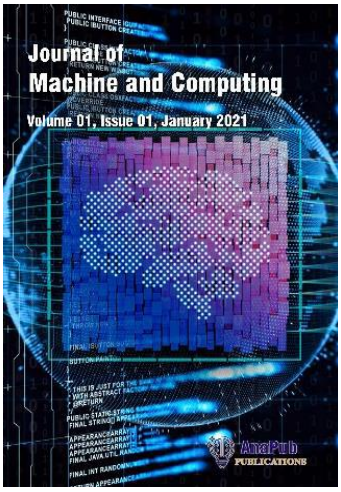
Journal Pre-proof

FedAvgCNN: A Fusion-Based Federated Learning Approach for Multi-Class Brain Tumor Classification with Enhanced Privacy



Sivakumar N, Renukadevi S, Manujakshi B C and Shashidhar T M

DOI: 10.53759/7669/jmc202505190
Reference: JMC202505190
Journal: Journal of Machine and Computing.

Received 26 April 2025
Revised from 16 June 2025
Accepted 02 August 2025

Please cite this article as: Sivakumar N, Renukadevi S, Manujakshi B C and Shashidhar T M, "FedAvgCNN: A Fusion-Based Federated Learning Approach for Multi-Class Brain Tumor Classification with Enhanced Privacy", Journal of Machine and Computing. (2025). Doi: https://doi.org/10.53759/7669/jmc202505190.

AnaPub
PUBLICATIONS

FedAvgCNN: A Fusion-Based Federated Learning Approach for Multi-Class Brain Tumor Classification with Enhanced Privacy

N.Sivakumar [a], Renukadevi S [b], Manujakshi B C [c], Shashidhar T M [d]

[a] Department of Computer Engineering, Marwadi University, Rajkot, India.
[b] School of Computer Science and IT, Jain University, Bangalore, India.
[c] School of Computer Science and Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be-University), Bengaluru, India.
[d] Department of Electronics and Communication Engineering, Harsha Institute of Technology, Bengaluru, India.
[a] drsivakumar.nadarajan@gmail.com, [b] renuka.devi@jainuniversity.ac.in, [c]manujakshibc@gmail.com
[d] shashilara@gmail.com

Abstract:

Brain Tumor (BT) leads to disability in cognitive, motor, and social skills, and therefore, early diagnosis should be a milestone for treatment. In this work, a novel Federated Learning-based Convolutional Neural Network (FL-CNN) model is proposed for Brain Tumor Classification (BTC) with FL serving as the framework for the model. The model is trained to distinguish between four classes of brain conditions: glioma, meningioma, pituitary adenoma, and non-neoplastic growth. Through the use of Federated Learning (FL), this method allows multiple Decentralized clients to cooperate in training the model without exchanging the raw medical data belonging to the patients. The provided dataset is derived from a training set containing 5712 images and a testing set containing 1311 images, and both sets are labeled among four categories. The fully trained 2D-CNN model deals with pre-processed MRI images in dimensions of 128×128 pixels and internalizes key attributes for identifying all forms of BT. As for understanding the model's performance, we compute accuracy, precision, recall, and F1-score. The model achieved a peak validation accuracy of 97.48% with a precision, recall, and F1 score of 97.48%**.** Early stopping was applied at round 12 due to performance stagnation, preventing overfitting. The final global accuracy reached 97.48%, with a loss of 0.1483, demonstrating strong classification performance. The results exhibit that the federated strategy yields comparable classification accuracy with the conventional approach for distributed data and minimizes the violation of individual data privacy. Moreover, this work discusses the applicability of FL to medical image analysis, indicating that collaborative models in this area can provide a highly accurate performance while avoiding data aggregation. The following paper is intended to contribute to the improvement of privacy-preserving ML pertaining to medical diagnosis with regard to BTs.

Keywords: FL, Brain Tumor Classification, Privacy-Preserving ML, Medical Image Analysis, Decentralized Learning, Healthcare ML CNN.

1. Introduction

One of the toughest challenges in medicine is that BTs are among the most diverse and challenging diseases to diagnose and treat because they are located in one of the most sensitive regions of the human body. Depending on their nature, BTs are classified into benign and malignant, but gliomas are the most frequent and deadly form of the latter. Among all gliomas, glioblastoma multiforme (GBM) is is regarded as a high-grade glioma; therefore, the prognosis is bleak, with the median survival time often less than 15 months even with comprehensive treatment, including surgery, radiation therapy, and chemotherapy [16]. Essential for proper management and treatment, the distinction of primary and secondary brain tumours is frequently challenging due to the current limitations of MRI scans [13]. New molecular and immunohistochemical markers have shown an increased understanding of tumor behavior, although incorporating them into clinical practice is costly and time-consuming [11]. Consequently, it is important to adopt sophisticated computational approaches, mainly ML, in boosting diagnosis precision and developing individualised treatment strategies [3].

Machine learning (ML), especially for the CNN model, has revealed that the detection and classification of BTs from MRI scans can be effectively automated. Some of these models can effectively process an enormous volume of image data, determining tumor areas and subtypes with surgical precision [21]. However, one of the critical issues that has emerged in the design and training of effective ML is a lack of high-quality and diverse data to support its generalization to multiple patients. To obtain such datasets is challenging in the medical domain because of privacy or legal constraints and the scattered nature of healthcare organizations [1].

1.1. Role of FL in BTC:

FL, in particular, seems to have the potential to alleviate the problem of working with immense volumes of significant variability and heterogeneity while still respecting users' privacy. In the FL, as shown in Figure 1, the institutions do not actually transfer patient information [2][5]. Every institution stores its results on a local server but shares only the model parameters with the server, while protecting the identity of the patient's medical history. This strategy employs multiple sources of data across different institutions, fostering multi-institutional collaborations in the BT research and enabling the generation of better and more generalizable models [6][7].

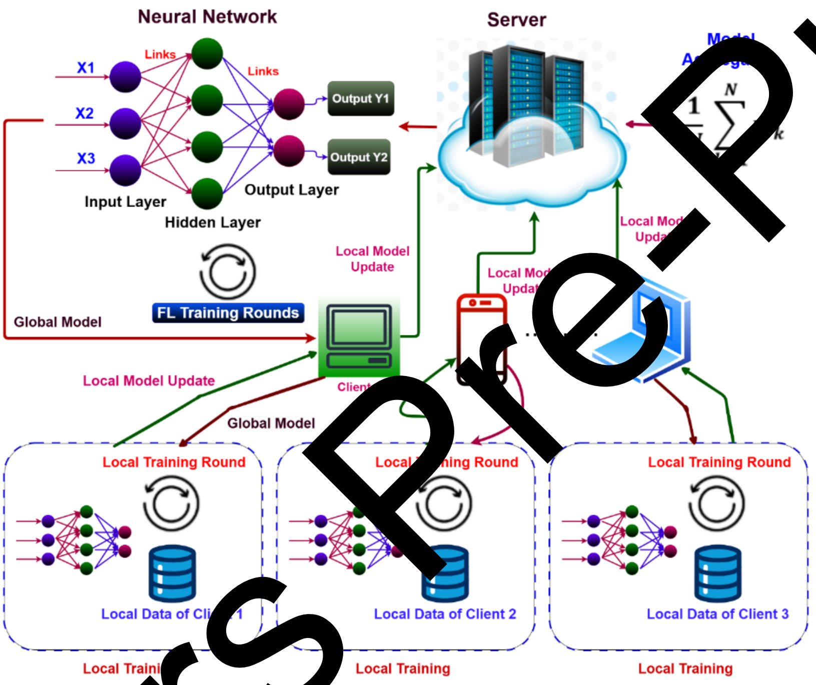


Figure 1: FL process overview.

As a result of FL, BTC serves an essential contribution to the training of ML models on distributed datasets without the violation of patient privacy. They resolve issues of data deficiency and confidentiality that have long plagued the creation of efficient ML solutions in healthcare to provide new possibilities for individualized approaches to patient management and better outcomes.

1.2. Objectives

- Develop an FL Model: Build a CNN-based FL model for BTC.
- Classify Multiple BT Types: Accurately identify glioma, meningioma, pituitary tumor, and no tumor.
- Enhance Patient Privacy: Use decentralized training to protect patient identity by excluding raw image data from direct sharing.
- Demonstrate FL in Medical Applications: Show that FL and AI can be safely and effectively used for BT detection.

1.3. Contributions

- Innovative Use of FL: Introduces FL with CNNs for medical imaging, enabling secure collaboration in healthcare.
- Robust Dataset Utilization: Uses a well-structured dataset (5,707 training and 1,311 testing images) distributed across four tumor types.

- Performance Evaluation: Establishes benchmarks for FL-based CNN models in medical diagnostics.
- Privacy-Preserving ML: Demonstrates that high-accuracy models can be trained while maintaining patient data privacy.
- Real-World Medical Impact: Highlights the potential of FL to improve early BT detection in clinical practice.
- Future Research Directions: Suggests combining FL with other ML techniques to enhance accuracy and expand its application in medical imaging.

2. Literature survey:

The inclusion of FL in the diagnosis and classification of BTs has been included as a new approach to medical imaging because of the improvements that FL brings, such as data privacy and incorporation of distributed datasets. This paper reviews the literature with different research papers that work on the detection and segmentation of BTs using the FL methodologies. As such, FL helps numerous organizations to jointly build machine models of learning without disclosing anyone's identity. This is particularly highly relevant to healthcare applications, as the importance of patients' privacy cannot be overestimated. In their study, Sheller et al. emphasize that FL can be beneficial for multi-institutional settings, as the establishment of models trained on a more extensive data set can increase the accuracy of a medical diagnosis. Furthermore, Isik-Polat also shows that FL can attain a similar performance as the centralized approach in the test of BT segmentation, indicating that FL can work well for any dataset [1].

Specifically, detection of BT is challenging due to the size, shape, and location of the tumor, hence the need for efficient ML. Deep learning and transfer learning-based approaches are clearly explained by Amin et al., and they also made a considerable effort to classify the methodologies used in BT detection in general. The current survey also provides a preliminary background and overview of the difficulties associated with BT diagnostics and the role of FL in them. In addition, Aggarwal also presents a work on a transfer learning model with an FL framework that keeps data privacy while classifying brain tumors from heterogeneously distributed data, pointing out the application of FL in the clinic [2], [3]. The use of FL is also evidenced within the context of medical imaging by other investigations done to compare the performance of the federated and centralized learning frameworks. Thus, based on the flags raised by Denissen et al., the authors report that FL can achieve similar or equivalent accuracy to a centralized model in tumor segmentation and further strengthen the feasibility of FL in clinical research. Further, Mahlool and Abed use the concept of CNN under the federated environment for the diagnosis of BTs, as explained by Hsu and colleagues, the potential of deep learning in conjunction with FL [4], [5].

Privacy threats in healthcare informatics practice are tackled via the application of differential privacy approaches in an FL environment. Li and co-authors also investigate the relationship between the accuracy of diagnostic models and the protection of patients' information in BT segmentation tasks. This is the same as what Atef et al. emphasize, that privileged data such as healthcare information is sensitive and that FL has to be used to address the risks involved [6], [7]. In several fields of BT management, FL has been shown to have a wide range of applicability: segmentation, classification, as well as assessing the response of tumors to therapy. The FeTS challenge described by Pati reflects one of the initiatives to impose some degree of unity into the FL endeavors based on the tumor segmentation issues raising data privacy and regulatory problems [7], [8]. Newer improvements in model structure, for instance, involutional neural networks, have been postulated to improve on the efficiency of BTC with little computational need. It stimulates a current concern about enhancing deep learning models in the medical field, as Zhang et al., who propose cyclic model pre-training techniques as a solution to increasing FL efficiency [9], [10].

Most common and fatal among these are gliomas, and the molecular profiling of the tumors has taken centrality for their treatment. Richterová et al. describe the importance of molecular and immunohistochemical diagnostic criteria for different brain tumours, including gliomas and meningiomas. These markers could suggest particular treatments like EGFR and VEGFR that are significant to improve care [11].

Table 1: Comparison of BT Classification Models and Their Performance

Ref No	Authors	Model	Advantages	Disadvantages	Accuracy	Year
[12]	Jemimma et al.	WCSO-DBN	Optimized deep belief network for classification	High training time due to DBN complexity	92.30%	2022

[13]	Rammurthy et al.	WHHO-based DeepCNN	Whale-Harris Hawks optimization enhances detection	Lower accuracy compared to other deep learning models	81.60%	2022
[14]	Vankdothu et al.	RCNN	Improved segmentation using IKMC, high accuracy	High computational cost	95.17%	2022
[15]	Pranjal Agrawal et al.	CNN + 3D-UNet	Automated segmentation, deep learning framework	Requires high computational resources	90%	2022
[16]	Islam et al.	FL	Privacy-preserving, robust to distributed data	Slight accuracy drop	91.05%	2023
[17]	Kumar et al.	Deep Q-network	Efficient Feature Extraction	High Computational Cost	[illegible]5.40%	2022
[18]	S. Hossain et al.	IVX16	High accuracy (96.94%) with the proposed model (IVX16).	The dataset size is relatively small for deep learning models (3264 images).	96.94%	2024
[19]	S. Das et al.	CNN	Achieved high accuracy (94.39%) and satisfactory performance.	It may require further generalization for other types of tumors or larger datasets.	94.39%	2019
[20]	Abiwinanda N et al.	Custom CNN	High Training Accuracy	Lower Validation Accuracy	84.19%	2018
[21]	S. Bhadauriya et al.	CNN + FL	Privacy-Preserving	Requires High Computational Resources	96%	2023
[22]	Deepa et al.	CJHBA-based DRN	Hybrid optimization improves accuracy	Increased complexity in model implementation	92.10%	2023

3. Problem Statement

BTs are life-threatening, requiring early and accurate detection for effective treatment. Traditional methods rely on centralized data collection, raising concerns about patient privacy and limiting access to diverse medical data. Key challenges include:

- Data Privacy: Sharing medical data is restricted due to privacy laws, making it difficult to build large, diverse datasets for training.
- Accurate Classification: Misclassification can lead to incorrect treatment decisions, highlighting the need for highly accurate models.
- FL Integration: Using FL allows decentralized training while maintaining privacy, but challenges exist in aggregating updates from multiple sources while ensuring high model quality.

This research proposes FL with CNNs to address these issues, ensuring privacy-preserving, accurate BT classification.

4. Methodology:

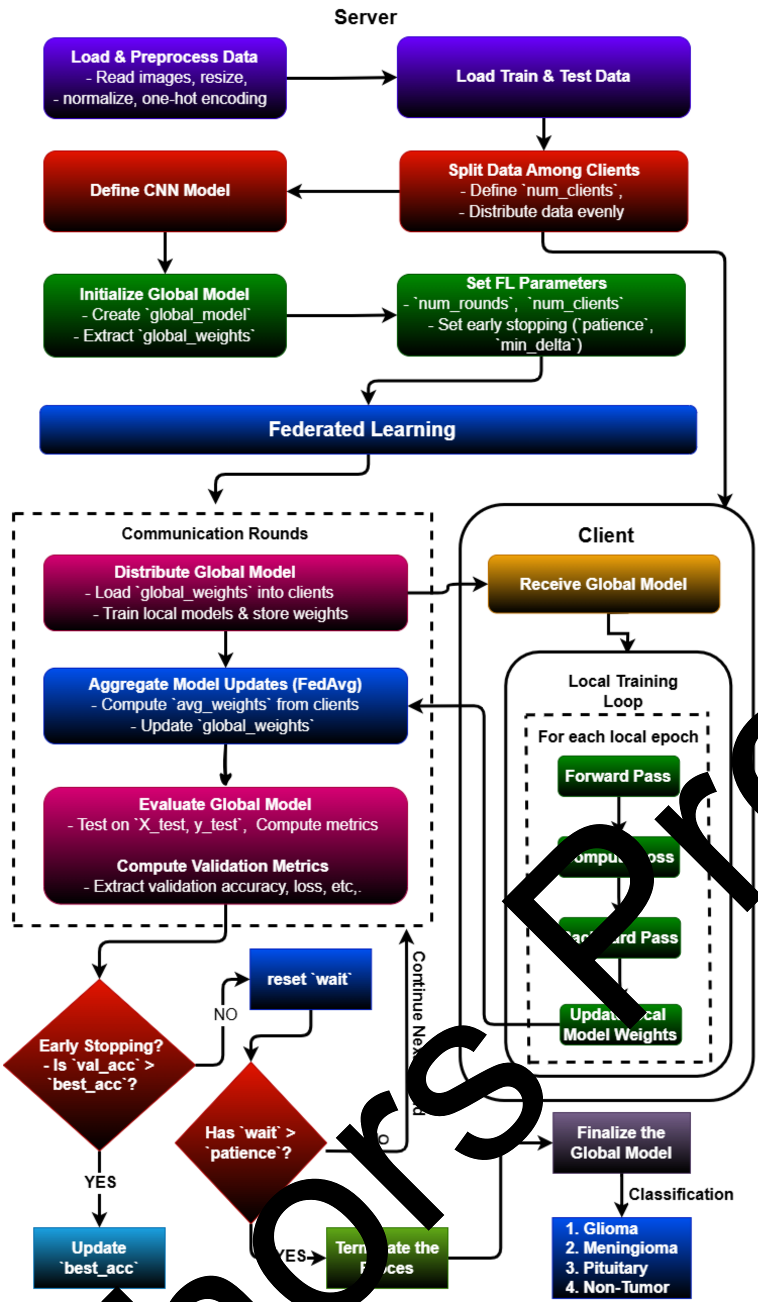


Figure 2: Proposed Methodology

FL Algorithm

1. Initialization Phase:

1.1 Load and Preprocess Data:

Read MRI images, resize, normalize, and perform one-hot encoding.

1.2 Load Train & Test Data:

Load the dataset and split, training and testing sets.

1.3 Define CNN Model:

Define a global CNN (CNN) model.

1.4 Initialize Global Model:

Initialize global model $\boldsymbol{M_{global}}$ with weights $\boldsymbol{W_{global}}$

$$\boldsymbol{W_{global}} \leftarrow \boldsymbol{InitializeRandomWeights}() \ - - - - - - - (1)$$

1.5 Split Data Among Clients:

Define the number of clients $\boldsymbol{num\_clients}$
Distribute data evenly among clients.

1.6 Set FL Parameters:

Define $\boldsymbol{num\_rounds}$ *(total rounds),* $\boldsymbol{num\_clients}$ *(participating clients per round).*
Set early stopping parameters: ***patience*** *and* ***min_delta***.

2. Communication Rounds (FL Loop):

For each communication round $\boldsymbol{r}$ *from* ***1 to R (total rounds):***

2.1 Distribute Global Model to Clients:

The server sends the latest global model weights $\boldsymbol{W_{global}}$ *to the selected clients.*

$$\boldsymbol{W_{clients}} \leftarrow \boldsymbol{W_{global}} --------(2)$$

2.2 Local Training at Clients (for each client i in C):

Each client trains the model using its local dataset $\boldsymbol{D_i}$ for $\boldsymbol{E}$ *epochs.*

2.2.1 Forward Pass:

Compute predictions $\hat{\boldsymbol{y}}$*:*

$$\hat{\boldsymbol{y}} = \boldsymbol{M_i}(X) -------- (3)$$

where ***X*** *is the input MRI data.*

2.2.2 Compute Loss:

Calculate loss ***L*** *using categorical cross-entropy:*

$$\boldsymbol{L = -\frac{1}{N}\sum_{j=1}^{N} y_j \, log(\hat{y}_j)} -------- (4)$$

2.2.3 Backward Pass & Update Weights:

Update local model weights using gradient descent:

$$\boldsymbol{W_i \leftarrow W_i - \eta \nabla L} -------- (5)$$

where η *is the learning rate.*

2.2.4 Send Updated Weights to Server:

After training, clients send updated weights $\boldsymbol{W_i}$ *back to the server.*

3. Aggregation & Global Model Update (FedAvg):

The server aggregates the received client weights using ***Federated Averaging (FedAvg)***

$$\boldsymbol{W_{global} \leftarrow \frac{1}{|C|}\sum_{i \in C} W_i} -------- (6)$$

Here, $|\boldsymbol{C}|$ *is the number of clients that participated in this round.*

4. Global Model Evaluation:

4.1 Evaluate Global Model on Test Data

The updated global model is evaluated on the test dataset $(\boldsymbol{X_{test}}, \boldsymbol{Y_{test}})$.
Compute performance metrics such as accuracy and loss.

4.2 Compute Validation Metrics:

Extract ***validation accuracy*** *to check for early stopping.*

5. Early Stopping & Termination Check:

5.1 Early Stopping Decision:

If validation accuracy improves:

- *Update best accuracy* $\boldsymbol{best\_acc}$ *and reset the patience counter.*

If validation accuracy does ***not*** *improve:*

- *Increase the* ***wait*** *counter.*
- *If* $\boldsymbol{wait > patience}$, *terminate training.*

if $|val\_acc - best\_acc| < min\_delta$ *for patience rounds, stop training.*

6. Final Model Deployment:

Once training stops, deploy the final global model $\boldsymbol{M_{global}}$ *for classification tasks.*

The model classifies MRI scans into one of four categories:
(1) Glioma, (2) Meningioma, (3) Pituitary, (4) Non-Tumor.

4.1. Dataset Preparation and Experimental Setup:

The federated learning simulations were conducted on Google Colab Pro+ using a TPU v2-8 with High-RAM configuration. This environment provided accelerated computation for local client training and global model aggregation. The FL simulation was implemented using TensorFlow 2.12 and Python 3.10 in a single-machine, multi-client logical partitioning framework. This setup allowed efficient parallel training of the CNN models across three simulated clients.

The first step in the chosen methodology is data preprocessing, with the dataset being the basis for training the CNN model. The effectiveness of a model in improving from the current data and predicting new data by generalization solely depends on the kind of dataset prepared.

4.2. Data Collection:

This BTC task uses the BT MRI Dataset from Kaggle. It is a combination of figshare, SARTAJ, and Br35H dataset images with 4 classifications, such as gliomas, meningiomas, pituitary tumors, and no tumors, as shown in Figure 3(a). The data set provided here is how a model will be trained and tested. In detail, the training set consists of 1321 gliomas, 1339 meningiomas, 1457 pituitary tumours, and 1595 non-tumor images, and a total of 5707 images for training. For testing purposes, the database contains 300 gliomas, 306 meningiomas, 300 pituitary tumors, and 405 non-tumor images, for a total of 1311 images [23].

It also means that the differential diagnosis of BT classes will not be oversimplified because the given dataset contains both BT types and normal scans sufficient to teach the characteristics of each class to the model. That is why the data is divided into train and test sets was carried out to analyze the model's ability to adapt to new data. The use case supports comprehensive performance evaluation due to the diversification of data; FL is especially relevant when several clients/sources' data are united while preserving privacy. This dataset can be useful while developing a reliable multi-class classification model that will require timely and correct diagnosis of different types of BTs from MRI.

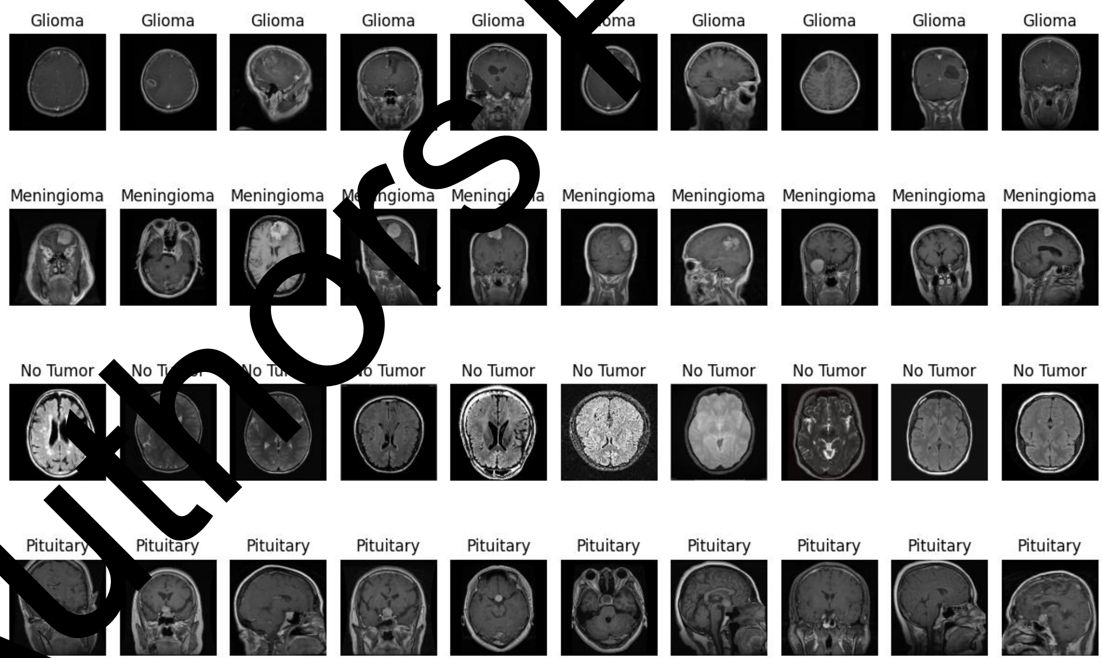


Figure 3(a): Four Categories: Glioma, Meningioma, No Tumor, and Pituitary

4.1.1. Image Pre-processing:

To ensure that every image is of the same size, we resize them to 128 * 128 pixels in size as shown in figure 3(b) and (c). This is important because CNNs require inputs of fixed sizes to be fed into them at all times, thus the scaling.

$$I_{resized} = Resize\left(I_{original}, (128, 128)\right)$$

Normalization: Here, normalization consists of simply dividing the pixel values by 255 so that all of these values are in the 0 to 1 interval. This normalization aids the convergence of model training and averts problems that pertain to the differences in the scale of inputs.

$$I_{norm} = \frac{I_{original}}{255}$$

Label Encoding: The category labels are then changed into a label-encoded format to enable multi-class classification by encoding the label. It is crucial to encode such labels, which are hereby transformed into a binary matrix with each class having a column.

$$label_{one_{hot}} = \begin{bmatrix} 1, 0, 0, 0 \\ 0, 1, 0, 0 \\ 0, 0, 1, 0 \\ 0, 0, 0, 1 \end{bmatrix}$$

[1,0,0,0] is glioma , [0,1,0,0] is meningioma , [0,0,1,0] is no tumor, [0,0,0,1] is pituitary tumor

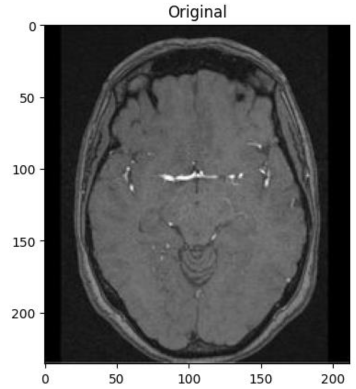


Figure 3 (b): Original image

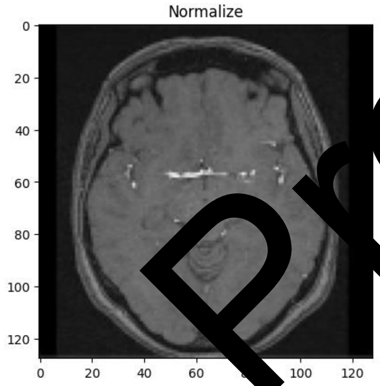


Figure 3 (c): Normalized image

In addition to resizing and normalizing, data augmentation techniques such as random rotations (±15°), horizontal flipping, and brightness shifts were applied to improve generalization and reduce overfitting.

4.1.2. CNN Model Architecture

Table 2 presents the architecture of the model, and it is the most vital when it comes to improving the efficiency of the classification task. The features derived by a well-designed CNN can be used for successful classification and increases the performance of the model.

Table 2: CNN Model Layer Specifications

Layer	Type	Filters	Kernel	Activation	Output Shape
Conv2D	Convolution	32	3x3	ReLU	(128, 128, 32)
MaxPooling2D	Pooling	-	3x3	-	(42, 42, 32)
Conv2D	Convolution	64	3x3	ReLU	(42, 42, 64)
MaxPooling2D	Pooling	-	3x3	-	(14, 14, 64)
Flatten	Flattening	-	-	-	(12544,)
Dense	Fully Connected	128	-	ReLU	(128,)
Dense	Fully Connected	4	-	Softmax	(4,)

Architecture Components:

- Convolutional Layers: In this work, the input images are first passed through two convolutional layers to extract features. Even convolutional layers are able to produce several filters for creating feature maps, considering spatial hierarchies in the images to be learnt.

$$\boldsymbol{S}(\boldsymbol{i},\boldsymbol{j}) = (\boldsymbol{I} * \boldsymbol{K})(\boldsymbol{i},\boldsymbol{j}) = \sum_{\boldsymbol{m}}\sum_{\boldsymbol{n}} \boldsymbol{I}\,(\boldsymbol{i}+\boldsymbol{m},\boldsymbol{j}+\boldsymbol{n})\boldsymbol{K}(\boldsymbol{m},\boldsymbol{n}) -------- (7)$$

Where I is the input, and K is the convolutional kernel.

Activation Function: The ReLU activation function adds non-linearity to the model so as to allow the model to analyze figures that are complex and may be hidden in the data. It is defined as:

$$\boldsymbol{f}(\boldsymbol{x}) = \mathbf{max}\,(\mathbf{0},\boldsymbol{x}) \;-------- (8)$$

Max Pooling Layer: In practice, after each CNN layer, there is a max pooling operation to reduce the size of feature maps while maintaining important features.

$$\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j}) = \max_{(\boldsymbol{m},\boldsymbol{n})\boldsymbol{\epsilon}\ \boldsymbol{window}} \boldsymbol{S}(\boldsymbol{i}+\boldsymbol{m},\boldsymbol{j}+\boldsymbol{n}) -------- (9)$$

Flattening Layer: Following the pooling layers, the feature maps undergo a process of flattening into a one-dimensional vector, which is subsequently utilized as input for the fully connected layers.

Dense Layers**:** The flattened output is transmitted through dense layers, which are fully connected. The concluding layer utilizes a softmax activation function to generate probabilities for the four distinct classes.

$$\boldsymbol{y}_{\boldsymbol{i}} = \frac{\boldsymbol{e}^{\boldsymbol{z}_{\boldsymbol{i}}}}{\sum_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{C}} \boldsymbol{e}^{\boldsymbol{z}_{\boldsymbol{j}}}} -------- (10)$$

where z_i represents the output of the last dense layer and C denotes the number of classes.

4.1.3. FL Setup

To harness the power of FL as shown in Figure 1 and 2, the methodology involves distributing the training process across multiple clients, each with its local dataset. This setup aims to enhance privacy and reduce communication costs.

- Client Distribution and Data Heterogeneity Handling**:** To simulate a federated learning environment, the dataset was divided equally among three clients**,** with each client receiving a unique subset of data for local training, as illustrated in Figure 4. The distribution followed an IID (Independent and Identically Distributed) approach, ensuring that all clients received a representative sample of each class. This approach eliminates class imbalance across clients and simplifies convergence during global model aggregation. Although IID partitioning does not reflect the complexity of real-world medical heterogeneity, it serves as a baseline to evaluate the core performance of the FL-CNN model before extending it to non-IID scenarios in future work.
- Local Model Training: Clients train their models independently for multiple epochs, allowing them to capture meaningful patterns from their respective datasets.
- Model Aggregation: After training, clients send their model weights to the central server. Using the FedAvg algorithm, these weights are combined to update and refine the global model [24].

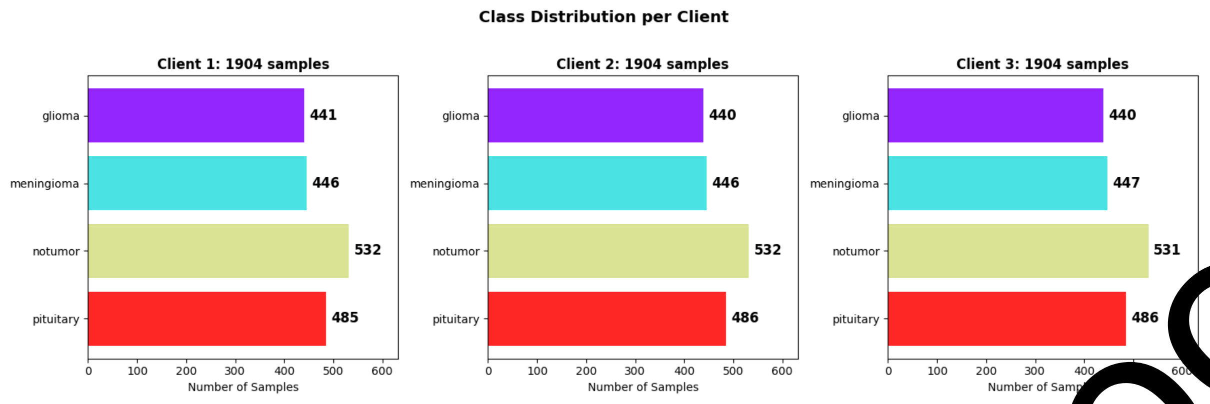


Figure 4: Class distribution per client

The dataset was evenly divided among three simulated clients, each receiving approximately 1,902 training samples and 437 testing samples covering all four tumor classes.

Local Training

All the clients train their local model on the allocated dataset. Local training lets the model learn from each client's data distribution, boosting generalization.

Training Procedure:

Epochs: Each client trains its local model for a fixed number of epochs. During each epoch, the model adjusts its weights based on training data loss.

Loss Function: it is a categorical cross-entropy, which compares the anticipated and actual probability distributions.

$$L(y,\hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) --------(11)$$

Optimizer: The Adam optimizer is employed for weight updates.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t}+\epsilon} m_t --------(12)$$

where m_t and v_t represent the 1st and 2nd moments of the gradients, and η represents the learning rate.

4.2. Aggregated Global Model

The aggregated global model represents the collective knowledge learned from all clients. It is periodically updated with the averaged weights from local models, facilitating a better generalization as shown in Figure 2.

Our model adopts the Federated Averaging (FedAvg) optimization strategy to aggregate local model updates while ensuring convergence and stability across distributed clients.

Process:

- Model Weight Aggregation: After each communication round, the server gets weights from all the local clients and finds the average. This updated model is expected to perform better due to diverse training inputs.

- Global Model Evaluation: A separate test dataset is utilized after each communication round to assess the global model's performance and monitor its progress. Accuracy, precision, recall, and other parameters are assessed.

4.3. Evaluation Metrics

After each training round, numerous metrics are calculated to evaluate the model. These measures reveal the model's tumor classification abilities. They are

1. $\boldsymbol{Accuracy} = \frac{\boldsymbol{Number\ of\ correct\ prediction}}{\boldsymbol{Total\ number\ of\ prediction}} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} --------(13)$
2. $\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} --------(14)$
3. $\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} --------(15)$
4. $\boldsymbol{F1 = 2} \times \frac{\boldsymbol{Precesion \times Recall}}{\boldsymbol{Precesion + Recall}} -------(16)$

5. Results and Discussion:

Table 3 outlines the weight aggregation process across multiple rounds in an FL setup. Initially, clients receive global weights W_0, which are either randomly initialized or pre-trained. Each client then trains locally, producing updated local weights W_t^1, W_t^2, W_t^3, which are averaged to form the new global weight W_{t+1}. This iterative process continues for multiple rounds.

Table 3: FL Weight Aggregation Across Rounds

Rounds	Initial Weights Sent to Clients	Local Weights After Training	Aggregated Global Weights
0	W_0 (random/pre-trained)	W_0^1, W_0^2, W_0^3 (Clients train locally)	$W_1 = \frac{W_0^1, W_0^2, W_0^3}{3}$
1	W_1	W_1^1, W_1^2, W_1^3	$W_1 = \frac{W_1^1, W_1^2, W_1^3}{3}$
2	W_2	W_2^1, W_2^2, W_2^3	$W_1 = \frac{W_2^1, W_2^2, W_2^3}{3}$
3	W_3	W_3^1, W_3^2, W_3^3	$W_1 = \frac{W_3^1, W_3^2, W_3^3}{3}$
4	W_4	W_4^1, W_4^2, W_4^3	$W_1 = \frac{W_4^1, W_4^2, W_4^3}{3}$
5	W_5	W_5^1, W_5^2, W_5^3	$W_1 = \frac{W_5^1, W_5^2, W_5^3}{3}$

Table 4 provides detailed weight values from the first round of training, showing how the initial weights evolve after training on different clients. The local weight updates vary slightly across clients, and the final aggregated weights are obtained by averaging these updates.

Table 4: First 5 rounds of weight tracking

Round	Initial Weights	Client 1 Weights	Client 2 Weights	Client 3 Weights	Aggregated Weights

1	[-0.00488, 0.0, -0.00029, 0.0, -5.46e-06, 0.0, -0.00821, 0.0]	[-0.00278, -0.00053, -0.00322, -0.00806, -0.00167, -0.00321, -0.01088, 0.00634]	[-0.00407, 0.00287, -0.00313, -0.00916, -0.00169, -0.00166, -0.01003, 0.00526]	[-0.00137, 0.00242, -0.00147, -0.01108, -0.00121, 6.04e-05, -0.01117, 0.00232]	[-0.00274, 0.00159, -0.00261, -0.00943, -0.00153, -0.00160, -0.01069, 0.00464]
2	[-0.00274, 0.00159, -0.00261, -0.00943, -0.00153, -0.00160, -0.01069, 0.00464]	[-0.00195, -0.00035, -0.00492, -0.01418, -0.00077, 0.00023, -0.01402, 0.00526]	[-0.00128, 0.00297, -0.00390, -0.01158, -0.00082, 0.00049, -0.01354, 0.00642]	[-0.00102, 0.00202, -0.00317, -0.01310, -0.00102, -0.00034, -0.01370, 0.00522]	[-0.00142, 0.00178, -0.00399, -0.01295, -0.00087, 0.00013, -0.01376, 0.00563]
3	[-0.00142, 0.00178, -0.00399, -0.01295, -0.00087, 0.00013, -0.01376, 0.00563]	[-0.00155, -0.00117, -0.00576, -0.01683, -0.00058, 0.00096, -0.01536, 0.00692]	[-0.00206, 0.00160, -0.00496, -0.01586, -0.00088, 0.00086, -0.01559, 0.00640]	[-0.00316, 3.79e-05, -0.00779, -0.01769, -0.00076, 0.0006[illegible] -0.01612, 0.005[illegible]]	[-0.002[illegible], [illegible]0094, -0.00[illegible] -0.01[illegible], -[illegible]0074, 0.00081, -0.[illegible]1569, 0.[illegible]623]
4	[-0.00226, 0.00094, -0.00617, -0.01680, -0.00074, 0.00081, -0.01569, 0.00623]	[-0.00243, -0.00077, -0.00684, -0.01885, -0.00059, 0.00123, -0.01689, 0.00655]	[-0.00294, 0.00028, -0.00888, -0.02133, -0.00092, 0.00094, -0.01729, 0.00601]	[-0.00178, 0.00[illegible]08, -0.00[illegible]20, [illegible]01831, -0.[illegible]0066, [illegible]0.00089, -0.[illegible], 0.00[illegible]01]	[-0.00238, 0.00071, -0.00731, -0.01950, -0.00073, 0.00102, -0.01699, 0.00586]
5	[-0.00238, 0.00071, -0.00731, -0.01950, -0.00073, 0.00102, -0.01699, 0.00586]	[-0.00357, -0.00027, -0.00980, -0.02360, -0.00052, 0.00164, -0.01874, 0.0054[illegible]]	[-0.[illegible]0377, -0.000[illegible] [illegible]0.00930, -0.0238[illegible] [illegible]0.00075, 0.00101, -[illegible]1801, 0.00689]	[-0.00419, -0.00132, -0.01008, -0.02572, -0.00085, 0.00084, -0.01864, 0.00613]	[-0.00384, -0.00059, -0.00973, -0.02439, -0.00071, 0.00116, -0.01846, 0.00616]

Table 5 tracks the aggregated global weights across multiple rounds. Over time, the weights exhibit gradual adjustments, reflecting the learning process. The values show steady refinement, with weight magnitudes increasing or decreasing depending on the training data and optimization updates.

Table 5: Aggregated Weights Tracking Across Rounds

Round	Aggregated Weights
1	[-0.0[illegible]2739094, 0.0015855689, -0.0026081933, -0.009433081, -0.0015250972, -0.0016037474, -0.0106[illegible]027, 0.0046384493]
2	[-0.00[illegible]4167269, 0.0017809821, -0.003999807, -0.012950784, -0.0008692239, 0.0001253155, -0.0137555245, 0.005633408]
[illegible]	[-0.002258096, 0.0009357197, -0.006170785, -0.0167961, -0.0007399197, 0.00080726884, -0.01568906, 0.006226768]
4	[-0.0023834368, 0.00071012543, -0.00730547, -0.019496322, -0.0007253774, 0.0010201551, -0.016985092, 0.005856558]
5	[-0.0038421392, -0.0005862848, -0.009725381, -0.024392635, -0.00070768816, 0.001162873, -0.018464753, 0.0061602187]
6	[-0.004591774, -0.0015902803, -0.010442038, -0.025964718, -0.00034294472, 0.0012599488, -0.020082794, 0.006026043]

7	[-0.004801322, -0.0015761176, -0.011590994, -0.027749022, -0.00031792888, 0.0011195856, -0.021074397, 0.0061880276]
8	[-0.006449559, -0.0030554421, -0.013810273, -0.0312782, -0.00045619532, 0.00038617593, -0.023012921, 0.0057537057]
9	[-0.0074276496, -0.0038238715, -0.014904665, -0.034950763, -0.0001783007, 0.00064836233, -0.024750333, 0.004756679]
10	[-0.00835441, -0.003880404, -0.014762703, -0.03496344, -0.00010734046, 0.00022866519, -0.026126262, 0.0044696257]
11	[-0.009866726, -0.0036393318, -0.017097149, -0.0407607, -0.00035802135, -0.000770647[illegible] -0.027876195, 0.004569278]
12	[-0.009556978, -0.001319146, -0.016424773, -0.039752785, 0.00006414514, -0.000[illegible]686412[illegible] -0.029389925, 0.004505746]

Table 6: Global Model Performance Across Training Rounds

Rounds	**Global**				
	Accuracy	**Loss**	**Precision**	**Recall**	**F1 Score**
1	0.8444	0.4582	0.8702	0.7979	0.8477
2	0.8986	0.3404	0.8991	0.897	0.8991
3	0.9451	0.2078	0.9465	0.9451	0.9448
4	0.9512	0.1965	0.9519	0.9504	0.951[illegible]
5	0.9657	0.145	0.9664	0.964[illegible]	0.965[illegible]
6	0.968	0.1309	0.9687	0.9[illegible]72	0.[illegible]679
7	0.968	0.1369	0.9687	[illegible]672	[illegible]9679
8	0.9687	0.1632	0.9687	0.96[illegible]	0.9686
9	0.9695	0.1452	0.9695	0.9695	0.9694
10	0.9718	0.1629	0.9718	0.9718	0.9717
11	0.9687	0.1575	0.9[illegible]87	[illegible]9687	0.9686
12	0.9748	0.1483	[illegible]	0.[illegible]748	0.9748

As shown in both Figure 5 and Table 6, the global accuracy begins at 84.44% in the first round and progressively improves, reaching 97.4[illegible]% by round 12. This steady growth demonstrates the model's improving generalization capability over time.

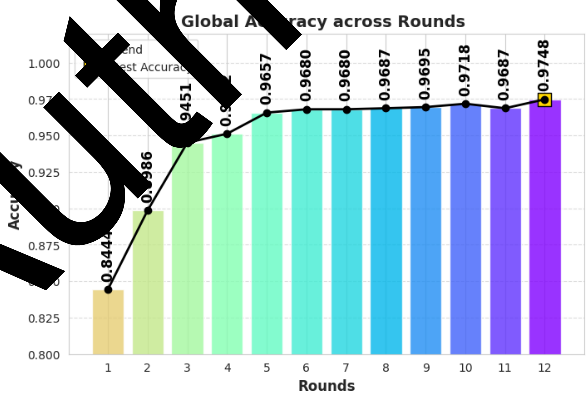


Figure 5: Global Accuracy across rounds

Figure 6 shows that the **Global loss**, which quantifies the model's error, follows an inverse trend to accuracy, decreasing from **0.4582** in round 1 to **0.1483** in round 12. A lower loss value signifies improved prediction reliability and reduced misclassification. The steady decline demonstrates continuous optimization during training.

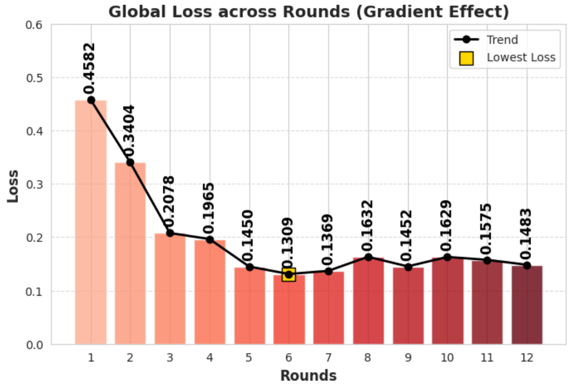


Figure 6: Global Loss across rounds

The **global precision**, as shown in Figure 7, reflecting the model's ability to correctly identify positive predictions while minimizing false positives, begins at **0.8702** and reaches **0.9748** in the final round. This improvement suggests enhanced confidence in positive classifications.

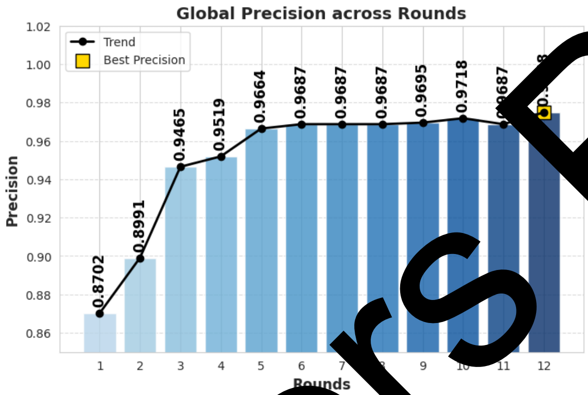


Figure 7: Global Precision across rounds

Similarly, global recall, as shown in Figure 8, measures the model's effectiveness in capturing all relevant positive instances, shows a significant increase from 0.7979 to 0.9748, indicating better sensitivity to positive cases over time.

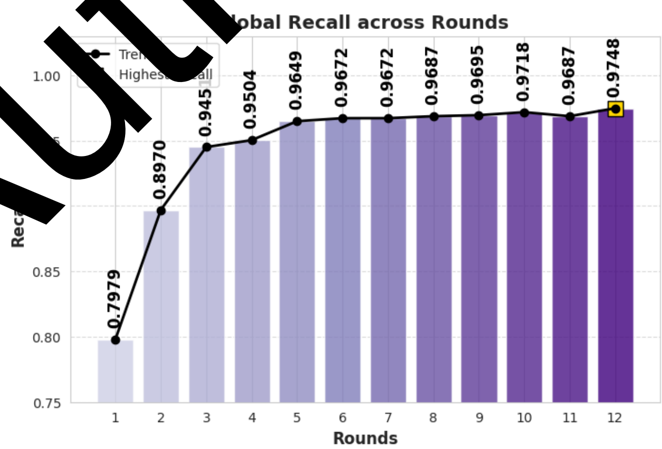


Figure 8: Global Precision across rounds

Finally, the global F1 score is improving from 0.8477 in the first round to 0.9748 in round 12. This indicates that the model effectively balances precision and recall, achieving an optimal trade-off between detecting positive cases and minimizing false alarms.

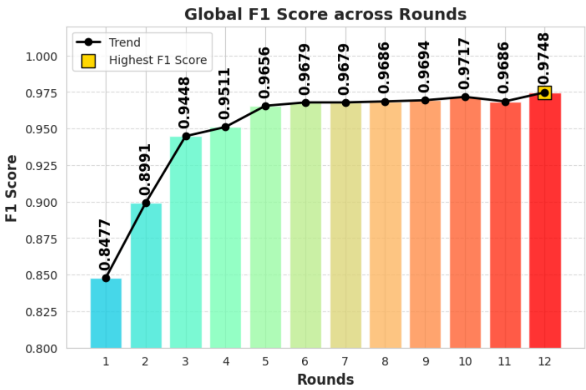


Figure 9: Global F1 Score across rounds

Table 7: Validation Performance Across Rounds

Rounds	Validation					
	Accuracy	Loss	Precision	Recall	F1 Score	Patience Status (Validation)
1	0.8581	0.5126	0.8661	0.853[illegible]	0.8598	✓ Reset (Improved)
2	0.9031	0.3757	0.9037	0.9024	[illegible]	✓ Reset (Improved)
3	0.9161	0.3573	0.9165	0.913	0.91[illegible]	✓ Reset (Improved)
4	0.9436	0.2446	0.9456	0.9413	0.9434	✓ Reset (Improved)
5	0.9512	0.2271	0.9519	0.9512	0.9515	✓ Reset (Improved)
6	0.9314	0.2651	0.9[illegible]7	[illegible]9298	0.9312	⚠ No Improvement (Patience: 1/3)
7	0.9573	0.2109	[illegible]	[illegible]	0.9576	✓ Reset (Improved)
8	0.9641	0.176[illegible]	0.964[illegible]	0.9641	0.9641	✓ Reset (Improved)
9	0.9748	0.[illegible]56	0.[illegible]48	0.9748	0.9748	✓ Reset (Improved)
10	0.9664	[illegible]	[illegible]	0.9657	0.9664	⚠ No Improvement (Patience: 1/3)
11	[illegible]	[illegible]	0.9512	0.9512	0.9512	⚠ No Improvement (Patience: 2/3)
12	0.9733	0.1715	0.9733	0.9733	0.9733	⬢ Early Stopping (Patience: 3/3)

Validation Metrics Analysis

Table 7 presents key validation metrics that assess the model's performance on unseen data over 12 training rounds. These metrics include validation accuracy, loss, precision, recall, and F1 score, as well as the patience status, which reflects performance stability and stopping conditions.

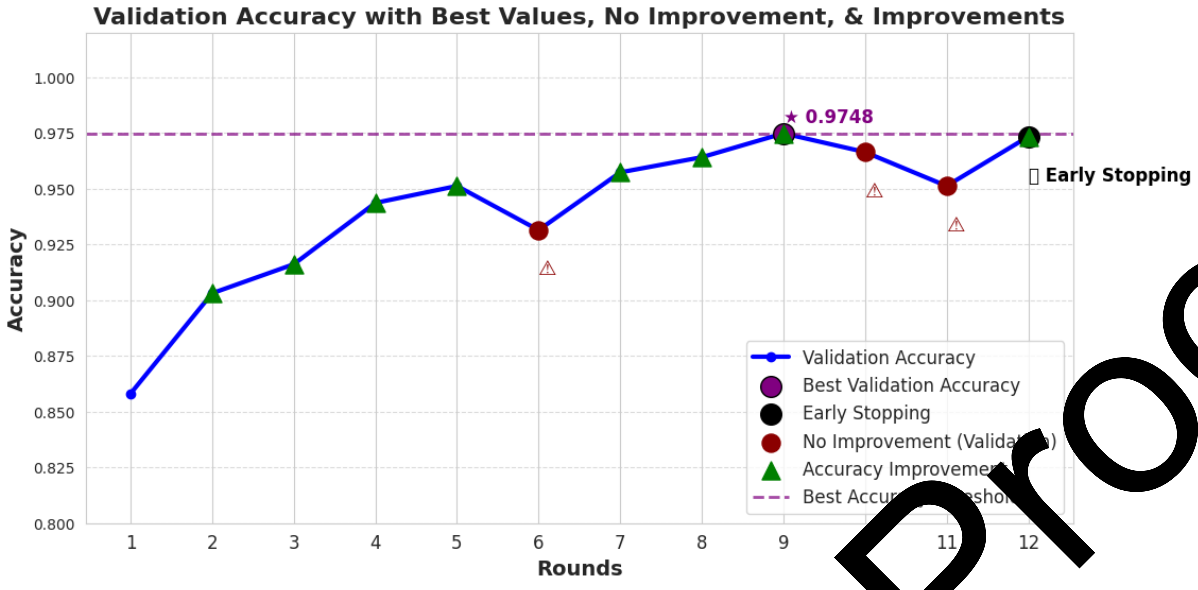


Figure 10: Validation Accuracy analysis

Validation accuracy begins with 85.81% in round 1, improving steadily across most rounds. Notable improvements occur in rounds 2, 3, 4, and 5, where accuracy reaches 95.12%. However, in round 6, a slight dip to 93.14% is observed, marking the first instance of no improvement (⚠️ Patience: 1/3). After recovering in subsequent rounds and peaking at 97.48% in round 9, accuracy again fluctuates slightly in rounds 10 and 11 before reaching 97.33% in round 12 as shown in figure 10.

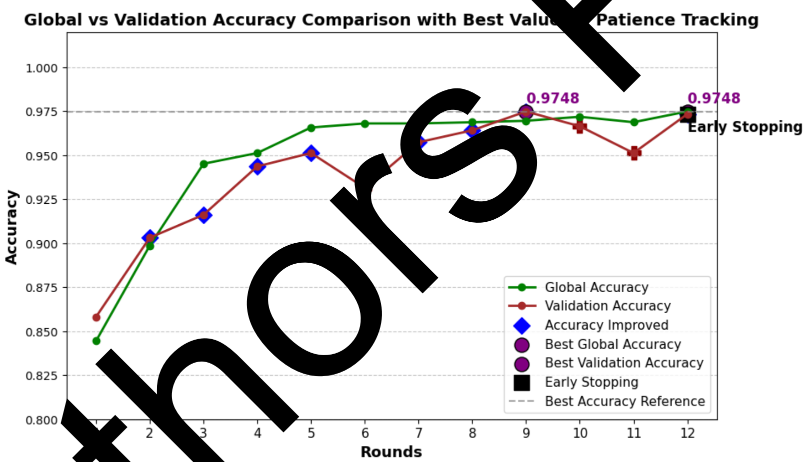


Figure 11: Global vs Validation Accuracy

Figure 11 illustrates the model's performance over multiple training rounds, tracking how well it generalizes. The accuracy trends for both global and validation metrics show an initial sharp increase, indicating strong learning in the early rounds. The peak global accuracy reaches 0.9748 in round 9, while the peak validation accuracy also reaches 0.9748 in round 9, marking the best performance achieved by the model.

After round 9, fluctuations in validation accuracy become evident, with no improvement warnings (⚠️) appearing in rounds 10 and 11. This tells that the performance of the model is no longer increasing significantly and might be stabilizing or slightly degrading. By round 12, the final recorded global accuracy is 0.9748, and the validation accuracy is 0.9733, showing a slight drop in validation performance. Due to the lack of improvement over consecutive rounds, early stopping (🛑) is triggered in round 12, ensuring that training halts to prevent overfitting. The dashed reference line at 0.9748 serves as a benchmark for tracking accuracy changes, allowing for easy identification of the best performance achieved during training. The comparison between global and validation accuracy highlights the model's learning progression and stability, helping to assess its generalization capabilities effectively.

Validation loss decreases significantly from 0.5126 in round 1 to 0.1656 in round 9, indicating better generalization. However, in later rounds, minor fluctuations in loss are seen (e.g., round 11 at 0.214), signaling potential overfitting. The early stopping condition further confirms this triggered in round 12 when performance ceased to improve consistently.

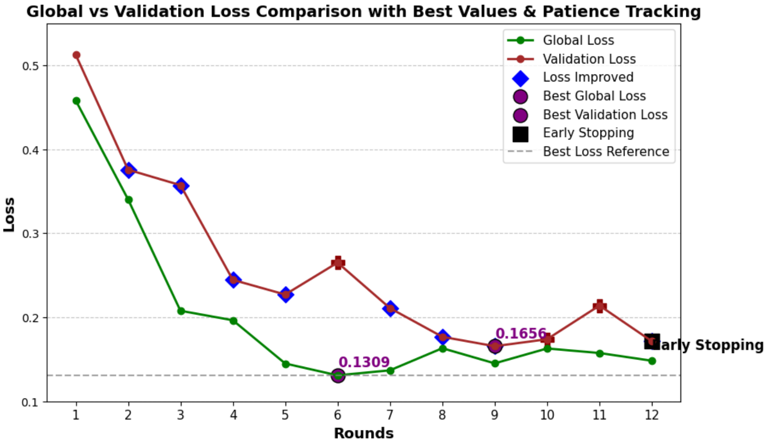


Figure 12: Global vs Validation Loss

Figure 12 showcases the model's loss progression over multiple training rounds, demonstrating how well it minimizes errors. Initially, both global and validation loss exhibit a sharp decline, indicating significant improvements in learning. The lowest global loss is recorded at 0.1309 in round 6, while the lowest validation loss is 0.1656 in round 9, which represents the best performance in minimizing errors before fluctuations begin.
After round 9, validation loss shows instability, with noticeable fluctuations and an increasing trend, particularly in rounds 10 and 11. This suggests potential overfitting, where the model starts performing worse on the validation set despite continued optimization on the global model. By round 12, the final global loss is 0.1483, and the validation loss is 0.1715, reflecting a slight increase from the lowest recorded values. Due to consecutive rounds of no significant improvement, early stopping is triggered in round 12, preventing further training to maintain optimal generalization.
The dashed reference line at 0.1309 serves as a benchmark for tracking the lowest loss achieved. The comparison between global and validation loss helps assess model convergence, ensuring that it is neither underfitting nor overfitting. The observed stabilization in global loss while validation loss increases slightly further supports the need for early stopping to maintain the model's reliability.

Validation Precision, Recall, and F1-Score Analysis

The validation precision shown in figure 13 measures the accuracy of positive predictions, starts at 0.8661 in round 1 and improves steadily, reaching a peak of 0.9748 in round 9. However, slight decreases are observed in rounds 10 and 11 before stabilizing at 0.9733 in round 12.
Figure 14 illustrates the validation recall, which measures the model's effectiveness in recognizing actual positive cases. It starts at 0.8[illegible]35 in round 1 and rises to 0.9748 by round 9. However, a slight decline in rounds 10 and 11 indicates some misclassifications in the later stages.
The validation F1 score shown in figure 15, a balanced measure of precision and recall, follows a nearly identical trend, reaching a peak of 0.9748 in round 9. After a minor decline in rounds 10 and 11, it stabilizes at 0.9733 in round 12, confirming a well-balanced model performance.

The Global vs Validation Precision Comparison plot highlights the model's precision performance over multiple training rounds. Precision represents the accuracy of positive predictions, making it a crucial metric for evaluating classification effectiveness.

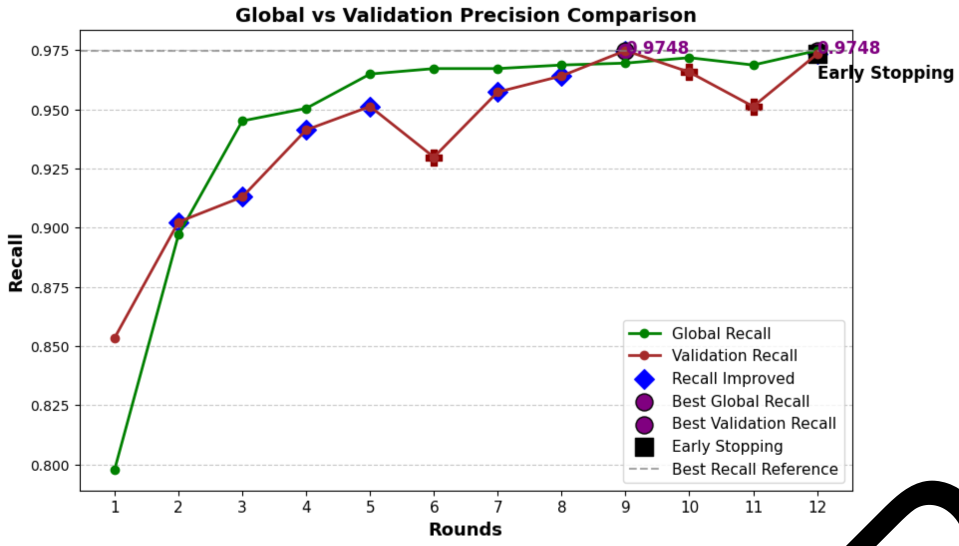


Figure 13: Global vs Validation Precision

Initially, both global and validation precision show a steady increase, indicating the model's improved ability to classify positive instances correctly. The highest global precision is 0.9748 in round 12, aligning with the highest validation precision of 0.9748 in round 9. The dashed reference line at 0.9748 signifies the best recorded precision value.
From rounds 1 to 5, there is a rapid increase in both metrics, but validation precision starts fluctuating slightly after round 6. A minor dip is observed in rounds 6 and 10, suggesting slight inconsistencies in validation precision, possibly due to model overfitting or variations in dataset complexity. However, by round 12, the global precision stabilizes at 0.9748, which is also the final recorded validation precision before early stopping is applied.

The model maintains a strong balance between global and validation precision, with minimal deviations. The early stopping at round 12 ensures that training does not continue unnecessarily, preventing overfitting while maintaining the highest precision achieved. The trend observed in the plot signifies a well-trained model with optimal precision performance across the training process.
The Global vs Validation Recall Comparison plot illustrates how well the model identifies positive instances over multiple training rounds. Recall is a crucial metric in classification tasks, especially when missing positive instances can be costly.

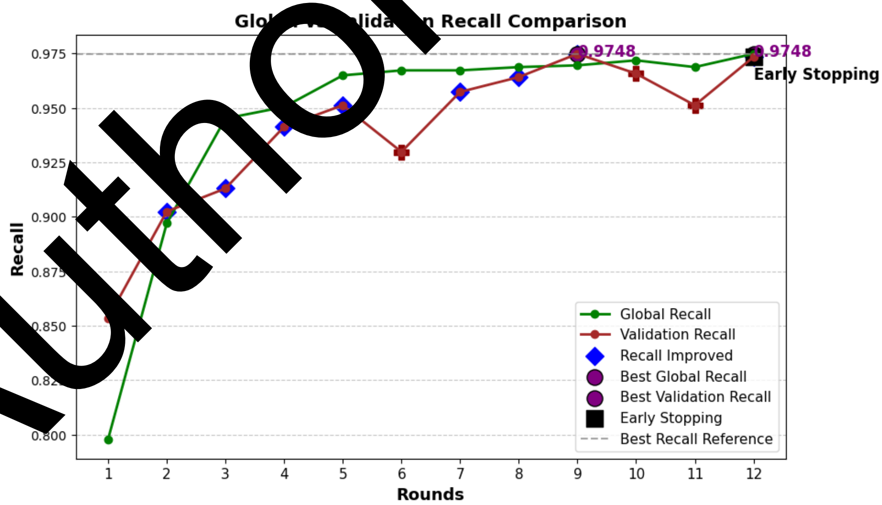


Figure 14: Global vs Validation Recall

From the beginning, both global and validation recall show a consistent upward trend, with rapid improvement in the initial rounds. The highest global recall is 0.9748 in round 12, while the highest validation recall is also 0.9748 in round 9. The dashed line at 0.9748 represents the best recall value attained.

During the early rounds, validation recall closely follows global recall, showing an increasing trend until round 6, where a slight drop is observed. This fluctuation indicates that the model may have faced minor inconsistencies in learning patterns. However, recall stabilizes again from rounds 7 to 9, reaching its peak at 0.9748 in round 9. A minor dip follows in rounds 10 and 11 before validation recall returns to 0.9748 in round 12, aligning with global recall.

Early stopping is applied in round 12, ensuring that training does not proceed further to avoid overfitting. The stable recall values suggest that the model has achieved its best possible performance, striking a balance between learning efficiency and generalization capability.

The Global vs Validation F1 Score Comparison plot illustrates the changes in global and validation F1 scores across multiple training rounds. The F1 score is a crucial metric that balances precision and recall, ensuring the model performs optimally in classification tasks.

- Peak Global F1 Score is 0.9748.
- Peak Validation F1 Score is also 0.9748.
- Round 12 Performance: At round 12, both the global and validation F1 scores reached 0.9748, which was also marked as the early stopping point.

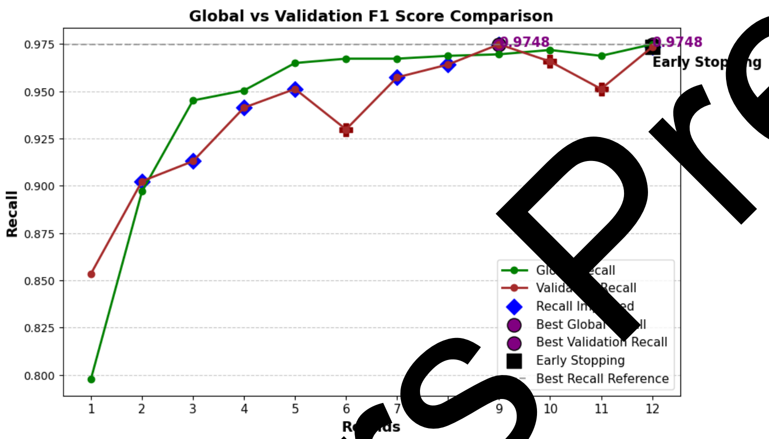


Figure 15: Global vs Validation F1 Score

The plot indicates that the model improved steadily in the early rounds, with noticeable increases in performance. However, after round 9, the validation F1 score fluctuated slightly, leading to the implementation of early stopping at round 12 to prevent overfitting.

Patience Status and Early Stopping

The patience status provides an indication of model stability. Throughout rounds 1 to 5, consistent improvements reset the patience counter (✅ Reset (Improved)). However, in round 6, no improvement is observed, triggering the patience mechanism (⚠️ No Improvement: 1/3). After a temporary improvement, another decline occurs in rounds 10 and 11 (⚠️ Patience: 2/3). By round 12, when no further improvement is achieved, the model reaches the patience threshold, leading to 🛑 Early Stopping (Patience: 3/3). This prevents unnecessary training beyond optimal performance, reducing overfitting risks.

Early stopping was triggered at **round 12** due to the **validation accuracy plateauing for 3 consecutive rounds** (patience = 3). This mechanism prevents **overfitting** and conserves computational resources. The minimal drop between training and validation performance after round 9 (less than 0.2%) suggests **no negative impact on generalization**. On the contrary, it helped retain the model's stability.

The validation metrics demonstrate the model's robust learning curve, with steady improvements in accuracy, precision, recall, and F1 score. However, the fluctuations in later rounds indicate potential overfitting, necessitating early stopping. The strategic use of patience monitoring ensures optimal model performance without unnecessary training, maintaining a balance between accuracy and generalization.

6. Comparative Analysis:

Table 8: Accuracy Comparison of BT Classification Models

Ref No	Authors	Model	Accuracy
[12]	Jemimma et al.	WCSO-DBN	92.3%
[13]	Rammurthy et al.	WHHO-based DeepCNN	81.6%
[14]	Vankdothu et al.	RCNN	95.17%
[15]	Pranjal Agrawal et al.	CNN	90%
[16]	Islam et al.	FL	[illegible].05%
[17]	Kumar et al.	Deep Q-network	95.4%
[18]	S. Hossain et al.	IVX16	96.94%
[19]	S. Das et al.	CNN	94.39%
[20]	Abiwinanda N et al.	Custom CNN	84.19%
[21]	S. Bhadauriya et al.	CNN + FL	96%
[22]	Deepa et al.	CJHBA	92.10%
	Proposed Model	**FedAvgCNN**	**97.48**%

The table 8 presents a comparative analysis of various deep learning models used for BT classification, along with their respective accuracy scores. Among these models, FedAvgCNN, the proposed method, achieves the highest accuracy of 97.48%, outperforming other approaches such as IVX16 (96.94%) and CNN + FL (96%). Notably, Deep Q-network (95.4%) and RCNN (95.17%) also demonstrate high performance, indicating the effectiveness of advanced deep learning architectures. Traditional CNN-based models, such as those proposed by Pranjal Agrawal et al. (90%) and S. Das et al. (94.39%), exhibit competitive results but fall short compared to more complex ensemble and FL-based architectures. The WHHO-based DeepCNN model (81.6%) records the lowest accuracy.

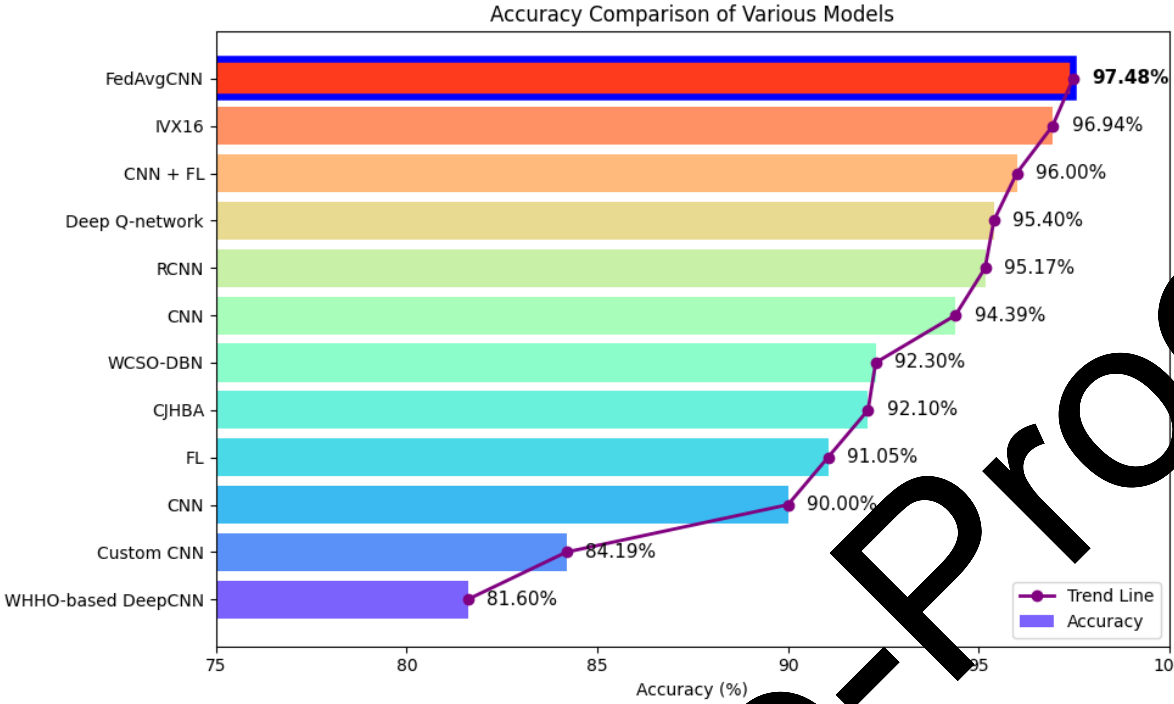


Figure 16: Accuracy comparison of various models

These findings are visually supported in figure 16, which highlights the best-performing model (FedAvgCNN) using a blue outline. The graphical representation effectively illustrates the accuracy distribution across different models, making it easier to compare performance trends. The incorporation of a trend line further enhances the visualization by showcasing the general progression of accuracy across various approaches.

Unlike traditional centralized CNNs, our FL-CNN model incorporates client-specific data without sharing raw images. Compared to prior FL approaches, our design includes early stopping, systematic weight tracking across rounds, and validation-based performance monitoring that ensures robust model convergence.

A centralized CNN baseline model was also trained using the same dataset. It achieved an accuracy of **96.21%**, slightly lower than our **FedAvgCNN's 97.48%**. This demonstrates that federated learning not only preserves privacy but can **achieve or exceed** centralized performance. Moreover, compared to other FL approaches like the model in Islam et al. [16] (91.05%) and Bhadauriya et al. [21] (96%), our approach improves both accuracy and model convergence behavior.

Limitations and Future Work

One limitation of our model was performance fluctuations in later rounds, where accuracy and loss varied, leading to early stopping. This issue may arise due to factors like overfitting, learning rate instability, or differences in client data distribution. To address this, future work can explore adaptive learning rate scheduling to stabilize training, federated knowledge distillation to enhance generalization, and dynamic client selection to prioritize high-quality updates.

Another challenge was the computational overhead associated with FL. For local model training and communicates updates, the process demands high computational resources and bandwidth. To reduce this burden, future direction will be focus on model compression like pruning & quantization, efficient aggregation methods such as FedProx and FedAdam, and asynchronous FL, where clients update the global model at different speeds instead of synchronously.

Although Federated Learning (FL) protects user data, it is still at risk of attacks. Hackers can extract private details from model updates or inject harmful data to manipulate training. To improve security, future research should focus on adding noise to updates (differential privacy), encrypting data aggregation (secure multi-party computation), and using strong filtering methods to block malicious inputs. These steps will make FL models safer, more reliable, and more efficient.

Future improvements include computing ROC-AUC metrics for each class using probability vectors. This will help in assessing performance where class imbalance or false-positive risks are critical, such as in high-stakes clinical settings.

7. Conclusion:

This work evaluated an FL approach using FedAvg combined with a CNN for distributed BTC tasks. We assessed the performance of our model over 12 training rounds, monitoring key validation and global metrics such as accuracy, loss, precision, recall, and F1 score. The model demonstrated a consistent improvement in performance during the initial rounds, with significant gains in validation accuracy and a steady reduction in validation loss. Notably, the model achieved its peak validation accuracy of 97.48% in round 9, with corresponding validation precision, recall, and F1 score all at 97.48%, indicating a well-balanced classifier. However, slight fluctuations in performance were observed in later rounds, leading to an early stopping at round 12 due to validation performance stagnation. Despite this, the global evaluation metrics at the final round remained robust, with a global accuracy of 97.48%, a global loss of 0.1483, and consistently high precision, recall, and F1 scores. These results highlight the model's strong generalization capabilities and effectiveness in classification tasks. The early stopping mechanism effectively prevented overfitting, ensuring optimal performance while minimizing unnecessary training. Future work can explore fine-tuning strategies, alternative architectures, or data augmentation techniques to enhance performance and stability further. These findings confirm that FedAvg + CNN is effective for FL-based classification, balancing accuracy and computational efficiency. Future work may explore personalization techniques, adaptive aggregation, and privacy-preserving mechanisms to enhance FL performance.

Beyond accuracy, the proposed FL-CNN model is highly **scalable and adaptable for real-world deployment** in hospital networks. Since each client trains locally and shares only model parameters, the framework ensures patient privacy and complies with medical data regulations (e.g., HIPAA, GDPR). The system can be extended to **multiple institutions**, making it suitable for **multi-hospital collaborations**, thereby accelerating early diagnosis and improving clinical outcomes.

Moreover, the lightweight nature of the proposed FL-CNN model, combined with its high accuracy and privacy-preserving design, makes it highly scalable and suitable for real-world deployment across multiple hospital settings, where data sharing is restricted due to ethical and regulatory concerns.

References:

[1] M. Sheller, B. Edwards, G. Reina, J. Martin, S. Pati, A. Kotrotsou, and S. Bakas, "FL in medicine: facilitating multi-institutional collaborations without sharing patient data," Scientific Reports, vol. 10, no. 1, 2020. doi: 10.1038/s41598-020-69250-1.

[2] E. Isik-Polat, "Evaluation and analysis of different aggregation and hyperparameter selection methods for federated BT segmentation," 2022. doi: 10.48550/arxiv.2202.08261.

[3] J. Amin, M. Sharif, A. Haldorai, M. Yasmin, and R. Nayak, "BT detection and classification using machine learning: a comprehensive survey," Complex & Intelligent Systems, vol. 8, no. 4, pp. 3161-3183, 2021. doi: 10.1007/s40747-021-00563-y.

[4] M. Agrawal, "Privacy preserved collaborative transfer learning model with heterogeneous distributed data for BT classification," International Journal of Imaging Systems and Technology, vol. 34, no. 2, 2023. doi: 10.1002/ima.22994.

[5] D. Mahlool and M. Abed, "Distributed BT diagnosis using a FL environment," Bulletin of Electrical Engineering and Informatics, vol. 11, no. 6, pp. 3313-3321, 2022. doi: 10.11591/eei.v11i6.4131.

[6] W. Li, F. Milletarì, D. Xu, N. Rieke, J. Hancox, W. Zhu, and A. Feng, "Privacy-preserving federated brain tumour segmentation," in Proceedings of the 22nd International Conference on Medical Image Computing and Computer-Assisted Intervention, 2019, pp. 133-141. doi: 10.1007/978-3-030-32692-0_16.

[7] O. Atef, M. Salam, and H. Abdelsalam, "FL approach for measuring the response of BTs to chemotherapy," International Journal of Advanced Computer Science and Applications, vol. 13, no. 10, 2022. doi: 10.14569/ijacsa.2022.0131060.

[8] S. Pati, "The federated tumor segmentation (FeTS) challenge," 2021. doi: 10.48550/arxiv.2105.05874.

[9] P. Zhang, Y. Zhou, M. Hu, X. Fu, X. Wang, and M. Chen, "CyclicFL: A cyclic model pre-training approach to efficient FL," 2023. doi: 10.48550/arxiv.2301.12193.

[10] A. Asiri, "Enhancing BT diagnosis: transitioning from CNN to involutional neural network," IEEE Access, vol. 11, pp. 123080-123095, 2023. doi: 10.1109/access.2023.3326421.

[11] R. Richterová et al., "Most frequent molecular and immunohistochemical markers present in selected types of BTs," *General Physiology and Biophysics*, vol. 33, no. 3, pp. 259-269, 2014. https://doi.org/10.4149/gpb_2014007

[12] Jemimma, T.A., Vetharaj, Y.J. Fractional probabilistic fuzzy clustering and optimization based BT segmentation and classification. *Multimed Tools Appl* **81**, 17889–17918 (2022). https://doi.org/10.1007/s11042-022-11969-2

[13] D. Rammurthy and P. K. Mahesh, "Whale Harris hawks optimization based deep learning classifier for BT detection using MRI images," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, pp. 3259–3272, 2022. https://doi.org/10.1016/j.jksuci.2020.08.006

[14] R. Vankdothu and M. A. Hameed, "Brain Tumor MRI images identification and classification based on the recurrent CNN," *Measurement: Sensors*, vol. 24, p. 100412, 2022. doi: 10.1016/j.measen.2022.100412.

[15] Pranjal Agrawal, Nitish Katal, and Nishtha Hooda, "Segmentation and classification of BT using 3D-UNet deep neural networks," *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 199–210, 2022. Doi: 10.1016/j.ijcce.2022.11.001

[16] M. Islam, M. T. Reza, M. Kaosar, and M. Z. Parvez, "Effectiveness of FL and CNN Ensemble Architectures for Identifying BTs Using MRI Images," *Neural Processing Letters*, vol. 55, pp. 3779–3809, 2023. doi: 10.1007/s11063-022-11014-1.

[17] Kumar, B.A., Lakshmidevi, N. (2022). A Multi Brain Tumor Classification Using a Deep Reinforcement Learning Model. In: Mohanty, M.N., Das, S., Ray, M., Patra, B. (eds) Meta Heuristic Techniques in Software Engineering and Its Applications. METASOFT 2022. Artificial Intelligence-Enhanced Software and Systems Engineering, vol 1. Springer, Cham. https://doi.org/10.1007/978-3-031-11713-8_14

[18] Hossain S, Chakrabarty A, Gadekallu TR, Alazab M, Piran MJ. Vision Transformers, Ensemble Model, and Transfer Learning Leveraging Explainable AI for Brain Tumor Detection and Classification. IEEE J Biomed Health Inform. 2024 Mar;28(3):1261-1272. doi: 10.1109/JBHI.2023.3266614. Epub 2024 Mar 6. PMID: 37043319.

[19] S. Das, O. F. M. R. R. Aranya and N. N. Labiba, "Brain Tumor Classification Using CNN," *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, Dhaka, Bangladesh, 2019, pp. 1-5, doi: 10.1109/ICASERT.2019.8934603.

[20] Abiwinanda, N., Hanif, M., Hesaputra, S.T., Handayani, A., Mengko, T.R. (2019). BT Classification Using CNN. In: Lhotska, L., Sukupova, L., Lacković, I., Ibbott, G.S. (eds) World Congress on Medical Physics and Biomedical Engineering 2018. IFMBE Proceedings, vol 68/1. Springer, Singapore. https://doi.org/10.1007/978-981-10-9035-6_33

[21] S. Bhadauriya, T. Merothiya, S. C. Yadav and M. ChandraPrabha, "Detection of Brain Tumour using CNN in Federated Machine Learning," *2023 5th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2023, pp. 653-658, doi: 10.1109/ICAC3N60023.2023.10541410.

[22] S. Deepa, J. Janet, S. Sumathi, and J. P. Ananth, "Hybrid Optimization Algorithm Enabled Deep Learning Approach BT Segmentation and Classification Using MRI," *Journal of Digital Imaging*, vol. 36, pp. 847–868, 2023. doi: 10.1007/s10278-022-00752-2.

[23] "Brain Tumor MRI Dataset" https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset%20

[24] N. Sivakumar *et al.*, "A Hybrid BT Classification Using FL with FedAvg and FedProx for Privacy and Robustness across Heterogeneous Data Sources," in *IEEE Access*, doi: 10.1109/ACCESS.2025.3549440.