# Semi Supervised Federated Learning with pseudo-labeling

*A Project Report*

*submitted by*

**KAVYA GUPTA**

*in partial fulfilment of the requirements*
*for the award of the degree of*

**MASTER OF TECHNOLOGY**

ELECTRONICS AND COMMUNICATION ENGINEERING
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

**Date of Thesis Completion**

# THESIS CERTIFICATE

This is to certify that the thesis titled **Semi Supervised Federated Learning with pseudo-labeling**, submitted by **Kavya Gupta**, to the Indraprastha Institute of InformationTechnology, Delhi, for the award of the degree of **Master of Technology with specialisation in Machine Learning**, is a bona fide record of the research work done by her under the supervision of Prof. Dr. Ranjitha Prasad. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

**Prof. Dr. Ranjitha Prasad**
Thesis Supervisor
Associate Professor
Dept. of Electronics and Communication
IIIT Delhi, 110020

Place: New Delhi

Date: 26th March 2024

# ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Dr. Ranjitha Prasad, for all her help and support during this research project. Additionally, I would want to express my gratitude to IIIT Delhi for providing the tools and space needed to finish this thesis. My family deserves special recognition for their constant love and support. Furthermore, I would want to express my gratitude to each and everyone who helped me with this endeavor in any way. This thesis has greatly benefited from your support.

# ABSTRACT

KEYWORDS: Federated Learning ; Semi-supervised learning ; Pseudo-labeling ; Autoencoder ; Joint learning

In order to efficiently learn from small amount of labeled data, this study presents pseudo- labeling using semi-supervised learning in a federated setting (Pseudo-FedSSL), a novel approach to semi-supervised federated learning that makes use of autoencoder-derived latent vectors and pseudo-labeling. Using this method, latent vectors from labeled data are aggregated to create unique vectors for every class. Subsequently, the unlabeled data is pseudo-labeled by calculating the distance between each distinct vector obtained from the labeled data and its latent vector. The class with the smallest distance determines the pseudo- label assignment, enhancing the model's capacity to efficiently label unannotated samples. Pseudo-FSSL makes use of training an autoencoder and its transfer learning capacity to capture complex data representations and relationships. In addition to adding to the expanding body of federated learning approaches, the suggested pseudo-FSSL method offers a dependable and scalable alternative for semi-supervised learning, along with increasing classification accuracy.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABBREVIATIONS

| | |
|---|---|
| **FL** | Federated Learning |
| **AE** | Autoencoder |
| **SSL** | Semi-supervised Learning |
| **ML** | Machine Learning |
| **FSSL** | Federated Semi-supervised Learning |

# NOTATION

| | |
|---|---|
| $D'_k$ | Local dataset |
| $(x_k, y_k)$ | Labeled datapoint of client k |
| $D_L^k$ | Labeled dataset of client k |
| $D_U^k$ | Unabeled dataset of client k |
| $L$ | Dimension of latent space |
| $M$ | Dimension of input space |
| $Z$ | Latent space |
| $X$ | Input space |
| $z_i^U$ | Unlabeled latent vector |
| $z_i^L$ | Labeled latent vector |
| $\mu_c$ | Unique latent vector of a particular class c |
| $d_i^c$ | Euclidean distance between unlabeled latent vector and unique latent vector of a particu |
| $\tau$ | Confidence threshold used in pseudo-labeling |
| $y_i'^U$ | Pseudo-label given to unlabeled datapoints |
| $N_{\textbf{combined}}$ | Total no. of datapoints in pseudo-labeled and already labeled dataset |

# CHAPTER 1

# INTRODUCTION

Federated data representation is a vital step for efficient communication, privacy protection, and flexibility. Since the models are trained on multiple decentralised edge devices in federated learning (FL), good data representation ensures that raw data stays local, thereby resolving privacy issues. Reduced bandwidth consumption from enhanced data representations permits the model changes to be provided without straining the network's capacity. Additionally, appropriate data representations allow models to adjust to the heterogeneity of various edge devices, promoting efficient and customised machine learning (ML) while adhering to particular device limits. Wen et al.'s thorough survey [1] explores the shortcomings and uses of federated learning and provides insights into its real-world applications. Effective representation guarantees significant model understanding, which enables group learning without sacrificing privacy. Strong data models enable FL to extract meaningful information from several sources, improving overall performance and generalisation as depicted by Zhang et al. [2],. To put it briefly, data representation plays a critical role in making federated learning successful in distributed and privacy-sensitive situations.

Various semi-supervised FL techniques frequently combine both supervised and unsupervised learning strategies. In addition to making effective use of the large amount of unlabelled data, federated semi-supervised learning (FedSSL) allows for the efficient use of very scarce labelled data. Methods like pseudo-labelling[3], co-training, and self-training are used to leverage the combining of labelled and unlabelled samples, resulting in improved model performance. Autoencoders[4] are essential in this domain as they improve model understanding by decoding unlabelled data which is clearly portryed by Zhao et al [5]. AE makes use of latent representations to interpret complex patterns in unlabelled datasets, allowing robust learning over edge devices. By increasing the use of available information, integrating autoencoders promotes better model performance.

FL, being a promising approach, has numerous drawbacks. A major obstacle is the privacy issue[6], since maintaining the secrecy of local data makes global model

aggregation more difficult. Restrictions on communication could make it difficult for edge devices and the central server to exchange data effectively. Managing heterogeneity across different client data distributions is a significant difficulty that hampers the performance of the global model. The complexity is increased by addressing some of the potential problems, like the non-identically distributed [7] (non-IID) nature of client data and the effect of client dropout during training due to connectivity issues. Also, the main obstruction encountered by semi-supervised setting in FL is the availability of very scarce amount of labelled data

By giving labelled data priority, supervised federated learning appears as a workable answer to the above problem. Model initializations are better due to the explicit use of labelled samples, which leads to quicker convergence and higher accuracy. It also reduces the possibility of biassed representations resulting from the incorporation of a large amount of unlabelled data and offers a more regulated and efficient method that overcomes the drawbacks of its semi-supervised version and promotes upgraded cooperative learning in dispersed settings. Relying only on labelled input ensures a more uniform learning process across dispersed nodes and reduces variability in model training.

In my proposed work, two major principles can be observed:

1. Transfer Learning: Here, the information from one task, which is training the autoencoder, is being utilized and applied to a different task of pseudo-labelling. The autoencoder takes labelled data and acts as a feature extractor, pointing out the important patterns and representations. The target job of pseudo-labelling unannotated data in a semi-supervised federated learning environment is then applied using this knowledge. This type of setup where one method is leveraged to gain insights on another was depicted in FedDC[8] too. Such setup helps in generating distinct latent vectors for every class and improving the model's capacity to efficiently pseudo-label unlabeled samples and generalize.

2. Distance-based classification: This approach makes use of the presumption that comparable latent vectors belong to similar classes, allowing for accurate and efficient pseudo-labelling. The distances[9] between unannotated sample and unique vectors representing each class are calculated in order to assign pseudo-labels to the samples. For the unannotated sample, the pseudo-label belongs to the class with the smallest

distance

# CHAPTER 2

# RELATED WORKS

## 2.1 Pseudo-labeling

In the setting of semi-supervised federated learning, pseudo labelling becomes an essential strategy that helps in dealing with the difficulties caused by a lack of labelled data on a distributed scale. Pseudo labelling provides a realistic way to mark unlabelled data in a federated system without consolidating sensitive data, where data privacy is of maximal significance. In [10], pseudo labeling is employed through the creation of pseudo histopathology pictures from every center, which helps to train the main network. The process, as it is highlighted in the context of Pseudo-FSSL, is based on the creation of distinct vectors for every class through the use of autoencoder-derived latent vectors. With this method, the model can independently pseudo-annotate unlabeled samples on different devices by measuring the separations between their latent vectors and class-specific vectors obtained from locally labelled data. Using the average of the pseudo-labels generated separately by every client, DS-FL[11] presents an ensemble pseudo-labeling technique which utilises the consolidated insights from various clients thereby establishing a robust method. Pseudo labelling is pivotal as it permits devices to autonomously contribute to the learning process by utilizing the most of the labelled data that is available. This not only ensures data privacy concerns but also improves model accuracy by integrating insights from numerous local datasets, but it also fosters a collaborative learning environment without jeopardising the originality of individual data.

## 2.2 Transfer Learning

In the field of machine learning, transfer learning is elemental since it alters model training by leveraging the knowledge obtained from one job and applying it to another. This paradigm change in learning methodology has shown to be invaluable in

a variety of fields, allowing models to more productively establish and adapt to new tasks. PfedHN[12] dynamically creates personalised models by modifying pre-existing knowledge to certain input conditions in the federated learning condition through the use of Hyper-networks, a type of transfer-learning. Transfer learning[13] often operates by pre-training a model on a source task, where it picks up useful characteristics, features, and representations. After applying these newly gained insights to a specific task, the model is able to improve its performance with possibly less labelled data by enlarging on its knowledge base and improvising its comprehension. This procedure is especially useful in situations when there is a very scarcity of labelled data since it enables models to leverage the knowledge that already exists to improve performance on new and related tasks.

## 2.3 Unsupervised Learning

In the reference of unsupervised learning, an algorithm's main goal is to find patterns and structures in data without the need for definite instructions or labelled samples. When working with large and complicated information, this method is especially helpful since it can bring out hidden patterns and insights that may be tedious for human experts to see. In federated settings, unsupervised learning methodologies can be leveraged to cluster and study data distribution among different participating devices. FedUL[14] converts unlabeled data into surrogate labelled data by utilizing altered models that are trained by supervised federated learning for each client. Federated unsupervised learning reduces privacy concerns while allowing collaborative model training[15][16] by picking out patterns and similarities without disclosing individual data points. This strategy finds use in a number of industries, including IoT, healthcare, and finance due to its ability of bringing out important insights without jeopardising the privacy of individual data, fostering a safer and more collaborative machine learning procedure.

## 2.4 Autoencoders

Neural network designs called autoencoders are meant for use unsupervised learning. They compress input data into a lower-dimensional space and reconstruct the original input from this compressed representation. This process permits autoencoders to learn essential representations of data. FedMAE[17] utilizes autoencoder for pretraining part wherein it generates encoding from masked images i.e. only 25 percent of full image for asynchronous training. Autoencoders are specifically useful in the area of federated learning because of their ability to capture important aspects of the data. Compact representations of local data can be created on individual devices via autoencoders, terminating the need to transmit raw data to a central server. They facilitate relevant features extraction from unlabeled input without the requirement for explicit labels. In federated systems, where data among different devices may have different qualities, this technique seems to be extremely helpful.

**Novelty in the work:** This approach offers a number of innovative developments in the field of semi-supervised federated learning. It makes use of federated learning in co-existence with semi-supervised learning techniques. The method deals with the hassle of training models on datasets with a scarcity of labeled data while utilizing the huge amounts of unlabeled data available across several clients by integrating pseudo-labeling into the federated learning framework. This connection makes it possible to use data resources more effectively and may even enhance model performance. First of all, this novel strategy of using autoencoders to produce latent vectors from labeled and unlabeled data is noteworthy as it aids in capturing meaningful representations of data in an unsupervised way. This, not only encodes both types of data but also makes it easier to extract higher amount latent features using the encoding along with its labels. It highly helps in classification tasks. Second, the new aspect in pseudo-labeling method is that it uses a threshold value to label unlabeled data points according to how far away they are from the centroids of labeled data clusters in the latent space. This mechanism for evaluating the degree of confidence in pseudo-label assignments manages the sensitivity of label assignments by selecting an appropriate threshold, which may enhance the labeling process's accuracy.

# CHAPTER 3

# MATHEMATICAL PRELIMINARIES

FL[18] utilizes separate local labelled datasets $D'_k$ to optimize a global model, parameterized by $\phi \in R^{d\times 1}$ among various decentralised clients. The global objective function $F_k(\phi)$ measures the quality of the collective model. Using the loss function $l(\phi;\cdot)$, every client, indexed by k, computes a local objective function $f_k(\phi)$ given by

$$f_k(\phi) = \frac{1}{N_k}\sum_{i=1}^{N_k} l(\phi; x_{k,i}, y_{k,i})$$

where $(x_{k,i}, y_{k,i}) \in D'_k$ and $\sum_{k=1}^{K} N_k = \mathrm{N}$ . When dealing with FL, we have to keep in mind the possibility of unavailability of labels which means absence of $\mathrm{y}_{k,i}$. Having the knowledge of labels can be an expensive task. Here we are particularly concerned with semi-supervised learning methodology which means we have very less amount of labelled data and we have to leverage it to label the huge amount of unlabelled data. Hence, it means that every client has some amount of labelled data represented by $\mathrm{D}_L^k = \{\mathrm{x}_{k,i}, y_{k,i}\}_{i=1}^{N_k}$ and large amount of unlabelled data represented by $\mathrm{D}_U^k = \{\mathrm{x}_{k,i}\}_{i=1}^{N_k}$.

Since we are dealing with limited data availability, we try to obtain as much information as possible from this data itself. Autoencoder[19] is one such model which can be pre-trained in an unsupervised manner with the existing unlabeled data. Without explicit labels, the pre-trained autoencoder learns meaningful representations from the incoming data in the role of a feature extractor. Important aspects of the data are captured in these representations, which can help with semi-supervised learning tasks later. Better performance results from the model's ability to generalise more effectively to unlabeled input by utilising the wealth of information contained in the latent space. An autoencoder is composed of an encoder and a decoder. Input data is mapped to a lower-dimensional latent space by the encoder, which is usually a set of completely connected layers. The function mapping $\mathrm{g}(\cdot;\phi) : X \to Z$ where X is M dimensional input space and Z is L dimensional latent space $\mathrm{L} \ll M$. The low dimensionality of

latent space ensures that only the relevant features of the input data is extracted thereby ensuring that neural networks don't deal with huge data with much more complexity.

This means for $z_i \in Z$ and $x_i \in X$; $z_i = g(x_i; \phi)$ where $z_i$ is the latent space representation of input $x_i$. Similarly, the decoder model can also be expressed mathematically. It is parameterised by $\psi$ and can be understood as function mapping from latent space to input space expressed by $h(\cdot; \psi) : Z \rightarrow X$. Together, the encoder and decoder forms an autoencoder, whose quality depends on how well the input can be reconstructed. The main aim of an autoencoder is to reduce the reconstruction error $L(\cdot)$. This error represents the difference between the input and output of an autoencoder. The goal of an autoencoder is to efficiently find the mapping functions $h(\cdot; \psi)$ and $g(\cdot; \phi)$ such that $\phi, \psi = \sum_{i=1}^{N} L(x_i, h(g(x_i; \phi); \psi))$

In this work, the concept of joint training comes into effect as each client has very small amount of labeled data and huge amount of unlabeled data. The scarce amount of these labels can be leveraged by the autoencoder to learn the representations better. The idea of using a classifier loss along with the reconstruction loss of an autoencoder is of much use for learning the representations better. A scaled cross entropy loss for classifier can be added with the reconstruction loss and this joint loss can be used for training the model. Now, classification using only the highly scarce amount of labelled data might tend the model to overfit hence the idea of pseudo-labelling can also be incorporated alongside joint training. Pseudo-labelling technique in this setup would improve learning by offering more training instances. This makes the model more robust and more generalizable. A confidence-threshold based pseudo-labelling technique is used where unlabeled latent vector $z_i^U$ is provided some pseudo-label of the unique latent vectors belonging to some class 'c'. This unique latent vector is obtained from labeled latent vectors $z_i^L$ given by

$$\mu_c = \frac{1}{N_c} \sum_{i=1}^{N_c} z_i^L$$

. Here, $\mu_c$ represents unique latent vector of a particular class 'c' and $N_c$ is the total number of labeled samples belonging to that class. The mean of all the labeled latent vectors $z_i^L$ , belonging to class c represents the unique vector of that class. For any $z_i^U$, Euclidean distance $d_i^c$ is calculated with $\mu_c$,

$$d_i^c = \|z_i^U - \mu_c\|$$

and a pseudo-label $\text{y'}_i^U$ is given such that

$$y_i'^U = \begin{cases} c, & \text{if } d_i^c < \tau \\ \text{unlabeled}, & \text{otherwise} \end{cases}$$

where $\tau$ is fixed confidence threshold value. The combined dataset $\text{D}_{\text{combined}}$ prepared by pseudo-labeled dataset $\text{D}_{\text{pseudo}} = \{(x_i^U, y_i'^U)\}$ and already labeled dataset $\text{D}_L = \{(x_i, y_i)\}$ is used for classification task of joint training. A cross entropy loss[20] is calculated for $\text{D}_{\text{combined}} = (x_i^U, y_i'^U) \cup (x_i, y_i) = (x_i^{com}, y_i^{com})$ The labels of the combined dataset are considered as true labels. So, for a total of $\text{N}_{\text{combined}}$ datapoints and C no. of classes at any client, the cross entropy loss can be defined as

$$L_{CE} = -\frac{1}{N_{\text{combined}}} \sum_{i=1}^{N_{\text{combined}}} \sum_{j=1}^{C} y_{i,j}^{com} \log(\hat{y}_{i,j}^{com})$$

with $\text{y}_{i,j}^{com}$ as true labels and $\hat{y}_{i,j}^{com}$ as predicted labels.

## 3.1 Proposed work

It describes how can one utilise sparse amount of labels from the labeled data to understand the features of a whole lot of unlabeled data in a decentralised setup that too in a privacy-preserved manner.

In this federated learning framework, the process initiates at the client side, where each client performs pseudo-labeling using its autoencoder network. First encodings are generated for all the data samples $(x_{k,i})$ using the encoder function $g(\cdot;\phi)$. The unique latent vectors $\mu_c$ for each class 'c' are calculated by taking the mean of all labeled latent vectors $(z_i^L)$ belonging to that class. Thereafter, for each unlabeled sample $(x_i^U)$, the Euclidean distance $d_i^c$ from its latent vector to each class's unique latent vector $\mu_c$ is calculated. Based on a fixed confidence threshold $\tau$, pseudo-labels $y_i'^U$ are assigned to the unlabeled samples. This process guarantees that unlabeled data is pseudo-labeled with the most relevant class label, easing out joint training with labeled data.

Mathematically, the computation of $d_i^c$ can be given as

$$d_i^c = \|z_i^U - \mu_c\|$$

, and the assignment of pseudo-labels $y_i'^U$ is as follows

$$y_i'^U = \begin{cases} c, & \text{if } d_i^c < \tau \\ \text{unlabeled}, & \text{otherwise} \end{cases}$$

where $c$ represents the class label, and $\tau$ is the confidence threshold. After pseudo-labeling, a combined dataset $D_{\text{combined}}$ is obtained.

Meanwhile, the server side receives and processes the aggregated datasets from all clients. The server leverages a surrogate encoder and an actual decoder to generate representations and reconstruct the data, respectively. Here the actual decoder is fed with the latent space $\mathrm{z}_{k,i}$ of all samples obtained as an output from the actual encoder at every client. A representation space $\mathrm{y}_{k,i}$ is obtained which is again passed through the surrogate encoder of client to get reconstructed output $\mathrm{z}_{k,i}^s$. The mean squared error (MSE) loss is computed between the latent space $\mathrm{z}$ and reconstructed output $\mathrm{z}^s$ aiding in the training of the autoencoder.

$$L_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(z_i - \{z_i^s)^2$$

Simultaneously, at each client, autoencoder network i.e the actual encoder- surrogate decoder pair is being trained. The input data $x_{k,i}$ is being fed to the actual encoder at each client to obtain latent embeddings $z_{k,i}$. The reconstructed output received from server side is passed through the surrogate decoder of every client to get $x_{k,i}^s$. An MSE loss is then computed between $x_{k,i}$ and $x_{k,i}^s$.

$$L_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(x_{k,i} - x_{k,i}^s)^2$$

Furthermore, there is a classifier head present at each client which is trained using the combined dataset, consisting of both labeled and pseudo-labeled data, $(x_i^{com}, y_i^{com})$. This dataset is further used for training the, where a cross-entropy loss function $L_{CE}$ is

calculated. The cross-entropy loss measures the dissimilarity between true labels $y_{i,j}^{com}$ and predicted labels $\hat{y}_{i,j}^{com}$ for the combined dataset. Mathematically, the cross-entropy loss is defined as

$$L_{CE} = -\frac{1}{N_{\text{combined}}} \sum_{i=1}^{N_{\text{combined}}} \sum_{j=1}^{C} y_{i,j}^{com} \log(\hat{y}_{i,j}^{com})$$

, where $N_{\text{combined}}$ denotes the total number of datapoints in the combined dataset and $C$ represents the number of classes. So, the overall loss at client side is given by

$$L_{\text{final}} = a \times L_{\text{MSE}} + b \times L_{\text{CE}}$$

Thus, at client side, classifier and autoencoder, both are trained concurrently resulting in utilisation of the sparse labeled data at client side also. Due to this coordinated effort in between the various clients and server, federated learning enables the combined processing of the global model while maintaining data privacy and amongst the network of clients.

## 3.2 Algorithm

The objective functions at server and client side can be given as below,

Server:

$$\min_{\psi,\phi^s)} F(\psi, \phi^s) = \sum_{k=1}^{K} \sum_{i=1}^{N_k} L(z_{k,i}, g^s(y_{k,i}; \phi^s))$$

Client:

$$\min_{\phi_k,\psi_k^s,\theta_k} f_k(\phi_k, \psi_k^s, \theta_k) = \sum_{i=1}^{N_k} L_k(x_{k,i}, h_k^s(z_{k,i}^s; \psi_k^s)) + \sum_{i=1}^{N} L_{CE}(y_i^{com}, f_{class}(x_i^{com}; \theta))$$

For dataset $D_k$ at the $k$-th client (K number of total clients), $T$ - number of rounds, $E$ - number of local epochs and $\eta$ - learning rate, parameters are initialised at every client.
For $t = 1$ to $T$ and $e = 1$ to $E$,
each client $k \in K$ and Sample mini-batch $B_k \subset D_k$
Update local parameters: $\{\phi_k, \psi_k^s, \theta_k\} \leftarrow \{\phi_k, \psi_k^s, \theta_k\} - \eta \nabla f_k(\phi_k, \psi_k^s, \theta_k)$

Update global parameters: $\{\psi, \phi^s\} \leftarrow \{\psi, \phi^s\} - \eta \nabla F(\psi, \phi^s)$

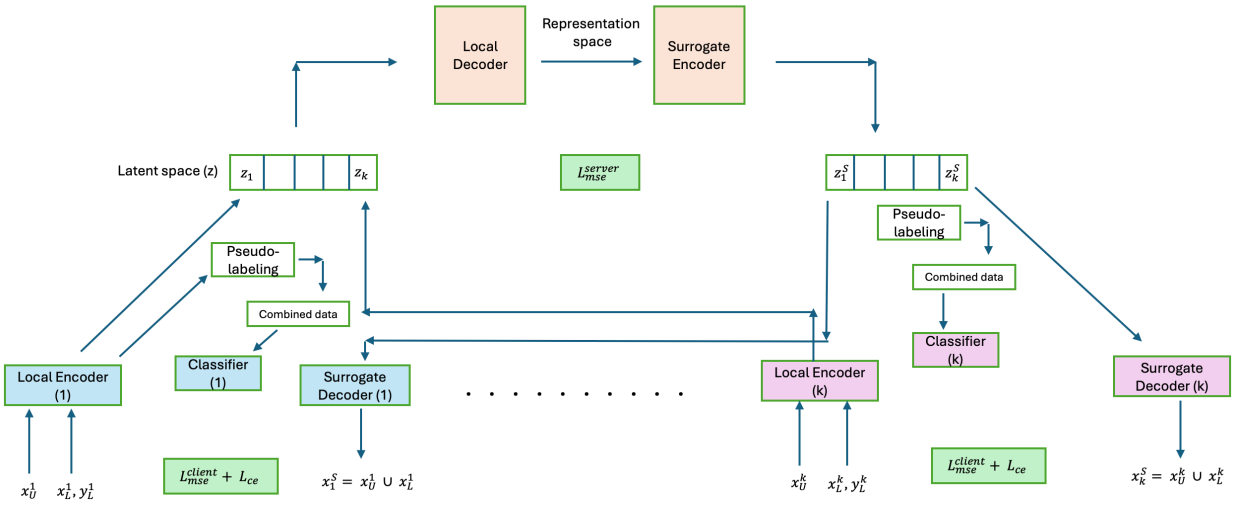


Figure 3.1: Architectural block diagram for proposed scheme, pseudoFSSL. The small amount of labeled data at each client helps in pseudo-labeling giving combined dataset made of already labeled data and pseudo-labeled data. This combined dataset is used to train the classifier and helps in Encoder-classifier joint training. The client side is trained using a scaled combination of both MSE loss and Cross entropy loss. The server side is trained using the MSE loss.

# CHAPTER 4

# EXPERIMENTAL SETUP AND RESULTS

A set of experiments have been carried out in this section to closely examine the vitality and performance of the proposed method under multiple settings. In the proposed scheme, the inherent model architecture used for autoencoder and classifier is of Resnet18 having four layers for convolutional and pooling operations, followed by residual blocks and an average pooling layer and finally ending with a fully connected layer. The decoder is formed using a linear and convolution area. The linear section comprises of fully connected layers with ReLu activations and the convolution area has three transposed convolution layers. An MSE loss function is used at server side while client side uses weighted average of MSE loss and Cross-entropy loss. Adam optimiser and Reduce LR on plateau learning rate scheduler is used on both client and server side. The various experiments are seen by changing different hyperparameters like learning rate, weight decay, no. of participating clients in each communication round, no. of local epochs and changing the percentage of labeled data at each client.

## 4.1 Choice of confidence threshold

A confidence threshold is used in the pseudolabeling process, experimentally the most suitable value for it has been found. A similar setup was kept intact and experiment was carried out on multiple values of confidence threshold. The data used here in CIFAR10, having a total of 60k samples, from which 50k are used as training data and rest as test data and each client from a total of 5 clients has 10% labeled data which makes 1000 samples as labeled at each client. To keep the comparision fair, experiment is conducted with 20 rounds and 2 local epochs. Learning rate used is 0.0001, a weight decay of 0.01 with batch size 8 and Adam optimiser. Different accuracies were obtained for different confidence threshold values between 0.15 to 0.195 and the optimum one was chosen. Initial values showed low accuracy indicating that a very minimal number of samples were getting pseudo labeled and reason for less accuracy for later values is

that the samples were getting wrongly pseudo-labeled. The best accuracy was obtained at threshold $\bar{0}.165$ which can be seen in Figure 4.1.

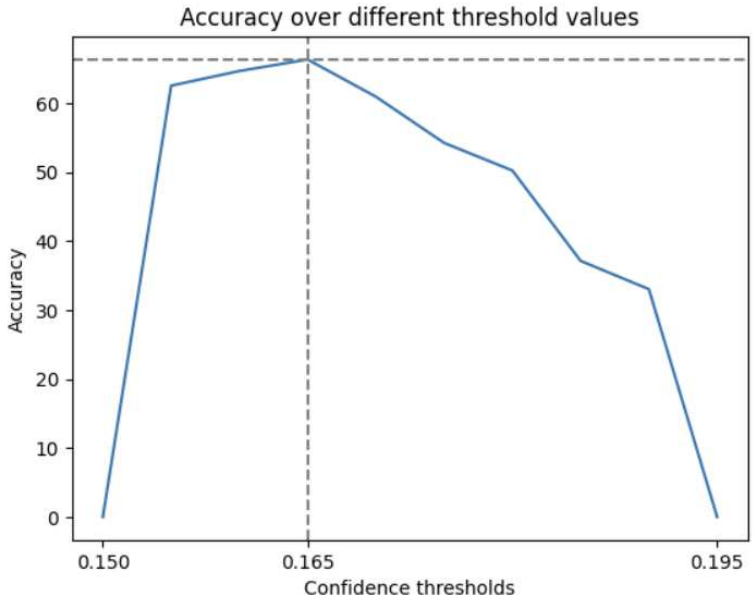


Figure 4.1: Best confidence threshold value for Cifar10 under IID setting

## 4.2 Comparision with different datasets

To check on the working and vitality of the proposed method, two datasets are used and results have been reproduced for both of them. MNIST dataset in dataset having images of handwritten numbers from 0 to 9. It comprises of of 28x28 grayscale images of handwritten digits along with corresponding labels pointing at each digit. It acts as a benchmark dataset for developing, exploring and evaluating algorithms in tasks such as recognition and classification. It has a total of 60k training samples and 10k testing samples. The Fashion MNIST (FMNIST) dataset, which includes grayscale images divided into ten different categories, is heavily utilized for image classification goals. All of the images in FMNIST are 28x28 pixel representations of multiple fashion items, including dresses, shoes, pants, and t-shirts. The dataset can be leveraged for both training and analysing machine learning models because it is split into two sets: a test set with 10,000 examples and a training set with 60,000 examples. In the world of computer vision and pattern recognition, FMNIST is frequently used as a benchmark dataset to test various algorithms. It provides a more polished option to the traditional MNIST dataset while still being useful and relevant for research and development.

Table 4.1 lists out accuracies under IID and Non-IID setting for the same setup of pseudoFSSL. The experiment is run for a total of 50 rounds and 2 epochs in each round. With respect to dataset partitioning,, only 10% data is labeled i.e for both MNIST and FMNIST dataset, each client has a total of 12000 training samples out of which a scarce amount of 1200 are labeled. The comparision curve can be seen in Figure 4.2, Figure 4.3 and Figure 4.4. Resnet-18 model architecture with four layers for convolutional and pooling operations, followed by residual blocks and an average pooling layer and finally ending with a fully connected layer. The decoder is formed using a linear and convolutional area. The linear section comprises of fully connected layers with ReLu activations and the convolution area has three transposed convolution layers. Adam optimiser with learning rate $= 0.0001$ and weight decay $= 0.01$ for both autoencoder and classifier network. An MSE loss function is used at server side while client side uses weighted average of MSE loss and Cross-entropy loss. with Reduce LR on plateau learning rate scheduler on both client and server side.

Table 4.1: Comparison of IID and NonIID accuracies on Fashion-MNIST and MNIST for pseudo-FSSL

| **Setting** | **MNIST %** | **F-MNIST %** |
|---|---|---|
| IID Setting (3 clients) | 95.761 | 93.32 |
| IID Setting (5 clients) | 91.348 | 89.74 |
| Non-IID Setting (5 clients) | 53.621 | 37.294 |

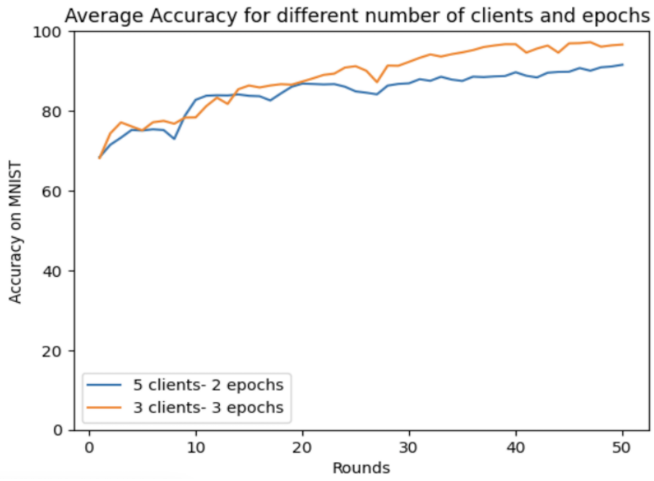


Figure 4.2: Accuracy curve for MNIST for different number of clients and epochs under IID setting (alpha= 100000.0)

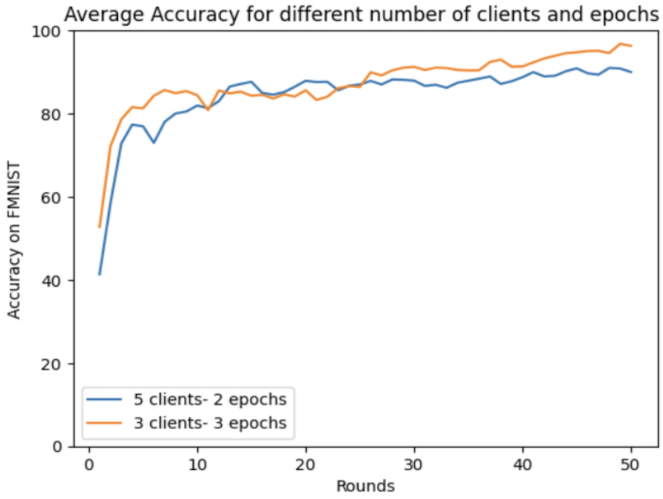


Figure 4.3: Accuracy curve for Fashion-MNIST for different number of clients and epochs under IID setting (alpha= 100000.0)

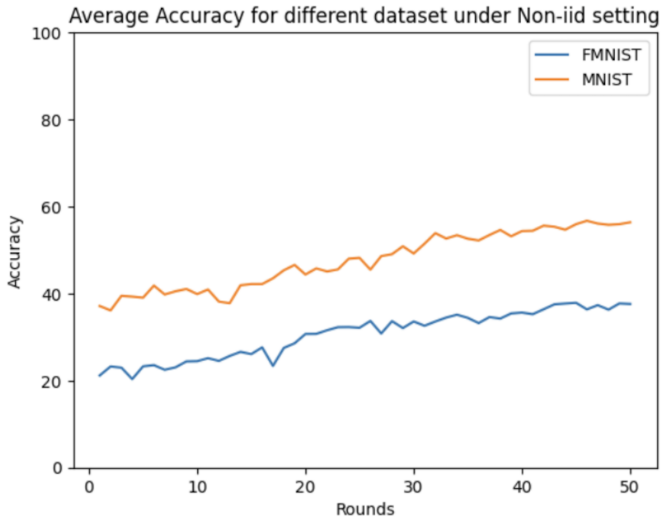


Figure 4.4: Comparison between MNIST and F-MNIST dataset under Non-IID setting (alpha=0.001)

## 4.3 Papers Read and Baselines Implemented

This section briefs about the papers read and baselines implemented in the realm of Federated Learning with multiple techniques like fully supervised, semi-supervised and unsupervised. The dataset used for all methods is CIFAR-10. It is a dataset with a total of 60,000 images of various categories like airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks, where all images are coloured with 32 X 32 dimension.

### 4.3.1 FedMatch

'Federated semi-supervised learning with inter-client consistency and disjoint learning' deals with the issue of learning with partially labeled or unlabeled data. Clients update the server with their newly acquired knowledge after this local training. It uses a parameter decomposition technique and inter-client consistency, inter-client consistency is a technique that forces multiple clients' learned models to consistently produce the same predictions. And by dividing model parameters into two sets, parameter decomposition splits the learning of supervised and unsupervised tasks. The proposed method i.e pseudo-FSSL and pure-FSSL is compared with labels at client scenario. For a fair comaprision, the proposed method is run on hyperparameters same as of Fedmatch (labels at client scenario). For the data partitioning, only 5% of the data is labeled at each client and a total of 5 clients particiate in each round. The total number of rounds considered is 50 where learning rate is set to 0.001 and weight decay of 0.0001 across all three methods. Table 4.2 presents the consolidated results for the same.

Table 4.2: Comparison of proposed work with FedMatch

| **Method** | **IID** | **NonIID** |
|---|---|---|
| pseudo-FSSL (proposed) | 47.17% | 40.26% |
| pure-FSSL (variation of proposed) | 44.32% | 38.13% |
| FedMatch (labels at clients) | 35.22% | 38.57% |

### 4.3.2 FedAvg

The paper titled "Communication-Efficient Learning of Deep Networks from Decentralized Data" lists out experiments involving the Federated Averaging algorithm. These studies were planned to evaluate the algorithm's performance using multiple datasets like CIFAR10, MNIST, and model architectures. The goal of the tests was to evaluate Federated Learning's persistency against non-identical, unbalanced, and non-independent data sets (non-IID) that are common in decentralised data environments, like those noticed on mobile devices. The focus lies mostly on communication efficiency, and the outcomes showcased a significant decrease in the number of communication rounds. The percentage of clients chosen for training each round (C), the number of local epochs completed by each client prior to updates (E), and the local batch size (B) used for the client's updates were the three signifocant hyperparameters used in this paper.

The trade-off between the effectiveness of communication and local computation was enhanced by the authors through experimentation with varied settings of these hyper-parameters. The main objective of these trials was to convey that federated learning is achievable in practice for deep network training on decentralised data while minimising privacy.

### 4.3.3 FedAux

The principal objective of the research is to analyse federated learning algorithms for text and picture classification tasks. To do this, the researchers leverage large-scale models like Tiny-Bert for word tasks and ResNet for image tasks. It uses auxiliary data to pre-train feature extractors, which form the substructure of the federated learning process that comes after. The auxiliary data is first splitted into two subsets: negative data to compare with client data, and distilled data for finding model confidence scores. By utilising a pre-trained model as a feature extractor, clients improvise the federated model's learning task. Clients later solve a regularized optimization problem to train a binary logistic regression classifier to separate out local data from the negative set. Following pre-training, ensemble distillation creates a new model that integrates the collective knowledge of the group, combining the best features of multiple models. On the contrary, most of the federated studies utilise averaging of the models which might not be useful in case of varied distribution of data over clients.

### 4.3.4 SemiFed

A novel approach, SemiFed, deals with limited labeled data in collaborative contexts for semi-supervised federated learning. The architecture utilizes a mix of consistency regularization and pseudo-labeling techniques with ResNet-50 models. While pseudo-labeling gradually labels unlabeled data based on model predictions, consistency regularization makes sure that the model output stays stable in the event of data disruptions. The process entails sending local models for pseudo-label agreement and well-laid data augmentation for perturbations. SemiFed successfully leverages unlabeled data and client participation to improve model accuracy in federated learning scenarios by combining both elements. The framework's usefulness is demonstrated by experimental

findings on the CIFAR-10 and SVHN datasets, which showcase increased performance over classic federated learning approaches under both homogeneous and heterogeneous data distributions.

Table 4.3: Comparison of accuracies on CIFAR-10 from different methods under same hyperparameter setting (ResNet18 model, 10 clients, 2 epochs, 20% labeled data for semi-supervised methods, lr=0.0001, wd=0.01, batch size= 8)

| **Experiment** | **Accuracy in iid setting (%)** | **Accuracy in non-iid setting (%)** |
|---|---:|---:|
| Pseudo-FSSL (proposed) | 65.43 | 47.57 |
| Pure-FSSL (variation) | 62.34 | 43.28 |
| FedAvg | 50.28 | 37.43 |
| SemiFed | 52.63 | 46.56 |
| FedAux | 52.9 | 48.78 |

# REFERENCES

[1] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," Int. J. Mach. Learn. Cybern., vol. 14, pp. 513–535, 2023

[2] Z. Zhang et al., "Benchmarking Semi-supervised Federated Learning," in International Conference on Machine Learning (ICML), 2023

[3] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated Semi-Supervised Learning with Inter-Client Consistency and Disjoint Learning," in Proceedings of the International Conference on Learning Representations (ICLR 2021)

[4] Yoshua Bengio, Aaron Courville, and 438 Pascal Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.

[5] Yuchen Zhao, Hanyang Liu, Honglin Li, Payam Barnaghi, and Hamed Haddadi. Semi-supervised federated learning for activity recognition. arXiv, 2020

[6] Jahid Hasan, "Security and Privacy Issues of Federated Learning," arXiv preprint arXiv:2307.12181, 2023.

[7] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated Learning with Non-IID Data," arXiv preprint arXiv:1806.00582, 2018.

[8] M. Kamp, J. Fischer, and J. Vreeken, "Federated Learning from Small Datasets," arXiv preprint arXiv:2110.03469, 2021.

[9] Liwei Wang, Yan Zhang and Jufu Feng, "On the Euclidean distance of images," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1334-1339, Aug. 2005, doi: 10.1109/TPAMI.2005.165

[10] L. Che, Z. Long, J. Wang, Y. Wang, H. Xiao and F. Ma, "FedTriNet: A Pseudo

Labeling Method with Three Players for Federated Semi-supervised Learning," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021

[11] S. Itahara, T. Nishio, Y. Koda, M. Morikura and K. Yamamoto, "Distillation-Based Semi-Supervised Federated Learning for Communication-Efficient Collaborative Training With Non-IID Private Data" in IEEE Transactions on Mobile Computing

[12] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik, "Personalized Federated Learning using Hypernetworks," arXiv preprint arXiv:2103.04628, 2021

[13] F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," in Proceedings of the IEEE, vol. 109, no. 1, pp. 43-76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555

[14] N. Lu, Z. Wang, X. Li, G. Niu, Q. Dou, and M. Sugiyama, "Federated Learning from Only Unlabeled Data with Class-conditional-sharing Clients," in *International Conference on Learning Representations*, 2022

[15] Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, Shuai Yi; Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 4912-4921

[16] M. Servetnyk, C. C. Fung and Z. Han, "Unsupervised Federated Learning for Unbalanced Data," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020

[17] N. Yang, X. Chen, C. Z. Liu, D. Yuan, W. Bao, and L. Cui, "FedMAE: Federated Self-Supervised Learning with One-Block Masked Auto-Encoder," arXiv preprint arXiv:2303.11339, 2023

[18] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017

[19] Pierre Baldi. Autoencoders, unsupervised 433 learning, and deep architectures. In Proceedings of ICML workshop on unsupervised and transfer learning, pages 37–49. JMLR Workshop and Conference Proceedings, 2012

[20] Andreieva, Valeria  Shvai, Nadiia. (2021). Generalization of Cross-Entropy Loss Function for Image Classification. Mohyla Mathematical Journal. 3. 3-10. 10.18523/2617-7080320203-10.