Towards Real World Federated Learning

Iman Morovatian
Politecnico di Torino
Turin, Italy
iman.morovatian@studenti.polito.it

Mahdi Naderibeni
Politecnico di Torino
Turin, Italy
mahdi.naderibeni@studenti.polito.it

Mohammad Hossein Ehteshami
Politecnico di Torino
Turin, Italy
mohammadhossein.ehteshami@studenti.polito.it

Abstract

Federated learning (FL) is a privacy-preserving machine learning technique that trains a shared model through the collaborative participation of clients and a central server. In this technique, clients train local models using local data, and to preserve the privacy of these clients, only locally trained parameters are sent to the server. This report investigates the preliminaries of FL and introduces some challenges, such as statistical heterogeneity, the availability of clients for participation, and the generalization ability to unknown target data (e.g., a new client). Subsequently, proposed solutions for these challenges are implemented. Specifically, the power-of-choice strategy is implemented to address the availability of clients, while the FedSR and FedDG methods are implemented to tackle the generalization ability challenge. The code of the report can be accessed via this link: https://github.com/imanmorovatian/MLDL-Final-Project

1. Introduction

Federated Learning (FL) is a machine learning paradigm that has been gaining significant attention in recent years and allows for decentralized learning across multiple devices, enabling the utilization of vast amounts of data generated by edge devices while maintaining data privacy. This concept was first introduced by [14], who highlighted the potential of FL to revolutionize the way machine learning models are trained, by shifting from a centralized to a decentralized approach. However, despite the promising potential of FL, several challenges persist in the field, including the availability of clients, data heterogeneity, and domain generalization. This report aims to study and review these challenges by examining the effectiveness of various methods, including FedAvg, Power of Choice, FedSR, and FedDG.

The FedAvg algorithm, proposed by [19], is a widely used baseline for FL. It provides a mechanism for training a global model using local updates from participating devices. The algorithm has been instrumental in the development of FL, providing a foundation upon which subsequent methods have been built. However, FedAvg assumes that all clients are always available for training, which is often not the case in real-world scenarios. This unrealistic assumption has led to the exploration of alternative methods that provide a more realistic approach to client availability in FL, such as the Power of Choice method.

The availability of clients is a critical factor in Federated Learning (FL) due to its decentralized nature. Unlike traditional machine learning where data is centrally located, FL relies on data from multiple devices, each of which may have different availability patterns. This variability can significantly impact the efficiency and effectiveness of the learning process. For instance, some devices may be offline or in a low-connectivity area during a training round, making them unavailable for participation. The Power of Choice method, introduced by [21], addresses the issue of client availability by allowing the server to choose a subset of available devices for training at each round. This method provides a more flexible approach to FL, accommodating the dynamic availability of clients. The Power of Choice method has shown promising results in terms of model performance and communication efficiency, making it a valuable area of research in FL.

Data heterogeneity is another significant challenge in FL. Due to the decentralized nature of FL, the data across different devices can vary greatly in terms of distribution, volume, and quality. This heterogeneity can lead to a discrepancy between the local and global models, affecting the overall performance of the learning process. A large number of various methods such as SCAFFOLD [12], FedDyn [1], and FedSpeed [24] have been proposed to address this issue which by the way are not the concern of this report.

Domain generalization, i.e., the ability of the global model to perform well across different local domains, in FL. This is particularly challenging due to the aforementioned data heterogeneity across different devices. The goal is to ensure that the global model performs well not only on the domains seen during training but also on unseen domains. FedSR [20] uses a shared representation learning approach to enhance the global model's performance across different domains. the method leverages the commonalities across different domains to learn a shared representation, which is then used to train the global model. Also, FedDG [17] employs a domain generalization technique to improve the robustness of the global model to domain shifts. This method aims to ensure that the global model performs well not only on the domains seen during training but also on unseen domains.

This report provides a comprehensive review of the above-mentioned methods, examining their effectiveness in addressing the challenges of client availability and domain generalization in FL. The rest of the report delves into the details of each method, discussing their strengths, and weaknesses, and also provides an analysis of these methods, highlighting their performance in various scenarios.

2. Related Work

In this section, some existing challenges of literature related to federated learning are reviewed. First heterogeneity is discussed, and then the availability of clients is studied. Finally, domain generalization is investigated.

2.1. Heterogeneity

In Federated Learning systems, heterogeneity is one the most significant challenges. [8] This challenge can be studied from multiple aspects. One is Data Space Heterogeneity [8] which means that some clients use data from different feature spaces and so they cannot train a shared model. Another aspect is Statistical Heterogeneity [8] meaning that considering the data is collected from the same data space but it can be non-i.i.d (independent and identically). Most of the studies concerning data heterogeneity in FL address this type of heterogeneity. The other aspect of heterogeneity is System Heterogeneity [8] and happens when the devices participating in the system are equipped with different hardware, power supply, network connectivity and etc. In order to overcome the data heterogeneity in FL the majority of the studies focus on incorporating regularization techniques during local optimization or enhancing the model aggregation mechanism at the server level [18]. But in order to go beyond this [1, 12, 24] have proposed methods that delve deeper into the statistical distributions of the data, resulting in more robust and compelling outcomes.

2.2. Availability of Clients

In each FL setup, a large number of clients participate which sometimes are available and sometimes are not. The unavailability of each client can be a result of facing troubles in the network, available resources and etc. In most of the studies, the contribution of clients is measured according to the size of the dataset they are working on [5]. In some recent studies, a relative score is assigned to each client based on certain conditions, resulting in the selection of each client with a specific probability. This approach enables a more realistic client selection process. [5].

2.3. Domain Generalization

Domain generalization (DG) tries to train a model using multiple different source domains in order to have the ability to generalize directly to an unseen target domain [7, 17]. Some DG methods aim to learn domain-agnostic representations by minimizing domain shifts [9, 10]. Some others rely on the meta-learning paradigm [3, 16] in which training data is divided into meta-train and meta-set sets. The model is trained using the meta-train sets with the aim of improvement of performance on the meta-test sets [27]. However, the mentioned approaches cannot be used in FL due to the fact that they usually require central access to all domains, which violates the privacy of clients [7, 17, 20, 26]. Meanwhile, there are methods such as [13], [11], [23], and [25] making use of deep neural networks, heuristics, data augmentation, and self-supervision, respectively. These methods respect the privacy of clients, but they hardly take advantage of rich data distributions across domains [17]. Also, they do not cope with imbalances [7].

3. Methodologies

In order to tackle the introduced challenges, a considerable number of methods have been proposed. This section reviews the FedAvg algorithm, one of the pioneers in this field, which has been used as the baseline in most studies. In addition, FedSR [20] and FedDG [17] which have been proposed to address domain generalization are discussed.

3.1. FedAvg

One of the most simple and straightforward methods in implementing Federated Learning with the purpose of privacy-preserving and also using the data spread between different clients is the FedAvg [19]. In this approach, each client has its own local dataset and updates the global model locally, and sends the updated model to the server, without sending the local data to the server. The data sent to the server from each client is as minimal as possible. Also, it is worth noting that in each round of training, only a few of the clients are selected for training, since not all clients are always available [19]. In general, FedAvg is done in the following steps:

1. Server selects the clients and sends the current global model to them.
2. Each client starts with the global model and trains the model using its own local data. The local training is done using the SGD algorithm.
3. Updated Model's parameters are sent to the server from clients.
4. Server updates the global model by averaging the parameters sent by the clients. This can also be done using a weighted average to take the different dataset sizes into account.
5. These steps repeat until convergence.

Since in the context of FL, the cost of communication between clients and server is relatively high in comparison to the centralized algorithms [22], usually, clients train more than one epoch in each round in order to increase computations on each client and reduce the communication.

3.2. Power of Choice

The length of each client's dataset is used as a measure in traditional FL client selection methods to determine the likelihood of selecting that client for a train round. However, in the Power of Choice method, each client returns a measure of itself that is utilized by the server to determine the probability that each client will be chosen in the upcoming train round. This metric could be the training loss, the training accuracy, or any other combination of related metrics. According to the approach suggested in [5], choosing a client should be based on a probability that is calculated in three steps:

1. The central server creates a candidate set of clients. The probability of being selected as a candidate for each client is related to the size of its dataset.
2. The server sends the current global model to each of the candidate clients and they send back their losses to the central server.
3. The server calculates a new probability for each of the clients in the candidate set according to their loss values. A client with a higher loss value will have a higher value of probability being selected.

The power of choice method attempts to mitigate the impact of clients with high loss values using the three steps outlined above and to converge the global model towards more promising accuracy.

3.3. FedSR

FedSR is a representation learning framework enabling domain generalization while respecting the privacy of clients in federated learning settings. The algorithm uses two locally-computable regularizers: a domain-specific regularize named L2R and a domain-invariant regularize named conditional mutual information (CMI). The former encourages the model to learn features that are specific to a particular domain, while the latter encourages the model to learn shared features across different domains [20]. More specifically, in federated learning, it is tried to minimize the following objective function

$$f(w) = \frac{1}{K}\sum_{i=1}^{K} f_i(w) \tag{1}$$

Where $f_i(w)$ is the local objective function and K is the number of clients. FedSR adds the two mentioned regularizers and changes the global objective function to the following:

$$f(w) = \frac{1}{K}\sum_{i=1}^{K} f_i(w) + \alpha^{L2R}\ell_i^{L2R} + \alpha^{CMI}\ell_i^{CMI} \tag{2}$$

Where α^{L2R} and α^{CMI} are two hyperparameters.

3.4. FedDG

FedDG is a domain generalization technique based on meta-learning used in federated settings. This technique tries to transfer the distribution across clients while protecting their privacy in order to exploit completely the distributions of source domains. To achieve this purpose, FedDG proposes the exchange of distribution information in the frequency space [17]. More specifically, given the i$^{\text{th}}$ image from the k$^{\text{th}}$ client ($x_i^k \in \mathbb{R}^{H\times W\times C}$), the frequency space signal of this image can be obtained through the use of fast Fourier transform($\mathcal{F}(x_i^k)$)(formula 3).

$$\mathcal{F}(x_i^k)(u,v,c) = \sum_{H-1}^{h=0}\sum_{W-1}^{w=0} x_i^k(h,w,c)e^{-j2\pi(\frac{h}{H}u+\frac{w}{W}v)} \tag{3}$$

After that, the signal can be decomposed into an amplitude spectrum ($\mathcal{A}_i^k \in \mathbb{R}^{H\times W\times C}$) indicating low-level distribution (e.g. style) and a phase spectrum ($\mathcal{P}_i^k \in \mathbb{R}^{H\times W\times C}$) reflecting high-level semantics (e.g. object) of the image. To provide the circumstance for the distribution exchange, firstly, a distribution bank $\mathcal{A} = [\mathcal{A}^1, \ldots, \mathcal{A}^K]$ is created, where each $\mathcal{A}^i$ contains all amplitude spectrums of images from the i$^{\text{th}}$ client showing the distribution of it. Afterward, for each local image of a client, some low-frequency components of its amplitude spectrum are replaced with ones from the distribution bank, while its phase spectrum is unaffected to preserve the semantic content. Consequently, for a local image of a client, there will be images with transformed appearances illustrating distributions of other clients. In more detail, assume there are N clients, using the distribution bank, for the i$^{\text{th}}$ image of the k$^{\text{th}}$ client (x_i^k), an amplitude spectrum from each external client is randomly sampled ($\mathcal{A}_j^n$ where $n = 1 \ldots N$ and $n \neq k$). For each pair of ($\mathcal{A}_i^k, \mathcal{A}_j^n$), using Formula 4, a newly synthesized amplitude spectrum is generated ($\mathcal{A}_{i,\lambda}^{k \to n}$).

$$(1-\lambda)\mathcal{A}_i^k * (1-\mathcal{M}) + \lambda A_j^n * \mathcal{M} \qquad (4)$$

Where $\mathcal{M} = \mathbb{1}_{h,w \in [-\alpha H:\alpha H, -\alpha W:\alpha W]}$ is a binary mask controlling the scale of low-frequency component within amplitude spectrum to be exchanged, and its value is 1 at the central region and 0 elsewhere. Also, λ is the interpolation ratio adjusting the amount of distribution information contributed by ($\mathcal{A}_i^k, \mathcal{A}_j^n$). Finally, the combination of the newly generated amplitude spectrum ($\mathcal{A}_{i,\lambda}^{k \to n}$) and the original phase spectrum ($\mathcal{P}_i^k$) is passed through the inverse Fourier transform $\mathcal{F}^{-1}$ to generate the transformed image. The transformed image is named counterpart and is denoted by t_i^k [17].

The situation for meta-learning is presented after producing counterparts. In each iteration, the process begins with the meta-train phase, where the model is trained on local images (x_i^k) using cross entropy as the loss function, and its parameters are adjusted. Subsequently, in the meta-test phase, the updated parameters are used to train the model on counterparts of the local images (t_i^k) [17].

4. Experiments

In this section, the implementations and results of experiments are discussed. The performance of any model trained in a federated fashion is upper bounded by the results obtained in the centralized setting. Firstly, a simple convolutional neural network (CNN) is tuned in the centralized situation. Then, to create a baseline, this CNN and FedAvg are used as the model in clients and as the aggregator in the server, respectively. Following that, some initial experiments are conducted on the baseline. Next, the results of experiments related to the situation in which all clients are not uniformly available for participation are indicated. Finally, the results of FedSR [20] and FedDG [17] are reported.

4.1. Dataset

The experiments are carried out on the FEMNIST (Federated Extended MNIST) dataset [4]. It is created by partitioning the data in the EMNIST (Extended MNIST) dataset based on the writer of the digit/character [6,15]. The dataset contains images of size 28x28 with **62 classes** including 26 lowercase and 26 uppercase characters of the English alphabet and numbers from 0 to 9. The dataset can be used in two scenarios: i.i.d and non-i.i.d.

- i.i.d. scenario: the likelihood of being sampled for each data point is identical, so all clients have the same underlying data distribution.
- non-i.i.d. scenario: classes and handwriting of the local datasets of clients are different.

4.2. Centralized Model

The model used in the centralized setting as well as in clients in the federated setting has the following structure:

- Two 5x5 convolutional layers with 32 and 64 output channels, respectively.
- Each convolutional layer is followed by both a 2x2 max pooling layer and an activation layer (ReLU).
- Two fully connected layers, the first one mapping the input to 2048 output features, and the second one mapping to the number of classes.

To tune the hyperparameters of the model in the centralized setting, Optuna framework [2] was used. Also, some changes were tested manually. The best accuracy over the EMNIST [6, 15] dataset was **87.37%** achieved by the following hyperparameters:

- Number of Epochs = 50
- Batch Size = 64
- Optimizer = SGD
- Learning Rate = 10^{-3}
- Momentum = 0.9
- Weight Decay for CNN Layers = 10^{-5}
- Weight Decay for Fully Connected Layers = 10^{-3}

4.3. Baseline

Some experiments were conducted on the baseline to inspect the effect of the number of local epochs, the number of participating clients, and being i.i.d or no-i.i.d. The values: **1, 5, 10** and values: **5, 10, 20** were used for the number of local epochs and the number of participating clients, respectively. The experiments were performed for both i.i.d and non-i.i.d scenarios, and results are shown in Fig. 1, 2.

According to the figures, in the i.i.d scenario, increasing the number of local epochs improves the performance of the model, while in the non-i.i.d. scenario, the growth of the number of local epochs from 1 and 5 to 10 leads to the drop of the model's performance. Moreover, in the i.i.d scenario, the rise in the number of clients makes the performance of

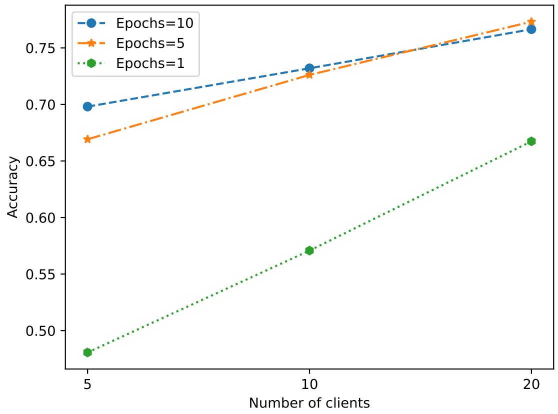

Figure 1. Effect of the number of local epochs and the number of clients in the FedAvg and **i.i.d** scenario. Accuracy refers to the mean of values of overall accuracy over the last 400 rounds.

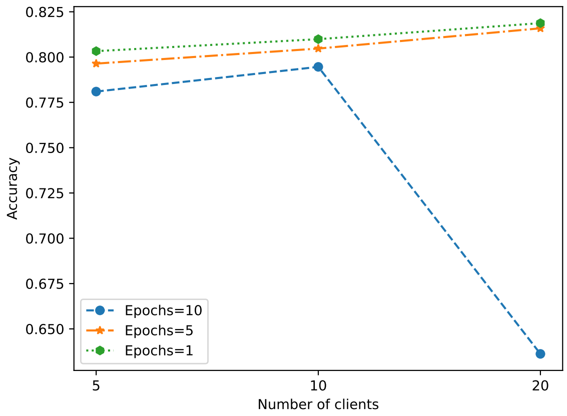

Figure 2. Effect of the number of local epochs and the number of clients in the FedAvg and **non-i.i.d** scenario. Accuracy refers to the mean of values of overall accuracy over the last 400 rounds.

the model better. The same trend can be seen in the non-i.i.d scenario with an exception in which increasing the number of clients from 10 to 20 decreases the performance when the number of local epochs is equal to 10.

The best accuracy of FedAvg in both i.i.d and non-i.i.d scenarios and the best accuracy of the centralized model are compared in table 1. Based on it, the performance of FedAvg is lower than the centralized model.

4.4. Client Selection

Towards a more realistic situation, two different conditions are adopted and each one is examined by different parameters.

The random strategy method considers choosing a portion of the clients with a specific probability and all the other clients with the remaining probability value. Considering the total number of clients is denoted by C and s denotes the portion of the clients that are going to be selected with the probability of p and $G1$ reveals the probability of each client in the first group and $G2$ denotes the probability of each client in the second group:

Model	Accuracy
Centralized Model	87.37%
Best FedAvg in i.i.d scenario	77.31 %
Best FedAvg in non-i.i.d scenario	81.87%

Table 1. Comparison between FedAvg and the centralized model

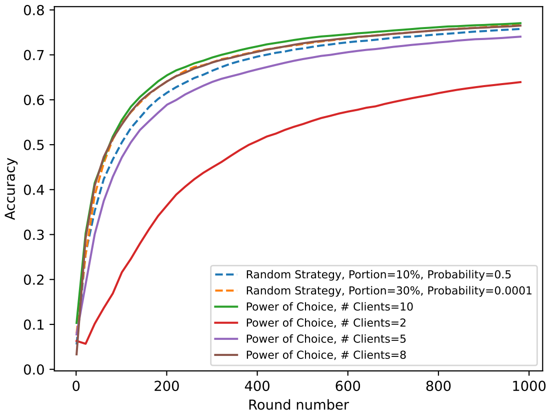

Figure 3. Comparison among client selection methods in the **i.i.d** scenario. For each one of the selection methods with its specified parameters, a learning curve is plotted.

$$\begin{aligned} P(G1) &= \frac{p}{C*s} \\ P(G2) &= \frac{1-p}{C-C*s} \end{aligned} \tag{5}$$

The power of choice method considers running the FedAvg paradigm with a specified number of clients to be selected from the candidate set.

As can be seen from Fig. 3, in the i.i.d scenario, besides the power of choice method with 2 clients in each round, the random strategy method and power of choice converge to almost the same value. However, in the power of choice method, increasing the number of clients to be chosen from the candidate set, increases the accuracy of the global model. In the case of random strategy, selecting 30% of clients with probability 0.0001 has better results in comparison with selecting 10% of clients with probability 0.5. The reason could be when 10% of clients are selected with probability 0.5, the remaining 90% of clients have a very low chance to be selected and so there is the chance that

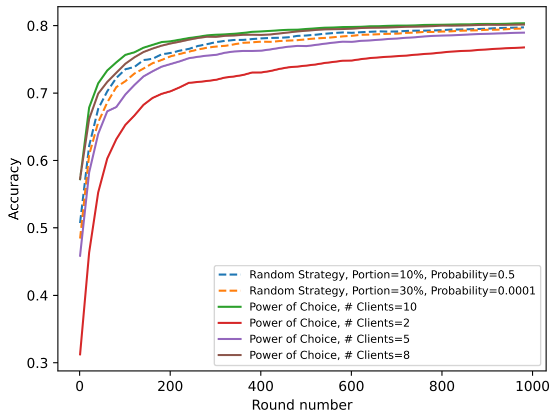

Figure 4. Comparison among client selection methods in the **non-i.i.d** scenario. For each one of the selection methods with its specified parameters, a learning curve is plotted.

the global model does not see the domain of the 90% of the clients which will result in lower accuracy. But in the first case, although 30% of the clients are selected with just probability 0.0001, the remaining 70% of the clients have more chance to be selected and so there is more chance the global model will be trained on a larger space that covers more domains.

In the non-i.i.d scenario, as depicted in Fig. 4, for the power of choice method the discussion is the same as in the i.i.d scenario. But in the random strategy method, the two cases which are selecting 30% of clients with probability 0.0001 and selecting 10% of clients with probability 0.5, converge to almost the same accuracy value. The reason could be that in the non-i.i.d scenario, the classes present in the FEMNIST data are not equally distributed in each client and so still there is no guarantee that if 70% of the clients are selected with a high probability, it can cover all the data targets. But at the same time, it can be possible that if 10% of the clients are chosen with a high probability, it can cover the data targets as they are covered in the first case.

4.5. Domain Generalization

To make the dataset appropriate for domain generalization, 1000 random clients were randomly sampled from the FEMNIST dataset. Then, they were partitioned into six groups and clients' images in each group were rotated 0°, 15°, 30°, 45°, 60°, and 75°clockwise, respectively. These groups were considered as different domains. Following one domain left-out strategy, one of the six domains was used as the test data and the other five domains as the training data. Among existing algorithms for domain generalization in federated settings, FedSR [20] and FedDG [17] algorithms were implemented, and applied to the dataset.

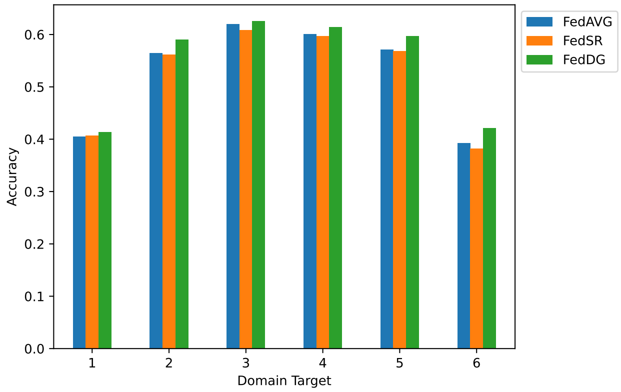

Figure 5. Comparison of FedDG [17], FedSR [20], and FedAvg [19] on the Roated FEMNIST dataset. Accuracy refers to the mean of values of accuracy over the last 400 rounds

Also, the FedAvg [19] algorithm was examined. Experiments were repeated six times for each algorithm and all domains were treated as the domain test. The results are visualized in Fig. 5. Considering the figure, FedDG has the best performance. Meanwhile, approximately, FedAvg and FedSR have the same performance, while in some cases FedAvg is marginally better. Another interesting point is that when the domain that is rotated 30°or 45°is used as the test data, alogoritlms have higher values of accuracy.

5. Conclusion

In this report, firstly, a brief introduction to federated learning (FL) is made, and preliminaries of the field are studied. Following that some existing challenges including heterogeneity, availability of clients, and domain generalization are introduced. After that, among available techniques tackling domain generalization in FL, FedSR, and FedDG are implemented. Also, the power of choice as a solution for the availability of clients is implemented. Finally, extensive experiments are conducted to compare implemented methods with each other and a baseline as well as investigate the effect of different factors in the methods.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021. 2

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019. 4

[3] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-

regularization. *Advances in neural information processing systems*, 31, 2018. 2

[4] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, J Konecnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. arxiv 2018. *arXiv preprint arXiv:1812.01097*, 2019. 4

[5] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020. 2, 3

[6] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017. 4

[7] Lidia Fantauzzo, Eros Fanì, Debora Caldarola, Antonio Tavera, Fabio Cermelli, Marco Ciccone, and Barbara Caputo. Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11504–11511. IEEE, 2022. 2

[8] Dashan Gao, Xin Yao, and Qiang Yang. A survey on heterogeneous federated learning. *arXiv preprint arXiv:2210.04505*, 2022. 2

[9] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015. 2

[10] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. *arXiv preprint arXiv:1711.10125*, 2017. 2

[11] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer, 2020. 2

[12] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020. 2

[13] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12*, pages 158–171. Springer, 2012. 2

[14] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016. 1

[15] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 4

[16] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 2

[17] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1013–1023, 2021. 2, 3, 4, 6

[18] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021. 2

[19] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 6

[20] A Tuan Nguyen, Philip Torr, and Ser Nam Lim. Fedsr: A simple and effective domain generalization method for federated learning. *Advances in Neural Information Processing Systems*, 35:38831–38843, 2022. 2, 3, 4, 6

[21] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019. 1

[22] Osama Shahid, Seyedamin Pouriyeh, Reza M Parizi, Quan Z Sheng, Gautam Srivastava, and Liang Zhao. Communication efficiency in federated learning: Achievements and challenges. *arXiv preprint arXiv:2107.10996*, 2021. 3

[23] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018. 2

[24] Yan Sun, Li Shen, Tiansheng Huang, Liang Ding, and Dacheng Tao. Fedspeed: Larger local interval, less communication round, and higher generalization accuracy. *arXiv preprint arXiv:2302.10429*, 2023. 2

[25] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In *European Conference on Computer Vision*, pages 159–176. Springer, 2020. 2

[26] Liling Zhang, Xinyu Lei, Yichun Shi, Hongyu Huang, and Chao Chen. Federated learning with domain generalization. *arXiv preprint arXiv:2111.10487*, 2021. 2

[27] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2