Grenze International Journal of Engineering and Technology, January Issue

MedFed: A Federated Learning Framework for Medical Imaging

Swati Joshi[1], Vaishali Rajput[2] and Prajakta Musale[3]
[1-3]Vishwakarma Institute of Technology, Pune
Email: swatijoshi2@gmail.com

***Abstract*— The medical imaging data gathered by the hospitals is not utilized to the fullest for research and analysis. The reason is strict Privacy policies for patients' data. This is also limiting the potential applications of the available data. One of the Solution for the issue is Federated learning. Typically, when data availability is high but computational power is limited Federated learning concept is used, e.g. mobile environments on the contrary, ample of computational resources are available at various medical institutions but the required data is unavailable. This issue can be addressed by a federated learning framework. The proposed solution recommends to use Django and TensorFlow for building the framework which allows large-scale model training within the available hospital infrastructure yet, patient privacy is not required to be compromised. The proposed approach encourages the collaboration among medical institutions by utilizing their computational resources leading to model performance improvisation. This improved the diagnostic accuracy too. An accuracy of 83% is achieved by the global model after training for five epochs across five clients, surpassing the individual accuracy of all participating clients. Here the test data set is hosted on the server. This technique strengthens collaborative medical research as well provides a basis for broader federated learning applications where data privacy is very important.** **Keywords—Federated Learning, Data Privacy, data availability, medical imaging.**

I. INTRODUCTION

All the domains - including healthcare are being notably influenced by the galloping advancements in artificial intelligence (AI) and machine learning (ML). This is exhibiting considerable enhancement in diagnostic precision and improving patient outcomes. Health care domain when Integrated with AI and ML experience many technical challenges. One of the major issues is the effective utilization of medical imaging data, which is restricted due to strict privacy policies for disclosing patient data outside the medical institutions.

Voluminous medical data is accumulated by Medical facilities, including hospitals and pathology centers, x ray clinics and still it grows exponentially. However, the use of this data beyond institutional boundaries is strictly controlled. The Health Insurance Portability and Accountability Act (HIPAA) enforces strict compliance to protect sensitive patient's data in USA. Likewise, in India, according to the Right to Information Act protects patient's medical documents. Patient's signed Permission that is consent is required to access the medical data. Digital Information Security in Healthcare Act (DISHA) initiative further strengthens data access by the third party. This imposes a significant barrier for obtaining and using medical data to train conventional machine learning models.

Grenze ID: 01.GIJET.12.1.32

To overcome above said constraints, Federated learning is very helpful. It preserves the data privacy and yet allows collaborative model training across multiple institutions. The approach is totally decentralized in contrast to the conventional machine learning methodologies which needs centralized data aggregation. Federated learning permits individual hospitals and medical institutions to retain the control over their datasets. Federated learning model updates are shared to the local servers of the organizations. Patients' data is not at all migrated which ensures the compliance with privacy policies like Digital health or DISHA regulations. Federated learning is typically used widely in mobile device applications, where data is easily available but computational power is limited, its application in healthcare. Federated learning can effectively address this issue .

The system presented here proposes a federated learning framework especially designed for medical imaging applications. To mitigate challenges related with data sharing and privacy , the proposed framework uses the decentralized nature of federated learning. Here, a detailed discussion of the system's design and implementation is provided. The paper gives description of the system to explain how the system facilitates collaborative model training across medical institutions yet preserving patient confidentiality. In addition, the framework's performance evaluation is presented, which shows improved diagnostic accuracy. Experimental results also indicate that the federated learning model obtains a global accuracy of 85% on a test dataset, surpassing the performance of individual models.

The transformative potential of federated learning within the health sector is highlighted by the experimental findings. The collaborative approach for utilization of medical data supports advancements in diagnostic methodologies and patient care. The proposed framework provides the foundation for future research and more real-world applications. This may lead to the development of more effective AI-driven healthcare solutions and accurate medical outcomes on a global scale.

II. Literature Review

Federated learning is an iterative process in which as a first step local servers at the institute download the pretrained shared prediction model . Initially this shared prediction model is trained on the proxy data. Then using data available on the local server/device the model does new learning and modifies itself. With those modifications updates are sent to the cloud. On the cloud modifications done by all sites are summarized and broadcasted to all the sites. This process is repeated until the high quality model is developed.

Gboard is the pioneer application which has started using federated learning (FL) [1] at Google, Using raw data as typing content from the multiple user's mobile devices collaboratively, the model was trained and now has been deployed on millions of Devices. On device training model uses light weight version of the TensorFlow. Gboard uses variety of differential Privacy preserving and enhancing techniques .Trusted Execution environments (TEEs) is playing major role in design of Next generation Gen AI models.

Significant studies on federated learning and its applications in medicine and healthcare have been done by the scientific research community [2-6]. Most of the recent research papers elaborate the applications of federated learning in medical imaging [7-8]. Various aspects like Challenges in medical imaging, Privacy protection in medical imaging, Federated learning problem formulation and its processing steps and available frameworks and datasets are discussed in those papers [8].

Privacy protection is a major concern in this digital Era. Selling or misusing patient data may cause unauthorised data leakage and may have impact on patients' lives. To address these concerns, Numerous legislations to protect patients' rights to digital health privacy, such as HIPAA in the U.S. and DISHA in India , make it difficult to obtain large-scale medical imaging data. Also, such data collection and labelling needs specific expertise and hence becomes expensive too. In addition, when it comes to real-life data rather than curated datasets, distribution methods differ causing it difficult to train robust models.

Federated learning offers a solution to these challenges. As described in [8], the federated learning problem is as follows:

"Suppose there are N independent clients (sites) with their own datasets. Each of the clients (sites) cannot access others' datasets. Federated learning (FL) aims to collaboratively train a machine learning model by gathering information from these N clients (sites) without exchanging or sharing their raw data. The ultimate output of FL is the learned model, which is broadcast to each client for deployment. The generalizability of should outperform each local model (typically with the same model architecture as) trained through local training [8]. "

The typical federated learning process involves:

- Client Selection: Identifying potential clients with data and labels that can contribute to the model's success.
- Client Initialization: Creating a local model structure for clients based on the server's aggregation policy and client data.

- Local Training: Training models locally at each client site.
- Model Upload: Sending model parameters such as weights, biases, and model structures to the server.
- Aggregation: Aggregating the uploaded models from clients using an aggregation policy to ensure fair and effective combinations, resulting in an updated global model.
- Broadcast: Broadcasting the updated global model to all clients.
- Iteration and Convergence: Repeating steps 3 to 7 until required performance metrics are met, followed by model deployment.

Federated learning model expects that medical imaging is similarly distributed. But practically this is not possible since, the data at one site (Clinic/institute) is not the same in nature or quality from data at another as the data at another site. This is called as cross-site data distribution. This variance is caused dur to different brands image scanners , different resolutions of the images and scanning protocols. This diversity is referred thee "domain shift" problem [12] or "client shift" in FL systems. The cross site data distribution, client shift and domain shift may negatively affect model accuracy, stability and reliability. For example, even if two hospitals are scanning the same organ (say, lungs), the resulting images can look quite different. This can affect the performance of the FL Model. Techniques such as Domain adaptation can be used to resolve these differences, ultimately having better model convergence and accuracy in FL systems. But if the global model is trained on the variegated data , it may affect local site model's Accuracy.

Handling client shift by locally fine-tuning the global model using client data. This requires Domain-Specific Learning Known as personalized FL, Using this method, model can accommodate each client's specific data properties. Feng et al. [17] suggested an encoder-decoder structure for magnetic resonance (MR) image reconstruction within an FL framework. This requires maintaining a shared encoder on the server to learn variety of representations of the images. and client-specific decoders can be executed locally. Similarly, Chakravarty et al. [8] suggested a combination of Convolutional Neural Network (CNN) and a Graph Neural Network (GNN) for chest X-ray classification, CNN weights globally shared and does fine-tuning local GNNs per client . Xu et al. [13] introduced an ensemble-based framework comprising a global model, personalized models, and a model selector to address client shift in medical image segmentation. Jiang et al. [14] proposed training locally adapted models by combining global and local gradients to optimize client-specific performance. For histopathological image harmonization, Ke et al. [15] developed an FL framework using a Generative Adversarial Network (GAN), where local discriminators capture client-specific styles, and a global generator produces domain-invariant images. Wagner et al. [16] presented a similar GAN-based method but assumed access to a reference dataset for local GAN training.

Domain Adaptation is another strategy to mitigate client data distribution differences. Li et al. [18] aligned functional MRI data distributions using domain adaptation, adding noise for privacy protection and training domain discriminators to reduce shifts. Dinsdale et al. [19] addressed scanner-induced domain shifts using a framework that assumes Gaussian-distributed medical image features with shared means and standard deviations among clients. Andreux et al. [20] employed group normalization in deep neural networks to deal with client shifts in histopathology datasets. Guo et al. [21] used intermediate latent feature alignment for MRI reconstruction, harmonizing distributions across clients using a reference site.

Image Harmonization employs image-to-image transformation models to standardize diagnostic images from different sites. Qu et al. [22] utilized a generative replay strategy with variational autoencoders to generate images resembling input data, allowing clients to train local classifiers with both real and synthesized data. Yan et al. [23] used cycleGAN [24] to harmonize medical images by aligning them to a reference site's style, reducing client shift. Jiang et al. [25] proposed frequency-based harmonization by transforming images into the frequency domain, sharing and normalizing amplitude components while retaining local phase data.

On the server side, domain shift among clients can lead to biased global models with uneven performance. Hosseini et al. [26] addressed this issue with a novel optimization technique inspired by balanced resource distribution ensuring uniform model performance across clients by weighting clients with inferior performance more heavily in the loss function. Fan et al. [27] proposed a guided-gradient optimization method, using only positive aggregated weight values to guide global gradient descent. Luo et al. [28] introduced federated learning with shared label distribution (FedSLD), which mitigates label distribution differences among clients by employing a weighted cross-entropy loss computed based on batch and global label distributions during local training.

Federated learning platforms and frameworks have also emerged. Among the most influential are Flower and TensorFlow Federated (TFF). Much of the research on federated learning focuses on minimizing the computational resources required to access vast data from mobile devices [26]. However, this is not the case in hospitals, where data is limited, but computational power is ample. Our solution to these problems ensures domain shift is handled effectively, enabling researchers to train large models with traditional frameworks like PyTorch and TensorFlow.

III. METHODOLOGY

The proposed approach delineates the architectural framework and operational mechanisms of a federated learning system implemented through Django applications, where edge devices function as clients and a central system operates as the server.

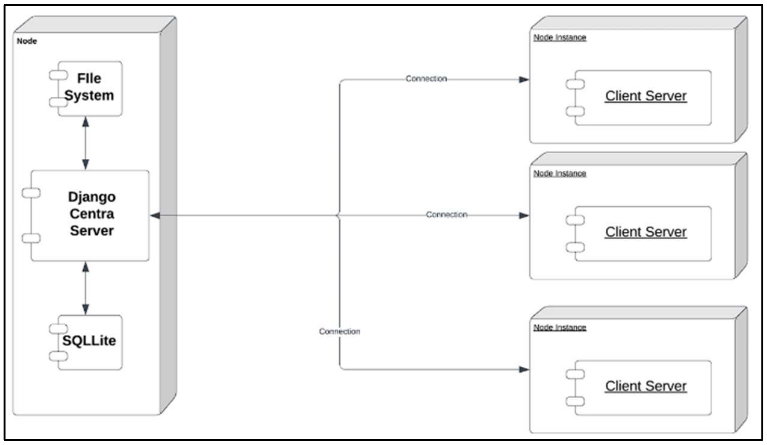

Figure 1 Deployment Diagram

A. System Architecture

The system consists of two primary components: the Client and the Server. The Client functions as a synchronous application responsible for model training, whereas the Server operates asynchronously to oversee the overall workflow. Multiple clients are capable of connecting to a single server, as illustrated in the deployment diagram (Figure 1).

Client application Design

The Client application is structured using an event-driven architecture, leveraging Django signals to initiate events upon the completion of each training step. It stores image data along with corresponding labels, which undergo preprocessing and are efficiently managed. To minimize loading times for recurrent training sessions, images are stored in the TFRecord format, enabling rapid access and significantly reducing loading times from minutes to seconds.

Upon successful initialization and preprocessing, clients transmit a registration request to the server, signalling their availability for participation in federated learning rounds. Once registered, clients execute model training based on global model parameters received from the server and subsequently transmit updated model parameters in .h5 file format. The server aggregates these parameters and returns an updated global model to the clients. This iterative process continues until either the target accuracy is reached or the predefined number of federated learning rounds are completed.

Server application Design

The Server application is implemented as an asynchronous system responsible for handling client registration requests, which contain crucial details such as IP address, port number, and unique identifiers. This information is stored within an SQLite database on the server (depicted in Figure 2).

To facilitate client-server communication, dynamically generated URLs adhere to the following format:

https://{clientip}:{clientport}/{function_path}

The server encompasses several core functions that define its operational framework:

- create_model() – Defines the architecture of the model intended for client-side training. An empty model file is dispatched to clients for this purpose.
- Round_manager(epochs, rounds) – Specifies the number of epochs for client training and determines the number of federated learning rounds.
- select_clients() – Establishes criteria for selecting clients to participate in each training round.

```
Function RoundManager(request):
   selected_clients = startRound()   // Start training
round, get clients who accepted

   For each epoch in request data:
      total_clients = count of selected_clients

      Try:
         Wait for all client files with a timeout of 50
seconds
      Catch TimeoutError:
         Print "Timed out. Not all clients sent their files."

      broadcast_path = aggregate client files
      responses = broadcast aggregated files to
selected_clients
      Print responses
```

Figure 2 Round Manager Function

The server architecture ensures that system operators interact solely with these functions, managing training rounds through a dedicated interface. An illustrative implementation of the round manager function is presented in Figure 2.

The outlined approach effectively defines the interactions and roles of client and server components within a federated learning framework utilizing Django applications. By integrating event-driven mechanisms and employing optimized data storage formats such as TFRecord, the system enhances training efficiency while maintaining centralized control at the server level. The iterative nature of model training and aggregation facilitates continuous refinement until the desired performance metrics are attained.

TABLE I: DESCRIPTION OF SEQUENCE DIAGRAM

Message	Type, Payload	From	To
Register request with client info	Synchronous, Carries client data like Ip address, port number and unique registration name	Clients	Server
Start Training	Asynchronous, Architecture of model to be trained in current round, in form of .h5 file.	Server	Clients
Model Training Finished	Synchronous, Updated Client model file in .h5 format.	Clients	Server
Aggregated model	Asynchronous, Aggregated server model in .h5 file format	Server	Clients

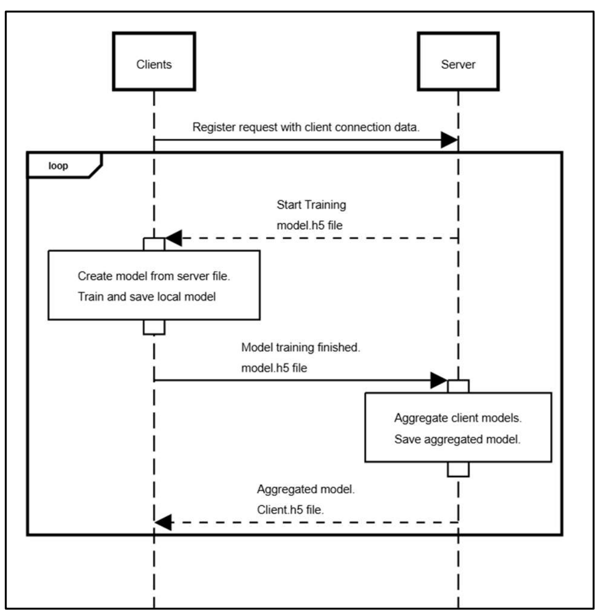

Figure 3 Sequence Diagram Communication

Sequence diagram (Figure 3) shows flow of communication between client and server. The messages are described in (Table 1).
The proposed approach effectively defines the roles and interactions between the client and server components within a federated learning framework implemented using Django applications. By utilizing event-driven mechanisms and optimized data storage formats such as .tfrecord, the system enhances model training efficiency while ensuring centralized control at the server. The iterative nature of model training and parameter aggregation facilitates continuous refinement, enabling the achievement of optimal performance metrics.

IV. IMPLEMENTATION

To assess the efficacy of the proposed approach, experiments were conducted utilizing the Breast Histopathology dataset available on Kaggle. This dataset comprises 162 whole-mount slide images of Breast Cancer (BCa) specimens, scanned at a magnification of 40x. From these whole-slide images, a total of 277,524 image patches, each measuring 50 × 50 pixels, were extracted. Among these, 198,738 patches were identified as IDC-negative, while 78,786 patches were labeled as IDC-positive [34]. The dataset adheres to a structured naming convention, where each patch's filename follows the format:

u_xX_yY_classC.png

For instance, a file named 10253_idx5_x1351_y1101_class0.png encodes the following details:

- 10253_idx5: Patient identification number
- x1351: X-coordinate of the patch within the whole-slide image
- y1101: Y-coordinate of the patch within the whole-slide image
- class0: IDC classification, where 0 denotes non-IDC and 1 indicates IDC presence

A visual examination of the dataset highlights significant colour variations between IDC-positive and IDC-negative samples. To create a non-IID (non-Independent and symmetrically Distributed) data distribution, the dataset was partitioned as demonstrated in Figure 4.
Client applications must load the image patches and convert them into .tfrecord format at initialization or upon the introduction of new data into the system. This transformation is facilitated through an event-driven architecture leveraging Django signals. The workflow proceeds as follows:

- Data Ingestion: Upon system startup, image patches are retrieved from storage.
- Conversion to .tfrecord: The dataset is serialized into the .tfrecord format to enhance storage efficiency and processing speed.
- Database Update: Upon completion of the transformation, metadata is updated within the database to maintain an index of available samples.
- Event-Based Processing: Django signals trigger updates upon the addition of new data. However, given the static nature of the experimental setup, this process occurs exclusively at startup.

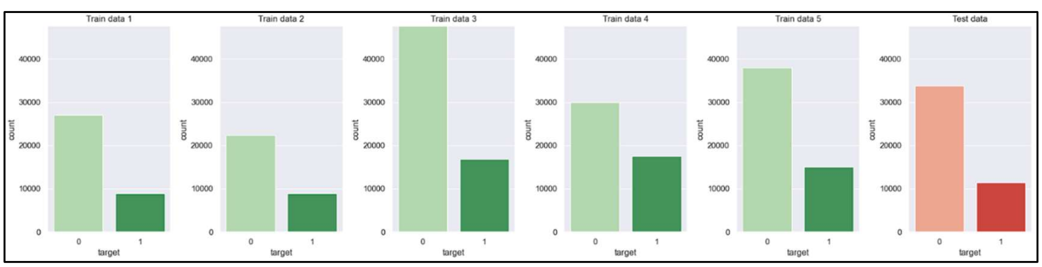

Figure 4 Non-IId Data Distribution

Upon data preparation, client applications initiate a registration request to the central server. This registration process establishes a secure communication channel between the client and the server, facilitating effective dataset management and ensuring seamless coordination for model training. Upon request acknowledgment, the server permits federated learning-based model training to be adopted, which keeps the data in a decentralized structure to protect the user privacy and reduce the risk of data disclosure.
While we are focusing on a static dataset in this study, future work can be done to include the real-time data acquisition systems supporting the continuous learning and updating of the model. Furthermore, additional optimizations could involve advanced data augmentation techniques, enhanced federated averaging strategies, and domain adaptation methods to improve model robustness and generalizability.

The proposed approach establishes a structured framework for managing histopathology image data within a distributed environment. By utilizing Django's event-driven architecture alongside the .tf record format, the system optimizes data handling, ensuring scalability and privacy preservation in machine learning applications for medical imaging.

V. Results

In this study, we evaluated the performance of our System. The system was trained across five rounds, and we measured accuracy and loss metrics at each step to assess model performance. Below, we present a summary of our findings.

The federated learning model demonstrated consistent improvements in accuracy across the training rounds. In Round 1, the model achieved an initial test accuracy of 0.2581. However, as training progressed, accuracy significantly improved, reaching 0.8296 in Round 5. The corresponding loss values also decreased over time, indicating improved convergence of the model.

VI. Round-Wise Global Model Performance

For the global model test accuracy was: 0.2581 and 0.8336 in the 1st and 2nd rounds respectively. This indicated that the early rounds effectively performed the exchange in weights and model adaption. After Round 3, precision was basically plateaued, with slight fluctuations around 0.83, suggesting convergence. Similarly, the loss metric exhibited a significant drop between Round 1 and 2, and then it kept the same low level among later rounds. This tendency indicates that the proposed model learned well from distributed data sources and showed the stable performance. The global models derived from the single datasets were of mixed quality, with accuracy ranging from 0.7253 to 0.8402 across rounds. The gain in accuracy there was in turn well correlated with the overall model's error reduction, underlining the validity of federated training in boosting collective model performance.

The experimental results in Figure 5, shows that, federated learning achieves good knowledge aggregation across distributed sources that in turn results in a well-generalized global model.

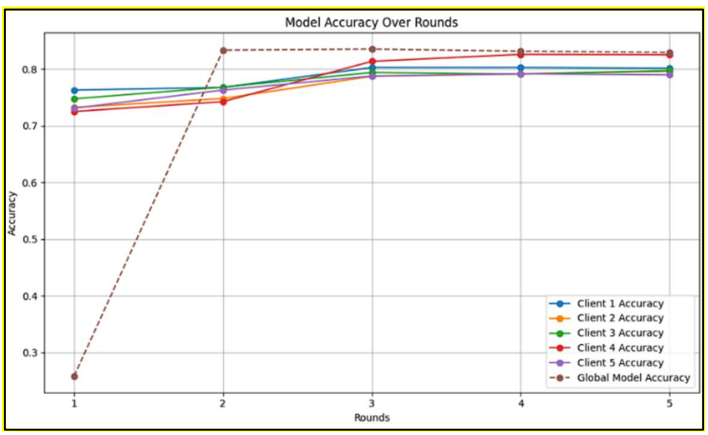

Figure 5 Round-wise Accuracy

In the early rounds, rapid improving of the accuracy is observed; and on the later ones, the convergence maintains stable, demonstrating the efficiency of MPHVAE. These results indicate the potential of federated learning in high-performance model training with data privacy in decentralized data.

VII. Conclusion

Our findings suggest that using large models in federated learning can be very effective given the large performance boosts we achieved in the early rounds of training. The global model quickly converged and this rapid convergence indicates that federated learning is an effective learning method for the decentralized data to maintain privacy.

One of the interesting observation we make from our implementation is that fix epoch-1 for each client model actually benefited the model’s overall performance. It has assisted in enhanced client learning with lesser overfitting at client level. Moreover, our results indicate that federated learning can achieve a similar performance to centralized models, while leveraging decentralized training.

Our results also demonstrate the significance of stability in federated learning. The model converged after an initial increase in accuracy. This behavior is consistent with the idea that federated learning systems should allow themselves to be content with slow and steady, not overfitting local model updates that might introduce variance.

In conclusion, our findings corroborate the applicability of federated learning to large-scale models and the preservation of privacy by the mechanism. With the adoption of a well-designed training strategy, we were able to reach high accuracy models on the original federated ABDL benchmark dataset without centralized data pooling, which again proof the feasibility of federated learning in practical systems.

FUTURE SCOPE

In the future, we can further develop the federated learning framework to strengthen the security, efficiency, and scalability. Enforcing stronger and secure aggregation policies (e.g., differential privacy and secure multi-party computation) we can avoid client data leaks while preserving model accuracy. Furthermore, networking security measures such as encrypted communication channels and authentication protocols will be vital for defense against adversarial attacks and to enforce data integrity.

To maximize resource, compression algorithms can be adopted to cut down on communication overhead, which reduces bandwidth consumption while preserving the model accuracy. Quantization and scarification are effective methods to improve training efficiency, which can make federated learning more practical for deployment in large scale.

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” *arXiv preprint*, arXiv:1602.05629, 2023.

[2] R. S. Antunes, C. A. da Costa, A. Küberle, I. A. Yari, and B. Eskofier, “Federated learning for healthcare: Systematic review and architecture proposal,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1–23, 2022.

[3] S. Rajendran et al., “Cloud-based federated learning implementation across medical centers,” *JCO Clinical Cancer Informatics*, vol. 5, pp. 1–11, 2021.

[4] D. C. Nguyen et al., “Federated learning for smart healthcare: A survey,” *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–37, 2022.

[5] B. Pfitzner, N. Steckhan, and B. Arnrich, “Federated learning in a medical context: A systematic literature review,” *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 2, pp. 1–31, 2021.

[6] N. Rieke et al., “The future of digital health with federated learning,” *NPJ Digital Medicine*, vol. 3, no. 1, p. 119, 2020.

[7] M. F. Sohan and A. Basalamah, “A systematic review on federated learning in medical image analysis,” *IEEE Access*, vol. 11, pp. 28628–28644, 2023.

[8] H. Guan, P.-T. Yap, A. Boozki, and M. Liu, “Federated learning for medical image analysis: A survey,” *arXiv preprint*, arXiv:2306.05980, 2024.

[9] **California Consumer Privacy Act (CCPA)**, CCPA, 2018. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[10] **General Data Protection Regulation (GDPR)**, GDPR, 2019. [Online]. Available: https://gdpr-info.eu/

[11] **Right to Information (RTI)**, RTI, [Online]. Available: https://rti.gov.in/

[12] H. Guan and M. Liu, “Domain adaptation for medical image analysis: A survey,” *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 3, pp. 1173–1185, 2022.

[13] A. Xu et al., “Closing the generalization gap of cross-silo federated medical image segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20866–20875.

[14] M. Jiang, H. Yang, C. Cheng, and Q. Dou, “IOP-FL: Inside-outside personalization for federated medical image segmentation,” *IEEE Transactions on Medical Imaging*, vol. 42, no. 7, 2023.

[15] J. Ke, Y. Shen, and Y. Lu, “Style normalization in histology with federated learning,” in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2021, pp. 953–956.

[16] N. Wagner et al., “Federated stain normalization for computational pathology,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2022, pp. 14–23.

[17] C.-M. Feng et al., “Specificity-preserving federated learning for MR image reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 42, no. 7, 2023.

[18] X. Li et al., “Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results,” *Medical Image Analysis*, vol. 65, p. 1–14, 2020.

[19] N. K. Dinsdale, M. Jenkinson, and A. I. Namburete, "Fedharmony: Unlearning scanner bias with distributed data," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2022, pp. 695–704.
[20] M. Andreux et al., "Siloed federated learning for multi-centric histopathology datasets," in *MICCAI Workshop on Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning (DART)*, Lima, Peru, 2020, Springer, 2020, pp. 129–139.
[21] P. Guo et al., "Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2423–2432.
[22] L. Qu et al., "Handling data heterogeneity with generative replay in collaborative learning for medical imaging," *Medical Image Analysis*, vol. 78, p. 102424, 2022.
[23] Z. Yan et al., "Variation-aware federated learning with multi-source decentralized medical image data," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 7, pp. 2615–2628, 2020.
[24] J.-Y. Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.
[25] M. Jiang et al., "Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 1087
[26] Hosseini, S. Maryam et al. "Layer-wise Federated Learning for Mobile Networks." 2024 IEEE Middle East Conference on Communications and Networking (MECOM) (2024): 229-234.
[27] Li Yuxi Fan, Mike Tse, Kuo-Yi Lin, "A review of applications in federated learning", Computers & Industrial Engineering, Volume 149, pp. 106854, 2020
[28] Luo J, Wu S. "FEDSLD: federated learning with shared label distribution for medical image classification". Proc IEEE Int Symp Biomed Imaging. 2022 Mar;2022:10.1109/isbi52829.2022.9761404. doi: 10.1109/isbi52829.2022.9761404. Epub 2022 Apr 26. PMID: 37818224; PMCID: PMC10563702.