International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Federated Machine Learning Performance Evaluation with Flower: MNIST vs CIFAR-10

Singanamalla. Jaya Mohnish[1], Gandra. Shiva Krishna[2], Kota. Venkata Narayana[3], Jonnalagadda. Surya Kiran[4], Thatavarti. Satish[5], Inbarajan.P[6]

Submitted: 26/01/2024 **Revised**: 04/03/2024 **Accepted**: 12/03/2024

Abstract: This paper delves into Federated Learning's advantages, an innovative approach to distributed machine learning that enables model training using decentralized data, eliminating the necessity for centralized aggregation. The primary objective involves conducting extensive experiments with image datasets to discern the superior model training methodology between federated and non-federated approaches. Flower Federated Learning, an open-source framework integrated seamlessly with the TensorFlow machine learning platform, is utilized in the investigation. The pivotal point of this investigation revolves around a comparative analysis of federated and non-federated configurations, utilizing the renowned MNIST and CIFAR-10 datasets. Focus canters on key performance metrics such as precision, recall, and F1 score. In this paper, the experiment is conducted with a federated setup having two clients, each containing only 50\% of random data from each dataset class. These clients are connected to a server where the global model is stored and updated as part of the experiment. The non-federated setup involves a single client for model training and testing. The federated setup achieves 92%-96% for MNIST and around 45%-50% for CIFAR-10 datasets over various rounds and 25 to 50 epochs. In contrast, the non-federated setup achieves around 96% for MNIST and 85% for CIFAR-10 datasets. These results represent a brief experimentation phase on a smaller client setup, with the potential for similar updates for a higher number of clients. However, it's crucial to note that the field of federated learning is evolving rapidly, promising advancements that may narrow, if not bridge, the existing performance gap. The empirical analysis concludes that, as of now, traditional setups exhibit superior performance in comparison to federated configurations. Nevertheless, the evolving nature of federated learning warrants continued investigation and experimentation, as it holds significant potential to transform the landscape of distributed machine learning soon.

Keywords: Federated Learning, Flower Federated Framework, Machine Learning, TensorFlow, MNIST, CIFAR-10.

1. Introduction

Federated Learning has gained considerable attention as a distributed machine learning approach that enables training models with decentralized data, eliminating the need for centralized aggregation.[9] This study addresses the gap in comprehensive comparisons between Federated Learning and traditional methods, particularly in the context of image datasets where the data between the clients is limited and yet to learn from the existing parameters. By conducting a comprehensive comparison between Flower Federated Learning, integrated with TensorFlow, and traditional non-federated approaches. The focus is on image datasets, specifically MNIST and CIFAR-10 where in each experiment the specific dataset is split among the two clients in 50% split and other variations of 75:25 or 25:75 split among the clients. Flower is an open-source framework that facilitates decentralized model training across multiple nodes. Standard metrics (precision, recall, F1 score) are used to evaluate performance. By emphasizing highly distributed datasets, the goal is to pinpoint scenarios where federated learning excels, contributing valuable insights to the understanding of its strengths in comparison to conventional methodologies. This research bridges a crucial gap in the understanding of federated learning's efficacy by showcasing its ability to traditional machine learning. The findings serve as a valuable guide for researchers and practitioners, aiding in the judicious selection of learning methods for image datasets. Considerations such as privacy, data distribution, and model performance are emphasized, offering insights that contribute to informed decision-making in the ever evolving landscape of machine learning methodologies [18]. In this paper, the same developed model is used in both the experiment setups with just changes in the input layers as per the dataset dimensions. A comparison between this non-federated setup and federated learning is conducted, focusing on parameters such as accuracy and loss. The key emphasis lies in understanding the correlation between the number of epochs and accuracy in model training evaluation. Evaluations on diverse datasets, including MNIST and CIFAR-10, reveal insights into the performance of the traditional setup with specifying epoch values.

2. Literature Survey

Considering the existing work for our research purposes has helped in identifying the research gaps and development areas of the current research work. The details of the literature survey are given in the below table:

[1,2,3,4,5] *Department of computer science and Engineering, Koneru Lakshmaiah Education foundation, Andhra Pradesh, India (522502)*
[6] *Samsung R&D Institute Bangalore, India*
[1] *Email of corresponding author*: sjayamohnish@gmail.com

S. No	Paper Title	Year of Publication	Objective	Limitations
1	A machine sound monitoring for predictive maintenance focusing on very low frequency band [14]	2021	Introduces a machine sound monitoring system tailored for predictive maintenance, with a specific focus on the very low frequency (VLF) band.	The accuracy and reliability of sensors capturing very low-frequency sounds might vary, potentially impacting the quality and consistency of the collected data
2	Data Poisoning Attacks on Federated Machine Learning [12]	2022	Unleash the Effects of Data Poisoning on a IOT Based Federated ML Model.	Untargeted attacks reduce the overall performance of the main task, they are easier to be detected.
3	Secure Aggregation with Failover and Encryption [8]	2021	Secure Aggregation algorithm targeted at cross-organizational federated learning applications with a fixed set of participating learners.	A failure is detected by setting a timeout on getting the result. Clearly, the best timeout to set depends on the number of nodes and features in the aggregation.
4	Data privacy preservation algorithm with k-anonymity [10]	2021	Experiment results are presented that the proposed algorithm could effectively preserve data privacy and also reduce the number of visited nodes for ensuring privacy protection.	Algorithm applies the global recording generalization which causes more information loss due to all values in the same quasi-identifier need to be generalized to the same level in the generalization hierarchy.
5	Secure Aggregation for Federated Learning in Flower [8]	2022	The proposed secure aggregation method, employing secure multi-party computation (MPC) techniques, aims to ensure the privacy of data in federated learning.	Improving the efficiency of the secret-sharing functions, extending Salvia for clients running in other programming languages
6	Federated Machine Learning: Survey, Multi-level, Classification, Desirable Criteria and Future Directions in Communications and Networking Systems [16]	2021	A three-level categorization system that classifies federated learning approaches into categories based on the high-level concerns they each address, the sub-challenges inside each high-level challenge category, and the strategy used to tackle each related sub-challenge.	Limitation in sharing data with the central cloud model that consumes huge bandwidth as it must share with various federated models that performs the required multi-level classifications and criteria based.
7	Privacy preservation techniques in big data analytics: a survey [11]	2018	Authors conduct a thorough survey encompassing diverse privacy preservation techniques within the domain of big data analytics.	The utilization of big data raises pertinent privacy concerns, potentially exposing sensitive information about individuals or organization.

3. Methodology

In traditional machine learning, data gathered from various sources such as laptops and cellphones are typically centralized on a single server for training. Machine learning algorithms utilize this data to self-train before making predictions based on new data. However, with federated learning, a distributed approach to machine learning, multiple users can collaboratively train a machine learning model without the need to exchange raw data [9]. This method allows for model updates from different clients to be sent to a central server, which oversees the training process without accessing the raw data itself.

The experimental setup for federated machine learning involves two clients where the data is partitioned into different splits like 50:50, 25:75, and 75:25. The training updates from these experiments are transmitted to the server at the conclusion of each round of training, which is executed for a specific number of epochs—either 25 epochs or 50 epochs in the study.

The project's workflow is illustrated in Fig. 1, with detailed explanations provided beneath the diagram, outlining the sequence of steps and interactions within the federated learning framework.

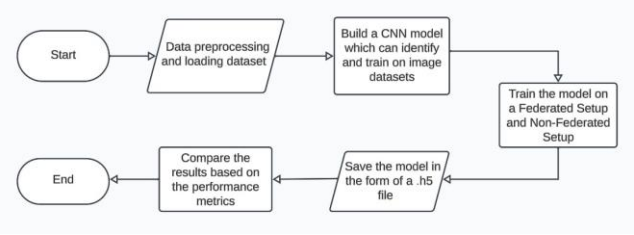

Fig. 1. Flowchart of Experimentation

In the data preprocessing phase for the MNIST dataset, comprising 28x28 grayscale handwritten digit images, standard preprocessing steps are applied. This includes normalizing pixel values to the range [0, 1], flattening images to 784-dimensional vectors, reshaping input dimensions, and converting categorical labels into one-hot encoded vectors. Similarly, for the CIFAR-10 dataset, which contains 32x32 color images across ten classes, preprocessing involves normalizing pixel values, applying data augmentation techniques for improved generalization, resizing images, reshaping input dimensions, and encoding class labels into numerical representations. In the subsequent sections, the description of the dataset and model architecture will be provided in detail.

In this paper, a performance analysis of Federated Machine Learning and Traditional Machine Learning using MNIST and CIFAR-10 datasets is being conducted. The data is split into specific percentages for training the entire dataset. In the later stages of this paper, the computation results section will showcase the performance analysis.

3.1. Dataset Description

3.1.1. MNIST Handwritten Digits

The MNIST database consists of handwritten digits with 60,000 training images and 10,000 testing images which are 28x28 pixel and grey scale with 10 classes from digits 0 to 9.
The MNIST database has become a widely adopted standard for rapidly testing theories and algorithms in the fields of pattern recognition and machine learning. It was derived from the original MNIST database and specifically modified for this purpose, hence the name MNIST. The database offers a relatively simple yet effective dataset for quick evaluation of theories and algorithms.
In the study, the objective is to compare the performance of a specific model in both Federated and Non-federated setups utilizing the MNIST dataset. An advantageous aspect of using MNIST is that the dataset's handwritten digits have already undergone preprocessing steps like segmentation and normalization. This pre-processed nature of the data saves valuable time and effort that would otherwise be necessary for preprocessing and formatting the data [1].

3.1.2. CIFAR-10

The CIFAR-10 dataset is another widely utilized benchmark in the field of computer vision for evaluating pattern recognition and machine learning algorithms. It consists of 60,000 images divided into 10 distinct classes, with each class containing 6,000 images.
The dataset is split into two main subsets: a training set and a testing set. The training set comprises 50,000 images, evenly distributed across the 10 classes, providing ample data for training and model development. The remaining 10,000 images form the testing set, which is used for evaluating the performance and generalization of the trained models.
Each image in the CIFAR-10 dataset is of size 32x32 pixels and contains various types of objects and scenes. The dataset covers a diverse range of image categories, including but not limited to animals (e.g., birds, cats, dogs), vehicles (e.g., automobiles, trucks), and everyday objects (e.g., planes, ships, chairs). This diversity makes the CIFAR-10 dataset well-suited for assessing the robustness and accuracy of algorithms across different visual domains.
Researchers often rely on the CIFAR-10 dataset due to its established status as a standard benchmark, allowing for meaningful comparisons and reproducibility of results.[2]

3.2. Model Description

The same model is used for training in both cases to maintain the same architecture for comparison purposes, just with a change in the input shape based on the dataset trained and tested upon. The results shown in this experimentation are performed on the same model as in Fig. 2.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 32)        832
 max_pooling2d (MaxPooling2D  (None, 14, 14, 32)       0
 )

 dropout (Dropout)           (None, 14, 14, 32)        0

 conv2d_2 (Conv2D)           (None, 14, 14, 64)        18496

 conv2d_3 (Conv2D)           (None, 14, 14, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 64)         0
 2D)

 dropout_1 (Dropout)         (None, 7, 7, 64)          0

 flatten (Flatten)           (None, 3136)              0

 dense (Dense)               (None, 128)               401536

 dropout_2 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 10)                1290
```

Fig. 2. Model used for Training

Fig. 2 provides an overview of the model's structure, illustrating the various layers employed in our Federated Setup to train the CNN Model on both the MNIST Dataset and CIFAR-10 dataset. The optimizer utilized in this setup is Adam, an extension of stochastic gradient descent that dynamically adjusts neural network parameters, thereby enhancing both speed and accuracy in real-time [3]. Employing Sparse Categorical Cross entropy as the loss function is advantageous given the class-based nature of the target values within the data. Notably, the model comprises 484,714 parameters.

4. Computational Experiments

4.1. Implementation of Traditional Machine Learning

The Traditional Model Approach refers to a conventional Centralized Model building Architecture where numerous data sources for machine learning models are centralized within a repository. This prevalent system is widely adopted by practitioners and has been extensively researched. For our experiment, we intend to employ this approach and contrast it with the federated setup. In this non-federated setup, parameters such as accuracy, loss, evaluation accuracy, and evaluation loss will be utilized for comparative analysis. The program is executed for 25 epochs and 50 epochs, allowing for comparison under varied conditions [2]

So, primarily, the primary focus is on the number of epochs versus accuracy for the model training evaluation. In the traditional machine learning setup, a single file houses the dataset loading, training, and testing procedures within itself. Fig. 3. shows how the model works in general for traditional training method.

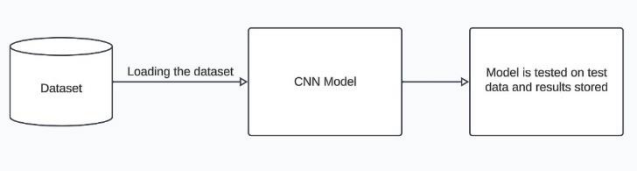

Fig. 3. Centralized Machine Learning Working

To obtain the results for the datasets, evaluations were initially conducted on various datasets, including MNIST and CIFAR-10. When the model was trained using the MNIST Hand-Written Images Dataset, testing was performed across different epochs, specifically 25 epochs and 50 epochs. Throughout these training sessions, evaluations were carried out on the models' accuracy. The recorded accuracy stood at 0.85173 after training the model for 25 epochs and increased to 0.96827 after training the same model for 50 epochs. Notably, the model underwent training using the traditional machine learning setup.

Moreover, the CIFAR-10 Image Dataset was utilized to both train and evaluate our model. To delve deeper into comprehending the impact of different training patterns on accuracy variations, the model underwent training using 25 and 50 epochs. After training the model for 50 epochs, the achieved accuracy stood at 0.85751, showcasing improvement over the accuracy of 0.62821 obtained when the model was trained for 25 epochs.
As this method of training has been followed for a long time it is obvious that the results are better in this type of training. There are a lot of proven results to this type of training on image datasets. [17]

Implementation of Federated Machine Learning using Flower Framework
In the Federated Learning Approach, the architecture of model building is decentralized, aiming for an efficient model that can adapt across various devices as needed. In this approach, separate models exist on both Client and Server Machines for training purposes. Initially, the model undergoes training on the Client Machine, and then the insights derived are transmitted to the Server model for aggregated training of the Server Model. Subsequently, the modifications are sent back to the Clients, enabling them to train their models based on the results obtained from the Server Model [5]. This approach is iterative, emphasizing continuous training for future developments to meet evolving requirements, rather than entirely replacing the model. The same approach is shown below in Fig. 4 where the averaging of updates is shown.

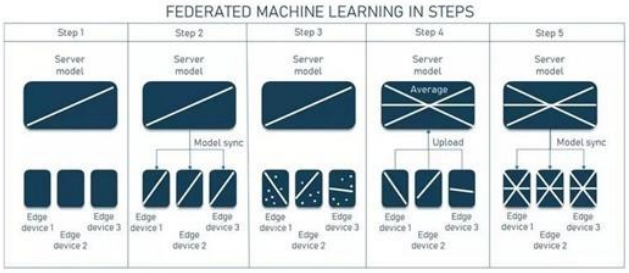

Fig. 4. De-Centralized Machine Learning Architecture

A complete FL framework that separates itself from other platforms by providing additional tools for executing large-scale FL experiments and analyzing richly diverse FL device situations.[4] Flower provides several features to support this process, including:

• Data Sharding: Flower can divide the data among the clients in a way that ensures that each client receives a representative sample of the overall data set.
• Model aggregation: Flower provides algorithms for aggregating the updates from the clients in a way that ensures the resulting model is accurate and unbiased.
• Communication protocols: Flower includes support for various communication protocols, such as HTTP and gRPC, to enable communication between the clients and the server.

The experimentation setup is shown in the below Fig. 5

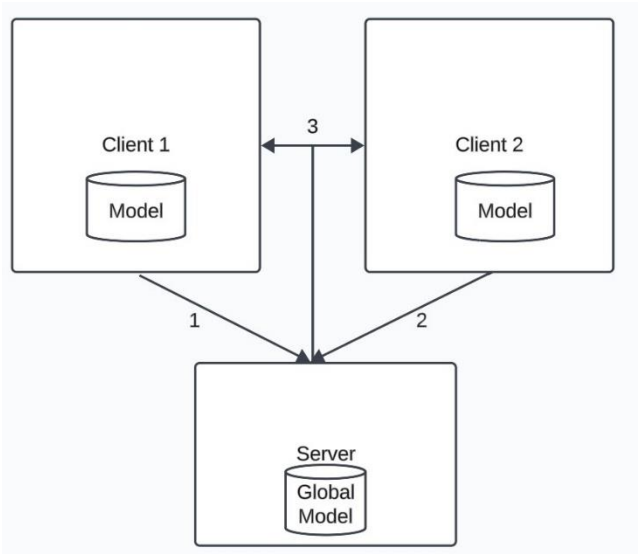

Fig. 5. Federated Learning Setup

As per the above Fig.4 there are 3 steps majorly at a high level. First the model is trained locally on the client and the model updates (learnings) are sent from the client to the server where the global model exists. Similar is the case with client 2, then in step 2 the learning is then updated to the global model on the server. In step 3 the updated learnings are sent to all the clients connected to the server. All this process is coordinated through the gRPC connection between them.

In this experimentation, the dataset has been divided into two equal halves, where each client possesses only 50% of the dataset. The distribution of the MNIST dataset among the clients is outlined below in Fig. 6.

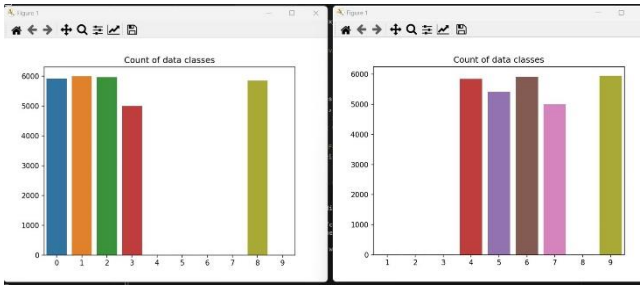

Fig. 6. Distribution of MNIST datasets among two clients

The experimentation results are shown below for the respective datasets –

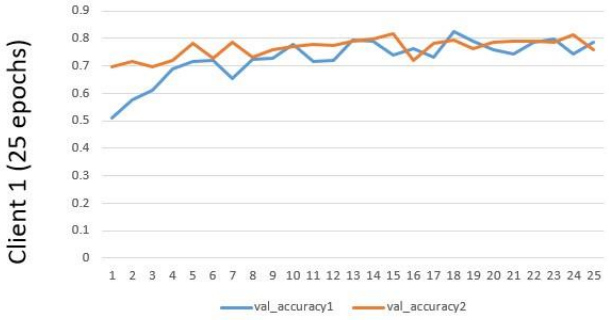

Fig. 7. Federated Setup for 25 epochs and 2 rounds on MNIST (Client 1)

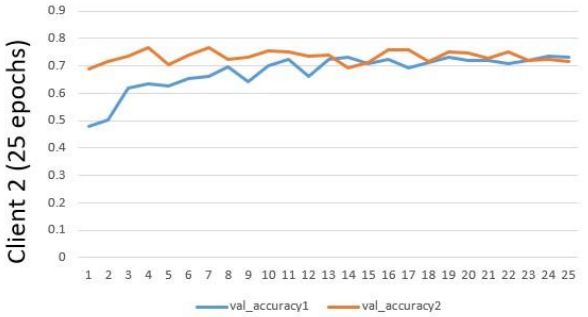

Fig. 8. Federated Setup for 25 epochs and 2 rounds on MNIST (Client 2)

Fig. 7 and Fig. 8 represent the training progress across 25 epochs through two rounds where val_accuracy1 represents the values accuracy during the first round of training and val_accuraccy2 represents the accuracy during the second round of training. We can clearly observe that the initial starting point of accuracy drastically increases from the second round.

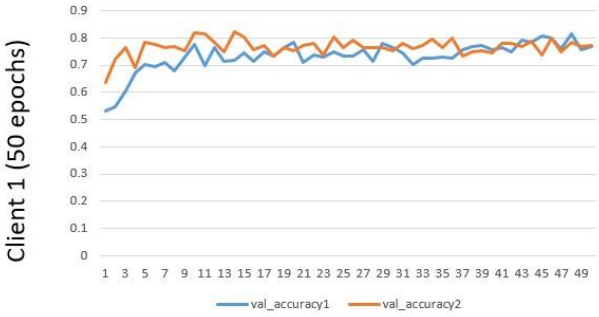

Fig. 9. Federated Setup for 50 epochs and 2 rounds on MNIST (Client 1)

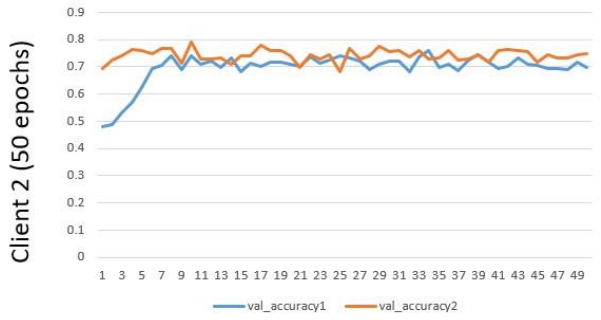

Fig. 10. Federated Setup for 50 epochs and 2 rounds on MNIST (Client 2)

Fig. 9 and Fig. 10 describe the improvement of accuracy from the start of training to 50 epochs for 2 rounds of training. Their values are like those of 25 epochs and show similar improvement. The values at the server side where the aggregation of model is performed with the help of FedAvg algorithm are 86.84% for 25 epochs training and 95.12% for 50 epochs of training which shows there is a gradual increase of performance with increment in rounds. This is the reason the experiment has not been conducted for a greater number of rounds as there is no major improvement in accuracy.

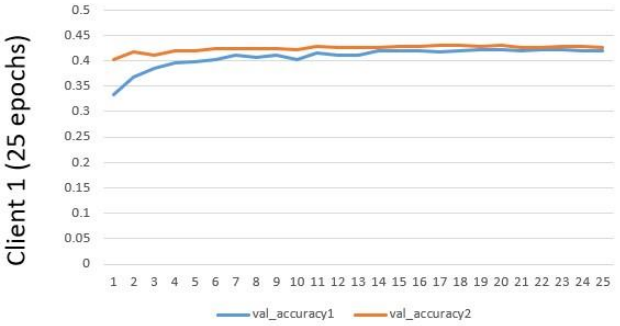

Fig. 11. Federated Setup for 25 epochs and 2 rounds on CIFAR-10 (Client 1)

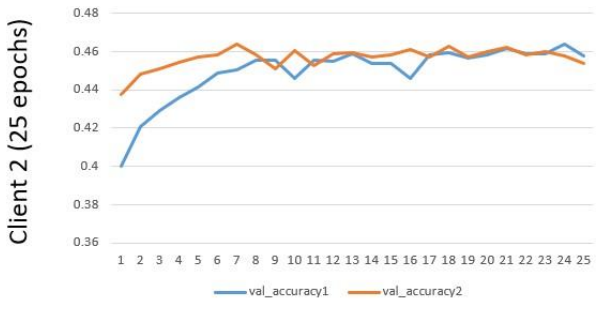

Fig. 12. Federated Setup for 25 epochs and 2 rounds on CIFAR-10 (Client 2)

Fig.11 and Fig.12 represent the federated setup results for CIFAR-10 dataset for 25 epochs and 2 rounds for Client 1 and Client 2 respectively. As we can see that there is not actually so much improvement happening in the case of this dataset, we can see that there is a small learning curve.

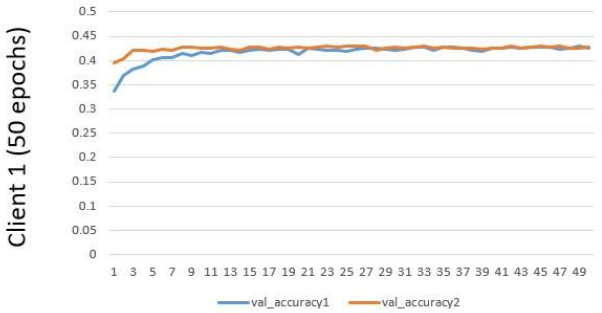

Fig. 13. Federated Setup for 50 epochs and 2 rounds on CIFAR-10 (Client 1)

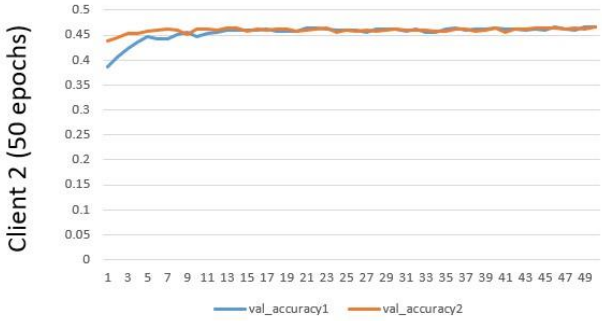

Fig. 14. Federated Setup for 50 epochs and 2 rounds on CIFAR-10 (Client 2)

Fig. 13 and Fig.14 represent the federated setup for 50 epochs and 2 rounds of training on CIFAR-10 dataset. These results are same as seen for 25 epochs and not much improvement as compared to MNIST dataset. The server-side values where model aggregation using the FedAvg algorithm is performed are 47.82% for 25 training epochs and 50.75% for 50 training epochs, which shows a gradual increase in performance as the number of training epochs increases.

In the implementation of the Federated Machine Learning Setup, various Aggregation Algorithms are employed to combine the values from the clients and train the server model. Specifically, the FedAvg Algorithm is utilized to aggregate the insights gathered from all client models and facilitate the training of the main model on the server. The FedAvg Algorithm plays a crucial role in efficiently managing client connections to the server for effective data sharing and communication with the clients, ensuring an optimized learning process.

FedAvg algorithm functions on the idea of iteratively averaging the model parameters from each client where data from each client is always kept private.[7]

Federated Learning is preferred where the data is to be distributed across clients and should have a secure communication for data transfer among the distributed nodes rather than centralizing the data with a single node.

4.2. Comparison of Traditional Machine Learning and Federated Machine Learning

The same model is used for training in both cases to maintain the same architecture for comparison purposes, just with a change in the input shape based on the dataset trained and tested upon.

The parameters considered for the comparison purposes are Precision, Recall, F1 Score.

Precision is considered as the model's efficiency to predict the positive instances, whereas Recall is used to measure how many positive instances that model can correctly identify among the available ones and finally F1-Score, it is generally stated as combination of precision and recall.

From Fig. 3 and Fig. 5 it can observed the difference between the working model of traditional machine learning training to Federated machine learning, the process for traditional machine learning is a straightforward whereas in federated machine learning the clients are connected to the server for global model updates.

The results obtained for Traditional Machine Learning Setup are as follows-

Class	Precision	Recall	F1-score
0	0.80	0.77	0.74
1	1.00	0.80	0.90
2	1.00	0.82	0.83
3	0.78	0.79	0.81
4	0.79	0.82	0.92
5	0.77	0.78	0.88
6	0.80	0.70	0.80
7	0.82	0.73	1.00
8	0.81	0.80	0.81
9	0.77	0.86	0.75

Table 1. Parameter Values for MNIST Non-Federated Setup

Recorded values for parameters such as Precision, Recall, and F1-Score are available. The values depicted in Table 1 aid in comprehending the model's accuracy in identifying images without errors effectively. These values were recorded after training the model for 25 epochs using the Traditional approach on MNIST Hand-Written Digits Image Dataset.

Class	Precision	Recall	F1-score
0	1.00	0.80	0.81
1	0.83	1.00	0.92
2	0.80	0.83	0.87
3	0.76	0.80	0.82
4	0.87	0.76	0.85
5	0.90	0.82	0.77
6	0.78	0.80	0.72
7	0.80	0.77	0.83
8	0.76	0.84	0.80
9	0.83	0.79	0.71

Table 2. Parameter Values for CIFAR-10 Non-Federated Setup

The values in Table 2 show us how accurate the model is and how efficient it is in detecting the positives. We have trained the model for better accuracy where the initial training was for only 25 epochs and later, we improved our model, and we later trained the models for 50 epochs, and we observed a decent growth in its accuracy and efficiency. These are the observations that we have obtained from training our models on MNIST and CIFAR-10 datasets using the Traditional Machine Learning Setup.

These values help us understand the way the model's efficiency is being increased and how to fine tune our model to accommodate the upcoming changes in the datasets.

Similarly, we have trained models of Federated Setup on both the datasets i.e., MNIST and CIFAR-10. Where we have two clients and a server model, we can configure the number of clients that can be attached to a server while configuring our server.

While experimenting, we configured our server to have two clients and we recorded the aggregated values from the server and analysed them. We tried out different variations in training the model i.e., 25 epochs for a single round and 50 epochs with 2 rounds and we have aggregated the values for each round and had a detailed analysis on how server is responding when we train for various patterns.

The comparison for 25 epochs of training for both MNIST and CIFAR-10 is mentioned below in details in tabular form. The training is for 25 epochs 2 rounds, 50 epochs 1 round and 50 epochs 2 rounds are discussed.

Class	Precision	Recall	F1-score
0	0.53	0.42	0.37
1	0.45	0.51	0.49
2	0.48	0.43	0.45
3	0.58	0.50	0.37
4	0.32	0.44	0.48
5	0.41	0.56	0.49
6	0.44	0.41	0.51
7	0.36	0.47	0.32
8	0.42	0.35	0.46
9	0.49	0.38	0.44

Table 3. Parameter Values for CIFAR-10 Federated Setup (25 epochs, 2 rounds)

Table 3 helps us understand the trends of efficiency and accuracy of the model. We have trained the model for 25 epochs and 2 rounds. All the values in the Image are from both clients after

training both the clients and the server contain the aggregated values. The accuracy of the above experiment comes to 47%.

Class	Precision	Recall	F1-score
0	0.31	0.48	0.48
1	0.49	0.34	0.45
2	0.54	0.38	0.52
3	0.48	0.41	0.46
4	0.41	0.45	0.39
5	0.39	0.38	0.47
6	0.46	0.41	0.50
7	0.50	0.57	0.48
8	0.48	0.38	0.45
9	0.35	0.44	0.38

Table 4. Parameter Values for CIFAR-10 Federated Setup (50 epochs, 1 round)

Table 4 helps us to understand the variations in various parameters of model when the model is trained for 50 epochs and a single round which is one of the variations of testing for improvement. The accuracy for the above task of training is 48%. Then to check the progress Table 5 helps to understand whether the model can save the results of end of round 1 and replicate to round 2 the start of training.

Table 5 describes when CIFAR-10 dataset is trained for 50 epochs and 2 rounds where the server aggregates both the client values 3 times, which is the result shown in the image.

Table 3, Table 4, Table 5 help us to analyse how model is performing when it is trained in different patterns. There is a decent difference of model's performance when we trained it for more epochs than the model that is trained for lesser epochs and when we increase the number of rounds there is a slight increase in performance of the model.

Class	Precision	Recall	F1-score
0	0.46	0.44	0.50
1	0.53	0.48	0.39
2	0.41	0.32	0.46
3	0.44	0.55	0.38
4	0.39	0.48	0.41
5	0.48	0.41	0.47
6	0.46	0.37	0.35
7	0.41	0.39	0.34
8	0.45	0.42	0.47
9	0.47	0.45	0.48

Table 5. Parameter Values for CIFAR-10 Federated Setup (50 epochs, 2 rounds)

In Comparison with Traditional Machine Learning Setup and the Federated Setup we can observe a difference in accuracy as the data is being distributed across clients and the aggregated values are used to train the central server, yet federated is preferred over Traditional Method as we can flexibly use data across various clients than centralizing the whole data with a single client which leads lo usage of heavy compute and storage resources. But the results show that there can be significant improvement and is not at par with the results to Traditional training.

Class	Precision	Recall	F1-score
0	0.70	0.67	0.85
1	0.77	0.80	0.81
2	0.80	0.77	0.78
3	0.81	0.89	0.80
4	0.82	0.78	0.88
5	0.62	0.87	0.83
6	0.79	0.69	0.81
7	0.72	0.75	0.77
8	0.78	0.74	0.78
9	0.71	0.82	0.82

Table 6. Parameter Values for MNIST Federated Setup (25 epochs, 2 rounds)

Table 6 helps us understand various values of model such as Precision, Recall, F1-Score and Accuracy. The Values are calculated for a model trained on MNIST Datasets for 25 epochs and 2 rounds. The accuracy obtained is 85.1%.

Class	Precision	Recall	F1-score
0	0.78	0.74	0.82
1	0.82	0.87	0.81
2	0.71	0.84	0.74
3	0.82	0.77	0.86
4	0.65	0.80	0.78

5	**0.70**	**0.79**	**0.84**
6	**0.72**	**0.82**	**0.84**
7	**0.75**	**0.81**	**0.78**
8	**0.77**	**0.74**	**0.77**
9	**0.69**	**0.76**	**0.65**

Table 7. Parameters MNIST Federated Setup (50 epochs,1 round)

We observed the difference in Various Parameters when we compared the model with the model that is trained for 50 epochs in Table 7 and there is a slight increase in model's efficiency. The accuracy obtained is 95.3% which is a drastic improvement when compared to 25 epochs of training.

Class	Precision	Recall	F1-score
0	**0.78**	**0.83**	**0.74**
1	**0.81**	**0.75**	**0.77**
2	**0.74**	**0.68**	**0.93**
3	**0.80**	**0.85**	**0.80**
4	**0.69**	**0.79**	**0.69**
5	**0.73**	**0.80**	**0.86**
6	**0.82**	**0.78**	**0.68**
7	**0.77**	**0.75**	**0.75**
8	**0.82**	**0.82**	**0.87**
9	**0.81**	**0.76**	**0.85**

Table 8. Parameter Values for MNIST Federated Setup (50 epochs, 2 rounds)

Table 8 is intended to show that there is a significant increase in the aggregated values when the time is increased for training as there are a greater number of epochs and since the aggregation at server is happening multiple times. The accuracy obtained is 96.8% on the server side.

For training our models we tried out 3 variations where one is 25 epochs with 2 rounds and the second is 50 epochs and single round and final variation is 50 epochs with 2 rounds. This experimentation is restricted to 2 rounds as with respect to CIFAR-10 data set there is no improvement or in some cases negligent improvement being shown to the training and with respect to MNIST data set it is at its best accuracy and reaches at par with the traditional training setup. When Compared to Traditional Machine Learning, Federated Machine Learning is a bit complex in implementation, but it has a great impact where data must be shared with various clients, whereas Traditional Machine Learning has fixed data in terms of input and cannot be changed as required. On Comparing various parameters, we could say that Federated Machine Learning was far more effective and efficient when the data nodes are more, whereas Traditional Machine Learning can be used where the data is fixed to be trained upon. When data is missing for a particular class, it is observed that federated machine learning can average the learning and provide better results compared to non-federated machine learning. By observing the records of variations in training, there is a gradual increase when we repeatedly train our model, and once we cross the bearing out, the model starts to overfit when it remembers the data and will not be able to predict the data. On analyzing the parameters, we can observe Traditional Setup gives us a more efficient model than compared to Federated Setup. It helps us to have more accurate outcomes as compared to Federated Setup. It also requires less compute power [12] as the data is not trained multiple times as in the case of federated Setup. When comes to comparison of the accuracy of predictions of models trained in various patterns, In Federated Learning the number of epochs and number of rounds increases, the accuracy of client's model increases, which helps us to gather precise learning and feed it back to the server to train the Server model than sharing its whole data and training our server's model on whole data set.[15]

In a comprehensive comparison of the performance between Federated and Traditional Setups of Machine Learning, the Traditional Setup demonstrates superior accuracy in predictions across all facets. The centralized nature of the Traditional Setup allows for enhanced precision and reliability in modeling outcomes. While the Federated Setup proves advantageous in scenarios where data is dispersed among numerous nodes, it is evident that the Traditional Setup outperforms the Federated counterpart even when data decentralization is not a factor.

5. Conclusion and Future Work

In our comprehensive experimentation, it became evident that the Traditional Machine Learning setup outperforms Federated Machine Learning, particularly in scenarios involving limited data and centralized data repositories. Even with a mere 50% data partition, the Traditional Machine Learning approach showcased initial superiority. Despite the competitive performance of the Federated Setup—achieving 92%-96% accuracy for MNIST and approximately 45%-50% for CIFAR-10 datasets across various rounds and 25 to 50 epochs—the non-federated setup exhibited higher accuracy, reaching around 96% for MNIST and 85% for CIFAR-10 datasets. These findings underscore the potential of Federated Machine Learning while acknowledging the necessity for ongoing refinement to enhance its effectiveness in decentralized machine learning environments.

However, the scope of this investigation extends beyond the confines of this paper. Future explorations aim to delve into Secure Aggregation and Data Privacy principles, emphasizing security measures to safeguard client data from unauthorized access.

The integration of IoT devices into Federated Machine Learning (FML), particularly for predictive maintenance, presents promising opportunities for device management improvement. Nevertheless, challenges such as handling diverse device types, ensuring security, and managing limited resources need to be addressed. Subsequent research endeavors should concentrate on developing robust methodologies that strike a balance between efficiency, security, and customization for the distinctive characteristics of IoT environments.

Acknowledgements

We express our gratitude to the publisher for their assistance in publishing this research article. The resources and platform offered by the publisher have allowed us to disseminate our discoveries to a broader audience. We value the diligent work of the editorial team in evaluating and refining our research, and we are

appreciative of the chance to make a meaningful contribution to the research field through this publication.

Author contributions

Singanamalla Jaya Mohnish and Gandra. Shiva Krishna: Participated in all experiments, implementing the research work, and writing the manuscript.

Kota Venkata Narayana: Designed the research plan and conducted organized study.

Jonnalagadda Surya Kiran, Thatavarti Satish and Inbarajan. P: Defining research play, study supervision, result analysis, manuscript reviewed discussion.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Chan, Y. H., & Ngai, E. C. (2021). FedHe: Heterogeneous models and communication-efficient federated learning. 2021 17th International Conference on Mobility, Sensing and Networking (MSN). https://doi.org/10.1109/msn53354.2021.00043

[2] Doon, R., Kumar Rawat, T., & Gautam, S. (2018). Cifar-10 classification using deep Convolutional neural network. 2018 IEEE Punecon. https://doi.org/10.1109/punecon.2018.8745428

[3] Ghosh, S. (2022). Comparative analysis of boosting algorithms over MNIST handwritten digit dataset. Evolutionary Computing and Mobile Sustainable Networks, 985-995. https://doi.org/10.1007/978-981-16-9605-3 69

[4] Grafberger, A., Chadha, M., Jindal, A., Gu, J., & Gerndt, M. (2021). FedLess: Secure and scalable federated learning using Serverless computing. 2021 IEEE International Conference on Big Data(BigData). https://doi.org/10.1109/bigdata52589.2021.9672067

[5] Ilias, C., & Georgios, S. (2019). Machine learning for all: A more robust federated learning framework. Proceedings of the 5th International Conference on Information Systems Security and Privacy.

[6] https://doi.org/10.5220/0007571705440551

[7] Kar, B., Yahya, W., Lin, Y., & Ali, A. (2023). Offloading using traditional optimization and machine learning in federated cloud–edge–Fog systems: A survey. IEEE Communications Surveys & Tutorials, 25(2),1199-1226. https://doi.org/10.1109/comst.2023.3239579

[8] Lee, C., & Lee, W. (2022). Participant selection scheme of federated learning in Non-IID data distribution environment. The Journal of Next generation Convergence Technology Association, 6(11), 2063-2075. https://doi.org/10.33097/jncta.2022.06.11.2063

[9] Li, K. H., De Gusm˜ao, P. P., Beutel, D. J., & Lane, N. D. (2021). Secure aggregation for federated learning in flower. Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning. https://doi.org/10.1145/3488659.3493776

[10] Lin, S., Zhou, Z., Zhang, Z., Chen, X., & Zhang, J. (2021). Edge intelligence via federated meta-learning. Edge Intelligence in the Making, 53-79. https://doi.org/10.1007/978-3-031-02380-4 3

[11] Mahanan, W., Chaovalitwongse, W. A., & Natwichai, J. (2021). Data privacy preservation algorithm with K-anonymity. World Wide Web, 24(5), 1551-1561. https://doi.org/10.1007/s11280-021-00922-2

[12] Ram Mohan Rao, P., Murali Krishna, S., & Siva Kumar, A. P. (2018). Privacy preservation techniques in big data analytics: A survey. Journal of Big Data, 5(1). https://doi.org/10.1186/s40537-018-0141-8

[13] Sun, G., Cong, Y., Dong, J., Wang, Q., Lyu, L., & Liu, J. (2022). Data poisoning attacks on federated machine learning. IEEE Internet of Things Journal, 9(13), 11365-11375. https://doi.org/10.1109/jiot.2021.3128646

[14] Sandholm, T., Mukherjee, S., & Huberman, B. A. (2022). Demo — SPoKE. Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. https://doi.org/10.1145/3548606.3563701

[15] Tsuji, K., Imai, S., Takao, R., Kimura, T., Kondo, H., & Kamiya, Y. (2021). A machine sound monitoring for predictive maintenance focusing on very low frequency band. SICE Journal of Control, Measurement, and System Integration, 14(1), 27-38. https://doi.org/10.1080/18824889.2020.1863611

[16] Turina, V., Zhang, Z., Esposito, F., & Matta, I. (2020). Combining split and federated architectures for efficiency and privacy in deep learning. Proceedings of the 16th International Conference on emerging Networking Experiments and Technologies. https://doi.org/10.1145/3386367.3431678

[17] Wahab, O. A., Mourad, A., Otrok, H., & Taleb, T. (2021). Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems. IEEE Communications Surveys & Tutorials, 23(2), 1342-1397. https://doi.org/10.1109/comst.2021.3058573

[18] Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. Pattern Recognition Letters, 141, 61-67. https://doi.org/10.1016/j.patrec.2020.07.042

[19] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning. ACM Transactions on Intelligent Systems and Technology, 10(2), 1-19. https://doi.org/10.1145/3298981