Federated Approach for Glioblastoma Radiogenomic Methylation Classification

Eriberto Andrea Franchi[1,*], Prof. Elena De Momi[1] and Prof. Alberto Redaelli[1]

[1]*Politecnico di Milano - Department of Electronics, Information and Bioengineering (DEIB), Via G. Ponzio, 34, 20133 Milan, Italy*

Abstract

The privacy and data management advantages introduced by the federated learning (FL) paradigm make it the perfect solution for medical data science. Yet, there is a lack of studies exploring the use of this technology in the context of radiogenomic classification, with only one relevant study analyzing the technology in this field. Previous studies explored the use of centralized learning (CL) in radiogenomics with no actual strategy for privacy preservation during the deployment of the analyzed models. The work reported in this paper aims at showing the feasibility of a reusable centralized-to-federated conversion framework, bringing the federated adaptation of a centralized submission to the RSNA-MICCAI Brain Tumor Radiogenomic Classification as a first use case, showing that results are comparable between the two paradigms under same constraints. To achieve these goals this work explores Flower as a FL framework for ease of use and potential for customization, and adapts an existent RSNA-MICCAI challenge submission using a DenseNet121 classification model and the middle coronal slice of the FLAIR acquisitions from each patient as training set. Both the CL and FL pipelines are validated locally on accuracy, F1-score, Cohen's kappa and Area-Under-Curve (AUC), generated over repeated runs to have enough data to be properly tested with a Wilcoxon Rank Test. Across ten repeated runs each for an increasing number of clients, the federated DenseNet121 achieved validation metrics (accuracy, F1-score, Cohen's kappa, and AUC) statistically indistinguishable from the centralized baseline for client counts of 3, 4, and 5 (Wilcoxon rank-sum $p > 0.05$). Only in the 2-client configuration did the FL setting exhibit significantly tighter metric distributions, thus significant differences in F1-scores and AUCs ($p \leq 0.05$). This artifact was attributed to deterministic, class-balanced splitting and fixed validation ordering in that extreme partitioning scheme. These results, over all, show the comparability of the validation metrics between the centralized and federated implementation of the DenseNet121 model, laying the groundwork for a more extensive "centralized-to-federated" adapter framework.

Keywords

Medical Informatics, Federated Learning, Centralized-to-Federated Migration, Privacy, OMOP Common Data Model (CDM)

1. State of the Art

Federated learning (FL) presents several key advantages compared to traditional centralized learning approaches. While this paradigm has already been applied in multiple fields, it is particularly beneficial in healthcare settings. By keeping patient data localized at each medical institution, FL mitigates privacy concerns, reduces the risk of data breaches, and simplifies compliance with regulations such as HIPAA and GDPR. Moreover, FL enables collaborative model development across diverse clinical sites without requiring the large-scale data transfers implied in centralized data collection, thereby lowering network bandwidth demands and avoiding the need for a central data repository. Finally, FL leverages on-site computational resources for model training, promoting scalability and efficiency when working with heterogeneous hardware environments.

Empirical benchmarks consistently show that under realistic non-IID conditions, FL can match or exceed centralized performance, provided appropriate aggregation, personalization, and privacy enhancements are applied [1, 2, 3, 4, 5, 6, 7].

Ital-IA 2025: 5th National Conference on Artificial Intelligence, organized by CINI, June 23-24, 2025, Trieste, Italy

*Corresponding author.

eribertoandrea.franchi@polimi.it (E. A. Franchi); elena.demomi@polimi.it (E. De Momi); alberto.redaelli@polimi.it (A. Redaelli)

https://nearlab.polimi.it/medical/eriberto-andrea-franchi/ (E. A. Franchi); https://www.deib.polimi.it/eng/people/details/172386 (E. De Momi); https://www.deib.polimi.it/eng/people/details/334534 (A. Redaelli)

0009-0009-5830-9573 (E. A. Franchi); 0000-0002-8819-2734 (E. De Momi); 0000-0002-9020-2188 (A. Redaelli)

Federated learning has already been deployed in different medical research areas. Notably, one medical field in which the application of FL has been investigated by only one relevant paper is radiogenomics.

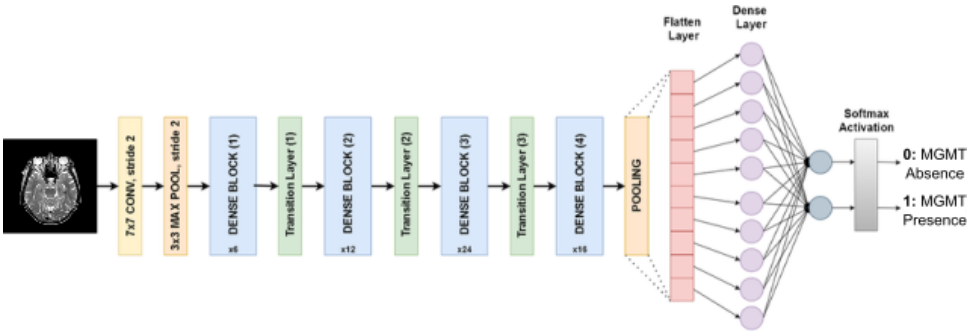

Figure 1: A diagram representing the neural network that was used in this work.

1.1. Federated Learning in Radiogenomic Classification

Up to date, only one paper has been found investigating applications of FL within the radiogenomic field. FL has been applied to the RSNA-MICCAI Brain Tumor Radiogenomic Classification (BraTS-2021) challenge, which tasks participants with predicting MGMT promoter methylation presence from multi-parametric MRI (mpMRI: FLAIR, T1w, T1wCE, T2w) on a hold-out test set. Atef et al. [8] implemented an FL pipeline using the OpenFL framework and an improved EfficientNet-B3 backbone. After 300 communication rounds among 10 collaborating sites, their FL model achieved 99.99% of the centralized model's accuracy on the BraTS-2021 dataset, demonstrating that data heterogeneity and imbalance across institutions did not significantly degrade performance.

1.1.1. Structure of the BraTS-2021 Dataset

This dataset consists of two folders for training and testing, both containing subfolders representing patients. Each patient's sub-folder contains itself folders for each of the mpMRI sequences, where the MRI volumes are collected as DICOM files for each volume slice. At the root level of the dataset, a `.csv` file stores the labels that describe the methylation status of the O^6-methylguanine-DNA methyltransferase (MGMT) promoter for each training subject: 0 for unmethylated, 1 for methylated.

1.1.2. Federated Learning Frameworks

FL requires predefined execution environments to ease data science developers from implementing communication and cybersecurity features. Atef et al. [8] relied on OpenFL [9], a FL framework developed by IBM. Yet, a variety of such environments exists, such as Kubernetes [10], NVFlare [11] or Flower [12]. Each of these environments, while providing a more or less user-friendly way to launch FL pipelines on-the-go, introduce some frictions in the implementation of advanced deep learning tools.

1.2. Deployment of Federated Learning

Real-world deployment challenges persist, with most studies still relying on simulations or small scale feasibility studies. Orchestrating a federated network, whether via Kubernetes [10], NVFlare [11] or Flower [12], demands robust error and network heterogeneity handling, along seamless integration with clinical data models like OMOP CDM so to refer to standardized data structures [13].

2. Aim of the Work

On the trail of the work of Atef et al. [8], this work implements a FL pipeline in the area of radiogenomics using Flower as a framework (Fig. 2). A submission to the RSNA-MICCAI Radiogenomic Classification

challenge carrying out the MGMT classification task in a centralized way [14], was chosen and adapted to a Flower project. This submission used a DenseNet121 architecture and used only the middle slice from FLAIR MRI acquisitions, de facto reducing the computational overhead introduced by loading DICOM data. The final goal of the study is to show the comparability of the validation metrics resulting from the centralized and federated implementations, to show that comparability can be achieved even with less than the computationally intensive and time consuming 300 communication rounds used by Atef et al..

3. Methods

3.1. Dataset

This study rely on the publicly available BraTS 2021 multi-parametric MRI dataset, which provides preoperative scans and associated MGMT promoter-methylation labels for each patient. For consistency and computational efficiency, we extract only the single middle coronal slice from each patient's FLAIR volume (like the input image in Fig. 1). Each slice is then resized to 128 × 128 pixels and its intensity values linearly rescaled to the [0, 1] range. Any further data augmentation was not applied, so that the dataset seen by our model exactly matches slice characteristics from the RSNA-MICCAI challenge submission we chose as a benchmark. In the federated setting, we partition this same preprocessed pool of slices into N virtual clients (ranging from two to six) using Flower's PathologicalPartitioner, that allows each client to receive a class-balanced shard and performs an internal 80/20 split into local training and validation subsets.

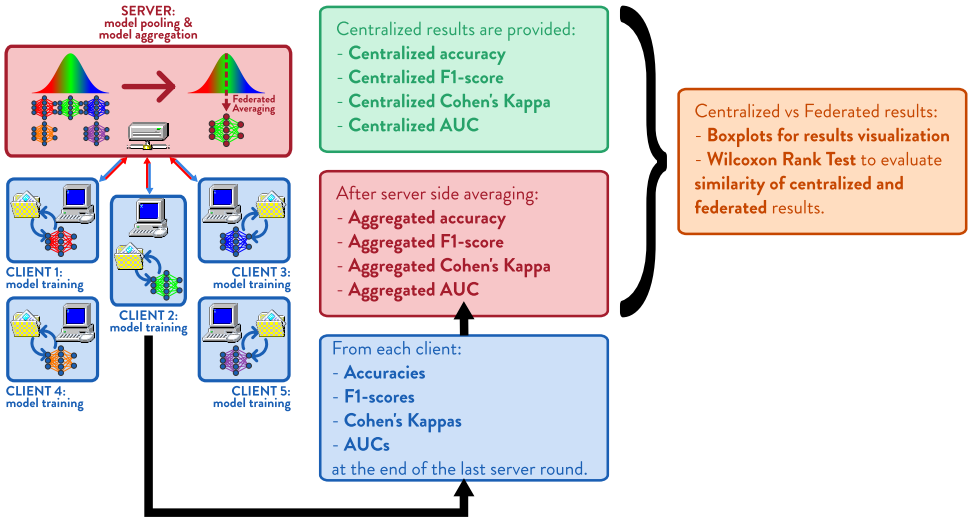

Figure 2: Diagram the setup of this study: a maximum number of clients (potentially hospitals) has been set at 5, starting from 2. Once a federated learning run has generated a global model after 20 server rounds, a new client is introduced and a new run is started. The validation metrics from 10 runs for each client number are collected, and used to check comparability with the validation metrics from 10 centralized runs with Wilcoxon Rank Testing.

3.2. Architecture

The model chosen in this study is a two-dimensional DenseNet121 (Fig. 1) with pre-trained parameters, followed by a two-way softmax head for methylation status (0 = unmethylated, 1 = methylated). To confirm comparability under identical initialization, a fixed random seed run (seed = 42) in both the centralized and federated settings were conducted to verify that their predictions match.

3.2.1. Centralized Pipeline

In the centralized configuration, all slices and labels are loaded onto a single workstation equipped with an Intel Core i9 (8 cores), 32 GB RAM, and an NVIDIA GeForce GTX 1660 with 22 GB VRAM. Training

runs for ten epochs with a batch size of 16, using the Adam optimizer (learning rate = 1×10^{-4}) and categorical cross-entropy loss. Four validation metrics are generated with every run (accuracy, F1-score, Cohen's kappa and AUC). Ten centralized runs were conducted so to have enough data to be statistically tested against the federated results.

3.2.2. Federated Pipeline

A simulation of a privacy-preserving federation was then set up using Flower 1.18.0. For each of the 20 communication rounds (chosen to assess whether comparability of the validation results could be achieved in less than the time-prohibitive 300 communication rounds set by Atef et al. [8]), each of the N clients trains locally for ten epochs under the same optimizer, learning rate, batch size, and loss settings as the centralized run. After each local training phase, client parameters updates are sent to the central server, where they are averaged via the foundational Federated Averaging (FedAvg) algorithm [1] and the aggregated global model is broadcast back for the next round (Fig. 2).

3.3. Test

The performance of the federated pipeline was evaluated against the centralized baseline by first executing ten independent centralized training runs (with unfixed randomness) and recording four validation metrics (accuracy, F1-score, Cohen's kappa, and AUC) on the 20% hold-out split. For each federated configuration, ten independent FL simulations were carried out, with the same metrics extracted at the conclusion of the 20th communication round. The client number has been increased every 10 independent FL simulations (from 2 to 5), to collect enough data points for statistical testing for every client number. For each metric, a box-plot was generated containing six distributions: one for the centralized baseline and one for each client-count configuration, illustrating the variability arising from data shuffling and client heterogeneity. Statistical comparisons between the centralized and federated distributions were made using the Wilcoxon rank-sum test (α = 0.05), treating each set of ten runs as independent samples (Fig. 2). Resulting p-values were overlaid directly onto the corresponding box-plots to highlight significant differences.

4. Results

4.1. Centralized & Federated accuracy comparison

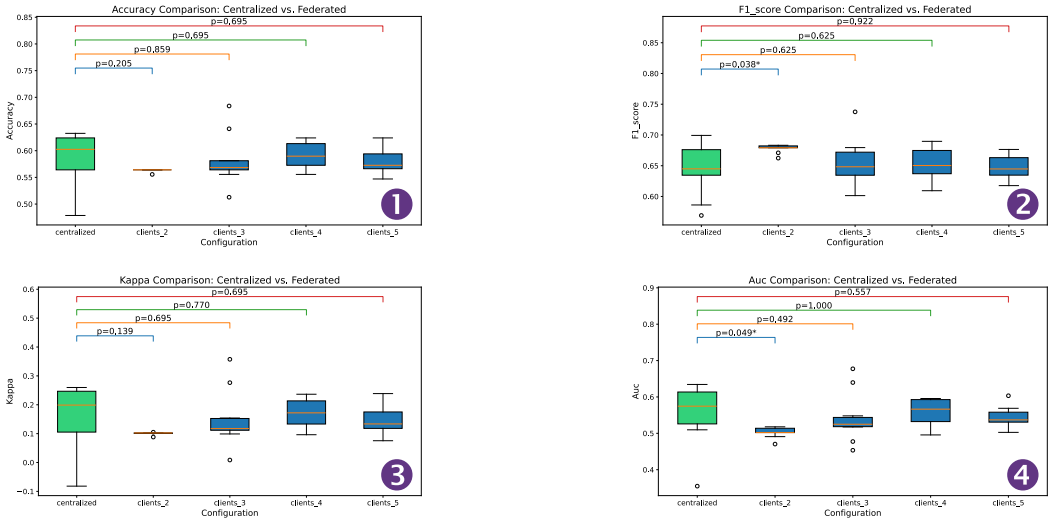

Figure 3: Boxplot of the centralized (green) accuracies (Fig. 3.1), F1-scores (Fig. 3.2), Cohen's Kappas (Fig. 3.3) and AUCs (Fig. 3.4) from 10 runs, along the boxplots from the federated (blue) 10 metrics from runs with increasing clients. The bars report the p-value resulting from the Wilcoxon Ranked Test. An asterisk after the p-value signals a significant difference ($\alpha \leq 0.05$)

5. Discussion

As previously reported, the centralized and federated paradigms in this work uses the same model architecture and deep learning configurations (e.g.: batch sizes, optimizers, etc.), but are setting less computationally intensive and less time consuming federated hyperparameters (e.g.: number of communication rounds, number of clients, etc.).

As initially envisioned, the boxplots comparing the validation metrics from the centralized case against the federated cases show that it is possible to reach comparable validation performances as most of the comparisons between the centralized case and the correspondent federated cases show no significant difference; yet notably, in the 2-client federated setting, the distribution of validation metrics displayed significantly lower variability (narrower interquartile ranges) compared to all other configurations. This resulted in statistically significant differences relative to the centralized baseline in F1-scores and AUCs (Fig. 3). A possible explanation lies in the chosen deterministic behavior of the PathologicalPartitioner used for client data splitting: with only two clients and two target classes, the deterministic policy likely produced near-identical client distributions across all ten runs. In contrast, for client counts greater than two, achieving consistent class balance deterministically becomes mathematically underconstrained, inducing greater variability in clients data composition. Furthermore, the absence of batch shuffling during validation (implemented to maintain consistency with the centralized setup) may have further contributed to the reduced variability observed in the 2-client case. Without shuffling, each client processes validation data in the same sequence across runs, leading to highly consistent evaluation conditions. This combination of deterministic partitioning and fixed validation data order likely resulted in the observed statistical significance, reflecting an artifact of reduced variability.

6. Limitations & Future Work

6.1. Computational power

Training a DenseNet121, even on single 2D slices, required substantial run time. The previously reported hardware, each 10-epoch training job took approximately 3 minutes in the centralized example, and from 5 to 15 minutes while training on two to six clients; these times are low in a single run, yet quickly scale up to hours of training when multiplied by the ten repetitions (and the number of communication rounds in FL). Scaling to full 3D volumes or larger architectures would likely exceed such resources, constraining model complexity and the number of clients that can be simulated in parallel until better configuration parameters or more powerful machines are accessible.

6.2. Flower Boilerplate

Out of the box, Flower's project templates target classification or regression tasks and include boilerplate scripts for command-line parsing, configuration values management, and client/server set-up. Adapting these templates to a segmentation network demands deeper familiarity with Flower's internals and project structure. In the reported work, refactoring the template to load custom dataset partitions, swap in our DenseNet121 backbone, and handle specialized training loops added significant development effort, but will for sure be addressed in subsequent experiments.

6.3. From simulation to actual networks

While Flower's simulation backend allowed easier orchestration and debugging of the project, it abstracts away real-world networking challenges: latency, bandwidth variability, asynchronous updates, and fault tolerance. We did not model communication delays or partial client participation, which in a real multi-center deployment could slow global model convergence. More realistic simulations will be addressed in subsequent experiments, and actual multi-computer networks will be designed once the university IT department will greenlight the endeavour (due to internal cybersecurity policies).

6.4. OMOP connection

The reported work partitions a single dataset in memory. A natural extension would be to interface Flower directly with an OMOP Common Data Model (CDM) database [13] at each site, standardizing medical vocabulary and data schemas. This would permit true cross-institutional studies, leveraging OHDSI tools for cohort definition and ensuring semantic interoperability.

Declaration on Generative AI

During the preparation of this work, the author(s) used OpenAI GPT-4 web client for: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

Acknowledgements

This work was supported by the *Operational Health Plan* funds allocated by the Italian Ministry of Health, "**CAL**abria **HUB** per **R**icerca **I**nnovativa ed **A**vanzata (CAL.HUB.RIA)" project, T4-AN.09.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y. Arcas, Communication-efficient learning of deep networks from decentralized data, arXiv [cs.LG] (2016).
[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks (2018). arXiv:1812.06127.
[3] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, A. T. Suresh, SCAFFOLD: Stochastic controlled averaging for federated learning (2019). arXiv:1910.06378.
[4] X. Li, M. Jiang, X. Zhang, M. Kamp, Q. Dou, FedBN: Federated learning on non-IID features via local batch normalization (2021). arXiv:2102.07623.
[5] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, S. Choudhary, Federated learning with personalization layers (2019). arXiv:1912.00818.
[6] G. N. Gunesli, M. Bilal, S. E. A. Raza, N. M. Rajpoot, FedDropoutAvg: Generalizable federated learning for histopathology image classification (2021). arXiv:2111.13230.
[7] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, M. Valko, Bootstrap your own latent: A new approach to self-supervised learning (2020). arXiv:2006.07733.
[8] O. Atef, M. A. Salam, H. Abdelsalam, Federated learning approach for measuring the response of brain tumors to chemotherapy, International Journal of Advanced Computer Science and Applications 13 (2022). URL: http://dx.doi.org/10.14569/IJACSA.2022.0131060. doi:10.14569/IJACSA.2022.0131060.
[9] OpenFL &x2013; Linux Foundation Project — openfl.io, https://openfl.io/, ???? [Accessed 29-05-2025].
[10] Kubernetes Authors, Kubernetes, https://github.com/kubernetes/kubernetes, 2025. Accessed: 2025-05-09.
[11] H. R. Roth, et al., Nvflare: Nvidia federated learning application runtime environment (2022). URL: https://github.com/NVIDIA/NVFlare. doi:10.5281/zenodo.6780567, accessed: 2025-05-09.
[12] Flower Authors, Flower: A friendly federated learning framework, https://github.com/adap/flower, 2025. Accessed: 2025-05-09.
[13] G. H. Lee, J. Park, J. Kim, Y. Kim, B. Choi, R. W. Park, S. Y. Rhee, S.-Y. Shin, Feasibility study of federated learning on the distributed research network of OMOP common data model, Healthc. Inform. Res. 29 (2023) 168–173.
[14] V. Bachu, Densenet 121 new, https://www.kaggle.com/code/varshithbachu/densenet-121-new, 2021. Kaggle Notebook, RSNA-MICCAI Brain Tumor Radiogenomic Classification Challenge.