privacy Act. https://www.caprivacy.org/.
[10] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao. 2020. FedHealth: A Federated Transfer Learning Framework for Wearable Healthcare. *IEEE Intelligent Systems* (2020).
[11] Federal Communications Commission. 2020. *Measuring Broadband America Mobile Data*. https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-broadband-america-mobile-data.
[12] Bart Custers, Alan M Sears, Francien Dechesne, Ilina Georgieva, Tommaso Tani, and Simone Van der Hof. 2019. *EU Personal Data Protection in Policy and Practice*. Springer.
[13] Mark Diaz. 2019. Algorithmic Technologies and Underrepresented Populations. *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing* (2019).
[14] Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. 2020. Personalized federated learning with Moreau envelopes. *NeurIPS* (2020).
[15] Yuanrui Dong, Peng Zhao, Hanqiao Yu, Cong Zhao, and Shusen Yang. 2020. CDC: Classification Driven Compression for Bandwidth Efficient Edge-Cloud Collaborative Deep Learning. *IJCAI* (2020).
[16] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *ITCS*.
[17] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948* (2020).
[18] Boli Fang, Miao Jiang, Pei-yi Cheng, Jerry Shen, and Yi Fang. 2020. Achieving Outcome Fairness in Machine Learning Models for Social Decision Problems.. In *IJCAI*. 444–450.
[19] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.
[20] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* (2018).
[21] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. 2020. The non-iid data quagmire of decentralized machine learning. In *ICML*.
[22] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* (2019).
[23] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*.
[24] Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. 2019. Adaptive gradient-based meta-learning methods. In *NeurIPS*.
[25] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).
[26] Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. 2020. Survey of Personalization Techniques for Federated Learning. *arXiv preprint arXiv:2003.08673* (2020).
[27] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *NIPS* (2017).
[28] Fan Lai, Yinwei Dai, Xiangfeng Zhu, and Mosharaf Chowdhury. 2021. FedScale: Benchmarking Model and System Performance of Federated Learning. In *arXiv:2105.11367*.
[29] Fan Lai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. 2021. Efficient Federated Learning via Guided Participant Selection. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*.
[30] Lik-Hang Lee, Ngo-Yan Yeung, Tristan Braud, Tong Li, Xiang Su, and Pan Hui. 2020. Force9: Force-assisted Miniature Keyboard on Smart Wearables. In *ICMI '20: Proceedings of the 2020 International Conference on Multimodal Interaction*. ACM, International, 232–241. https://doi.org/10.1145/3382507.3418827 International Conference on Multimodal Interaction, ICMI ; Conference date: 25-10-2020 Through 29-10-2020.
[31] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2019. Fair Resource Allocation in Federated Learning. In *ICLR*.
[32] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. 2020. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials* (2020).
[33] Frank Po-Chen Lin, Christopher G Brinton, and Nicolo Michelusi. 2020. Federated Learning with Communication Delay in Edge Networks. *arXiv preprint arXiv:2008.09323* (2020).

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

[35] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR.

[36] Jeonghoon Mo and Jean Walrand. 2000. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on networking* (2000).

[37] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. 2019. Agnostic Federated Learning. In *International Conference on Machine Learning, ICML 2019*. 4615–4625.

[38] Takayuki Nishio and Ryo Yonetani. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC*. IEEE.

[39] Openmined. 2021. Openmined. https://www.openmined.org/.

[40] Samsung. 2020. *Use Automatic Workout Detection on your Samsung smart watch*. https://www.samsung.com/us/support/answer/ANS00083510/.

[41] Victor Sanh, Thomas Wolf, and Alexander M Rush. 2020. Movement Pruning: Adaptive Sparsity by Fine-Tuning. In *NeurIPS*.

[42] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated multi-task learning. *NIPS* (2017).

[43] Luping WANG, Wei WANG, and Bo LI. 2019. CMFL: Mitigating Communication Overhead for Federated Learning. In *ICDCS*.

[44] Jiacheng Xia, Gaoxiong Zeng, Junxue Zhang, Weiyan Wang, Wei Bai, Junchen Jiang, and Kai Chen. 2019. Rethinking transport layer design for distributed machine learning. In *APNet*.

[45] Jie Xu and Heqiang Wang. 2020. Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective. *IEEE Transactions on Wireless Communications* 20, 2 (2020), 1188–1200.

[46] Mengwei Xu, Feng Qian, Mengze Zhu, Feifan Huang, Saumay Pushp, and Xuanzhe Liu. 2019. Deepwear: Adaptive local offloading for on-wearable deep learning. *IEEE Transactions on Mobile Computing* 19, 2 (2019), 314–330.

Appendix A USER EVALUATION

We implemented an FCFL-supported sports application on a smartwatch that is highly characterized by the limited computational resource and the ease of user interaction [30]. The application features intelligent recognition of user activities during sports activity, with the following spotlights. First, the user with the application only needs to press the start button and immediately start the sports activities, while the application can automatically recognize the activity. Second, due to the automatic activity recognition, users do not need to manually set the activities once another activity type has been made, for instance, switching from running to cross-trainer activities. Third, when a user notices a wrong recognition, the user can conveniently click the result and select the correct one anytime during or after the activity. The result is immediately recorded in the database, and given a higher weight in next-round training to help model-tuning. With the aforementioned user-centered features, the applications were evaluated remotely by 17 participants.

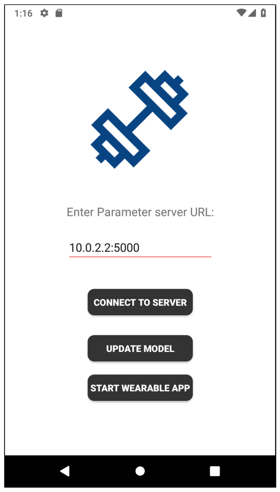
A System Prototype of FCFL – A sport application. We implemented a prototype of FCFL following the system architecture shown in Figure 5. Specifically, we built the parameter server using PyGrid[5] on Ubuntu 16.04; the clients as an Android smartphone app using KotlinSyft[6] on Android 9, the wearable as a sport monitoring app on WearOS 2.35. We deployed the server in a MSI GS65 Stealth 8SG laptop equipped with a 6-core I7-8750H CPU, 32GB of memory, and an Nvidia RTX 2080 Max-Q GPU; the client in a Huawei Mate9 Pro smartphone, equipped with a 2.4 (1.8) GHz octa-core HiSilicon Kirin 960 CPU and 4GB of memory; the wearable in a Suunto 7 smartwatch. Figure 13 shows the prototype including user interfaces of the client (a smartphone) and the wearable (a smartwatch). A **demonstration video**[7] shows the key functions of the FCFL-supported application. Remarkably, the FCFL-supported application offers a user feedback channel that allows users to report inaccurate activity recognition.

We first tested the algorithm and model with a public dataset[8]. The dataset was collected with 8 users, and each user was equipped with 5 devices on the body positions of the torso, right arm, left arm, right leg, and left leg. To align with our wearable use case, we used only the data collected from left-arm device, which consists of the X, Y, Z axis values of the accelerator, gyroscope, and magnetometer, i.e., 9 features. Splitting the dataset by 90/10 for training data and testing data, our model performs both training accuracy and test accuracy > 97% on average, as shown in Figure 14.

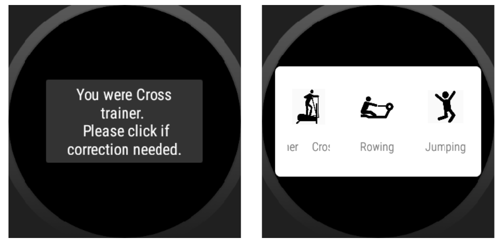
A.1 Study Design

We prepared two videos for the user interviews remotely via zoom, as follows: The first video shows the FCFL-support sport application, in which the machine learning algorithms can recognize the user activities automatically. The video has demonstrated that the user with FCFL can reduce the burden of selecting activities manually, and automatic activity switches from one to another. In contrast, the second video is a baseline application, namely Suunto. As Suunto does not support any intelligent sensing of human sports activity driven by machine learning algorithms, users in the video have to manually select the target activities, and re-select other activities during the activity switches. The two videos have pinpointed the differences in user interaction with smartwatches, especially when users have to select a new activity and switch from one activity to another (automatic vs. manual operations involving a series of tap-and-swipe operations on the touchscreen of a smartwatch). In particular, our videos contain several FCFL screenshots displaying activity recognition and switching in automatic manners. The two videos do not last longer than two minutes to ensure the user memory does not become a bottleneck to the user evaluation for both the application conditions.

[5] https://github.com/OpenMined/PyGrid
[6] https://github.com/OpenMined/KotlinSyft
[7] https://bit.ly/3jNL2w0
[8] https://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities

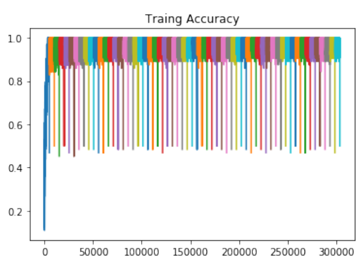


(a) Client UI on an Android smartphone. Optional functions in the UI includes manually requesting to participate in federated training, starting the companion app in the watch, and sending an updated model to the companion app.

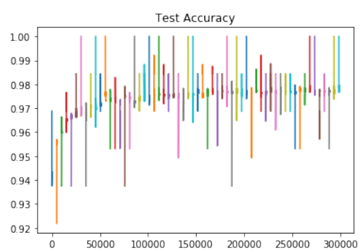


(b) Wearable UI on WearOS (i.e., smartwatch interfaces). The leftmost UI shows the inference result provision, which informs the user once the app is stopped. Upon the need for result correction, the user can simply click on the result which shows up the rightmost UI with available activity choices. By clicking on the activity icon, the app will record the correct result in the corresponding data file.

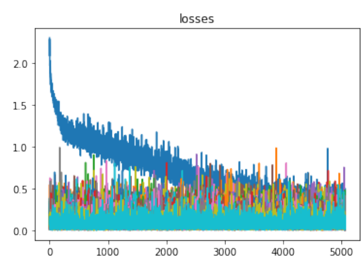
Fig. 13. User interfaces.



(a) Training accuracy.



(b) Test accuracy.



(c) Loss.

Fig. 14. Training performance with HAM dataset. X axis indicates *steps*.

A.2 Procedures

Due to the Covid-19 pandemic and the recent lockdown restriction in the local region, we conducted interviews remotely with our participants via Zoom. During the user interviews, we described the critical functions of the applications. Next, we showed the two videos representing two experimental conditions to the participants. Once a video display had been completed, we distributed a NASA Task Load Index questionnaire to the participants. Table 6 demonstrates the six user workloads on a 0–100 scale between the FCFL-support sport application and a standard sport application (the baseline) on smartwatches, where the lower the score, the higher the user preference. The two videos were selected and displayed randomized to alleviate carry-over effects causing any threats to internal validity. After finishing the questionnaire, another survey about user information and

Table 6. NASA Task Load Index (TLX) for an FCFL-supported sport application (FCFL), and a baseline sport application on a smartwatch, showing mean and standard deviation (SD) in the 2^{nd}, 3^{rd} columns, and statistical F-Critical values / p-values in the 4^{th} and 5^{th} columns, with a total of 17 participants. Statistical significance are depicted by numbers in the italic style.

Workload	FCFL	Baseline	F-Crit.–$F_{(1,32)}$	p-value
Mental	14.06 (15.37)	28.06 (23.87)	4.13	0.05
Physical	15.41 (15.76)	42.24 (30.58)	10.34	*<0.01*
Temporal	35.00 (34.60)	34.88 (23.90)	0.0001	0.99
Performance	20.12 (24.37)	35.12 (20.13)	3.82	0.06
Effort	16.41 (16.08)	36.41 (23.87)	8.21	*<0.01*
Frustration	14.94 (14.86)	31.35 (24.75)	5.49	*0.03*

technology literacy of smartwatches and sports application were presented to the participants. The entire interview lasts no longer than 20 minutes per participant.

A.3 Participants

We recruited a total of 17 participants from our university campus. Regarding the ages of the participants, 76.5% and 17.6% of them were ranged 21–30 and 31–40, respectively. The participants reported a variety of smartwatch usage frequencies: 'Daily' (41.2%), 'Usual' (5.9%), 'Rare' (11.8%), and 'Never Own a Smartwatch but Tried Before' (41.2%). Also, their frequencies of sports application usage are as follows: 'Daily' (35.3%), 'Weekly' (29.4%), 'Monthly' (11.8%), and 'Others' (23.5%), showing that the majority of participants own sufficient technology literacy to the purposes and functions of the standard sports applications. The participation was wholly voluntary and consent-based. The experimental protocols were approved by the university's institutional review board (IRB). We remunerated all participants with a compliment letter, under the premise of social distance, to appreciate their participation.

A.4 User Workload (Results)

We first checked the normality of the user responses with the Shapiro-Wilks Test, as the variance between conditions. Then, we ran a One-way ANOVA to analyze the user responses reflecting the six metrics. Table 6 shows the six metrics in terms of physical, mental, temporal, performance, effort, and frustration. The one-way ANOVA shows that statistical significance exists in physical, effort, and frustration, but not temporal. The results indicated that activity recognition during sports allows users to reduce the physical burdens from a series of tap-and-select operations during menus and buttons selection. In general, the users with the FCFL-supported sports application feel more manageable and less frustrated than the baseline application.

It is important to note that the metrics of mental and performance are sightly higher than the threshold of 0.05 (p-value). Albeit no statistical significance has been found, such metrics show improvements by 42%–49%. The key reason is that the user interactions on the standard application (Suunto) are highly simplified, i.e., less than five interaction costs (tap/switch) to begin or cease the activity recognition. It is expected that the user response will become distinguishable once the complexity of user interfaces increases. Surprisingly, there exists no statistical significance in the metric of temporal. Initially, we expect that the participant can realize the inconvenience of tap-and-swipe operations during a running task, i.e., unstable pointing on the small surface of a miniature-sized touchscreen on a smartwatch. However, most of the participants did not reflect such inconvenience in the video. We conjecture the study method of remote interviews (with video demonstration) limits the user experience.

If re-experiments are permitted after the lockdown situation, we expect users in outdoor scenarios to strongly sense the aforementioned hurdles and hence temporal demands.

Takeaway: FCFL on devices with insufficient computational resources (e.g., smartwatches) can achieve intelligent sensing of user activities, driven by machine learning algorithms. Such benefits can reduce the user's physical workload and alleviate the user's effort and frustration due to the inconvenient interactions with the miniature-size smartwatch. It is worthwhile to mention that FCFL can be further extended to other wearable devices like smart glasses and other applications such as the tracking/ monitoring of sleep patterns and health conditions.