Benchmarking Federated Learning Frameworks for Medical Imaging Tasks

Benchmarking Federated Learning Frameworks for Medical Imaging Tasks*

Samuele Fonio[1] [0009–0003–1870–4233]

1 - University of Turin, Italy
samuele.fonio@unito.com

Abstract. This paper presents a comprehensive benchmarking study of various Federated Learning (FL) frameworks applied to the task of Medical Image Classification. The research specifically addresses the often neglected and complex aspects of scalability and usability in off-the-shelf FL frameworks. Through experimental validation using real case deployments, we provide empirical evidence of the performance and practical relevance of open source FL frameworks. Our findings contribute valuable insights for anyone interested in deploying a FL system, with a particular focus on the healthcare domain—an increasingly attractive field for FL applications.

Keywords: Federated Learning · Medical Image Classification · Scalability · Usability · FL Frameworks · Benchmark · Real Case Deployment · Cross Silo

1 Introduction

Federated Learning (FL) [18] has emerged as a crucial area of research in the field of Machine Learning (ML) in response to growing concerns surrounding data privacy [2, 15]. This is especially relevant in the healthcare domain, where data is typically managed by hospitals and medical centers that must adhere to ethical and legal regulations, such as the General Data Protection Regulation (GDPR). Consequently, alternative approaches are necessary to address these data restrictions.

In this context, FL offers a valuable solution by enabling diverse data stakeholders to collaboratively train ML algorithms, overcoming the challenge of decentralized datasets. The core concept of FL involves training ML algorithms by aggregating clients’ models without sharing the underlying data. A central server (referred to as Centralized FL) receives the local models and broadcasts the aggregated model at each iteration. The strength of FL lies in its ability to

* This work receives EuroHPC-JU funding under grant no. 101034126, with support from the Horizon2020 programme (the European PILOT) and by the Spoke 1 ”FutureHPC & BigData” of ICSC – Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU

ensure data privacy, which aligns with the requirements of the healthcare domain. Moreover, FL proves highly effective for Deep Neural Networks (DNNs), particularly when models need to adapt to complex, non-linear patterns found in images or text. In fact, such DNNs often demand large quantities of data, making it challenging to aggregate multiple sources due to privacy concerns. FL facilitates the creation of shared models without compromising the privacy of local datasets, thus addressing this limitation.

However, deploying an FL system in a real-case scenario is not straightforward. Many Federated Learning Frameworks (FLF) are available and open-source, but they differ in many aspects: communication protocol, security, FL tools available, customization, and many others. The two main characteristics explored in this study are *scalability* and *usability*. At the best of our knowledge, literature lacks of works that compare FLFs regarding these two aspects. This gap hinders the research towards high-performing FLFs.

In this work, scalability refers to how computational time varies as the number of clients grows for a fixed problem size. As the number of clients grows, the time to complete the task is supposed to decrease, since the data volume for each party is smaller (*strong* scalability). On the other hand a growing number of clients usually brings more communication costs, which impacts on the performances of the FLF. Scalability is indeed regarded as an important future direction [3, 14, 26] for the design of FLFs.

Usability in the context of Federated Learning (FL) refers to the convenience and ease of deploying an FL system. However, a systematic literature review conducted by Witt et al. [26] highlights a significant limitation in the existing research. Among the 34 reviewed papers, only a small fraction (11 out of 34) considered a non-iid (non-independent and identically distributed) setting, while the majority focused on experiments with MNIST or CIFAR-10 datasets for classification tasks. This narrow focus suggests that FL frameworks may be optimized for specific datasets, making it challenging to adapt them to new datasets. It is essential to address this issue to ensure that FL frameworks can be effectively applied to a wide range of real-world scenarios. For this reason, the customization of FLF is a key aspect, and in the design of the architecture is often taken into account. We aim to provide valuable insights into the adaptability of FL frameworks, shedding light on this crucial usability concern.

To summarize, the contribution of this work is threefold, we:

1. study the scalability of FLFs;
2. provide insights about the usability of FLFs;
3. conduct experiments by deploying multiple FLFs in a realistic environment for the task of Medical Image Classification.

These contributions collectively enhance our understanding of FLFs, addressing the critical aspects of scalability, usability, and practical application in the healthcare domain.

2 Related works

There are already different benchmarks and surveys for the application of FL to the healthcare domain [14, 17, 20], but they usually concentrate on the Decentralized Federated Learning. In our case, we deal with Centralized Federated Learning, which uses a trusted server to deal with the clients. At the best of our knowledge, there aren't works treating the scalability of the FLFs, so we extend the insights suggested by the cited literature review providing experimental results on scalability for the centralized case.

For the specific task of Image Classification there are many studies available regarding FL approaches in general [24] [1] [6] [7] and for specific tasks: Brain tumor segmentation [16], Prediction of SARS-COV2 from Chest X-Ray [10], multi-desease X-Ray classification [18], Breast density classification [22]. Our work does not focus on performances of different FL algorithms, but we enrich the performance evaluation with results on scalability.

For a real-case deployment, there are many possible choice of FLF: OpenFL [21], NVFlare [23], FedML [12], FedScope [27], Flower [4], SecureBoost [8], Substra [11] and in particular for the healthcare domain [9]. In this work we compare only some of them: OpenFL, NVFlare, FedML and Flower. In future works extending this list is a key point to find similarities and differences that impacts on the communication cost.

3 Experiments

In this section we are going to present the experiments which are the core contribution of the proposed study. The task we choose is Image Classification on MedMNIST [28], and in particular on the organAMNIST dataset [5]. It consists of 58,850 images (MNIST-like, 28x28, grayscale) labeled with 11 classes, split into 34.581 for training, 6.491 for validation and 17.778 for test (available to all the clients). The transformations used for data augmentation are normalization, random flip and random rotation. The real case deployment was experimented on a cluster with 10 nodes (each node provided with a Tesla T4 GPU) and a frontend node (with no GPU). The frontend node was used as aggregator and/or administrator (for the frameworks that required it), and other nodes were used as clients of the FL system. This experimental setting mimics a realistic scenario where all clients and the aggregator are on different machines that can only communicate via network requests.

The FL algorithm used is FedAvg [18], with an iid split of data among the clients. As backbone network we chose ResNet18 [13] trained from scratch using the Adam optimizer with an initial learning rate set to 0.0001. A total of 100 FL rounds were performed, with 1 epoch of local training performed by each client using a batch size of 64. The communication protocol used is gRPC [25].

Table 1: Execution times with different frameworks and different numbers of clients.

Number of clients	Frameworks			
	OpenFL	NVFlare	Flower	FedML
2	01:11:24	00:55:43	00:51:08	00:45:38
4	01:07:55	00:40:27	00:38:52	00:36:48
6	01:21:22	00:35:30	00:34:30	00:33:44
8	01:40:13	00:34:18	00:33:07	00:34:15
10	02:12:43	00:33:30	00:29:27	00:28:33

The results obtained are reported in Table 1 and displayed in Fig. 1. There are plenty of architectural details that impact on the performance and usability of a FLF. We are going to discuss about them highlighting the similarities and differences among FLFs that may impact on the performances in terms of scalability and usability.

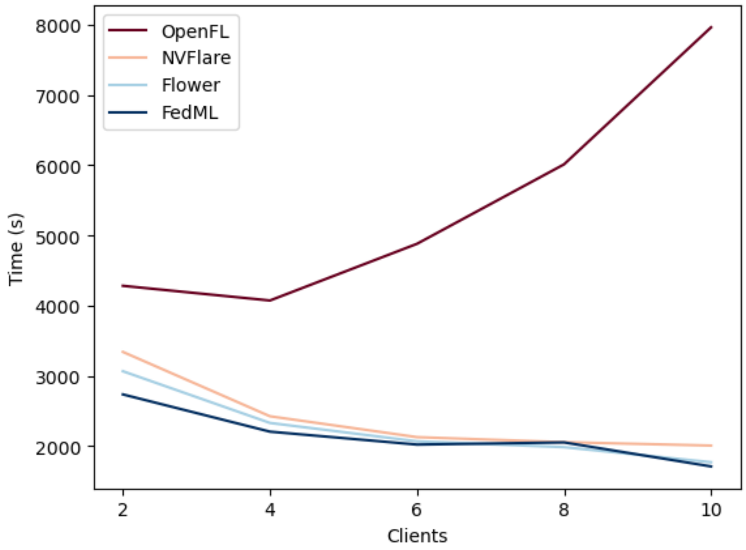

Fig. 1: Execution times for different number of clients.

As we can see, OpenFL does not scale efficiently after 4 clients. On the contrary, the computational time increases when the number of clients goes beyond 4. As highlighted by [19], changing the code at a very low level provided some improvements in the time performances, but the scaling behavior remains inefficient with a growing number of clients.

Because of these low performances we decided to deepen the investigation for OpenFL.

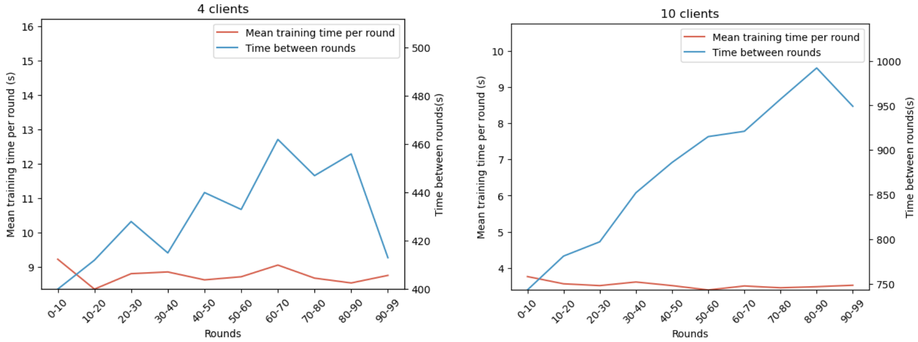

Fig. 2: Execution time with OpenFL for completing windows of 10 rounds along the experiment (blue line) and average training time for each client in each window (red line).

In Fig. 2 we plot the execution time for completing 10 rounds along the experiment and the average training time in the examined window of rounds. It's clear that the training time is almost constant, showing that the problem stands in the communication cost. In fact, we can see that the time needed for completing the first 10 rounds is way less than the time needed for completing the last ones. This behavior was investigated for 4 and 10 clients. This latter scenario shows a linear trend, which clearly indicates the presence of communication overhead. The structure of OpenFL is clear and simple, using gRPC for connecting aggregators and collaborators and transport layer security (TLS) for network connection. A task based programming interface is used, focusing on the whole workflow design rather than the single client customization. We will see in the following that a similar approach is used by NVFlare, but with different results. As a consequence, the detected communication overhead must be investigated properly to see what is the reason of the low performances. An effort has already been done by [19], but more studies are needed to avoid this behavior in the future FLFs.

On the contrary, Flower shows a very good scaling behavior. In fact, execution time decreases with a growing number of clients, highlighting a good implementation for the communication system. With these results, we confirm the good scaling behavior of this FLF, which has been presented in [4] as one of the goal in its design. Comparing Fig. 3 and 2 we can see that the execution time every ten rounds does not increases, as expected by an efficient scaling. From an architectural perspective, this is probably due to the Virtual Client Engine: a tool that virtualizes the Flower client in order to maximize the utilization of its hardware capacities. This decision helps to address the resource consumption, which is the bottleneck when large-scale experiments are conducted. In addition, they developed the *Strategy* module for describing the FL workflow chosen, which makes the customization of the experiments straightforward. From this analysis, we may advocate that having resource-aware agents improves the scalability of a FLF.

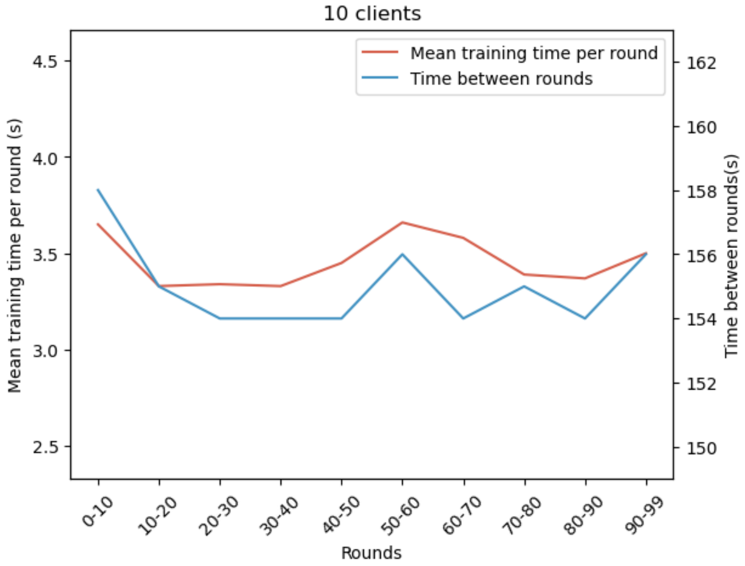

Fig. 3: Execution time with Flower for completing windows of 10 rounds along the experiment (blue line) and average training time for each client in each window (red line).

In a similar manner, FedML relies on a worker-oriented architecture, avoiding the description of the entire training procedure. In order to do so they introduced the *WorkerManager* class, which utilizes an API system to manage the communication, instead of using a training procedure-oriented programming. In particular FedML-API and FedML-core are the main innovative modules of this framework. The first module is responsible to provide the customization of the algorithms, making the implementation of new FL scenarios straightforward. The second module separates the training engine and the communication system, enabling the customization of the whole procedure at many levels. This architectural choice makes the framework flexible and robust, providing good results for what concerns the scalability. In addition, it is the only one among the FLFs presented that provides different communication backends: MPI, mqtt, tRPC, gRPC. The analysis of FedML enforces the idea that a worker-oriented architecture is useful to make the FLF scalable. The use of APIs and the separation between communication and training may be considered as a winning strategy for the design of FLFs both in terms of scalability and usability.

We conclude with NVFlare. This FLF presents a very good scaling behavior. Their architectural focus is on the controller-worker interaction rather than a worker-oriented structure. Similarly to FedML, they implemented an API controller interface, which supports the typical controller-client interaction making the configuration of the workflow very practical. However, the central concept of collaboration stands in the notion of *task*, similar to OpenFL. On the contrary, they use a *Shareable* object to store a different information (like the model weights) and API at an architectural level. As highlighted with FedML, this seems to be a key aspect to take into account for developing scalable and flexible FLFs, and it may shed light on the different performances with respect to OpenFL.

4 Results

With these experiments we provided empirical evidence of the performance of open source FLFs for what concerns the strong scalability, proposing a comparison that is new in the literature. In addition, all along the experiments we provided insights about the design of the FLFs which impacts on scalability and usability and may help in developing and deploying FL systems in a real world scenario.

In particular we can recognized two possible patterns in designing FLFs: *client-oriented* programming and *training procedure-oriented* programming. The former is developed by Flower and FedML in different fashions, obtaining in both cases good results of scalability. The *training procedure-oriented* results effective for the customization of the workflow, but some architectural choices may impact heavily on the scalability of the framework, as highlighted by the difference in performances between NVFlare and OpenFL. A more detailed study is needed to understand what hinders the performances of OpenFL.

For what concerns usability, the implementation of an API system makes the framework very functional, and its customization straightforward. However, the same result can be obtained using particular abstractions like Flower does with the *Strategy* module. In addition, the sharp separation between communication system and training procedure developed by FedML results to be effective both for scalability and usability.

5 Conclusions

In conclusion, we have presented a study that compares different Federated Learning Frameworks (FLFs) when accomplishing the task of Medical Image Classification. The task, which may initially appear simple, has been challenging in multiple aspects. In the end our contribution is threefold:

i) we have provided insights about the usability of open-source FLFs. We examined their implementations and discussed key aspects that make a FLF flexible.

ii) We have tested the scalability of FLFs, which is a crucial aspect of their future development, through detailed experiments. In addition, we highlighted possible key features for designing scalable FLFs.

iii) We conducted a real-case deployment, which makes this study useful also from a practical perspective.

To conclude, scalability remains a critical focus for further advancements in FLFs. Our research provides empirical results on the performances of some of the main open-source FLFs available, with an additional focus on the usability that gives a practical impact to this work. This proposed study represents a starting point in an unexplored area and has the potential to provide valuable insights leading to FLFs improvement.

6 Future works

This work represents a preliminary research for deepening our knowledge of FLFs. In particular the empirical results are useful when it comes to choosing which FLF suits the best for the task needed. However there are many aspects that we did not touch and may have relevance in the future.

First of all, a bigger number of clients should be considered, as much as a detailed study of the architectures both from empirical and theoretical points of view. In fact, recording the time needed for every computational step may shed lights on the positive and negative design choices of an FLF. If this is matched with a theoretical treatment of the FLF, then the study would bring important advancements to FLFs' development.

Furthermore, we have considered only the *strong* scalability, which provides an analysis when the amount of operations needed decreases with an increasing number of clients. On the other hand the *weak* scalability provides results when the amount of work is constant and the number of clients increases. Treating both strong and weak scalability would result in a more complete evaluation of the FLFs.

To conclude, a broader selection of FLFs would bring more comparisons between architectural choices in designing FLFs enabling a broader view on the possible directions towards more performative FLFs.

References

1. Adnan, M., Kalra, S., Cresswell, J.C., Taylor, G.W., Tizhoosh, H.R.: Federated learning and differential privacy for medical image analysis. Scientific reports **12**(1), 1953 (2022)
2. Al-Rubaie, M., Chang, J.M.: Privacy-preserving machine learning: Threats and solutions. IEEE Security & Privacy **17**(2), 49–58 (2019)

3. Beltrán, E.T.M., Pérez, M.Q., Sánchez, P.M.S., Bernal, S.L., Bovet, G., Pérez, M.G., Pérez, G.M., Celdrán, A.H.: Decentralized federated learning: Fundamentals, state-of-the-art, frameworks, trends, and challenges. arXiv preprint arXiv:2211.08413 (2022)
4. Beutel, D.J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K.H., Parcollet, T., de Gusmão, P.P.B., et al.: Flower: A friendly federated learning research framework. arXiv preprint arXiv:2007.14390 (2020)
5. Bilic, P., Christ, P., Li, H.B., Vorontsov, E., Ben-Cohen, A., Kaissis, G., Szeskin, A., Jacobs, C., Mamani, G.E.H., Chartrand, G., et al.: The liver tumor segmentation benchmark (lits). Medical Image Analysis **84**, 102680 (2023)
6. Casella, B., Esposito, R., Cavazzoni, C., Aldinucci, M.: Benchmarking fedavg and fedcurv for image classification tasks. In: Anisetti, M., Bonifati, A., Bena, N., Ardagna, C.A., Malerba, D. (eds.) Proceedings of the 1st Italian Conference on Big Data and Data Science (itaDATA 2022), Milan, Italy, September 20-21, 2022. CEUR Workshop Proceedings, vol. 3340, pp. 99–110. CEUR-WS.org (2022), `https://ceur-ws.org/Vol-3340/paper40.pdf`
7. Casella, B., Esposito, R., Sciarappa, A., Cavazzoni, C., Aldinucci, M.: Experimenting with normalization layers in federated learning on non-iid scenarios. arXiv preprint arXiv:2303.10630 (2023)
8. Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Papadopoulos, D., Yang, Q.: Secureboost: A lossless federated learning framework. IEEE Intelligent Systems **36**(6), 87–98 (2021)
9. Cremonesi, F., Vesin, M., Cansiz, S., Bouillard, Y., Balelli, I., Innocenti, L., Silva, S., Ayed, S.S., Taiello, R., Kameni, L., et al.: Fed-biomed: Open, transparent and trusted federated learning for real-world healthcare applications. arXiv preprint arXiv:2304.12012 (2023)
10. Flores, M., Dayan, I., Roth, H., Zhong, A., Harouni, A., Gentili, A., Abidin, A., Liu, A., Costa, A., Wood, B., et al.: Federated learning used for predicting outcomes in sars-cov-2 patients. Research Square (2021)
11. Galtier, M.N., Marini, C.: Substra: a framework for privacy-preserving, traceable and collaborative machine learning. arXiv preprint arXiv:1910.11567 (2019)
12. He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., et al.: Fedml: A research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518 (2020)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Joshi, M., Pal, A., Sankarasubbu, M.: Federated learning for healthcare domain-pipeline, applications and challenges. ACM Transactions on Computing for Healthcare **3**(4), 1–36 (2022)
15. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. IEEE signal processing magazine **37**(3), 50–60 (2020)
16. Li, W., Milletarì, F., Xu, D., Rieke, N., Hancox, J., Zhu, W., Baust, M., Cheng, Y., Ourselin, S., Cardoso, M.J., et al.: Privacy-preserving federated brain tumour segmentation. In: Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10. pp. 133–141. Springer (2019)
17. Lian, Z., Yang, Q., Wang, W., Zeng, Q., Alazab, M., Zhao, H., Su, C.: Deep-fel: Decentralized, efficient and privacy-enhanced federated edge learning for healthcare cyber physical systems. IEEE Transactions on Network Science and Engineering **9**(5), 3558–3569 (2022)

18. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
19. Mittone, G., Riviera, W., Colonnelli, I., Birke, R., Aldinucci, M.: Model-agnostic federated learning. Springer, Limassol, Cyprus (August 2023). https://doi.org/10.48550/arXiv.2303.04906
20. Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A., Srivastava, G.: A survey on security and privacy of federated learning. Future Generation Computer Systems **115**, 619–640 (2021)
21. Reina, G.A., Gruzdev, A., Foley, P., Perepelkina, O., Sharma, M., Davidyuk, I., Trushkin, I., Radionov, M., Mokrov, A., Agapov, D., et al.: Openfl: An open-source framework for federated learning. arXiv preprint arXiv:2105.06413 (2021)
22. Roth, H.R., Chang, K., Singh, P., Neumark, N., Li, W., Gupta, V., Gupta, S., Qu, L., Ihsani, A., Bizzo, B.C., et al.: Federated learning for breast density classification: A real-world implementation. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings 2. pp. 181–191. Springer (2020)
23. Roth, H.R., Cheng, Y., Wen, Y., Yang, I., Xu, Z., Hsieh, Y.T., Kersten, K., Harouni, A., Zhao, C., Lu, K., et al.: Nvidia flare: Federated learning from simulation to real-world. arXiv preprint arXiv:2210.13291 (2022)
24. Sheller, M.J., Reina, G.A., Edwards, B., Martin, J., Bakas, S.: Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In: Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4. pp. 92–104. Springer (2019)
25. Wang, X., Zhao, H., Zhu, J.: Grpc: A communication cooperation mechanism in distributed systems. ACM SIGOPS Operating Systems Review **27**(3), 75–86 (1993)
26. Witt, L., Heyer, M., Toyoda, K., Samek, W., Li, D.: Decentral and incentivized federated learning frameworks: A systematic literature review. IEEE Internet of Things Journal (2022)
27. Xie, Y., Wang, Z., Gao, D., Chen, D., Yao, L., Kuang, W., Li, Y., Ding, B., Zhou, J.: Federatedscope: A flexible federated learning platform for heterogeneity. arXiv preprint arXiv:2204.05011 (2022)
28. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Scientific Data **10**(1), 41 (2023)