Advancing Fairness and Privacy in Federated Models

Abstract

Federated Learning is a machine learning training methodology, which addressing challenges related to decentralized data, scalability, and privacy. The focus on user privacy, decentralized data sources, and collaborative learning across industries highlights its significance. As technology evolves, Federated Learning reshapes machine learning towards a decentralized and privacy-conscious future.

In this context, the aggregation of client model gradients by the server in Federated Learning's parameter servers is crucial, showing variations in techniques. Fairness is important to avoid biased outcomes, ensuring equitable representation and treatment of diverse user populations. This project aims to identify an optimal federated learning technique, specifically focusing on fairness and robustness. To align with real-world Federated Learning scenarios where privacy is highly valued, we incorporate differential privacy.

This project specifically examines the effects of gradient averaging techniques, such as FedAvg and Q-FedAvg, on the accuracy, wall-clock time, and fairness of a Convolutional Neural Network. Initially, the investigation focuses on the FEMNIST dataset, characterized by distinct heterogeneous users, to identify the optimal gradient averaging scheme. Additionally, the fairness of the chosen gradient averaging scheme is explored using the UCI Adult dataset, with a specific focus on fairness constraints such as Demographic Parity and Accuracy Parity.

1 Introduction

Deep Learning's popularity in the modern era can be attributed in part to its triumphs, which are facilitated by the availability of large datasets dispersed among numerous stakeholders. Conventional methods frequently require centralizing data in order to train models, which poses serious difficulties including scalability, privacy, and data security risks. Federated Learning is a novel methodology that has emerged in response to these challenges.

McMahan et al [1] have extensively elaborated on Federated Learning,a decentralized learning approach that takes advantage of edge data collection and computing environments. A centralized model is sent to an edge device in this method, allowing it to perform update steps on local data. The updated weights are then sent back to the server, where they are aggregated into a centralized model. Simply put, Federated Learning ensures that user data remains on the devices from which it was collected, with only gradients shared between the central 'server' model and the local 'client' model.

When compared to centralized learning, where a general model gains access to all client data despite potential privacy efforts that may not fully prevent data leakage, this provides a significant privacy advantage.

It is very common to encounter imbalances or biases in the information available to individual clients in the area of data utilization, deep learning models has the potential to amplify these biases. To elaborate, when these models are mainly trained to maximize accuracy, they tend to favor specific demographic groups, such as those classified by gender, age, or race. These demographic characteristics are known as *sensitive attributes*. To ensure the equitable and fair

deployment of deep learning model across diverse populations, it is critical to recognize and address these biases. There are numerous legal frameworks and regulatory measures in place to limit the incorporation of certain characteristics in the development of machine learning (ML) models. The EU General Data Protection Regulation, for example, expressly prohibits the collection of sensitive user attributes. In this project we evaluate fairness of the deep learning model trained using federated learning on the basis of fairness metrics namely Demographic Parity and Accuracy Parity.

Apart from fairness we also focus on two most important phenomenon's in the context of a Federated Setting: The First one is how client knowledge is aggregated to the server, the presence of Non-IID data on client devices, an uneven distribution of training examples among clients, a frequent mismatch between the number of participating devices in training and the number of training examples per client, and communication delays caused by device offline status. The presence of Non-IID data, as well as an uneven distribution of training examples, adds complexity to the aggregation of newly generated models from each client, with the goal of forming the final global model. This process necessitates a trade-off between simply minimizing aggregate client loss and ensuring more equitable performance across all clients. In This project we examine two most popular aggregation algorithms namely FedAvg and Q-FedAvg and evaluate them on the metrics of accuracy, wall-clock time, and fairness for a Convolutional Neural Network to determine which algorithm performs better in minimizing aggregate client loss and ensuring more 'fair' performance across all clients.

The Second Phenomenon is privacy, on first glance, it appears that privacy is preserved in Federated Learning (FL) because the aggregator does not have direct access to private or sensitive data. Various attacks, however, have revealed potential information leaks within the FL framework [2]. Researchers have used cryptographic approaches centered on intricate Partial Homomorphic Encryption (PHE) or embraced Differential Privacy (DP) to address this vulnerability. While Private FL solutions based on PHE [3],[4] struggle with computational inefficiency and vulnerability to post-processing attacks, In this project we focuses on leveraging the robust privacy guarantees provided by a differentially private solution for the privacy constraint.

The goal of this project is to investigate the performance metrics such as accuracy, robustness of various federated learning gradient averaging algorithms and also evaluate them on basis of ethical dimensions, specifically fairness.

2 Background and Preliminaries

2.1 FedAvg

Federated Averaging (FedAvg) is a key optimization algorithm within the Federated Learning framework. FedAvg solves the problem of aggregating model updates from distributed devices. The algorithm uses a weighted average of local model updates to enable collaborative model training while protecting user privacy. FedAvg has gained popularity due to its effectiveness in scenarios involving a large number of devices, such as mobile phones, edge devices, and IoT devices.

At its core, Federated Averaging works in a cyclical fashion, with local model training followed by global model aggregation. Participants train models on their local data, resulting in model updates. These updates are then safely aggregated to form a global model. The global model's weights are adjusted based on each device's contribution, with more weight given to devices with larger, more representative datasets. This iterative process is repeated until the global model converges on a solution that is representative of the decentralized network's collective knowledge.

One of the primary advantages of Federated Averaging is its inherent ability to preserve user privacy. Sensitive information remains on the user's device by keeping data localized and only transmitting model updates, reducing the risk of privacy breaches. Because the aggregation mechanism ensures

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
$m \leftarrow \max(C \cdot K, 1)$
$S_t \leftarrow$ (random set of m clients)
for each client $k \in S_t$ **in parallel do**
$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w): *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
for batch $b \in \mathcal{B}$ **do**
$w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

Figure 1: FedAvg Algorithm

that global model parameters are updated without exposing individual user data, it is a reliable solution for applications where data confidentiality is critical. Figure 1 shows the algorithm for Federated Averaging.

2.2 Q-FedAvg

While federated averaging has proven effective in training a global model that performs well in general, it overlooks an important factor: the diversity of data among edge devices. Essentially, the data is not always consistent or uniform across the devices that the model evaluates. In other words, while federated averaging can result in a shared global model with good overall performance, individual clients may experience significant performance differences.

To address this issue, there is a model aggregation technique known as Q-federated averaging. Its goal is to develop a global model that performs more equitably across the various data sets it encounters. This is accomplished by introducing a new objective loss function, known as the q-FFL objective, into Federated Learning. This function penalizes clients who suffer a higher loss functions more, ultimately minimizing the objective outlined in equation below .

$$f_q(w) = \sum_{j=1}^{J} \frac{p_j}{(q+1)} F_j^{(q+1)}(w)$$

Where q is a parameter to tune the amount of fairness we impose on the final centralized model.Due to the q+1 exponent term, the global loss in this federated loss scheme is no longer simply a weighted average of the sampled client losses. To address this, Li et al [5] proposed Q-Federated Averaging in Figure 2, in which the weights are derived from the upper bounds of the gradients' local Lipschitz constants.

2.3 Differential Privacy

Differential privacy emerges as a robust and rigorous framework for privacy preservation in data analysis. It provides a mathematical definition of privacy guarantees, offering a principled approach to balance the utility of data analysis with the protection of individual privacy.

At its core, differential privacy ensures that the inclusion or exclusion of any single data point does not significantly impact the outcome of a computation or analysis. This statistical notion of privacy

guarantees that individual contributions to the dataset remain indistinguishable, thereby protecting against inference attacks.

Algorithm 2 q-FedAvg

Input: $K, E, T, q, 1/L, \eta, w^0, p_k, k = 1, \cdots, m$
for $t = 0, \cdots, T-1$ **do**
  Server selects a subset S_t of K devices at random (each device k is chosen with prob. p_k)
  Server sends w^t to all selected devices
  Each selected device k updates w^t for E epochs of SGD on F_k with step-size η to obtain $\bar{w}_k^{t+1}$
  Each selected device k computes:

$$\Delta w_k^t = L(w^t - \bar{w}_k^{t+1})$$
$$\Delta_k^t = F_k^q(w^t)\Delta w_k^t$$
$$h_k^t = qF_k^{q-1}(w^t)\|\Delta w_k^t\|^2 + LF_k^q(w^t)$$

  Each selected device k sends Δ_k^t and h_k^t back to the server
  Server updates w^{t+1} as:

$$w^{t+1} = w^t - \frac{\sum_{k\in S_t}\Delta_k^t}{\sum_{k\in S_t} h_k^t}$$

end for

Figure 2: Q-FedAvg Algorithm

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.
Initialize θ_0 randomly
for $t \in [T]$ **do**
  Take a random sample L_t with sampling probability L/N
  Compute gradient
  For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$
  Clip gradient
  $\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i)/\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$
  Add noise
  $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
  Descent
  $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

Figure 3: DP-SGD Algorithm

Differential privacy is often characterized by two key parameters: ϵ (epsilon) and δ (delta). Epsilon quantifies the level of privacy protection, with lower values indicating stronger privacy guarantees. Delta is a parameter that allows for a small, controlled probability of deviation from perfect privacy.

In this project we used the most famous DP-SGD Algorithm, which was introduced in [6]. The authors of DP-SGD use Gaussian noise ($N(0,\sigma^2)$) to sanitize the gradients provided by the Stochastic Gradient Descent (SGD) algorithm. This step aims to limit the influence of training data on the training process. The Algorithm of the DP-SGD is shown in Figure 3.

2.4 DemoGraphic Parity

DemoGraphic Parity, often referred to as demographic fairness or demographic parity, is a fairness criterion that focuses on equalizing the impact of a machine learning model across different demographic groups. Specifically, it requires that the predictions or decisions made by a model exhibit parity among these groups, ensuring that outcomes are not systematically biased against any particular demographic.

Demographic attributes encompass a range of characteristics such as age, gender, race, ethnicity, and other socio-economic factors that define individuals or groups. In the context of DemoGraphic Parity, understanding the role and significance of these attributes is vital for assessing and addressing potential biases in machine learning models.

Mathematically, DemoGraphic Parity is often expressed as the requirement that the conditional probability of a positive prediction given a particular demographic attribute should be equal across all demographic groups. Achieving DemoGraphic Parity implies that the model's predictions are not influenced by an individual's demographic background.

The condition for demographic parity is, for all a,b ϵ A

$$P(C = 1|A = a) = P(C = 1|A = b)$$

2.5 Accuracy Parity

One critical metric of fairness is accuracy parity, which demands an equal level of accuracy across different demographic groups at its core, mandates that the accuracy of a predictive model should be consistent across diverse groups, irrespective of demographic attributes such as race, gender, or ethnicity. It serves as a mechanism to mitigate biases that may arise during the development and deployment of algorithms, ensuring that the system's performance does not disproportionately favor or disadvantage any particular group

While accuracy parity shares common ground with demographic parity, it addresses certain limitations inherent in the latter. Demographic parity aims for equal representation across groups but does not specifically focus on the accuracy of predictions. Accuracy parity, however, hones in on the precision and reliability of the algorithm's outcomes, thereby offering a more nuanced perspective on fairness.

The condition for Accuracy parity is for a,b ϵ A.

$$P(C = Y|A = a) = P(C = Y|A = b)$$

3 Experimental Setup

3.1 Datasets

3.1.1 FEMNIST

The First dataset we chose to work with is the Federated Extended MNIST dataset (FEMNIST) developed by Cohen et al [7]. in 2017 to conduct experiments that mimic a real-world scenario of federated learning. This dataset contains handwritten digits ranging from 1 to 10, organized by the people who wrote them. This one-of-a-kind arrangement enables us to capture the non-uniform behavior that is common in federated learning setups. Even though users are tasked with writing the same digits, variations in handwriting style, label distribution, and label quantity are to be expected.

We used the TensorFlow Federated API to access this dataset, which had already been preprocessed into 28x28 tensors with a single color channel. We used the test/train split provided by the TensorFlow Federated package for our training and evaluation processes. This split entails withholding 10 examples from each client at random for the test set. Following that, we divided each client's test set in half, randomly assigning one half for validation and the other half for determining final testing metrics.

3.1.2 UCI Adult

The Second dataset we choose is UCI Adult dataset, also known as the "Census Income" dataset, is a popular dataset in machine learning for classification and prediction tasks it was initially introduced by Ronny Kohavi and Barry Becker[8] in 1996 and is hosted by the UCI Machine Learning Repository.. It was derived from data from the 1994 United States Census and includes information about individuals such as age, education, marital status, occupation, and more. The target variable in this dataset is typically income level, which is classified as ">50K" or "=50K," indicating whether an individual earns more than $50,000 per year or not.

In the UCI Adult dataset the sensitive attribute is gender and which is available as either male or female. To conduct the federated learning experiment we preprocessed the dataset by performing data cleaning by removing rows with missing values and later we converted the input into one hot encoding for categorical variables and normalised the rest. Further we balanced the data a bit by duplicating minority samples which is females who is earning more than 50K USD in the existing data to get exactly 50k samples.We used the test/train split provided by the scikit learn for our training and evaluation processes we divided the dataset in 80:20 for training and testing. Following that, we further divided training dataset equally for 10 client's.

3.2 Model Architecture

3.2.1 Model Architecture for FEMNIST

The first model we used for the experiment is Convolutional Neural Network (CNN) model architecture for FEMNIST dataset. It basically contains 6 layers.

The first layer of the network, which is a convolutional layer. The layer comprises 32 filters (also known as kernels) with a size of 3x3. The activation function used is Rectified Linear Unit (ReLU). The input shape for this layer is specified as (28, 28, 1), indicating that the input images are grayscale with dimensions 28x28 pixels.

After the first convolutional layer, a max-pooling layer is added as second layer. Max pooling is a downsampling operation that reduces the spatial dimensions of the input data. In this case, it uses a pool size of 2x2, effectively reducing the width and height of the previous layer.

Third layer is again a convolutional layer with 64 filters of size 3x3. This layer operates on the output of the previous max-pooling layer.

Fourth layer is second max-pooling layer is added after the second convolutional layer, further reducing the spatial dimensions.

Fifth layer is the flatten layer is introduced to convert the 3D tensor output from the previous layer into a 1D tensor. This flattening step is necessary to transition from the convolutional and pooling layers to the fully connected layers.

Final layer is a fully connected layer with 128 neurons is added. This layer connects every neuron from the previous layer to each neuron in this layer. The activation function used is ReLU. The Figure 4 visualises the Model Architecture

3.2.2 Model Architecture for UCI Adult

The Second model we used for UCI Adult dataset is neural network model architecture implemented using the Keras framework [9] for deep learning. Its basically a two fully connected neural networks having the same architecture. Each network has two hidden layers with (50, 50) neurons and ReLU activation

The first layer is a dense layer, serving as the input layer. This layer has an activation function of Rectified Linear Unit (ReLU), a popular choice in neural networks for introducing non-linearity. The input shape for this layer is specified as (103), indicating that the layer expects input data with a feature size of 103. The number of neurons or units in this layer is 50.

The second layer is another dense layer, following the input layer. Similar to the input layer, it utilizes the ReLU activation function. The purpose of this layer is to capture complex patterns and representations within the input data. The number of neurons in this layer is 50. The hidden layers between the input and output layers are crucial for the network to learn hierarchical representations of the input data.

The third and final layer is the output layer, also a dense layer. Unlike the previous layers, it doesn't have a ReLU activation, it performs a linear transformation. The number of neurons in this layer is 2 as it is binary classification.The Figure 5 visualises the Model Architecture

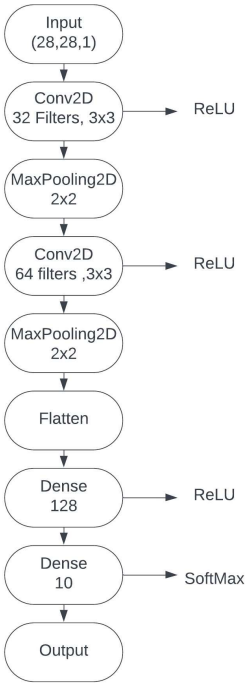

Figure 4: FEMNIST CNN Model Architecture

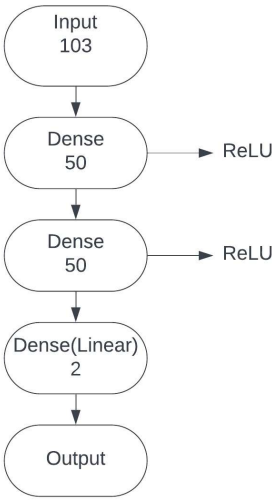

Figure 5: UCI Adult Dataset Model Architecture

3.3 Federated Learning Simulation

For the evaluation using the FEMNIST dataset we used the Flower Python framework [10] to implement the simulations of the various Federated averaging algorithms. Due to computation limitations we simulate our federated learning experiments using a distributed network of 100 unique clients. Both the server and client has the same model architecture mentioned above in 3.2.1

For the second part of evaluation on UCI Adult dataset we custom implemented the federated learning without using Flower as we had to use custom metrics and loss functions for our model to evaluate fairness which was causing issue's when used via flower framework specifically during deserialization and serialization of model. Basically we implemented simple FedAvg by taking layerwise weights and averaging it from the weights received from all clients reproducing what flower framework would typically do. Unlike FEMNIST setup we used only 10 clients for training and both the server and client share same model architecture as mentioned in 3.2.2.

4 Experiment Process and Results

To investigate the performance and fairness of gradient aggregation schemes, we conducted a thorough two-phase evaluation. We meticulously compared the accuracy, wall clock time, and fairness metrics associated with two distinct gradient aggregation schemes in the first phase. This evaluation was carried out by training a Convolutional Neural Network (CNN) model on the FEMNIST dataset using federated learning techniques. Following that, in the second phase, we concentrated on evaluating the fairness constraints, specifically DemoGraphic Parity and Accuracy Parity, using the superior gradient aggregation scheme discovered in the first phase. The UCI Adult Dataset and a Neural Network model were used in this evaluation. We prioritized privacy concerns throughout both phases by using the DP-SGD optimizer from the tensorflow_privacy package [11]. This strong approach enabled us to thoroughly investigate and understand the nuances of gradient aggregation schemes, their impact on model performance, and their adherence to fairness constraints while maintaining a commitment to privacy-preserving methodologies.

4.1 Evaluation Phase-1

During the first phase of our evaluation, we began by running federated learning experiments on a decentralized network with 100 distinct clients. As previously stated, each client has an identical model architecture that is securely stored locally, allowing for the seamless exchange of model weights with the central server. To simulate real world federated learning training process we Incorporated the integration of differential privacy measures, by use of Kera's DP-SGD optimizer. We discover that from trying noise multiplier values of .2, .4, .6, .8, 1, and 1.2, the upper end of which was used as a noise value in the canonical TensorFlow tutorial [12], that .2 is the highest we can set our noise to ensure consistent learning across epochs. For Q-FedAvg scheme we choose the q param as 0.001 for our experiment.

Table 1 provides an overview of the hyperparameters used throughout the experiment, shedding light on the intricacies of our research design for a more detailed insight into the experimental setup.

The training process started by , we randomly select 10 clients from a pool of 100 using a uniform distribution in each training round. These clients are trained for 5 epochs, and their models are then aggregated into the global model using the two gradient aggregation schemes FedAvg and Q-FedAvg. We execute 50 training rounds for both federated learning schemes, mainly tracking the training accuracy, loss and wall-clock time required to complete these 50 rounds of training and evaluation. To measure model fairness we measure the variance of the test set loss and accuracy across 10 uniform randomly sampled clients. In order to incorporate these measurements from a variety of clients, we average these values across the rounds of training for each federated aggregation strategy

The results we obtained are showed in Figure 6, 7 and Table 2, 3. From the results obtained we can clearly see the FedAvg outperforms Q-FedAvg both in terms of execution speed and variance of loss and accuracy.So we conclude that the FedAvg scheme is better compared to Q-FedAvg.

The code used to conduct the experiments are in the following github links. [13] and [14]

Dataset	Hyperparameters
FEMNIST	NUM_CLIENTS = 100 BATCH_SIZE = 8 MICROBATCH_NUM = 4 CLIENT_EPOCHS = 5 ROUNDS = 50 SGD_LEARNING_RATE = .1 SGD_NOISE_MULTIPLIER = .2 L2_NORM_CLIP = 1 Loss = Categorical Cross Entropy
UCI Adult	NUM_CLIENTS = 10 BATCH_SIZE = 1000 MICROBATCH_NUM = 1 CLIENT_EPOCH = 5 ROUNDS=20 SGD_LEARNING_RATE = 0.0001 L2_NORM_CLIP = 1.5 SGD_NOISE_MULTIPLIER = 1.3 Loss = Categorical Cross Entropy

Table 1: HyperParameters for Datasets

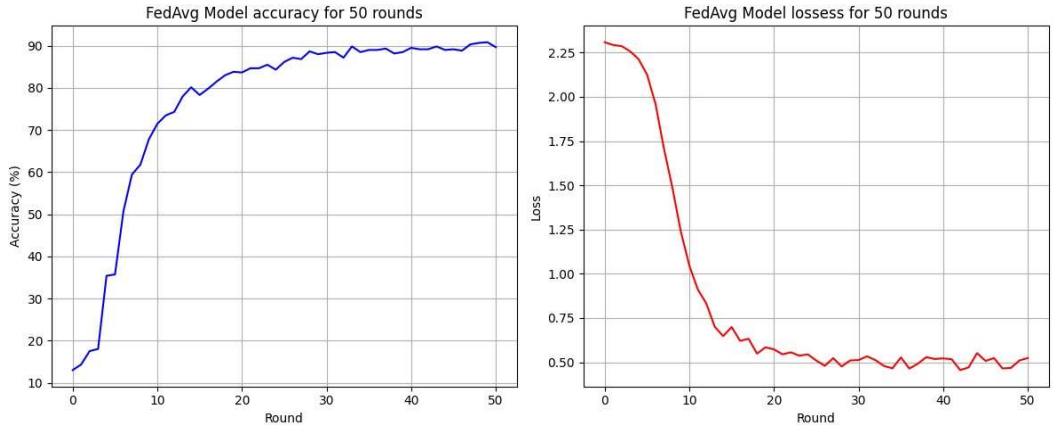

Figure 6: FedAvg Accuracy and Loss

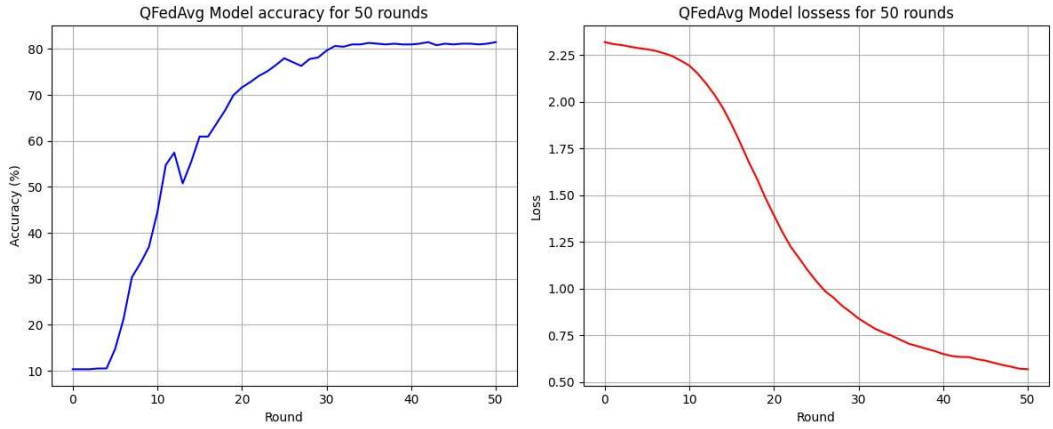

Figure 7: Q-FedAvg Accuracy and Loss

FL Scheme	WallClock Time
FedAvg	1551.21274
Q-FedAvg	1582.46569

Table 2: FL Schemes WallClock Times

FL Scheme	Avg Variance of Accuracy	Avg Variance of Loss
FedAvg	0.018275	0.000710
Q-FedAvg	0.036424	0.102536

Table 3: Fairness Evaluation by Variance of Accuracy and Loss

4.2 Evaluation Phase-2

During the second phase of our evaluation, We choose FedAvg as it performed better as per first phase results to conduct indepth analysis of fairness by measuring DemoGraphic Parity and Accuracy Parity. In this phase the dataset we choose is UCI Adult dataset.

Table 1 provides an overview of the hyperparameters used throughout the experiment, shedding light on the intricacies of our research design for a more detailed insight into the experimental setup.

The training process started by, we select 10 clients for each training round. These clients are trained for 5 epochs, and their models are then aggregated into the global model using the FedAvg gradient aggregation schemes FedAvg. We execute 20 training rounds for the federated learning schemes,and track the training accuracy, DemoGraphic Parity and Accuracy Parity.

The results for this phase are shown in Figure 8,9. The findings reveal a discernible inverse relationship between accuracy and DemoGraphic Parity, implying that FedAvg's fairness may be inadequately addressed. It is expected to remain at a minimum or constant across all rounds, according to the principles of DemoGraphic Parity. A detailed examination of accuracy parity across rounds, on the other hand, reveals noticeable fluctuations, highlighting the presence of bias within the model. This in-depth examination sheds light on the complexities of fairness considerations in the context of Federated Learning. It emphasizes the importance of gaining a more complete understanding of the complexities involved in achieving equitable model performance across diverse demographic groups.

The code for conducting this experiment in the following github link [15].

5 Conclusion

In conclusion, this project aimed to assess various federated aggregation algorithms based on their impact on accuracy,execution time and fairness by simulating real world like federated training using privacy. Through rigorous experimentation, FedAvg and QFedAvg were evaluated, with the goal of identifying the most effective one for collaborative model training across decentralized devices.

The results revealed that FedAvg algorithm outperformed QFedAvg in terms of achieving higher accuracy while maintaining reasonable execution times and lower variance of accuracy and loss among the rounds. This finding suggests that FedAvg can be a promising choice for federated learning implementations where balancing accuracy and efficiency is crucial.

However, the investigation did not conclude with the evaluation of accuracy and execution time alone. Recognizing the importance of fairness in machine learning models, especially in federated settings, an in-depth analysis of the FedAvg model's fairness was conducted using UCI Adult Dataset. Disappointingly, the examination uncovered instances where the model exhibited disparities in its predictions across different demographic groups, indicating a lack of fairness as seen in the results of Phase2 Evaluation.

By this project we want to convey that ongoing efforts on federated learning research should also focus on developing aggregation algorithms that not only optimize for accuracy and efficiency but also incorporate mechanisms to mitigate biases and ensure fairness in the model outcomes.

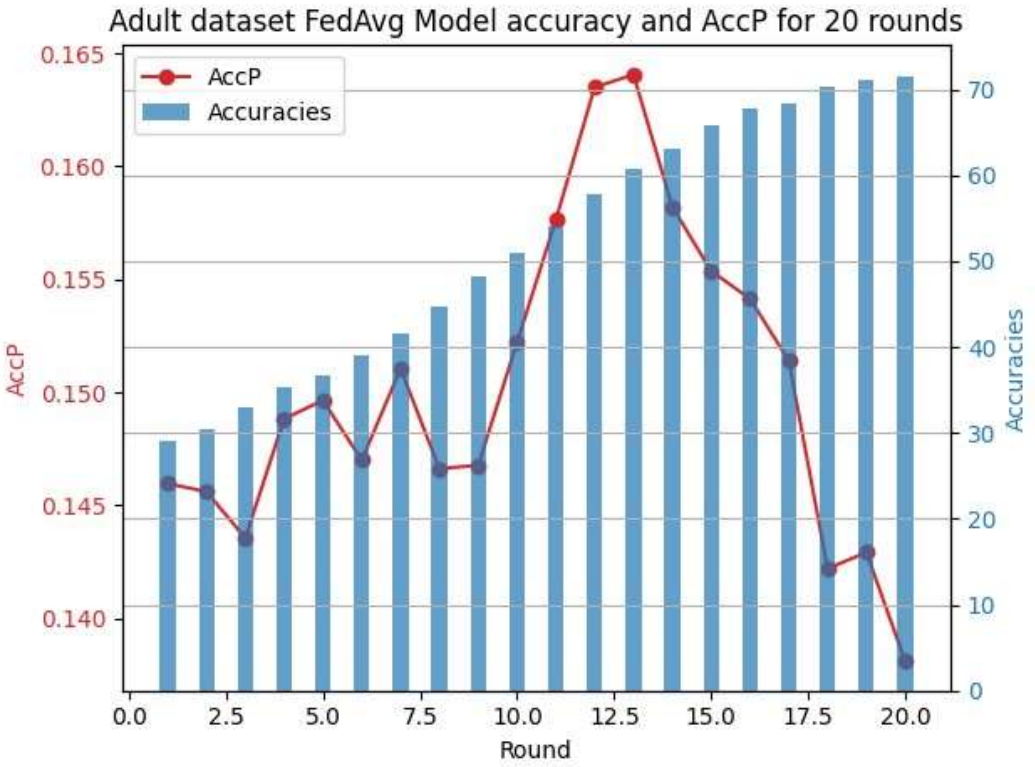

Figure 8: FedAvg Accuracy and Accuracy Parity for UCI Adult Dataset

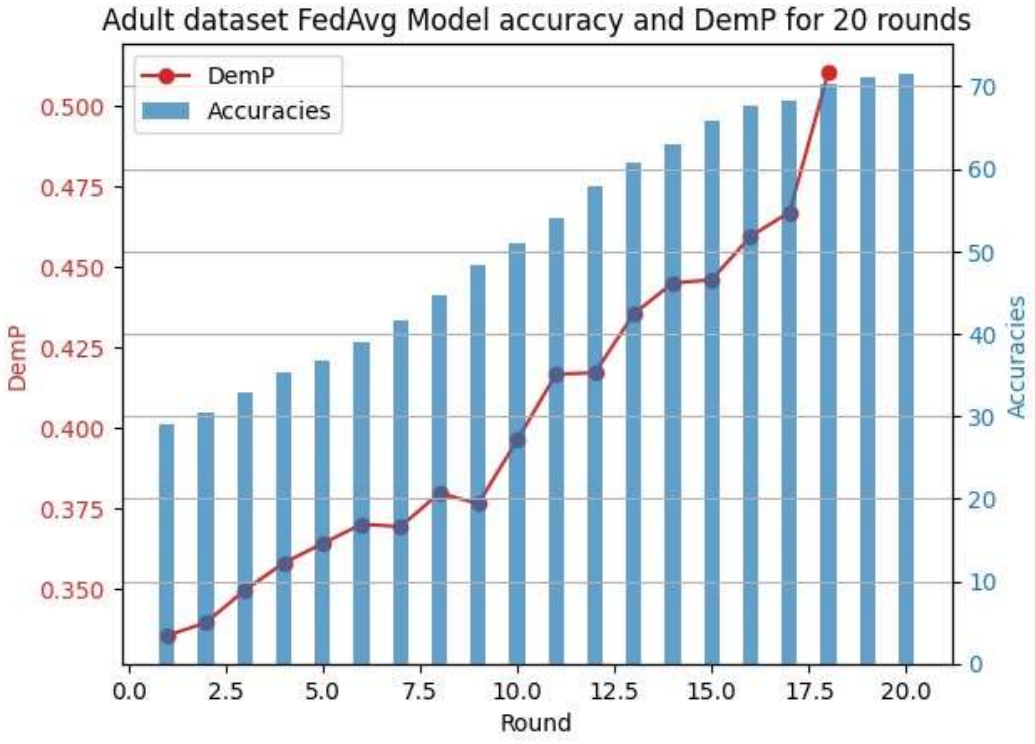

Figure 9: FedAvg Accuracy and DemoGraphic Parity for UCI Adult Dataset

References

[1] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023.

[2] Viraaji Mothukuri, Reza M. Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[3] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pages 493–506. USENIX Association, July 2020.

[4] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38, 2017.

[5] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning, 2020.

[6] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16. ACM, October 2016.

[7] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017.

[8] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[9] François Chollet et al. Keras. https://keras.io, 2015.

[10] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Hei Li Kwing, Titouan Parcollet, Pedro PB de Gusmão, and Nicholas D Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[11] tensorflow_privacy. https://github.com/tensorflow/privacy.

[12] tensorflow_privacy. https://www.tensorflow.org/federated/tutorials/federated_learning_with_differential_privacy.

[13] Fedavg evaluation results. https://github.com/PrabhakarKamathS/CS584-Machine_Learning_Project/blob/main/FedAvg_Evaluation.ipynb.

[14] Qfedavg evaluation results. https://github.com/PrabhakarKamathS/CS584-Machine_Learning_Project/blob/main/QFedAvg_Evaluation.ipynb.

[15] Fedavg phase2 evaluation results. https://github.com/PrabhakarKamathS/CS584-Machine_Learning_Project/blob/main/FedAvg_Phase2Evaluation.ipynb.