Understanding the Impact of Label Skewness and Optimization on Federated Learning for Text Classification

Sumam Francis
sumam.francis@kuleuven.be
KU Leuven
Leuven, Belgium

Kanimozhi Uma
kanimozhi.uma@kuleuven.be
KU Leuven
Leuven, Belgium

Marie-Francine Moens
sien.moens@kuleuven.be
KU Leuven
Leuven, Belgium

ABSTRACT

Federated Learning (FL), also known as collaborative learning, is a distributed machine learning approach that collaboratively learns a shared prediction model without explicitly sharing private data. When dealing with sensitive data, privacy measures need to be carefully considered. Optimizers have a massive role in accelerating the learning process given the high dimensionality and non-convexity of the search space. The data partitioning in FL can be assumed to be either IID (independent and identically distributed) or non-IID. In this paper, we experiment with the impact of applying different adaptive optimization methods for FL frameworks in both IID and non-IID setups. We analyze the effects of label and quantity skewness, learning rate, and local client training on the learning process of optimizers as well as the overall performance of the global model. We evaluate the FL hyperparameter settings on biomedical text classification tasks on two datasets ADE V2 (Adverse Drug Effect: 2 classes) and Clinical-Trials (Reasons to stop trials: 17 classes).

CCS CONCEPTS

• **Computing methodologies → Information extraction**; **Supervised learning by classification**; • **Security and privacy → Domain-specific security and privacy architectures**.

KEYWORDS

Federated Learning, Adaptive Optimizers, Privacy Secure Learning, non-IID, Text Classification

ACM Reference Format:
Sumam Francis, Kanimozhi Uma, and Marie-Francine Moens. 2023. Understanding the Impact of Label Skewness and Optimization on Federated Learning for Text Classification. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3543873.3587599

1 INTRODUCTION

Federated Learning (FL) [22] is a distributed machine learning approach where the training takes place across multiple clients without sharing data between the clients. FL allows multiple clients to collaboratively learn a shared prediction model without explicitly exchanging the data samples thus avoiding the need to store and share the private data.

Conventional centralized learning machines require that all training data located at different client locations are uploaded to a server for training a centralized model. This may give rise to serious privacy concerns. Especially, when dealing with sensitive data like patient data or personal information, privacy measures need to be considered and followed with due care. In the context of the web domain, these privacy measures are of utmost relevance as well since personal information found on the web together with other information of that person found at other locations could be used to infer sensitive information, hence the need for a privacy secure federated learning.

FL allows training the model in a decentralized way that allows multiple clients to collectively train a machine learning model that ensures training takes place across multiple clients without sharing data between the clients, thus keeping the data private. Such a setup is valuable, for instance, in the medical and financial domains when retrieving information from sensitive documents or determining similar patients from different hospitals while preserving the patient's privacy. The two key aspects of FL approaches include dealing with imbalanced and heterogeneous data and incorporating data privacy constraints.

The main challenge of FL is dealing with the data heterogeneity among the different clients (non-IID). One of the key research areas in FL is to understand how the shared prediction model performance is impacted by the varying level of heterogeneity (non-IID) in the client data [14, 15]. The data heterogeneity can arise due to label imbalance in the data (*label skew*; some clients have a certain set of labels which others do not have), or data quantity imbalance (*quantity skew*; the amount of data available per client can be vastly different) or both. There exist other challenges like communication costs and delays which arise due to the communication rounds between clients and global model and local training of the client models. These are important parameters that need to be analyzed in order to improve federated training.

Optimizers have a massive role in accelerating the learning process given the high dimensionality and non-convexity of the search space. A number of modifications have been proposed to traditional machine learning setups to accelerate the learning process. Adaptive optimizers like Adam, Adagrad, etc., have been shown to improve the model performance and have faster convergence rates. In this paper, we present our experiments to analyze hyperparameter optimization (HPO) to tackle data heterogeneity in an FL setting on biomedical classification tasks. We study the effect of different optimization methods when training neural networks in a federated manner both in IID and non-IID setups. We make use

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW '23 Companion, April 30–May 04, 2023, Austin, TX, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9419-2/23/04…$15.00
https://doi.org/10.1145/3543873.3587599

of 2 biomedical datasets: ADE V2 [24] (identity text with Adverse drug effect mentioned) and Clinical_trials (identify the reasons to stop a clinical trial from text).

This paper investigates several key issues that can be optimized for federated learning. Communication efficiency and data heterogeneity arising from data being highly imbalanced and coming from different distributions are important factors to be explored when studying federated setups. This work also studies the impact of incorporating adaptive optimizers with varying learning rates to federated model updates and validates their effectiveness in accelerating convergence. Several adaptive optimizers such as Adam [2], Adagrad [3], and Yogi [4] are compared and we analyze their convergence in the presence of IID and non-IID data in an FL setting. We also perform extensive experiments on varying levels of label skewness and quantity skewness in the non-IID data and show that the use of adaptive optimizers can significantly improve the performance of federated learning. Further, we analyze the impact of varying learning rates, local training epochs, client fractions, varying non-IID-ness, and freezing model parameters on federated model performance.

2 RELATED WORK

Several works explored the limited aspects of hyperparameter tuning in FL (e.g. step-size) on a general setting [7–9]. A monitoring scheme was proposed to reduce the negative effect of label skew by detecting the class imbalance in FL [10]. Nevertheless, this method relies heavily on auxiliary data and poses potential privacy issues. In contrast, we tune a wide range of hyperparameters in realistic federated networks on biomedical classification tasks to understand the influence of various optimization mechanisms.

Several methods were investigated to optimize the FL training process to address non-IID challenges. Federated Averaging (FedAvg) [20] is a predominant algorithm used for training in FL. It experiences performance degradation in terms of low accuracy, slow convergence, and divergence on non-iid data [14, 15]. Several alternate approaches such as FedAdagrad, FedYogi, and FedAdam models have been proposed to mitigate the data heterogeneity impact in FedAvg-based FL. The federated versions of adaptive optimizers, including Adagrad, Adam, and Yogi, were analyzed for their convergence in the presence of heterogeneous data for general nonconvex settings [19]. We experiment with the impact of using multiple adaptive optimizers on different non-IID distributions.

FedEx [16, 17] is an FL-HPO framework to accelerate a general HPO. It is important because the choice of hyperparameters (number of clients, local minibatch size, number of local epochs, and the learning rate) can have a dramatic impact on performance [18]. To further improve the efficiency of HPO, several approaches were proposed in the recent literature. [11] discard the worst configurations during tuning. The MENNDL framework [12] uses genetic algorithms [13]. In this work, we perform benchmark experiments on HPO addressing key properties of FL such as label and quantity skewness, identifying optimal learning rate, client sampling, FL with frozen model parameters, handling data heterogeneity, and privacy issues.

3 METHOD

3.1 Federated averaging

FL requires defining an aggregation strategy. The server aggregate function aggregates the model weights received from every client and updates the global model with the updated weights. The standard aggregation strategy used in FL is FedAvg [13]. The learning process is performed in rounds. The global server samples a fraction of C clients from a total of K clients during each round. These selected clients receive the current shared global model. These clients train this model by optimizing the loss on their local private training data using SGD for multiple epochs and updating the parameters of the model. After each communication round between the global server and sampled clients, local parameter updates are sent to the global model. The global server aggregates the locally updated parameters by performing a weighted average. Further, the updated global model with aggregated parameters are sent to the sampled clients in the next communication round. The above procedure is repeated until the algorithm converges.

The original FedAvg algorithm implicitly set the server and client optimization to be SGD with a fixed learning rate. Though SGD [1] is commonly used in optimization for federated averaging, it can sometimes lead to slow convergence [21].

FedAvg algorithm is a flexible algorithm that can be generalized to have a different optimization update rule for the client side and a different update rule for the global model. Following [25], the FedAvg algorithm can be parameterized by two gradient-based optimizers: a client optimizer with a client learning rate and a server optimizer with a server learning rate. Client optimizer is used to update the local models and the global model takes the aggregate of local model updates as a pseudo-gradient and updates the global model.

To deal with the high dimensionality and noisy gradients of the loss function during federated training, we experiment with adaptive optimizers like Adam, Adagrad, and Yogi, together with varying learning rates both at the client side and server side. We also look into the impact of local training, and client fraction which are important factors that influence the model performance and convergence.

The learning rate is an important factor in the learning process that influences the model performance and convergence. For the local training, we experiment with multiple learning rates with different adaptive optimizers Adam, Adagrad, and Yogi along with SGD. Since the number of communication rounds and the local training iterations play a significant role in determining the overall communication cost of the model, we also analyze the effect on model performance by varying these factors.

3.2 Data partitioning

To simulate the FL setup, the training dataset is partitioned among the K clients. We explore both the IID and non-IID distributions of data.

IID: The data is equally divided among the clients and represents the balanced (IID) case. Here the data points can be seen as representative of the overall data distribution.

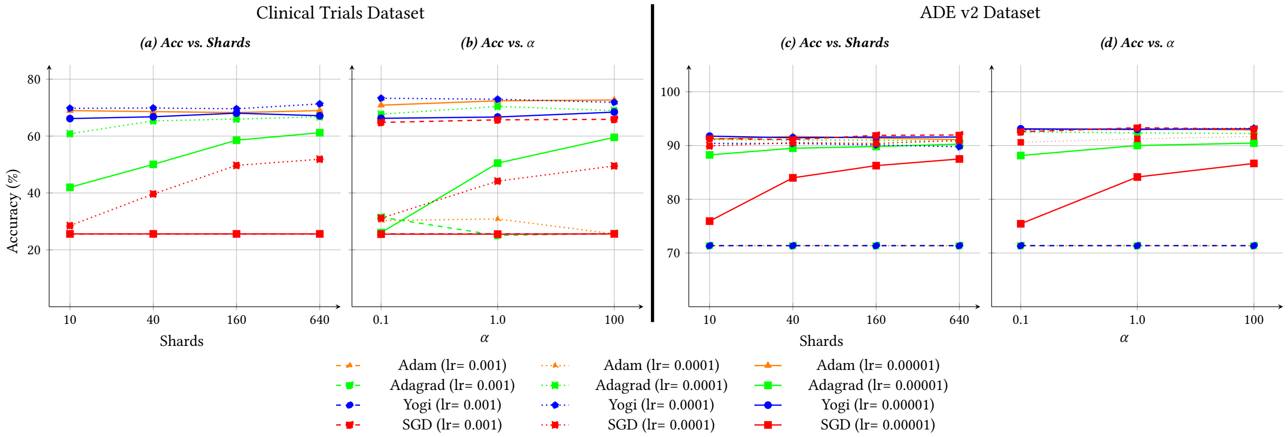

Figure 1: (a) Test accuracy vs non-IID shard sizes on (b) Test accuracy vs Alpha values on CT dataset(c) Test accuracy vs non-IID shard sizes (d) Test accuracy vs Alpha values on ADE dataset for 10 communication rounds with different optimizers using multiple learning rates.

non-IID: The data partitions for each client are drawn from a different distribution. The local data cannot be seen as a representative sample of the overall distribution. The data partitioning among clients can result in label skewness, quantity skewness, or both. For non-IID, we investigate the following setups.

nIID-1: We sort the dataset according to the label index. The dataset is then divided into n shards of equal size. Shard refers to multiple independent groups of input text. Given we have K clients and m training examples, n/K shards are assigned to each client randomly. Thus, when n is small the distribution will be more non-IID, and when n is closer to m it simulates an IID setting. In this approach, only the label skewness is considered and not the quantity skewness.

nIID-2: Secondly, we follow a more probabilistic approach where the samples are drawn based on Dirichlet distribution [5, 6], which we call $\mathbb{D}(\alpha)$, where α controls the skewness of the distribution. $\alpha = 100$ indicates a more IID distribution between clients with a relatively similar number of labels per client. An $\alpha = 0.1$ indicates a case of non-IID setup where there is a high probability that each client receives instances belonging purely to a single class. The smaller the α the more the non-IID nature of the distribution is accentuated. This setting takes both the label and quantity skewness into account.

4 EXPERIMENTAL SETUP

To analyze the influence of the optimizer algorithms on FL, we train models on heterogeneous client data using adaptive optimizers. We also look into communication costs incurred in the federated training setup. For this, we study parameters like the number of local updates performed on the client data before the global update and the client participation at each round. We perform FL on a 2-class and a 17-class biomedical classification task where given an input text, the goal is to correctly identify the class to which the input text belongs. The test dataset is created by holding out 20% of the training set. We report the accuracy of the shared global model on the test dataset. We use two biomedical datasets to benchmark our results.

4.1 Dataset

ADE-Corpus-V2 (ADE): is a dataset for classification tasks to identify if an input text is Adverse Drug Effect (ADE) related or not. The dataset comprises of 23516 sentences labeled with 2 classes.
Clinical_trial_reason_to_stop (CT): This is a classification dataset comprising of reasons why a clinical trial suffered an early stop. The dataset is available at the HuggingFace[1] datasets repository. The text has been extracted from clinicaltrials.gov, the largest resource of clinical trial information by Open Targets organization to provide data relevant to drug development. The dataset comprises 5000 examples labeled with 17 classes.

4.2 Setup

For the nIID-1 setup, n shards of data are assigned to each client where n can be 10, 40, 160, 640. For nIID-2 setup [2], we experiment with $\alpha = 0.1, 1, 100$ where α controls the distribution skewness. We run it over $K = 10$ clients in total, selecting only a fraction of $C = 0.5$ in each round for training. We experiment with modifying the client training with adaptive optimizers Adam, Adagrad, and Yogi along with SGD. Since learning rates have a significant influence on optimizer learning, we experiment with varying learning rates 0.01, 0.001, 0.0001, and 0.00001. We follow the same experimental setup for the IID case also. For the model, we used a batch size of 16, and a maximum sequence length of 128. Each experiment was repeated 3 times and accuracy scores are reported. The target models are evaluated on the development set every 750 step. The checkpoints are saved based on the accuracy obtained on the development set.

To understand the communication overhead of FL in a non-IID setup, we see how many local epochs (E) are sufficient for acceptable model performance and experiment with $E = \{1, 5, 10, 15\}$. We fixed the communication rounds to 10 rounds. Further, we also

[1] https://huggingface.co/datasets/opentargets/clinical_trial_reason_to_stop
[2] The dataset splits will be made publicly available after acceptance

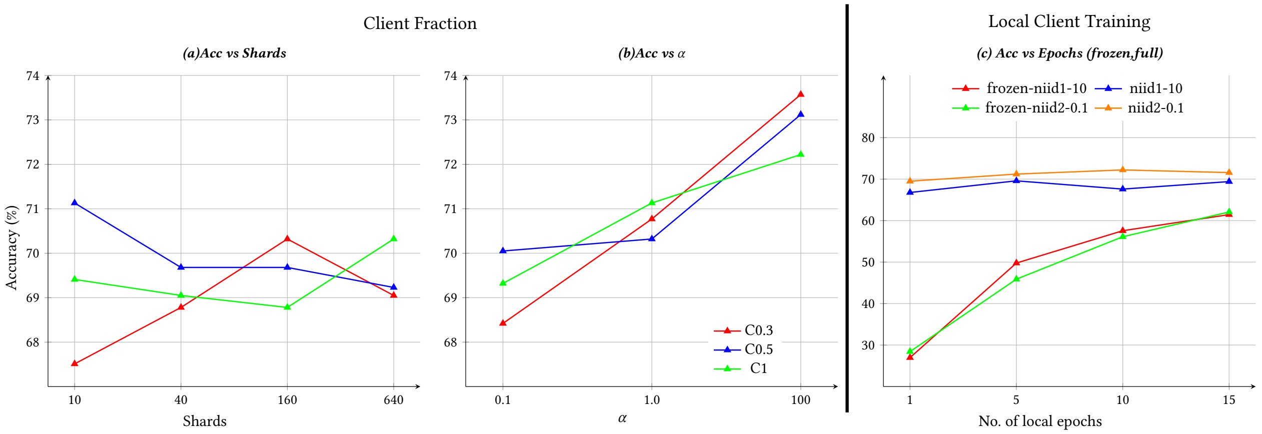

Figure 2: (a) Test accuracy vs non-IID shards for active clients sampled from $C = [0.3, 0.5, 1]$ from a total of 10 clients and (b) Test accuracy vs Alpha values for active clients sampled from $C = [0.3, 0.5, 1]$ from a total of 10 clients and (c) Test accuracy vs no. of local epochs per communication round. We perform 10 communication rounds. We experiment with both frozen and full model parameter fine-tuning settings. These experiments are evaluated on the CT dataset.

experiment with different client fractions $C = \{0.3, 0.5, 1\}$ to study the impact of the fraction of clients participating per communication round on the overall model performance. We further look into freezing the model parameter and fine-tuning only a few of the layers to see its effect on FL.

5 RESULTS AND DISCUSSION

5.1 Role of optimizers in dealing with data heterogeneity in a federated setup

The FL experiments using different optimizers with multiple learning rates are evaluated on the datasets described above. From Figure 1, it can be seen that the Yogi optimizer achieves an accuracy close to 70% in the nIID-1 setup with 10 shards (most non-IID setup) with the best configuration of hyperparameters ($lr = 0.0001$). Similarly the Adam optimizer achieves an accuracy close to 69% with ($lr = 0.00001$). Adagrad's performance reaches around 60% accuracy with the best configuration of hyperparameters ($lr = 0.0001$) while decreasing the learning rate ($lr = 0.00001$) leads to a dramatic drop in accuracy (to around 40%) in the most non_IID case.

In the nIID-2 setup label and quantity skewness in the data is simulated by varying the α value. From Figure 1, it can be seen that the Adam optimizer ($lr = 0.00001$) achieves an accuracy close to 71% in the nIID-2 setup with $\alpha = 0.1$ (most non-IID setup). This is followed by the Yogi optimizer achieving an accuracy of around 66% ($lr = 0.00001$) and the SGD optimizer achieving an accuracy of around 64% ($lr = 0.001$). However, it can be seen that the Adagrad optimizer ($lr = 0.00001$) performance drops below 30% in the extreme non-IID case ($\alpha = 0.1$) but further increases to 60% as the non-IID-ness decreases. The results show that Adam and Yogi demonstrate poor performance with higher learning rates ($lr = 0.01, 0.001$) reaching at most a 25% accuracy without further improvement, while SGD performs better at higher learning rates. Since Adagrad aggregates the entire history of the gradients it can lead to fast decay in the learning rate resulting in weak performance when dealing with a highly non-IID setup.

The results clearly show that incorporating momentum and adaptive optimization methods like Adam and Yogi are critical components for training federated deep neural networks. The results validate their effectiveness in accelerating convergence. The server-side momentum simulates the similar effect of increasing the number of clients selected every round. Due to the exponentially weighted averaging of pseudo-gradients, local model updates from selected clients in previous rounds also contribute to the global model updates in the present round.

The results elucidate that by incorporating server-side adaptivity using adaptive optimizers like Adam and Yogi, models achieve much faster convergence and higher performance. These adaptive optimizers are easier to tune and are more robust when training highly heterogeneous data with both label and quantity skew (CT dataset) thanks to their faster convergence and adaptivity and robustness to hyperparameters whereas SGD and Adagrad are more sensitive to the hyperparameters and lead to slower convergence. However, in cases where label skew is minimal (ADE dataset), the SGD optimizer performs quite well (See Figure 1(c),(d)).

5.2 Influence of local client training

We see that the performance of the model using federated averaging saturates or even degrades with the increased number of local epochs from 1 to 15 epochs. When clients perform n local model updates, the communication cost per client model update can be effectively reduced by a factor of n. However when client data are extremely non-IID, and when we use a greater number of local client model updates, the federated averaging algorithm communicates less frequently with the global model. This may hinder convergence since more local steps result in client updates in FedAvg Algorithm being biased towards the local minima. There is thus a trade-off between convergence and communication efficiency. In this setup, the best value of local model update steps is 5 epochs after which the model performance saturates. Optimizing the number of local client training steps is an important parameter in potentially reducing the overall communication cost of the training.

5.3 Impact of freezing the federated model parameters

From our experiments, freezing the model parameters for federated training and only fine-tuning the last 2 linear layers result in a degradation in performance in the extreme non-IID setup. This can be attributed to the fact that the initial model (BERT-base) needs more domain-specific examples and local training to fit to the biomedical domain. This is evident in Figure 2(c), that increasing the number of local epochs of client training from 1 to 15 for the frozen model improves the test accuracy from 20% to 60% on the CT dataset.

5.4 Effect of client fraction sampling to FL

For both non-IID setups, we observe that increasing the fraction of clients participating per communication round from 0.3 to 0.5 improves the model performance. This indicates that we can improve the learning performance of FL by selecting more clients in each round. This can be contributed to the fact that using too few clients per round can significantly increase the stochastic noise in the training process. However, using all the clients during each communication round does not lead to further improvement either. This can be due to the inclusion of clients with high label noise and a smaller dataset. Looking at it from the perspective of communication cost incurred, reducing the number of clients participating in each communication round by sampling a smaller subset can potentially reduce the communication cost and per-round communication delay that monotonically increases with the number of clients participating. The average of all client's computing delays is further increased by additional time for waiting for the slowest one. Including better client selection algorithms can help select optimal clients in each round. Reducing the number of participating clients to half at each round help in potentially reducing the communication overhead per round.

6 CONCLUSION

FL enables privacy-preserving collaborative training of deep learning models respecting data privacy in applications that use or retrieve information from documents dealing with sensitive data. The predominant factors of FL that are crucial to the model performance are investigated. We performed experiments on varying levels of label skewness and quantity skewness in the non-IID data and demonstrate that adaptive optimizers like Adam and Yogi are more robust to data heterogeneity. Adam and Yogi perform best at lower learning rates while SGD performs better if label skew is not significant and at higher learning rates due to its slower convergence.

Minimizing communication between the global server and participating clients is crucial both for system efficiency and to support the privacy goals of federated learning. Optimizing the number of local client training steps is an important parameter in potentially reducing the overall communication cost of the training. When the majority of the model parameters are frozen, increasing the local training is desirable to improve the model performance. Reducing the number of participating clients to half at each round help in potentially reducing the communication overhead per round and provides optimal performance.

The next step would be to look into aspects that further reduce the communication overhead of FL algorithms as well as training models with differential privacy that adds noise to the model in each round, thus preventing the model from memorizing every unique training example. Another promising research direction is to add a personalization layer to the federated learning framework by keeping track of the client state and further global model fine-tuning on the client side.

In the web domain, clients participating in an FL model setup can be huge. This can open doors to a variety of attacks and vulnerabilities on the privacy of the client and server models. A compromised communication channel could be addressed using a cryptography public key, which keeps message content secure and safe throughout the communication. The global server, the most critical part of the network, must be robust and secured to prevent intrusions access by unauthorized persons. To make reverse inferences of data from gradients more complex, the addition of noise to gradients and compression of gradients can be useful techniques.

In spite of overcoming some centralized machine learning issues, FL faces new challenges related to learning biases [26][27] which can impact model decisions and can yield discrimination of data partitions. Learning bias can in fact be aggravated by FL since bias can be seen mostly related to the non-IID nature of the data distributed across the clients. Here optimizing the parameters of FL like the optimizer algorithms, the number of participating clients and the communication costs incurred are quite prominent. Bias mitigation techniques must be adapted to the specific context of FL. Techniques such as detecting communities of clients that share similar behaviors and building up a global model that integrates all those populations in an equal way can reduce bias. Such an approach could also help detect specific outliers such as poisoning sources and thus increase security.

Thus future works in FL would involve the development of federated models that take into account the information security, confidentiality, and integrity of the participating clients and address the learning biases accentuated by the non-IID nature of the data.

REFERENCES

[1] L eon Bottou. 1998. Online learning and stochastic approximations. Online learning in neural networks 17, 9 (1998), 142.

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[3] John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. J. Mach. Learn. Res.12 (2011), 2121–2159.

[4] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In Advances in Neural Information Processing Systems, pp. 9815– 9825, 2018.

[5] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated Learning on Non-IID Data Silos: An Experimental Study. In 38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022. IEEE, 965–978.

[6] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan H. Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian Nonparametric Federated Learning of Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97), Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 7252–7261.

[7] Zhongxiang Dai, Kian Hsiang Low, and Patrick Jaillet. Federated bayesian optimization via thompson sampling. In Advances in Neural Information Processing Systems, 2020.

[8] Antti Koskela and Antti Honkela. Learning rate adaptation for federated and differentially private learning. arXiv, 2018.

[9] Hesham Mostafa. Robust federated learning through representation matching and adaptive hyper-parameters. arXiv, 2019.

[10] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Addressing class imbalance in federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 10165– 10173, 2021.

[11] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. The Journal of Machine Learning Research, 18(1):6765–6816, 2017.

[12] Steven R Young, Derek C Rose, Thomas P Karnowski, Seung-Hwan Lim, and Robert M Patton. Optimizing deep learning hyper-parameters through an evolutionary algorithm. In Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, pp. 1–5, 2015.

[13] Jian Wu, Saul Toscano-Palmerin, Peter I Frazier, and Andrew Gordon Wilson. Practical multi-fidelity bayesian optimization for hyperparameter tuning. In Uncertainty in Artificial Intelligence, pp. 788–798. PMLR, 2020a.

[14] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the Convergence of FedAvg on Non-IID Data. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net

[15] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated Learning with Non-IID Data. CoRR abs/1806.00582 (2018).

[16] Mikhail Khodak, Tian Li, Liam Li, M Balcan, Virginia Smith, and Ameet Talwalkar. Weight sharing for hyperparameter optimization in federated learning. In Int. Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML 2020, 2020.

[17] Mikhail Khodak, Renbo Tu, Tian Li, Liam Li, Maria-Florina Balcan, Virginia Smith, and Ameet Talwalkar. Federated hyperparameter tuning: Challenges, baselines, and connections to weight-sharing. arXiv preprint arXiv:2106.04502, 2021.

[18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial intel- ligence and statistics, pages 1273–1282. PMLR, 2017b.

[19] Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konec ˇ ny', Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In International Conference on Learning Representations, 2020.

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Net- works from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA (Proceedings of Machine Learning Research, Vol. 54), Aarti Singh and Xiaojin (Jerry) Zhu (Eds.). PMLR, 1273–1282.

[21] Sebastian Bischoff, Stephan Günnemann, Martin Jaggi, and Sebastian U Stich. 2021. On second-order optimization methods for federated learning. arXiv preprint arXiv:2109.02388 (2021).

[22] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. CoRR abs/1610.02527 (2016). arXiv:1610.02527 http://arxiv.org/abs/ 1610.02527

[23] Felbab, V., Kiss, P., and Horváth, T. (2019, September). Optimization in Federated Learning. In ITAT (pp. 58-65).

[24] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. J. Biomed. Informatics 45, 5 (2012), 885–892.

[25] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Kone, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In International Conference on Learning Representations, 2021.

[26] Gosselin, Rémi, Loïc Vieu, Faiza Loukil, and Alexandre Benoit. 2022. "Privacy and Security in Federated Learning: A Survey" Applied Sciences 12, no. 19: 9901.

[27] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... and Zhao, S. (2021). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 14(1–2), 1-210.