Device or User: Rethinking Federated Learning in Personal-Scale Multi-Device Environments

Hyunsung Cho*
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
hyunsung@cs.cmu.edu

Akhil Mathur
Nokia Bell Labs
Cambridge, UK
akhil.mathur@nokia-bell-labs.com

Fahim Kawsar
Nokia Bell Labs
Cambridge, UK
fahim.kawsar@nokia-bell-labs.com

ABSTRACT

We are witnessing a trend of users owning multiple data-generating wearable and IoT devices that continuously capture sensor data pertaining to a user's activities and context. Federated Learning is a potential technique to derive meaningful insights from this sensor data in a privacy-preserving way without revealing the raw sensor data to a central server. In this paper, we introduce a new problem setting in this multi-device context called *Federated Learning in Multi-Device Local Networks* (FL-MDLN). We identify core challenges for FL-MDLN in relation to its federation architecture, and statistical and systems heterogeneity across multiple users and multiple devices. Then, we introduce a new *user-as-client* (UAC) federation architecture, and propose various device selection strategies to counter statistical and systems heterogeneity in FL-MDLN. Early empirical findings show that our proposed techniques improve model test accuracy as well as battery power efficiency in FL. Based on these findings, we elucidate open research questions and future work in FL-MDLN.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computer systems organization** → **Sensor networks**; • **Computing methodologies** → **Cooperation and coordination**.

KEYWORDS

Federation Architecture, Federated Learning, Device Heterogeneity in FL, Data Heterogeneity in FL, Device Selection in FL

ACM Reference Format:
Hyunsung Cho, Akhil Mathur, and Fahim Kawsar. 2021. Device or User: Rethinking Federated Learning in Personal-Scale Multi-Device Environments. In *The 3rd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT 21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3485730.3493449

*This work was done while the author was an intern at Nokia Bell Labs Cambridge.

SenSys '21, November 15–17, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9097-2/21/11.
https://doi.org/10.1145/3485730.3493449

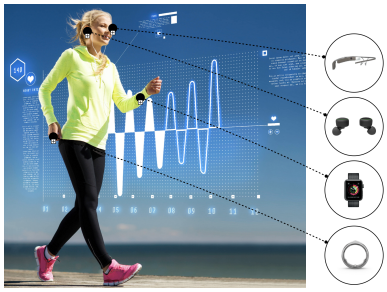


Figure 1: A user owning multiple data-generating devices in a body area network.

1 INTRODUCTION

Federated Learning (FL) has emerged as a promising technique to enable distributed training of machine learning models in networks of remote devices, while keeping the personal data on the devices private. Over the last few years, the research community has built upon the core idea of FL and explored several challenges in FL, including the design of novel system architectures such as hierarchical federated learning [1, 17] and vertical federated learning [20, 27]. There also exists a significant body of work aimed at studying the challenges of systems [15, 18, 26] and statistical [5, 6, 12, 15, 16] heterogeneity across clients in FL.

In this paper, we present a novel problem setting called Federated Learning in Multi-Device Local Networks (FL-MDLN) which touches on all the challenges discussed above. This problem setting is inspired by the current trend of increasing number of sensory devices that reside in local networks, offering intelligent, personalized services. For example, users these days own multiple sensor-enabled, data-generating devices, e.g., smartphone, wearables, smart speakers, and other IoT devices. Some studies (e.g., [21]) even estimate that by the year 2025, each person will own 9.3 connected devices on average. An example of this trend is shown in Figure 1 – here, a user is wearing multiple accelerometer-enabled devices in a body area networkwhich are simultaneously collecting sensor data while the user is performing an activity (e.g., running). IoT devices in smart home networks or smart camera networks are other popular examples of multi-device local sensor networks. In this paper, we focus on the example of personal-scale multi-device body area networks to showcase the problem setting and our approach.

Apart from the growing importance and practicality of this problem setting, it introduces the following core challenges for FL. The first challenge relates to the *federation architecture*. In existing FL approaches with N remote users, each user i is assumed to have

one data-generating device, which acts as a client in FL. However, in FL-MDLN, each user i can own multiple ($K \geq 1$) data-generating devices. This raises a question whether FL systems should consider all $N \cdot K$ devices as separate FL clients (*device-as-client*), or should each user serve as a representative of its local devices in FL (*user-as-client*). In §2, we explain why *user-as-client* is the better architecture for the FL-MDLN problem, and explain the technical challenges for FL in this architecture.

More specifically, we focus on the challenges of statistical and systems heterogeneity in FL-MDLN. Unique to this problem setting is the fact that there exists statistical heterogeneity not just across client (or user) datasets, but also within a client's local dataset. This 'local' statistical heterogeneity is caused by the differences in data distributions of the various devices owned by the user. For example, motion sensors placed at different positions of the body will capture the user's motion differently, thereby making the device datasets non-iid. In this scenario of heterogeneous local datasets, a FL system needs to decide which of the local devices from a user will contribute to the federated learning in a round. We call this problem *Device Selection* and elaborate on it in §3. We also explain why it differs from prior Client Selection approaches in FL, and propose various device selection strategies.

In addition to the statistical heterogeneity across devices, we also need to consider the resource characteristics (e.g., computational capabilities, battery power, network communication speeds) of each device in the Device Selection decision. We take the example of available battery power on each device as a selection metric and illustrate how we can design FL algorithms that balance both system resources and statistical heterogeneity across devices.

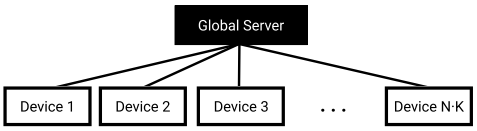
Considering the novelty of this problem setting and the goal of this workshop to identify new challenges in AI/ML for IoT, this paper aims to do a thorough analysis of the FL-MDLN setting, explain the technical challenges for federated learning involved herein, and highlight several open research questions. Although we present some early stage solutions, our goal in this paper is not to arrive at the most sophisticated state-of-the-art solution to these challenges. Instead, our proposed solutions are intended to throw light on potential research directions in this space.

In the following two sections, we introduce the key research challenges in the FL-MDLN problem setting.

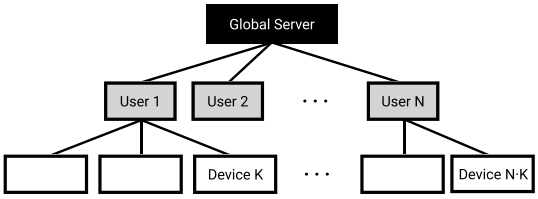
2 CHALLENGE 1: FEDERATION ARCHITECTURE

In conventional federated learning, the remote data-generating devices are considered as FL clients and interact directly with the FL server to perform distributed training. We call this architecture as *device-as-client* (DAC) and depict it in Figure 2(a). In FL-MDLN, however, the data-generating devices have a natural clustering, in that multiple devices are owned by the same user and capture the data-generating process simultaneously from different views. For example, when a user is engaged in a *jogging* activity, various accelerometer-enabled devices placed on their body will capture this physical phenomenon simultaneously, but from different perspectives.

This natural affinity between the devices of a user raises an important question: can we aggregate the data from all devices



(a) Device as Client (DAC)



(b) User as Client (UAC)

Figure 2: Two server-client architectures for FL-MDLN: Device-as-Client and User-as-Client.

belonging to the same user, and treat each user as a FL client (i.e., *user-as-client* in Figure 2(b))? In this architecture, a user becomes a representative for all its devices while participating in FL, in contrast to DAC where each device is an independent FL client.

We hypothesize that the *user-as-client* (UAC) architecture is apt for the FL-MDLN problem setting, both to achieve better accuracy across clients as well as to reduce the system overhead in FL. Below we provide arguments to support this hypothesis at a conceptual level, and later in §2.1, we provide empirical results to further justify this hypothesis.

Firstly, by grouping the devices from the same user, we can leverage the synergy in the temporally-aligned device datasets to learn better feature representations from the data. This form of temporal alignment between devices has been exploited for representation learning in other ML fields [22], but its exploration in FL has been missing. Secondly, data captured on some devices could be highly skewed towards a certain class, depending on the purpose and utility of the device. For instance, if a user prefers to wear a smartwatch while jogging, other devices such as a smartphone or AR glasses will rarely have the opportunity to collect jogging data, and the 'Jogging' class will be under-represented in their datasets. In the DAC architecture, such extreme class imbalance and data heterogeneity across devices can lead to accuracy loss as shown in prior work [8]. Instead, the UAC architecture mitigates this issue because the missing 'knowledge' on one device can be filled-in by the data from other devices. For instance, we can train a generative data translation model such as Pix2Pix [13] using paired and time-aligned samples from various devices, and use it to generate missing data on some devices.

Finally, from the perspective of minimizing the system overhead of FL, UAC allows for better management of system resources at the user-level by having awareness of each device's resource profile. As an example, we can design *personalized* resource-balancing algorithms at the user-level that decide which of the user devices

will participate in a given round of federated training based on their available system resources (e.g., battery power). This decision, for example, could be based on each user's device usage history and battery charging patterns.

The UAC architecture resembles the hierarchical structure of previously proposed vertical FL and hierarchical FL (Section 6) but is more tailored for the FL-MDLN problem setting. Vertical FL assumes situations in which features of the same data entity are distributed across nodes, for example, multiple banks having one user's banking information and history. Hierarchical FL leverages hierarchical structures involving intermediate servers for the purpose of distributing the computing and communication load, rather than exploiting the statistical similarity of devices owned by the same entity. The UAC architecture considers the fact that devices in the same user network are heterogeneous but have underlying statistical similarity under the possession of the same user, or under the same local network. Federated learning in the FL-MDLN also requires system-level consideration on how to manage the federation of constrained resources of mobile, wearable, and IoT devices.

So far, we have conceptually discussed the potential benefits of the *user-as-client* architecture. In the next section, we present a simple approach to implement this architecture and demonstrate its performance gains over DAC.

2.1 Exploring Device Datasets' Aggregation Approach

To illustrate the performance of UAC, we first consider a simple approach where we combine the datasets of all devices from the same user to create an aggregated 'user' dataset. This dataset is then used for federated training with the global server. This simple approach may not be the most sophisticated way to implement the *user-as-client* architecture, but it allows us to quantify the potential benefits of UAC. Later, we also discuss the pros and cons of this approach and discuss future research directions.

Dataset. For our experiments, we used the RealWorld dataset [25] for human activity recognition (HAR). This dataset contains labeled accelerometer and gyroscope traces recorded simultaneously on 7 sensor-enabled devices mounted at various body positions on a user (forearm, thigh, head, upperarm, waist, chest, and shin). In total, there are 15 users in the dataset, each owning 7 sensor-enabled devices. During the data collection exercise, each user performed 8 activities: *climbing stairs down and up, jumping, lying, standing, sitting, running/jogging, and walking*. Each activity was performed for 10 minutes on average by each user, except for *jumping*, which was done for ~1.7 minutes on average. The sampling rate for all the sensors was 50Hz.

Data Pre-Processing. The accelerometer and gyroscope traces are segmented into time windows of 3 seconds, without any overlap. This window length was chosen empirically to align with the duration of various human activities in the dataset. If a 3-second-long trace includes an activity transition, timestamp noise, or data points without labels, the trace gets discarded. The whole dataset is normalized to be in the range of -1 and 1. Finally, we use stratified splitting to divide the data from each device into two parts: training set (75%) and test set (25%).

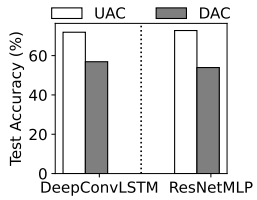


Figure 3: Accuracy comparison between UAC and DAC for two model architectures.

Experiment Setup. In the DAC setup shown in Figure 2(a), each device of each user participates as a client in the federated learning process (7 * 15 = 105 clients in total). Instead, in the UAC setup, each user – holding the aggregated data of all its devices – participates as a client in the learning process (15 clients in total). In this experiment, we do not perform any client selection and as such, all clients participate in each round of training.

Our system is implemented in TensorFlow 2.0 and uses the Flower framework [3] for federated learning. We use TF HParams library to tune FL hyperparameters, and arrived at the following set of hyperparameters: 20 FL rounds and 10 local epochs of training on each client in each round. We tested on two neural network architectures: DeepConvLSTM [23] and ResNetMLP [4]. DeepConvLSTM is a deep neural network with convolutional and LSTM recurrent units for human activity recognition in wearable devices. ResNetMLP has a convolutional neural network based on the ResNet [11] architecture and a multilayer perceptron (MLP). The local training used a learning rate of 0.001, batch size of 16, and the Adam optimizer. After we train a global model using FL, it is evaluated on the held-out test set from all devices. We report the average accuracy over the devices in each experiment setting.

Results. Figure 3 summarizes our results. For both DeepConvLSTM and ResNetMLP models, the performance of UAC exceeds that of DAC by 15% and 18.9% respectively. There could be two reasons for this performance gain: firstly, even though the total number of data samples used for FL are the same in both settings, UAC has more local samples on each client by virtue of its architecture design. This could lead to faster and better convergence of the local loss on each client and in turn, leads to higher accuracy for the global model. Secondly, in the UAC architecture, the local datasets are statistically heterogeneous, as they contain data from multiple devices, each with a different data distribution. We expect that local learning on these statistically heterogeneous (device) datasets is advantageous for learning more generalizable feature representations, which in turn improves the generalizability of the global model.

Open Questions. Our early results illustrate the benefit of the *user-as-client* federation architecture. However, there still remain a number of open research challenges:

- We assumed that data from all devices of a user can be aggregated before federated training for the purpose of showing the optimal benefit of considering heterogeneous data of the same user in training a model. However, this is a huge assumption that it is possible to aggregate data from all devices for a single user. This presents a clear practical drawback considering high data

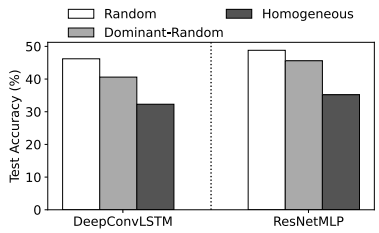


Figure 4: Model accuracy comparison across different device selection strategies.

communication cost and privacy concern over sharing raw data across different devices and vendors. This pragmatic constraint suggests that as opposed to data sharing, we may need to apply collaborative training approaches even among local devices to learn useful features or a local model from their data. How do we design FL systems wherein the training occurs both *across users* and *between the local devices of a user* is an open research problem.

- Selecting all K available devices from all N users for federated training in each round will be expensive from a system resource perspective, particularly in terms of overall energy consumption of FL. It is imperative that we design effective device-selection strategies that choose only a subset of devices from each user in a given round of FL. Moreover, it will be important to explore how such device-selection strategies can co-exist with other client sampling strategies, which in the case of UAC, will be used to sample a subset of users for training in each round.
- The proposed approach assumes that all devices of a user are available for federated training. This could be impractical as often devices are unavailable (e.g., out of battery) or have limited resources (e.g., they are already running expensive computations or they have low battery power). Such device-level factors need to be taken into account while designing the FL algorithm.

3 CHALLENGE 2: DEVICE SELECTION

In this section, we dig deeper into the open question of device selection, which is unique to the FL-MDLN problem setting. It is worth clarifying that Device Selection has a subtle difference from the problem of FL Client Selection studied in prior works [5, 15]. In Client Selection, the goal is to decide which of the c out of N clients (or users) will be selected for training in a given round of FL. Instead, *Device Selection* kicks in after a client is selected, and the goal here is to decide which of the k out of K devices at each client will contribute to federated training in a round. As we discussed earlier, the choice of devices used for training in each round of FL can have direct implications for error convergence as well as the system overhead of FL.

3.1 Strategies for Device Selection

We first explore three strategies for device selection that vary the statistical homogeneity of the devices selected across clients. The first strategy called *Homogeneous Devices* selects the same devices (e.g., a smartphone) on each client in a given round. This ensures that the device dataset(s) chosen on each client have a high statistical homogeneity across them. The second strategy called *Random Devices* selects k ($1 \leq k \leq K$) random device(s) on each client in each round and encourages the device datasets across clients to have statistical heterogeneity among them. The third strategy called *Dominant-Random Devices* also chooses k random devices on each client, however instead of uniform random sampling from all available devices, it samples from the p-largest device datasets (i.e., dominant devices) with a higher probability and from the remaining $K - p$ datasets with a smaller probability.

Results. We evaluate the three device selection strategies with the UAC architecture using the RealWorld HAR dataset. For *Dominant-Random Devices*, we set $p = 3$ and selected thigh, forearm, and head as the dominant devices, each representing a smartphone, smartwatch, and head-mounted device, respectively, in practical scenarios. For training, we use the same DeepConvLSTM and ResNetMLP models with the same FL hyperparameters in §2.1. After we train a global model using FL, it is evaluated on the held-out test set of all devices.

Our results are presented in Figure 4 and show that heterogeneous device selection (*Random Devices*) makes the model more generalizable and accurate across devices, as compared to the *Homogeneous Devices* and *Dominant-Random Devices* strategies. Comparing the heterogeneous and homogeneous selection strategies, we observe an accuracy difference of 13.9% and 13.6% for DeepConvLSTM and ResNetMLP models respectively. The key takeaway from this result is that we need not choose the same type of device (e.g., a smartphone) on each client. Instead, statistical heterogeneity between devices selected on each client is beneficial for generalizability of the global model.

Open Questions. A number of open questions still remain: are there better strategies than random sampling to achieve heterogeneous device selection? How do we account for the resource availability on each device (e.g., available battery power) in the device selection algorithm?

4 FEDMD: A DEVICE SELECTION STRATEGY TO ACCOUNT FOR STATISTICAL AND SYSTEMS HETEROGENEITY IN FL-MDLN

In this section, we present a new device selection strategy called FedMD, which addresses some of the open questions identified in the previous section. More specifically, FedMD is designed to be resource-aware, i.e., it takes into account the heterogeneities in system resources available on each device to make the device selection decision. At the same time, FedMD also considers the impact of statistical heterogeneities across device datasets – as opposed to randomly selecting devices on each client, FedMD samples the devices based on their local loss on the current global model. A *resource-aware* device selection approach should prioritize devices which have higher system resources to perform on-device training. For our use-case, we consider the *available battery level* as the metric on which devices are selected. In other words, we want to prioritize the selection of those devices which have a battery level higher than a certain threshold. On the other hand, we also want the global model to perform accurately for all the devices. As such, a *data-aware* selection approach should prioritize devices, such that

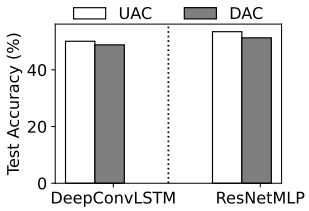
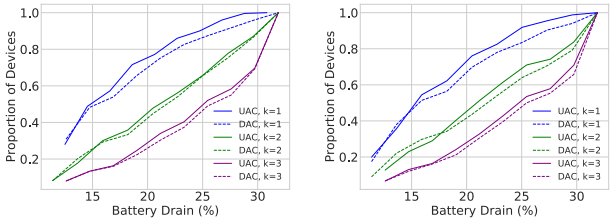


Figure 5: Model accuracy comparison using FedMD with UAC and DAC architectures



(a) DeepConvLSTM model (b) ResNetMLP model

Figure 6: CDF of the percentage of battery drain in devices that participated in the training process of (a) DeepConvLSTM and (b) ResNetMLP

training on them will improve the global model's accuracy on the test set.

With the goal of fulfilling both the *resource-aware* and *data-aware* device selection goals, we present a simple device selection scheme called Federated-Multi Device (FedMD). FedMD first picks a *candidate set* of devices for each user, by looking at their current battery level and checking if their battery level is above a threshold t. This step filters out those devices which do not have enough system resources to participate in FL. Next, the current global model is sent to all the devices in the *candidate set* to compute their local losses. Finally, for the *user-as-client* architecture, FedMD selects k devices with the highest local loss from the *candidate set* of each user. This technique is partially motivated by the empirical finding in prior FL works [5, 14] which show that sampling clients with higher local losses results in faster convergence of the global model.

In effect, FedMD does a two-stage filtering for devices: first it performs resource-aware selection of candidate devices (using their current battery levels), and thereafter it leverages the statistical characteristics of the candidate devices (using their local loss as a proxy) to finalize the choice of devices in each round of FL. It is worth noting that the use of battery levels and local loss in FedMD as proxies for systems and statistical heterogeneity is just one example. Future extension of FedMD can substitute them with other metrics such as update latency and recency of selection.

Experiment Setup. We evaluate FedMD on the RealWorld HAR dataset with the UAC architecture. For comparison, we also evaluate how FedMD performs in a DAC architecture. For this architecture, the scheme selects $k \cdot N$ devices with the highest loss out of the entire device set with battery levels higher than the threshold t.

To emulate battery drain during the federated training process, we make some simplifying assumptions. After each round, devices that participated in the training round are assumed to experience 2% battery drain, while non-participant devices are assumed to have 0.5% battery drain (due to other processes running on the device). These battery drain values were obtained after an offline profiling of the training process on an Nvidia Jetson Nano embedded device – in practice however, the battery drain numbers will be dynamic and highly dependent on the processes running on the device. All devices are initialized with random battery levels in order to simulate varying resource availability on each device. In the experiment, we use $k = 1, 2, 3$ and $t = 60\%$.

Results. We analyze the impact of device selection on statistical and system performance through accuracy and battery consumption results. First, we note that FedMD shows higher accuracy in comparison to the random sampling strategy *Random Devices* (§3.1) in all conditions. On average, FedMD improves the global model accuracy by 3%.

Next, we observe from Figure 5 that the on average, FedMD in a UAC architecture outperforms DAC by ~2%. It is notable that the accuracy gap between UAC and DAC here is smaller than in Figure 3. This result can be explained by the difference in size and heterogeneity of the local datasets in the two settings. While Figure 3 uses an aggregated dataset from all seven devices, here we use the data only from the selected k devices. This result also resonates with our finding in §3.1 about the benefit of having greater statistical heterogeneity between devices in local training. We believe one potential way to address this low accuracy when using a subset of devices is to use local collaborative training and model personalization during FL. We elaborate on these research directions in §5.

Another important goal in FL-MDLN setting is minimizing the battery drain due to local training on resource-constrained mobile and wearable devices. We compare the battery drain using FedMD in the UAC and DAC settings. We focus our analysis on a practical setting where an end-user or mobile OS developers may impose a target limit on the battery drain they can tolerate due to FL. Figure 6 shows a CDF plot for the amount of battery drain due to FL. The x-axis is the target amount of battery drain, and the y-axis shows the percentage of devices whose amount of battery drain (from randomly initialized battery level) is below each x.

We observe that applying FedMD in the UAC architecture results in more percentage of devices remaining above the battery drain limit than DAC. For instance, after training FedMD in UAC, 5.9% more devices have below 20% battery drain than DAC (k=1, model=DeepConvLSTM). This indicates that the UAC has a lower system overhead than DAC, while still achieving higher accuracy.

Our findings are early-stage in nature. While we observe improvements in both accuracy and system overhead due to FedMD, the performance gains could certainly be improved in this novel problem setting of FL-MDLN. In the next section, we introduce several research directions that we are currently investigating.

5 DISCUSSION

Our empirical results in Sections 2-4 hint at the potential benefits of the UAC architecture and the FedMD device selection strategy

in the FL-MDLN problem setting. In reflection of the early results, we discuss several directions for future work.

Device Selection Meets Client Selection. We explored various device selection strategies, however we did not perform any client (i.e., user) selection in each round. Client selection algorithms has been extensively explored in FL literature [5, 19], however the design of algorithms that jointly do client and device selection remains an unexplored problem. More specifically, client selection algorithms will need to consider that each client no longer has a single dataset, but a number of non-iid datasets collected from different devices.

Personalization in FL. We focused on the performance of the global model in the UAC architecture. However, prior works have shown that a global model does not always generalize well to clients with smaller datasets and whose data distribution varies from the global distribution. To address this lack of generalizability, personalization and local optimization techniques have been studied in FL [2, 16]. We hypothesize that the UAC architecture could benefit from personalization and local optimization. As described in Section 2, we can exploit the temporal alignment in the local datasets of different devices to learn robust local models as shown in prior work in self-supervised learning [22].

Local Collaborative Training. In §2.1, we showed that by aggregating the raw data from all devices and treating them as one common dataset yielded high performance in UAC. However, this may not be practical for two reasons: a) communication cost of sharing raw data between wireless devices is high, and b) device vendors may not be willing to share raw data with other devices. This pragmatic constraint suggests that as opposed to data sharing, we may need to design collaborative training techniques even among local devices to learn useful features or a local model from their data. One possible design to support local collaborative training is to run a two-depth federated learning that consists of global FL and local FL. The traditional way of collaborative learning in global FL can be transferred to the local setting as well. One device of a user, possibly one with the most computing resources, can act as a dummy "FL server" within the local setting. It can coordinate local FL among the devices to learn a local model. The weights of this local model are then sent for aggregation to the global server to train the global model.

6 RELATED WORK

Federated training across multiple heterogeneous clients requires handling statistical and system heterogeneities. One approach to improve accuracy of heterogeneous client training is to adapt the local models to client characteristics through personalization [6, 12, 16]. Another approach is to leverage this heterogeneity to build a more fair and robust global model through federated clustering [7, 9, 10, 24]. Client selection strategies further optimize statistical and system utilities based on clients' data and resource information [5, 15, 19]. As we explained in §3, the goals of client selection differ from the Device Selection problem in FL-MDLN, which necessitates further research.

Similar to UAC, new federation architectures have been proposed for various federation goals. Hierarchical FL places intermediate edge servers [1, 17] between clients and the central server to offload partial model aggregation and communication with the goal of reducing communication latency and accelerating the training process. Vertical FL jointly trains a model when data features of the same dataset are split into multiple parties, e.g., a patient's data in different hospitals [20, 27]. UAC differs from these approaches because it is intended to address both statistical and systems heterogeneity in a multi-device setup.

7 CONCLUSIONS

We presented technical challenges for FL in an emerging problem setting called FL-MDLN. In particular, we proposed solutions that included a new federation architecture and various device selection strategies. Our investigation, although in early stages, has thrown light on the numerous challenges and open questions related to statistical and systems heterogeneity in FL-MDLN. We hope to discuss them with the other attendees and experts of AIML in IoT systems at the workshop.

ACKNOWLEDGMENTS

We thank Daniel Beutel for his feedback on the paper, and the Flower team [3] for providing a scalable framework to run realistic FL experiments.

REFERENCES

[1] Mehdi Salehi Heydar Abad, Emre Ozfatura, Deniz Gunduz, and Ozgur Ercetin. 2019. Hierarchical Federated Learning Across Heterogeneous Cellular Networks. arXiv:1909.02362 [cs.LG]
[2] Claudio Bettini, Gabriele Civitarese, and Riccardo Presotto. 2021. Personalized Semi-Supervised Federated Learning for Human Activity Recognition. *ACM Transactions on Intelligent Systems and Technology* 37, 4 (2021), 1–19. arXiv:2104.08094 http://arxiv.org/abs/2104.08094
[3] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. 2020. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390* (2020).
[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709 [cs.LG]
[5] Yae Jee Cho, Samarth Gupta, Gauri Joshi, and Osman Yagan. 2020. Bandit-based communication-efficient client selection strategies for federated learning. *arXiv* (2020), 1–4. arXiv:2012.08009
[6] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. 2020. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461* (2020).
[7] Don Kurian Dennis, Tian Li, and Virginia Smith. 2021. Heterogeneity for the Win: One-Shot Federated Clustering. *arXiv preprint arXiv:2103.00697* (2021).
[8] Moming Duan, Duo Liu, Xianzhang Chen, Renping Liu, Yujuan Tan, and Liang Liang. 2020. Self-balancing federated learning with global imbalanced data in mobile systems. *IEEE Transactions on Parallel and Distributed Systems* 32, 1 (2020), 59–71.
[9] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *arXiv preprint arXiv:2006.04088* (2020).
[10] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. 2019. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629* (2019).
[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]
[12] Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos I. Venieris, and Nicholas D. Lane. 2021. FjORD: Fair and Accurate Federated Learning under heterogeneous targets with Ordered Dropout. (2021). arXiv:2102.13451 http://arxiv.org/abs/2102.13451
[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2018. Image-to-Image Translation with Conditional Adversarial Networks. arXiv:1611.07004 [cs.CV]
[14] Angelos Katharopoulos and François Fleuret. 2018. Not All Samples Are Created Equal: Deep Learning with Importance Sampling. In *ICML*.

[15] Fan Lai, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. 2020. Oort: informed participant selection for scalable federated learning. *arXiv* (2020). arXiv:2010.06081

[16] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and Robust Federated Learning Through Personalization. arXiv:2012.04221 [cs.LG]

[17] Shengzhong Liu, Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Huajie Shao, and Tarek Abdelzaher. 2020. GlobalFusion: A global attentional deep learning framework for multisensor information fusion. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (mar 2020), 1–27. https://doi.org/10.1145/3380999

[18] Bing Luo, Xiang Li, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. 2020. Cost-Effective Federated Learning Design. *arXiv preprint arXiv:2012.08336* (2020).

[19] Takayuki Nishio and Ryo Yonetani. 2019. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. *IEEE International Conference on Communications* 2019-May (may 2019), 1–7. https://doi.org/10.1109/ICC.2019.8761315 arXiv:1804.08333

[20] Daniele Romanini, Adam James Hall, Pavlos Papadopoulos, Tom Titcombe, Abbas Ismail, Tudor Cebere, Robert Sandmann, Robin Roehm, and Michael A. Hoeh. 2021. PyVertical: A Vertical Federated Learning Framework for Multi-headed SplitNN. arXiv:2104.00489 [cs.LG]

[21] Bardia Safaei, Amir Mahdi Hosseini Monazzah, Milad Barzegar Bafroei, and Alireza Ejlali. 2017. Reliability side-effects in Internet of Things application layer protocols. In *2017 2nd International Conference on System Reliability and Safety (ICSRS)*. IEEE, 207–212.

[22] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. 2018. Time-Contrastive Networks: Self-Supervised Learning from Video. arXiv:1704.06888 [cs.CV]

[23] Satya P. Singh, Madan Kumar Sharma, Aime Lay-Ekuakille, Deepak Gangwar, and Sukrit Gupta. 2021. Deep ConvLSTM With Self-Attention for Human Activity Decoding Using Wearable Sensors. *IEEE Sensors Journal* 21, 6 (Mar 2021), 8575–8582. https://doi.org/10.1109/jsen.2020.3045135

[24] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. Federated multi-task learning. *arXiv preprint arXiv:1705.10467* (2017).

[25] Timo Sztyler and Heiner Stuckenschmidt. 2016. On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE Computer Society, 1–9. https://doi.org/10.1109/PERCOM.2016.7456521

[26] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1205–1221.

[27] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.