Resource-Efficient Federated Learning

Ahmed M. Abdelmoniem [1] **Atal Narayan Sahu** [2] **Marco Canini** [2] **Suhaib A. Fahmy** [2]

Abstract

Federated Learning (FL) is a distributed training paradigm that avoids sharing users' private data. FL has presented unique challenges in dealing with data, device, and user heterogeneity which impact both model quality and training time. The impact is exacerbated by the scale of the deployments. More importantly, existing FL methods result in inefficient use of resources and prolonged training times. In this work, we propose *REFL*, to systematically address the question of resource efficiency in FL, showing the benefits of intelligent participant selection and incorporation of updates from straggling participants. *REFL* is a resource-efficient federated learning system that maximizes FL systems' resource efficiency without compromising statistical and system efficiency. *REFL* is released as open source at `https://github.com/ahmedcs/REFL`.

1 Introduction

Federated Learning (FL) has seen wide adoption by service providers such as Apple, Google, and Meta to train computer vision (CV) and natural language processing (NLP) models in applications such as image classification, object detection, and recommendation systems (tensorflow.org, 2020; Yang et al., 2018; FedAI, 2021; Team, 2017; Hsu et al., 2020; Hartmann et al., 2019). The life cycle of FL training is as follows: First, the FL operator builds the model architecture and determines hyper-parameters with a standalone dataset. Model training is then conducted on participating learners for a number of centrally-managed rounds until a satisfactory model quality is obtained. The main challenge in FL is the heterogeneity in terms of computational capability and data distribution among a large number of learners which can impact training performance (Bonawitz et al., 2019a; Kairouz et al., 2019).

Time-to-accuracy is a crucial performance metric which depends on both the statistical efficiency and system efficiency of training (Kairouz et al., 2019; Li et al., 2020; Yang et al., 2021; Wu et al., 2021; Lai et al., 2021). The number of learners, minibatch size, and local steps affect the former. It is common for these factors to be treated as hyper-parameters to be tuned for a particular FL job. A focus on training time has also resulted in schemes that are not robust to non-IID data distributions as they favor certain learner profiles (Li et al., 2022). Finally, learners also have varying availability for training (Kairouz et al., 2019; Bonawitz et al., 2019a; Li et al., 2022; Yang et al., 2021), which requires consideration when dealing with data heterogeneity.

Resource wastage is another major issue—where learners perform training work that does not contribute to enhancing model quality due to discarded late updates. This does not encourage users to participate in FL and makes scaling FL systems problematic. In this project, we design FL systems with the objective of optimizing resource-to-quality in heterogeneous settings.

Existing efforts aim to improve convergence speed (i.e., boosting model quality in fewer rounds) (Li et al., 2020; Wang & Joshi, 2019) or system efficiency (i.e., reducing round duration) (McMahan et al., 2017; 2018), or selecting learners with high statistical and system utility (Lai et al., 2021). These approaches ignore the importance of maximizing the utilization of available resources while reducing the amount of wasted work. Therefore, we introduce *REFL*, a resource-efficient FL system that optimizes resource efficiency without compromising statistical and system efficiency. *REFL* can be integrated as plug-in module into FL systems (Bonawitz et al., 2019a; Lai et al., 2022) and is compatible with existing privacy-preservation methods (Bonawitz et al., 2017; 2019b). In summary, we make the following contributions: 1. We highlight the importance of resource usage of learners' limited capability and availability in FL and present *REFL* to intelligently select participants and efficiently make use of their work. 2. We propose staleness-aware aggregation and intelligent participant selection algorithms to improve resource usage with minimal impact on time-to-accuracy. 3. We implement and evaluate *REFL* using real-world FL benchmarks and compare it with state-of-the-art solutions to show the benefits it brings to FL systems.

[1]School of EECS, Queen Mary University of London, UK and Assiut University, Egypt. [2]KAUST, KSA. Correspondence to: Ahmed M. Abdelmoniem <ahmed.sayed@qmul.ac.uk>.

Workshop of Federated Learning and Analytics in Practice, colocated with 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA.

2 Background

We consider the popular cross-device FL setting introduced in federated averaging (FedAvg) (McMahan et al., 2017; Bonawitz et al., 2019a). The FedAvg approach consists of a (logically) centralized server and distributed learners, such as smartphones or IoT devices. Learners locally maintain private data and collaboratively train a joint global model. The training of the global model is conducted over a series of rounds. Each participant trains the model on its local data for a specified number of epochs and produces a model update (i.e., the delta from the global model) which is sent to the server. The server waits until it receives a target number of participants' updates and aggregates them to update the global model. This concludes the current round and these steps are repeated in each round until a certain objective is met (e.g., target model quality or training budget).

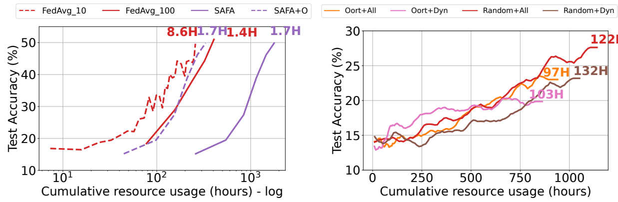
The FL setting is also distinct from conventional training because the learners may exhibit the following types of heterogeneity: 1) ***data heterogeneity:*** learners generally possess variable data points in number, type, and distribution; 2) ***device heterogeneity:*** learner devices have different training speeds owing to different hardware and network capabilities; 3) ***behavioral heterogeneity:*** the availability of learners varies across rounds and there may be learners that abandon the current round if they become unavailable.

3 The Case for Resource-Efficient FL

We motivate *REFL* by underlining the trade-off between system efficiency and resource diversity as key goals in FL. Current FL designs either aim to reduce time-to-accuracy (i.e., system efficiency) (Lai et al., 2021) or to increase coverage of the pool of learners to fairly spread the training data and workload (i.e., resource diversity) (Xie et al., 2020; Li et al., 2020; Wu et al., 2021), but do not consider the cost of wasted work by learners. The first goal results in a discriminatory approach towards certain categories of learners, either preferentially selecting computationally fast learners or learners with model updates of high quality (i.e., those with high statistical utility) (Li et al., 2020; Lai et al., 2021). The second goal entails spreading out the computations ideally over all available learners but at the cost of potentially longer round duration (Xie et al., 2020; Wu et al., 2021) and wasted work.

To strike a balance between the two extremes, FL systems should achieve a sufficient level of resource diversity without significantly sacrificing system efficiency. We first show that existing systems fail to achieve both these goals and result in significant wastage of resources. We also highlight the opportunities they present which we embrace in our design of *REFL*. We use an audio dataset of spoken words provided by Google, hereafter referred to as the Google Speech benchmark (Warden, 2018).

Asynchronous Update Aggregation: Taking inspiration from asynchronous methods (Ho et al., 2013; Xie et al.,



(a) SAFA vs FedAvg **(b) Oort vs Random**

Figure 1: The state-of-the-art methods versus their baselines

2020), SAFA allows straggling participants to contribute to the global model via stale updates. We first evaluate SAFA's resource usage (i.e., the time cumulatively spent by learners in training), and resource wastage (i.e., the time cumulatively spent by learners producing updates that are *not* incorporated into the model). We compare the performance of SAFA as described in Wu et al. (2021) against FedAvg with 10 and 100 clients and SAFA+O which is a version of SAFA that assumes a perfect oracle that knows which stale updates are eventually aggregated (i.e., will not exceed the staleness threshold). Figure 1a shows the resource usage and resulting test accuracy; the lines are annotated with the runtime to achieve the final accuracy. Notably, SAFA is inefficient in terms of resource usage, consuming nearly 5× the resources of Fedavg and SAFA+O to achieve the same final accuracy. By selecting all available learners, then eventually discarding a large number of the computed updates, SAFA wastes around 80% of learner resources.

Optimized Participant Selection: Many FL systems select participants using a uniform random sampler (Caldas et al., 2019; Yang et al., 2018; Bonawitz et al., 2019a). Oort (Lai et al., 2021) prioritizes fast learners which may have unfavorable consequences through biasing the model to a subset of the learners that can reduce data diversity. To see this in practice, we compare Oort with Random selection using the Google Speech benchmark for 1,000 training rounds. We consider two conditions: 1) all learners are available (AllAvail); 2) dynamic availability based on a real-world behavior trace (DynAvail) (Yang et al., 2021). Figure 1b shows that in the non-IID case, learners' availability has a significant effect on model accuracy (i.e., 10-point drop).

4 *REFL* Design

REFL's objective is to enhance the resource efficiency of the FL training process by maximizing resource diversity without sacrificing system efficiency. *REFL* achieves this by reducing resource wastage from delayed participants and prioritizing those with reduced availability. It leverages a theoretically-backed method to incorporate stale updates based on their quality which helps improve the training performance. It proposes a scaling rule for weighting aggregations to mitigate stale updates' impact.

4.1 Intelligent Participant Selection (IPS)

The IPS component allows the global model to capture a wide distribution of learners' data. Moreover, it can fur-

Algorithm 1: Priority Selection Algorithm

Input : N_t-Target number of participants
Output : S-List of selected participants
Initialize $S_t = \emptyset$, $P_t = \emptyset$, $a = (\mu_t, 2\mu_t)$;
on event *Learner_Check_In*:
  Send slot a to learner l;
  Receive learner l's availability probability p_l;
  $P_t = P_t \cup p_l$;
on event *End_Selection_Window*:
  Sort in ascending order P_t;
  Randomly shuffle P_t for probabilities with ties;
  Return S_t as the top N_t learners in P_t;

ther reduce resource wastage by adapting the number of participants in every round.

Least available prioritization: Algorithm 1 describes how the IPS component selects participants from the large pool of available learners. Each learner periodically trains a model that predicts its future availability.

Availability prediction model: We use Prophet (Taylor & Letham, 2017) to train a simple linear prediction model on the Stunner dataset (Szabó et al., 2019), which comprises device events (e.g., the charging state of the devices) from a large number of mobile users. Given a time window in the future, the model produces a probability for the availability of the device within the queried time window.

Adaptive Participant Target (APT): IPS optimizes resource usage by adapting the pre-set target number of participants N_0 selected by the operator. In large-scale scenarios, this could potentially further improve resource consumption. Note, irrespective of clients' availability, APT is an add-on feature that prevents over-committing of participants, further reducing resource consumption.

4.2 Staleness-Aware Aggregation (SAA)

This component enables the participants to submit their updates past the round deadline and processes these stale updates along with the fresh updates. Stale updates can be noisy since the model can drift significantly by the time a stale update arrives. In order to mitigate this impact, we adjust the weight of the stale updates.

Tackling staleness: We propose a stale-update weight that combines the staleness-based damping rule of DynSGD (Jiang et al., 2017) with a boosting factor. The boosting favors a stale update based on its deviation from the fresh updates' average and hence it does not require any information about learners' data. Let, $\Lambda_s = \frac{\|\hat{u}_\mathcal{F} - \frac{u_s + n_\mathcal{F}\hat{u}_f}{n_\mathcal{F}+1}\|^2}{\|\hat{u}_\mathcal{F}\|^2}$ be the deviation of the stale update u_s from the average of the fresh updates $\hat{u}_\mathcal{F}$. Let $\Lambda_{max} = \max_{s\in\mathcal{S}} \Lambda_s$. The boosting factor term scales a stale update s proportional to $1 - e^{-\frac{\Lambda_s}{\Lambda_{max}}}$. Finally, we compute the scaling factor as:

$$w_s = (1-\beta)\frac{1}{\tau_s+1} + \beta(1 - e^{-\frac{\Lambda_s}{\Lambda_{max}}}) \tag{1}$$

where β is a tunable weight for the averaging. For every fresh update $f \in \mathcal{F}$, we choose a scale value of one, i.e., $w_f = 1$. The final coefficients for weighted averaging are the normalized weights. That is, for an update $i \in \mathcal{F} \cup \mathcal{S}$, the final coefficient as: $\hat{w}_i = \frac{w_i}{\sum_{i\in\mathcal{F}\cup\mathcal{S}} w_i}$. Hence, $w_i < w_f$, meaning that weights applied to stale updates are strictly less than weights for new updates. More details and convergence analysis are in Abdelmoniem et al. (2023b).

5 Evaluation

Our evaluation addresses the following questions: 1. Is *REFL* able to achieve its resource-efficient federated learning design goal? 2. Is *REFL* scalable and future-proof?
We summarize the observations on *REFL*: 1. it achieves better models with significantly fewer resources compared to existing systems; 2. it scales well and is future-proof.

5.1 Experimental Setup

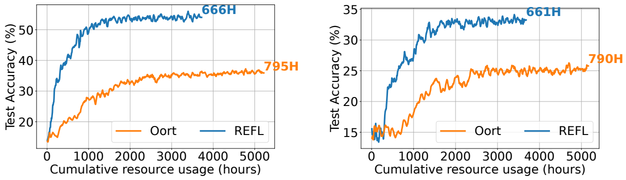
Our experiments simulate FL benchmarks consisting of learners using real-world device configurations and availability traces. Our experiments capture different scenarios, models, datasets and data distributions as detailed next. We use a cluster of NVIDIA GPUs to interleave the training of the emulated learners. The participants train in parallel on time-multiplexed GPUs using PyTorch v1.8.0.

Implementation: We implement *REFL* atop FedScale (Lai et al., 2022), a framework for emulation and evaluation of FL systems. The SAA and IPS are implemented as Python modules and integrated into FedScale's server aggregation logic and participant selection procedures, respectively.

Data partitioning: To account for realistic heterogeneous data, we partition the labeled training dataset among the learners using different methods. We introduce **label-limited mappings** where learners are assigned data samples drawn from a random subset of labels, with data samples per learner following particular distributions as follows. L1) **Uniform distribution**: using uniform random assignment of data points to labels on each learner; L2) **Zipf distribution**: Zipfian distribution with $\alpha = 1.95$ resulting in a higher level of label skew (popularity).

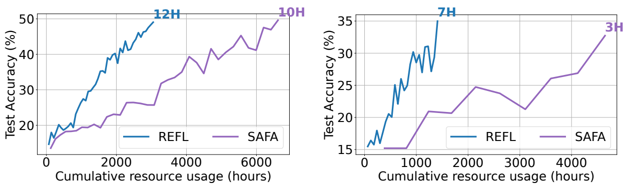
System performance of learners: Learners' hardware performance is assigned at random from profiles of real device measurements from the AI (Benchmark, 2021) and MobiPerf (M-Lab, 2021) benchmarks for inference time and network speeds of mobile devices, respectively. AI Benchmark catalogs inference times for popular DNN models (e.g., MobileNet) on a wide range of Android devices (e.g., Samsung Galaxy S20 and Huawei P40). The profiles include devices with at least 2GB RAM using WiFi, which matches the common case in FL settings (Yang et al., 2018; Bonawitz et al., 2019a; Lai et al., 2021).

Dynamic Availability: We use a trace of 136K mobile users from different countries over a period of 1-week (Yang et al., 2021). The trace contains ≈180 million events such as connecting to WiFi, battery charging, and (un)locking



(a) Uniform distribution (b) Zipfian distribution

Figure 2: Training convergence comparison under OC+DynAvail across different label-limited cases.



(a) 1K Clients (b) 3K Clients

Figure 3: Comparison against SAFA.

the screen. A device is available when it is plugged into a charger and connected to the network, similar to (Bonawitz et al., 2019a; Lai et al., 2021; Abdelmoniem et al., 2023a).

Hyper-parameter settings: The FL and learning hyper-parameters were the default values set by the FedScale framework and no further tuning was done. The common FL hyper-parameters were the same for all methods in the comparison. We used the recommended parameter settings for the evaluated methods, Oort and SAFA.

5.2 Experimental Results

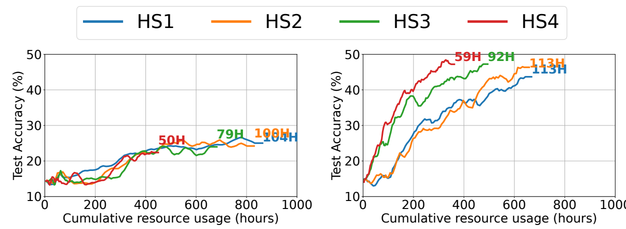
Here, we focus on Google Speech. For more in-depth results of other benchmarks and experimental settings, we refer the reader to Abdelmoniem et al. (2023b).

Selection algorithms: In this experiment, we run the FL training process for more rounds in the label-limited non-IID case and observe that *REFL* achieves superior performance thanks to the availability-based prioritization and aggregation of stale updates. Figure 2 shows that *REFL* converges to significantly higher accuracy than Oort, in less time and with lower resource usage.

Aggregation algorithms: Comparing SAFA and *REFL*, we use the **DL+DynAvail** setting with a total learner population of 1,000 and a round deadline of 100s. We use FedAvg as the underlying aggregation algorithm. *REFL* pre-selects 100 participants and the target ratio is set to 10% and 80% for SAFA and *REFL*, respectively. For both schemes, we set the staleness threshold to 5 rounds.

The results in Figure 3 show that run times of SAFA and *REFL* are comparable, but SAFA consumes significantly more resources. *REFL* improves accuracy by 10 points using $\approx$60% fewer resources compared to SAFA.

Large-scale federated learning: We show the impact of large populations on resource usage using the Google speech benchmark and $3\times$ the number of learners (3,000). As shown in Figure 3b, we observe that SAFA wastes many resources in the non-IID setting and *REFL* is able to produce models of the same quality with nearly $5\times$ less resources.



(a) Oort (b) *REFL*

Figure 4: Impact of future hardware advancements.

Future hardware advancements: We run the Google Speech benchmark in 4 settings using: current device configurations (*HS1*); device configurations with completion times (i.e., computation and communication) doubled for the top X percentile of devices (where X is 25% (*HS2*), 75% (*HS3*), and 100% (*HS4*)). As shown in Figure 4a, with realistic label-limited non-IID settings, *REFL* sees significant performance benefits due to the aggregation of stale updates and higher participant diversity.

6 Related Work

Heterogeneity in FL: A key challenge facing wider adoption of FL systems is uncertainties in system behavior due to learner, system, and data heterogeneity. Learners' computational capacity can restrict contributions and extend round duration (Li et al., 2020; Yang et al., 2021; Abdelmoniem et al., 2023a). System and algorithmic solutions to tackle heterogeneity have been proposed (Wang et al., 2020; Lai et al., 2021; Abdelmoniem & Canini, 2021). Heterogeneity in FL is particularly challenging because participants have varying data distributions and availability, as well as heterogeneous system configurations that cannot be controlled (Kairouz et al., 2019; Abdelmoniem et al., 2023a).

FL proposals: Improvements in FL systems include reducing communication costs (Konečný et al., 2016; Smith et al., 2017; Bonawitz et al., 2019a; Chen et al., 2020; Reisizadeh et al., 2020), improving privacy guarantees (McMahan et al., 2018; Melis et al., 2019; Bonawitz et al., 2019a; Nasr et al., 2019; Bagdasaryan et al., 2020), compensating for partial work (Li et al., 2020; Wang et al., 2020), minimizing devices' energy consumption (Li et al., 2019; M. Evans & Peacock, 1994), and personalizing global models trained by participants (Jiang et al., 2019).

7 Conclusion

We studied two key issues preventing the wider adoption of FL systems: resource wastage and diversity. We presented *REFL* that addresses these issues through two core components that encompass novel selection and aggregation algorithms. Compared to existing systems, *REFL* was shown, both theoretically and empirically, to improve model quality while reducing resource usage and training time. *REFL* is a vital step towards establishing a practical ecosystem for resource-efficient federated learning.

References

Abdelmoniem, A. M. and Canini, M. Towards mitigating device heterogeneity in federated learning via adaptive model quantization. In *EuroMLSys*, 2021.

Abdelmoniem, A. M., Ho, C.-Y., Papageorgiou, P., and Canini, M. A comprehensive empirical study of heterogeneity in federated learning. *IEEE Internet of Things Journal*, 2023a.

Abdelmoniem, A. M., Sahu, A. N., Canini, M., and Fahmy, S. A. REFL: Resource-Efficient Federated Learning. In *EuroSys*, 2023b.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. How To Backdoor Federated Learning. In *AISTATS*, 2020.

Benchmark, A. Performance ranking, 2021. URL https://ai-benchmark.com/ranking.html.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *CCS*, 2017.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, B., Van Overveldt, T., Petrou, D., Ramage, D., and Roselander, J. Towards Federated Learning at Scale: System Design. In *MLSys*, 2019a.

Bonawitz, K., Salehi, F., Konečný, J., McMahan, B., and Gruteser, M. Federated learning with autotuned communication-efficient secure aggregation. In *53rd Asilomar Conference on Signals, Systems, and Computers*, 2019b.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. LEAF: A Benchmark for Federated Settings. In *Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

Chen, Y., Sun, X., and Jin, Y. Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10), 2020.

FedAI. Federated AI Technology Enabler, 2021. URL https://www.fedai.org.

Hartmann, F., Suh, S., Komarzewski, A., Smith, T. D., and Segall, I. Federated Learning for Ranking Browser History Suggestions, 2019.

Ho, Q., Cipar, J., Cui, H., Kim, J. K., Lee, S., Gibbons, P. B., Gibson, G. A., Ganger, G. R., and Xing, E. P. More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server. In *NeurIPS*, 2013.

Hsu, T., Qi, H., and Brown, M. Federated Visual Classification with Real-World Data Distribution. In *ECCV*, 2020.

Jiang, J., Cui, B., Zhang, C., and Yu, L. Heterogeneity-Aware Distributed Parameter Servers. In *SIGMOD*, 2017.

Jiang, Y., Konečný, J., Rush, K., and Kannan, S. Improving Federated Learning Personalization via Model Agnostic Meta Learning. *arXiv 1909.12488*, 2019.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K. A., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. Advances and Open Problems in Federated Learning, 2019.

Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. In *Workshop on Private Multi-Party Machine Learning*, 2016.

Lai, F., Zhu, X., Madhyastha, H. V., and Chowdhury, M. Efficient Federated Learning via Guided Participant Selection. In *OSDI*, 2021.

Lai, F., Dai, Y., Zhu, X., and Chowdhury, M. FedScale: Benchmarking Model and System Performance of Federated Learning. In *ICML*, 2022.

Li, L., Xiong, H., Guo, Z., Wang, J., and Xu, C.-Z. SmartPC: Hierarchical Pace Control in Real-Time Federated Learning System. In *RTSS*, 2019.

Li, Q., Diao, Y., Chen, Q., and He, B. Federated Learning on Non-IID Data Silos: An Experimental Study. In *ICDE*, 2022.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In *MLSys*, 2020.

M. Evans, N. H. and Peacock, B. Statistical distributions, second edition. *Applied Stochastic Models and Data Analysis*, 10(4):297–297, 1994. doi: 10.1002/asm.3150100411. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/asm.3150100411.

M-Lab. MobiPerf: an open source application for measuring network performance on mobile platforms, 2021. URL https://www.measurementlab.net/tests/mobiperf/.

McMahan, B., Ramage, D., Talwar, K., and Zhang, L. Learning Differentially Private Recurrent Language Models. In *ICLR*, 2018.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. Exploiting Unintended Feature Leakage in Collaborative Learning. In *SP*, 2019.

Nasr, M., Shokri, R., and Houmansadr, A. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *SP*, 2019.

Reisizadeh, A., Mokhtari, A., Hassani, H., Jadbabaie, A., and Pedarsani, R. FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization. In *AISTATS*, 2020.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated Multi-Task Learning. In *NeurIPS*. 2017.

Szabó, Z., Téglás, K., Berta, A., Jelasity, M., and Bilicki, V. Stunner: A Smart Phone Trace for Developing Decentralized Edge Systems. In *DAIS*, 2019.

Taylor, S. J. and Letham, B. Forecasting at scale, 2017.

Team, D. P. Learning with privacy at scale, 2017. URL https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf.

tensorflow.org. Tensorflow federated: Machine learning on decentralized data, 2020. URL https://www.tensorflow.org/federated.

Wang, J. and Joshi, G. Adaptive Communication Strategies to Achieve the Best Error-Runtime Trade-off in Local-Update SGD. In *MLSys*, 2019.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *NeurIPS*, 2020.

Warden, P. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition, 2018.

Wu, W., He, L., Lin, W., Mao, R., Maple, C., and Jarvis, S. SAFA: A Semi-Asynchronous Protocol for Fast Federated Learning With Low Overhead. *IEEE Transactions on Computers*, 70(5), 2021.

Xie, C., Koyejo, O., and Gupta, I. Asynchronous Federated Optimization. In *Workshop on Optimization for Machine Learning*, 2020.

Yang, C., Wang, Q., Xu, M., Chen, Z., Bian, K., Liu, Y., and Liu, X. Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data. In *The Web Conference*, 2021.

Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., Ramage, D., and Beaufays, F. Applied Federated Learning: Improving Google Keyboard Query Suggestions, 2018.