Personalized Unsupervised Federated Autoencoders

A Project Report

submitted by

KANIKA ARORA

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

July,2023

THESIS CERTIFICATE

This is to certify that the thesis titled **Personalized Unsupervised Federated Autoencoders**, submitted by **Kanika Arora**, to the Indraprastha Institute of Information Technology, Delhi, for the award of the degree of **MASTERS OF TECHNOLOGY**, is a bonafide record of the research work done by her under our supervision. 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.

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

Place: New Delhi

Date: 19th January 2009

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my advisor Dr. Ranjitha Prasad for her supervision, guidance and constant support. Her expertise in Federated Learning has been invaluable in formulating the research methodology. I will be grateful to my advisor for providing me with this opportunity regardless of my lack of research experience.

I also thank current PhD student Nazreen Shah for her assistance and support in this research project. It gives me immense pleasure to thank all of the members of the Intellicom Lab for their inspiration and patient support.

I am pleased to collaborate with LightMetrics Technologies Pvt. Ltd for this research project.

Lastly, I would like to thank my family for their constant encouragement and motivation throughout this thesis. They stood beside me during my weak days. I also thank my friends with whom I could connect and discuss my problems most of the time.

ABSTRACT

An immense amount of data is generated daily using modern technologies in autonomous vehicles, IoT, smart grids, etc. But unfortunately, this data generated at the edge cannot be used for any machine learning model training due to privacy concerns or expensive computational costs. The data must be stored at the central server for any machine learning process to occur. To overcome the problem faced due to the traditional machine learning approach decentralized technique called Federated Learning started to gain popularity. Federated Learning allows multiple clients in the network to collaborate and learn a global machine learning model, which can be passed to all the edge devices for locally training a model while maintaining privacy since data is present at the edge device. The availability of annotated data is one of the challenges of supervised federated learning. Moreover, sometimes it is difficult for a global model to perform well for all the clients in the network due to the presence of heterogeneous data.

In this thesis, a novel Personalized unsupervised Federated AutoEncoders,pFedAE, is proposed with the main motivation that local and global latent space representations of all the clients in the network. The optimisation framework of the autoencoder is divided into two parts global and per-client local optimisation frameworks. We have adopted two evaluation strategies to evaluate the latent space representation at both global and local levels. We demonstrated that pFedAE under both evaluation strategies performed better than the other baselines. pFedAE, most importantly, leads to faster convergence, is scalable for different numbers of clients, is effective in varying data distribution across clients and is robust to different numbers of local epochs. Using t-SNE projections and angle histogram plots, a comparison of the pFedAE with other baselines for latent space is also demonstrated in the later part of the thesis.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS i

ABSTRACT ii

LIST OF TABLES v

LIST OF FIGURES 1

1 INTRODUCTION 2

1.1 Background and Motivation 2

1.2 Federated Learning 2

1.3 How does Federated Learning work? 3

1.4 Benefits of Federated Learning 5

1.5 Challenges of Federated Learning 5

1.6 Contributions of this Thesis 6

1.7 Thesis Structure 7

2 RELATED WORKS 8

2.1 Personalized Federated Learning 8

2.2 Unsupervised Federated Learning 9

2.3 Autoencoder in Unsupervised Federated Learning 10

3 METHODOLOGY 12

3.1 Mathematical Preliminaries 12

3.1.1 Supervised Federated Learning 12

3.1.2 Unsupervised Federated Learning 12

3.1.3 Autoencoder 13

3.1.4 Federated Autoencoder 14

3.2 pFedAE: PROPOSED SCHEME 15

4 EXPERIMENTS AND RESULTS 20

4.1 Datasets and Partitioning 20
4.2 Implementation details 21
4.3 Baselines 21
4.4 Linear and Semi-supervised Evaluation and Results 22
4.5 Results on Attributes of Federated Learning 24
4.5.1 Sensitivity to Statistical heterogeneity 24
4.5.2 Robustness to Local Epochs 25
4.5.3 Scalability 27
4.5.4 Convergence 29
4.5.5 Local Accuracy 29
4.6 Assessment of Latent Space and Discussions 30
4.6.1 Qualitative assessment via t-SNE 30
4.6.2 Quantitative Assessment via Inter-class difference 30

5 CONCLUSION **34**

LIST OF TABLES

4.1 Top-1 Global and local accuracy ($\%$) comparison under the Linear and Semi-supervised evaluation protocol for statistically heterogeneous ($\alpha = 0.1$) setting and the pathological (2-class) non-IID for CIFAR10 and CIFAR100 datasets. 23

LIST OF FIGURES

1.1 Federated Learning Framework 3

1.2 Working of Federated Learning 4

1.3 Benefits of Federated Learning 4

3.1 Overview of the pFedAE architecture (on the left): The figure shows the local encoder-surrogate decoder pair at K clients and the global decoder-surrogate encoder pair at the server. The local encoder $g_k(\cdot;\phi_k)$ transforms the input data to $\mathbf{z}_k$. The global decoder $h(\cdot;\psi)$ transforms $\mathbf{z}$ obtained by concatenating latent representations from each client. The surrogate encoder $g^s(\cdot;\phi^s)$ transforms $\mathbf{y}$ to obtain $\mathbf{z}^s$ which is then transformed into locally reconstructed data by $h^s_k(\cdot;\psi^s_k)$. Training occurs by optimizing the learning objectives in (3.13). The figure on the right gives a closer view of the learning strategy. The clients and server communicate $\mathbf{z}_k$ and $\mathbf{z}^s_k$. The calculation of loss at each client and server is carried out in every round for updating weights through backpropagation as detailed in Algorithm 1. 16

4.1 t-SNE projections of the latent representation from vanilla FL McMahan *et al.* (2017) where the underlying ML model is an autoencoder (FedAvg-AE), Orchestra Lubana *et al.* (2022) and the proposed pFedAE algorithm. PFedAE provides meaningful latent representations for different classes, albeit being an unsupervised technique. 25

4.2 Global and local accuracy convergence plots for different values of α for CIFAR10 and CIFAR100 datasets. 25

4.3 Sensitivity to statistical heterogeneity (varying values of α for CIFAR10 (left) and CIFAR100 (right) datasets. 26

4.4 Standard deviation plots for local and global accuracy across clients, for different values of α on CIFAR10 and CIFAR100 datasets. 26

4.5 Robustness to local epochs for CIFAR10 (left) and CIFAR100 (right). 27

4.6 Global and local accuracy convergence plots for different numbers of local epochs (first and second from the left) and different numbers of clients (third and fourth from the left) in a cross-silo setting. 28

4.7 Varying number of clients (scalability) for CIFAR10 (left) and CIFAR100 (right). 28

4.8 Convergence of pFedAE as compared to baseline methods using $E = 5$ local epochs and $C = 100$ rounds. 29

4.9 Local accuracy behaviour of FL attributes. 29

4.10 t-SNE projections of encoder models trained using pFedAE, Orchestra and FedAvg-AE using 1, 5 and 10 local epochs. 31

4.11 t-SNE projections of encoder models trained using pFedAE and single client training. 31

4.12 t-SNE projections of encoder models trained using pFedAE and single client training on the global test dataset and local test dataset with $\alpha = 0.1$ 32

4.13 Histogram on inter-class difference. 33

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Feature representations of the data that give meaningful information are used in various disciplines, including medical diagnosis Dash *et al.* (2019), wearable devices Brodie *et al.* (2018), etc., resulting in massive personalised data. These feature representations are important for extracting useful information for downstream tasks, including classification, prediction, object detection Girshick *et al.* (2014), landmark detection Wood *et al.* (2021), etc. When data is collected at the edge, transferring that data to a cloud-based central entity to develop a centralised machine learning (ML) model is prohibitively expensive. Because of privacy concerns, it may have legal issues. Federated learning (FL) has evolved to process input at all the clients in the network locally. Still, the model is trained on a server through client parameter updates. McMahan *et al.* (2017)

1.2 Federated Learning

Federated learning is a decentralized technique where multiple clients remotely share data to train a shared global machine-learning model. Each client trains a local machine-learning model on local data and shares the update with the central server. After receiving the updates from all the clients, the server aggregates them and shares the aggregated update across all clients. Iteration after iteration, the training continues until the global model is fully trained. Since model training is done locally at each client, federated learning prevents data breaches, which is one of the disadvantages of traditional machine learning techniques. Federated Learning also allows for smarter models because the models are trained on diverse datasets across each client. The communication cost of transferring the data to the central server is reduced because the training takes place

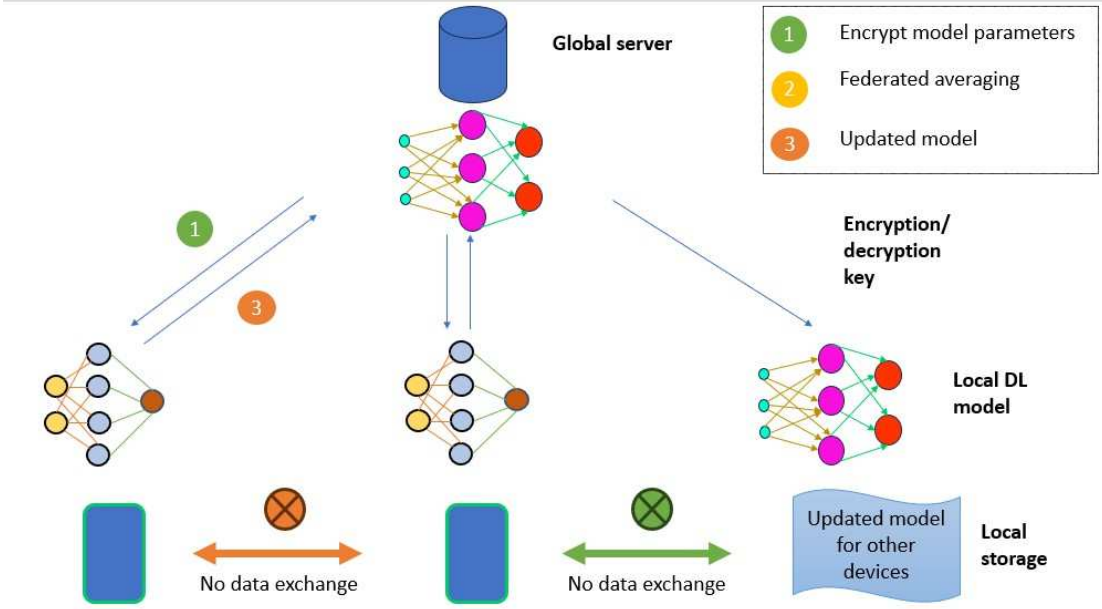

Figure 1.1: Federated Learning Framework

locally, and only the model parameter updates of each client are shared with the central server.

1.3 How does Federated Learning work?

For training a Federated Learning model, data must not be stored at the central server; instead, it uses the decentralised network to train a model.

- **Step 1 :**The first step required to train a federated learning framework is to choose a model that can be either pre-trained or not-trained, depending upon the user's requirement.
- **Step 2 :**After that, the chosen model is distributed among all the clients in the network.
- **Step 3 :**Following the model distribution across all clients, each client in the network trains the local model using local data.
- **Step 4 :**When each local model's training is completed, the updates from each client are shared with the central server.
- **Step 5 :**The server, on receiving updates from all the clients in the network, aggregates those updates and shares the best-performing model with all the clients for further training.

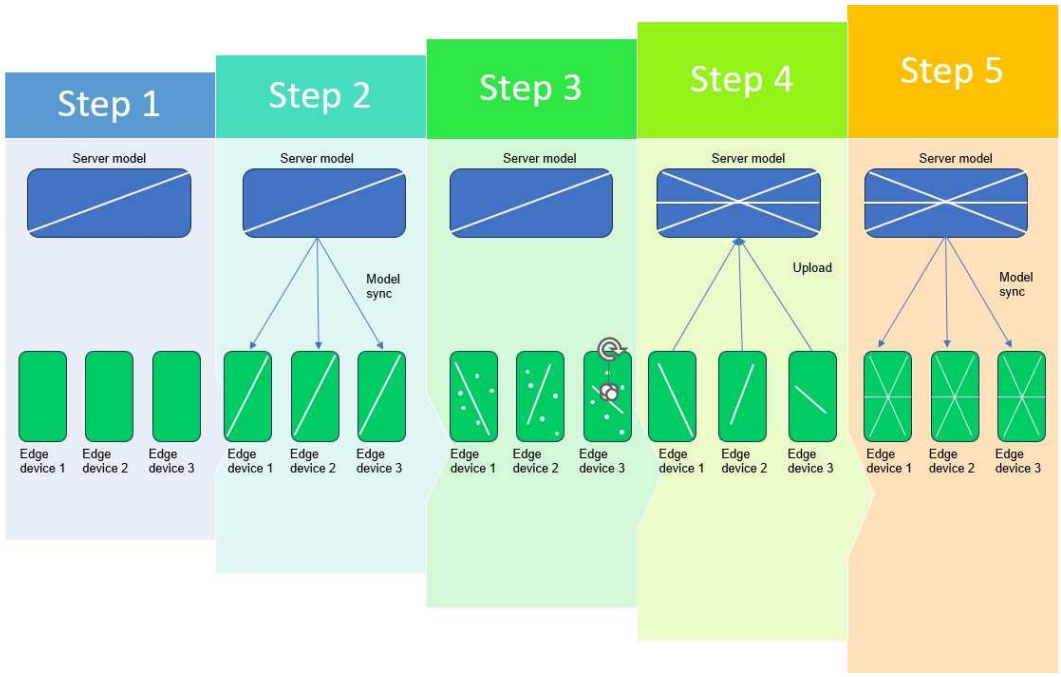

Figure 1.2: Working of Federated Learning

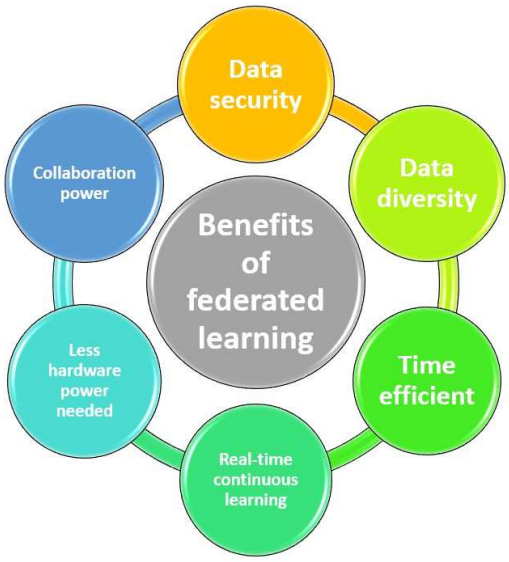

Figure 1.3: Benefits of Federated Learning

1.4 Benefits of Federated Learning

- **Data Security :** Unlike traditional machine learning methods, training in federated learning takes place locally at each client in the network using the local data, so data is not required to be stored at the central server, which makes federated learning a privacy-preserving approach.
- **Data Diversity :** The data used for training the federated learning model is from different clients, so it can be heterogeneous because the data present at each client locally depends on various conditions like geographical locations, different methods by which the data is captured, etc. As a result, even if the local client model is only trained on one type of data in a federated learning system after the models from all clients are aggregated, each client can perform well on a completely different test set.
- **Real Time Continuous Learning :** Training happens iteration after iteration in a Federated Learning setup model until the global model is fully trained.
- **Less Hardware Power Needed :**As model training takes place locally at each client in a Federated Learning setup, there is no requirement for complex hardware at the central server.
- **Collaboration Power :** Federated Learning aims to make a collaborative framework so that all the clients in the network can learn from a global model, which was trained from the local models trained at each client without sharing the local data at the server.
- **Time Efficient :**As previously stated, the data used to train Federated Learning models is trained locally without being shared with the central server. This saves a lot of time because there is no time lag while uploading data to the central server.

1.5 Challenges of Federated Learning

Federated Learning, like the traditional machine learning approach, comes along with its own set of challenges. The key challenges of Federated Learning are listed below:

- **Statistical Heterogeneity :** Federated Learning is a decentralised framework; the data used for training is from the local devices, which may or may not be of a similar type. It is a possibility that often the data that will be used for training models will be highly unbalanced and Non-IID because the clients can be from different time zones/geographic locations, and also, the preferences of each client may or may not be the same. Each client in the network may not possess the same set of data points. Because of the absence of IID data global model may not always perform well because it is learned from different local datasets from local clients. This problem can be solved by using the personalized model to train each client in a network and then aggregate the updates from those models to create one global model. Using Personalized Federated Learning will also improve the model's performance.

- **Communication Efficiency:**In a Federated Learning framework, communication between server and client is one of the major challenges. Therefore, it is a necessity for any Federated Learning technique to converge after a few rounds of iteration. To increase communication efficiency, it is very important for a Federated Learning framework to have more iterations of local training and few iterations of global aggregation; this will reduce the number of communication rounds and give a faster convergence.

- **Scalability with Number of Clients :**In federated learning, the communication overhead increases significantly as the number of clients increases. Each client in the network must share the local updates with the server, which is then aggregated by the server. This communication overhead can hamper the network bandwidth and lead to increased latency, potentially affecting the system's scalability. The scalability of clients will also require efficient resources both at the server and clients. The server should be able to handle the load of the larger client network model. Various new methods are being used to overcome the problems faced because of the scalability of clients, like the client selection approach, hierarchical federated learning, etc.

1.6 Contributions of this Thesis

Our main goal with this work is to create a client-specific ML model so that the client can handle heterogeneity within its computing competence. We learn personalised feature representations with preserving privacy by using unlabeled data for each client, a critical yet widely overlooked problem. In this thesis, We have touched on the generalisation and personalization of federated unsupervised feature extraction at clients in a privacy-preserving manner. The following are the key aspects of our work:

- The proposed pFedAE framework is built using the autoencoder principle. The autoencoder is split into two parts global and local optimization problem.
- To ensure SGD-based training, we incorporate a *surrogate* encoder at the server and surrogate decoders at the clients, which help us to incorporate local information at the server, and global information at the clients.

Our proposed method,pFedAE is, to the best of our knowledge, the first of its kind, based on a federated personalised autoencoder in an unsupervised federated learning environment. We have evaluated pFedAE on two public dataset CIFAR10 and CIFAR100. Zhuang *et al.* (2021); Lubana *et al.* (2022); Han *et al.* (2022).

1.7 Thesis Structure

This thesis is divided into six chapters. Following the Introduction (Chapter 1), the thesis embodies the following five chapters:

- **Chapter 2 :** It briefly overviews the literature review of the techniques used to propose our method.

- **Chapter 3 :** Mathematical Preliminaries for the proposed scheme have been explained in this chapter.

- **Chapter 4 :** It gives a detailed description of the methodology we proposed to create client-specific machine learning models that can overcome the challenges faced by most Federated Learning methods.

- **Chapter 5 :** Discussion on the Experimental results for our proposed framework, along with a comparison from the baselines, is presented in this chapter.

- **Chapter 6 :** The conclusion drawn from the work done for our proposed scheme and future works is listed in the last chapter.

CHAPTER 2

RELATED WORKS

This chapter presents a comprehensive review of existing techniques for Federated Learning. By examining the current methods for Federated Learning, we aim to identify the gaps and challenges that can be addressed in our methodology. We have also discussed Federated Learning and auto-encoder-based techniques to integrate them into our proposed method.

2.1 Personalized Federated Learning

Personalized Federated Learning develops personalized models for each client in the Federated Learning framework. Personalized Federated Learning develops a separate model for each client based on their requirements and characteristics. Personalised Federated Learning also addresses the issue of statistical heterogeneity in each client because all clients in the network will have personalised machine learning models, as opposed to the traditional Federated Learning framework, in which all clients have a generalised machine learning model. The training process in Personalized Federated Learning is similar to that of the traditional Federated learning method. A review of the existing works which can be employed along with Personalized Federated Learning, such as differential privacy, model distillation, etc., has been discussed further.

Personalised supervised FL was introduced to deal with data heterogeneity and build client-specific ML models. Tan *et al.* (2022). When non-IID data is utilised for training, the accuracy of FedAvg-based approaches decreases dramatically due to data discrepancy across clients. Using the PFL scheme, a global single model is trained first, followed by the learnt global Federated Learning model at each client. The local model must be retrained on the local dataset for local adaptation. Personalization model performance is closely linked to the generalisation performance of the client's global model. Various Personalised Federated Learning strategies aim to improve the

performance of global models by utilising data heterogeneity, thus boosting personalization performance. FedAvg is another Personalised Federated Learning technique inspired by MAML. Fallah *et al.* (2020). By conducting a few stages of a gradient-based method, this novel methodology tries to locate an initial point shared by all clients in a network that performs well when each client updates it for its own loss function. While the initial model is shared with all users, each user's final model is unique depending on their personal data. The MAML technique has the advantage of preserving the benefits of FL and capturing the distinction between customers as existing or new clients can take the solution of this new challenge as an initial point and slightly adjust it for their data. Instead of training a single global model for the whole network, the multitask learning (MTL) framework Smith *et al.* (2017) federated learns different models for each client in the network, allowing for the consideration of separate but related models. MTL also solves the scalability of network and node heterogeneity challenges. To mitigate the negative impacts of statistical heterogeneity, global training of neural network base layers using federated averaging is followed by local training of top layers using a stochastic gradient to generate a personalised model FedPer. Arivazhagan *et al.* (2019). All network clients have the same base layer weights and distinct personalization layers that may be easily adapted to individual data. The server is shared by all network clients, but the personalization layer is unique to each client. Other approaches to personalisation include knowledge distillation, transfer learning, a combination of models, meta-learning, and so on. Kulkarni *et al.* (2020). The authors of Shamsian *et al.* (2021) personalized personalise FL models utilising a central hyper network that generates unique ML models for each client. None of the preceding works can handle unlabeled or partially tagged data from customers.

2.2 Unsupervised Federated Learning

Unsupervised Federated Learning is a type of Federated learning which uses unlabeled data for training, similar to unsupervised machine learning to learn data representation without using labels. In Unsupervised Federated Learning, all the network clients trained a shared global model using unlabeled data for different purposes like dimensionality reduction, clustering, etc., while preserving privacy like the traditional Federated Learning setup. The main aim of Unsupervised Federated Learning is to use

unlabeled data to learn a global model that can be used for feature representation for all the clients in the network. This field of Federated Learning has great potential and can be used in various domains since unlabeled data is available in ample amounts compared to labelled data. A review of a few Unsupervised Federated Learning techniques has been discussed further, which can be used for various tasks like domain adaptation, anomaly detection, clustering, etc.

Unsupervised FL is very necessary, especially when clients only have unlabeled data. FedUL Lu *et al.* (2022) is an unsupervised FL technique for supervised FL that applies surrogate labels to unlabeled data. The problem emerges when the trained model for the original classification task is inferred from the model learned from the surrogate task. Surrogate class-posterior probabilities with specific injective transition functions were used to solve this challenge, which was accomplished by designing distinct transition layers to output each client-trained model. FedUL is a broad framework that, due to its flexibility, can operate as a wrapper that converts the original client to surrogate clients. Orchestra Lubana *et al.* (2022) was offered as another unsupervised FL technique for achieving uniform partition of client data for unsupervised clustering. FedAUX Sattler *et al.* (2021) is a federated distillation-based method that uses unsupervised pre-training on auxiliary data to build a model that can then be used for distributed training. Further, a global feature representation learning technique, FedU Zhuang *et al.* (2021), uses contrastive loss for training of an online encoder and predictor. FedX Han *et al.* (2022), an unsupervised FL approach based on FedAvg, uses contrastive loss to learn semantic representation from the local data. Through knowledge distillation, the bias at the central server is regularized. The difficulties of system and data heterogeneity are addressed by a method FedProx Li *et al.* (2020). Even though all these methods train the model using the unlabelled data of each client, they are still not considered personalized models.

2.3 Autoencoder in Unsupervised Federated Learning

In Unsupervised Federated Learning, Autoencoders play a very important role as it is used in representation learning from unlabeled data. Autoencoder makes use of encoder-decoder architecture, which is used to reconstruct the input data without any

data loss. In an Unsupervised Federated Learning setting autoencoder is used to learn data representation from the unlabeled data available at each client in the network. The encoder of the autoencoder network is used to transform input data into a smaller dimension of latent space. This latent space is then passed on to the decoder, which reconstructs it into the original input data. The autoencoder learns the unsupervised important data features by minimizing the error between the input and reconstructed data. Autoencoder in Unsupervised Learning will help clients learn diverse data representation from abundant available unlabeled data. Further, we have reviewed a few Autoencoder-based Unsupervised Federated Learning frameworks.

For unsupervised learning, autoencoders are important for feature extraction tasks Bengio *et al.* (2013); Baldi (2012). When labelled data is available, autoencoders can obtain useful feature representations of the client's data. Autoencoders in a Federated Learning framework are beneficial as encrypted data is passed to the server, and extracting the exact data representation of the original data at the server is impossible. Another unsupervised representation learning approach is proposed for feature extraction in a Federated Learning framework, which uses autoencoder van Berlo *et al.* (2020). Using unlabeled input, unsupervised representation learning is used to pre-train neural networks. The pre-trained model is then used to learn features; these learned features are useful in tasks with little labelled data. This pre-trained model approach is beneficial and has outperformed most supervised learning methods for downstream tasks. Semi-supervised federated learning method Zhao *et al.* (2020) has been proposed for learning in IoT-based systems for human activity recognition, where with the help of autoencoders, unsupervised learning training is done using unlabelled data across all the clients in the network to learn general representation and the central cloud server using the labelled data conducts supervised learning for the classifier. The autoencoders are present locally at each client, which sends features to the central cloud server, which is then aggregated to form a global autoencoder. For learning from multi-modal IoT data Zhao *et al.* (2022), multimodal and semi-supervised FL framework is proposed using autoencoders to extract general representation across all the clients in the network. All of the methods described above develop a common machine-learning model shared by clients and servers; thus, these methods are intrinsically generalized.

CHAPTER 3

METHODOLOGY

3.1 Mathematical Preliminaries

3.1.1 Supervised Federated Learning

In Supervised Federated Learning, all the clients have access to locally labelled dataset $\mathcal{D}_k^s$, which consists of N_k data samples. A global objective function $F(\boldsymbol{\phi})$ is considered, which using shared model parameters $\boldsymbol{\phi} \in \mathbb{R}^{d\times 1}$ collaborates across multiple clients. The objective function of Fl is given by

$$\min_{\boldsymbol{\phi}} F(\boldsymbol{\phi}) \triangleq \mathbb{E}_{\mathcal{D}_k^s}[f_k(\boldsymbol{\phi})], \tag{3.1}$$

where the local objective function is $f_k(\boldsymbol{\phi}) \triangleq {N_k}^{-1} \sum_{i=1}^{N_k} l(\boldsymbol{\phi}; \mathbf{x}_{k,i}, y_{k,i})$ $(k \in \mathcal{K} \triangleq [K])$ which measures the empirical risk as a function of user-specified loss function $l(\boldsymbol{\phi}; .)$ where $(\mathbf{x}_{k,i}, y_{k,i}) \in \mathcal{D}_k^s$ and $\sum_{k=1}^{K} N_k = N$.

A significant challenge in a supervised Federated Learning (FL) framework, as described in equation (3.1), is annotated data is not available at the time of learning. Annotated data involving labelling the data is often time-consuming and expensive; therefore, it is not accessible at the edge devices. Inspired by machine learning, it can be said that even if the data at different clients is heterogeneous, common patterns and features will still exist that can be learned in a Federated Learning framework Bengio *et al.* (2013). This approach of Federated Learning, where the model focuses on learning shared representation, can be beneficial because this will reduce the need of annotates data at the clients.

3.1.2 Unsupervised Federated Learning

In the previous section, we discussed Supervised Federated Learning and the challenges which occur because of the unavailability of labelled data. To overcome those data

challenges we focus on the unsupervised Federated Learning aspect where unlabelled data is present at the client. The unlabelled data at the k-th client is represented as

$$\mathcal{D}_k = \{\mathbf{x}_{k,i}\}_{i=1}^{N_k} \ \forall \ k \in 1, \ldots, K, \tag{3.2}$$

where $\mathbf{x}_{k,i} \in \mathbb{R}^M$. In an unsupervised approach latent variable of the client is used for the downstream task.

3.1.3 Autoencoder

One popular model for learning latent space representation is autoencoder Baldi (2012), Hinton and Salakhutdinov (2006). The autoencoder comprises two main components: an encoder and a decoder. The encoder is expressed as a function mapping $g(\cdot;\boldsymbol{\phi}) : \mathcal{X} \rightarrow \mathcal{Z}$ from the input space $\mathcal{X}$ (M dimensional) to the latent space $\mathcal{Z}$ (L dimensional, $L << M$) as a function of parameters $\boldsymbol{\phi}$. Hence, for $\mathbf{z}_i \in \mathcal{Z}$, we have $\mathbf{z}_i = g(\mathbf{x}_i;\boldsymbol{\phi})$, where $\mathbf{z}_i$ is the latent variable representation of the input $\mathbf{x}_i \in \mathcal{X}$.The decoder is parameterized by $\boldsymbol{\psi}$, is expressed as a function mapping from the latent space back to the original input space given by $h(\cdot;\boldsymbol{\psi}) : \mathcal{Z} \rightarrow \mathcal{X}$. The unknown function mapping is characterised by the parameters $g(\cdot;\boldsymbol{\phi})$ and $h(\cdot;\boldsymbol{\psi})$. Therefore, training an autoencoder makes it essential to entail finding $g(\cdot;\boldsymbol{\phi})$ and $h(\cdot;\boldsymbol{\psi})$ such that

$$\boldsymbol{\phi}, \boldsymbol{\psi} \leftarrow \underset{\boldsymbol{\phi},\boldsymbol{\psi}}{\arg\min} \sum_{i=1}^{N} \mathcal{L}(\mathbf{x}_i, h(g(\mathbf{x}_i;\boldsymbol{\phi});\boldsymbol{\psi}), \tag{3.3}$$

where, the reconstruction error is given by $\mathcal{L}(\cdot)$. The main objective is to minimize the reconstruction error between the original input data and the reconstructed output. Basically, reconstruction loss is the mean squared error between the input data and the reconstructed output. The above-discussed framework is referred to as centralized autoencoder training, as we have assumed that data is available at a single location. The main objective of the autoencoder is to learn latent space representation $\mathbf{z}_i$ that has essential input data information. The latent space representation is compressed to a lower dimension, leading to a more concise representation than the input $\mathbf{x}_i$ in the latent space while storing the same information as the input.

Lemma 3.1.1 *Assume a setting in which the quadratic loss $\mathcal{L}(\cdot)$ is used, the represen-*

tations $g(\cdot)$ and $h(\cdot)$ is a linear invertible projection onto a L-dimensional and M-dimensional subspace, respectively, we have the following:

$$\mathcal{L}(\mathbf{x}_i, h(g(\mathbf{x}_i; \boldsymbol{\phi}); \boldsymbol{\psi}) \leq \frac{1}{2}||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2 + \frac{\lambda}{2}||\mathbf{z}_i - \hat{\mathbf{z}}_i||^2, \tag{3.4}$$

where $\mathbf{z}_i = g(\mathbf{x}_i, \phi)$ and λ is a constant that depends on the decoder transformation.

Proof: For linear invertible projections onto L-dimensional and M-dimensional given by $\mathbf{A}$ and $\mathbf{B}$ respectively, we have the following:

$$\begin{aligned} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2 &= ||\mathbf{x}_i - \mathbf{A}\mathbf{B}\mathbf{x}_i||^2 = ||\mathbf{A}(\mathbf{A}^{-1}\mathbf{x}_i - \mathbf{B}\mathbf{x}_i)||^2 \\ &\leq ||\mathbf{A}||^2||(\mathbf{A}^{-1}\mathbf{x}_i - \mathbf{B}\mathbf{x}_i)||^2, \end{aligned} \tag{3.5}$$

where, using Cauchy-Schwartz inequality, we obtained the inequality. Essentially, the reconstruction error will be zero if $\mathbf{B} = \mathbf{A}^{-1}$. But in autoencoders, $\mathbf{z}_i = \mathbf{A}^{-1}\mathbf{x}_i$ gives latent space representation whereas, $\mathbf{B}$ is given by $\hat{\mathbf{z}}_i = \mathbf{B}\mathbf{x}_i$ gives the encoded vector.Therefore, by setting $\lambda = ||\mathbf{A}||^2$, we get

$$||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2 \leq \lambda(\mathbf{z}_i - \hat{\mathbf{z}}_i)||^2. \tag{3.6}$$

By Adding $||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2$ to both sides of the above expression and diving both sides by 2, we get the result.

3.1.4 Federated Autoencoder

In this section, we have discussed how to utilize autoencoders in an unsupervised Federated Learning setting. Using autoencoders in unsupervised Federated Learning models can learn useful latent space representation without the need for labelled data at the client. The mathematical formulation for Federated Autoencoder is given below. Consider unlabeled data at the k-th client $\mathcal{D}_k = \{\mathbf{x}_{k,i}\}_{i=1}^{N_k}$ where $\mathbf{x}_{k,i} \in \mathbb{R}^M$. We generalize Lemma. 3.1.1 to Federated autoencoder training across K clients. The overall loss function in (3.3) can be rewritten as

$$\sum_{k=1}^{K}\sum_{i=1}^{N_k} \mathcal{L}(\mathbf{x}_{k,i}, h(g(\mathbf{x}_{k,i}; \boldsymbol{\phi}); \boldsymbol{\psi}) = \sum_{k=1}^{K} \frac{1}{N_k} \sum_{i=1}^{N_k} ||\mathbf{x}_{k,i} - \hat{\mathbf{x}}_{k,i}||^2, \tag{3.7}$$

Assuming λ to be a hyperparameter, and using (3.4), we write (3.7) as

$$\sum_{k=1}^{K}\sum_{i=1}^{N_k}\mathcal{L}(\mathbf{x}_{k,i}, h(g(\mathbf{x}_{k,i};\boldsymbol{\phi});\boldsymbol{\psi}) \leq \sum_{k=1}^{K}\frac{1}{2N_k}\sum_{i=1}^{N_k}\underbrace{(||\mathbf{x}_{k,i}-\hat{\mathbf{x}}_{k,i}||^2}_{Local}+\lambda\underbrace{||\mathbf{z}_{k,i}-\hat{\mathbf{z}}_{k,i}||^2)}_{Global}. \tag{3.8}$$

Based on the above discussion, we can drive federated learning training using a federated autoencoder, considering both global and local aspects. In cases where the clients share their latent space representations $\mathcal{Z}_k = \{\mathbf{z}_{k,i}|i \in [N_k]\}$, the server combines all the information obtained from each client by optimizing $\sum_{k=1}^{K}\sum_{i=1}^{N_K}||\mathbf{z}_{k,i}-\hat{\mathbf{z}}_{k,i}||^2$ according to the upper bound in (3.8). Since the clients do not share data, they locally compute $\sum_{i=1}^{N_K}||\mathbf{x}_{k,i}-\hat{\mathbf{x}}_{k,i}||^2$, where $\hat{\mathbf{x}}_{k,i}$ is the globally encoded data as shared by the server with all clients k. In the next chapter, we propose a federated autoencoder approach based on the principles discussed in Lemma. 3.1.1 and (3.8).

3.2 pFedAE: PROPOSED SCHEME

This section will discuss pFedAE, our proposed personalised Federated Learning framework. pFedAE is put together by carefully dividing an autoencoder's optimisation objective into K local objectives for clients and a global objective to be optimised at the server. Data is partitioned so that data accessible to clients determines the local goal function, while information exchanged by the client with the server defines the global goal function

One feature of pFedAE that differentiates it from other Federated Learning methods is that client-server communication occurs through exchanging latent space representation rather than model parameters. Many Federated Learning approaches have the disadvantage of increasing model complexity at the client, necessitating a huge communication bandwidth. Because the size of the latent representation is a user-defined variable, using latent space representation can help to relieve this problem. Because the latent space representation of pFedAE is in a lower-dimensional space than the input data, reconstruction is impossible without data. Furthermore, we could extend pFedAE as a privacy enhancing strategy based on autoencoder, which is effective in maintaining

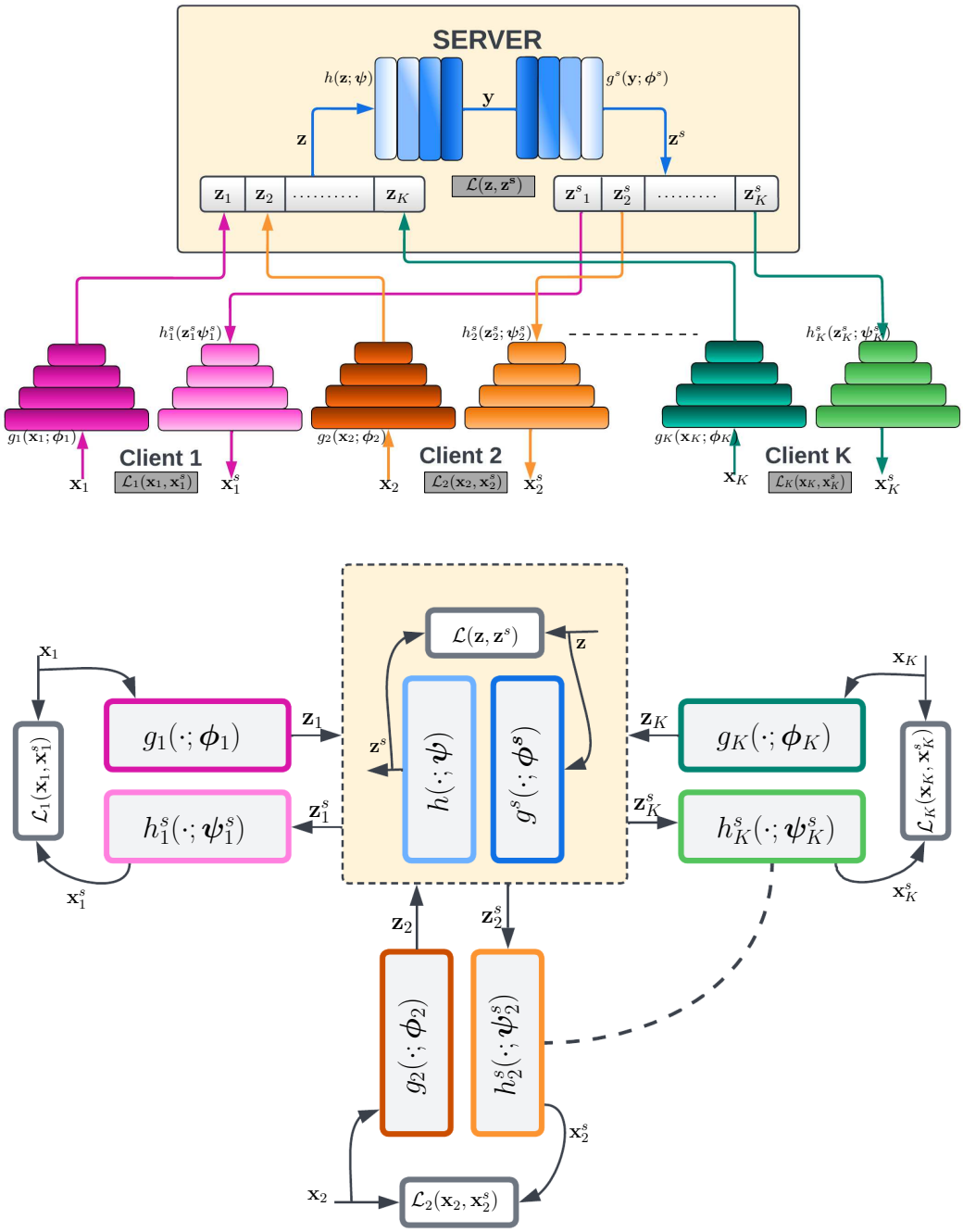

Figure 3.1: Overview of the pFedAE architecture (on the left): The figure shows the local encoder-surrogate decoder pair at K clients and the global decoder-surrogate encoder pair at the server. The local encoder $g_k(\cdot;\phi_k)$ transforms the input data to $\mathbf{z}_k$. The global decoder $h(\cdot;\psi)$ transforms $\mathbf{z}$ obtained by concatenating latent representations from each client. The surrogate encoder $g^s(\cdot;\phi^s)$ transforms $\mathbf{y}$ to obtain $\mathbf{z}^s$ which is then transformed into locally reconstructed data by $h^s_k(\cdot;\psi^s_k)$. Training occurs by optimizing the learning objectives in (3.13). The figure on the right gives a closer view of the learning strategy. The clients and server communicate $\mathbf{z}_k$ and $\mathbf{z}^s_k$. The calculation of loss at each client and server is carried out in every round for updating weights through backpropagation as detailed in Algorithm 1.

latent space features such as Hajihassani *et al.* (2020); Malekzadeh *et al.* (2019, 2018).

For the architecture of pFedAE we have used local autoencoders at each client k. The objective function is given by the following:

$$\boldsymbol{\phi}_k, \boldsymbol{\psi}_k \leftarrow \underset{\boldsymbol{\phi}_k, \boldsymbol{\psi}_k}{\arg\min} \sum_{i=1}^{N_k} \mathcal{L}_k(\mathbf{x}_{k,i}, (h_k(g_k(\mathbf{x}_{k,i}; \boldsymbol{\phi}_k); \boldsymbol{\psi}_k). \tag{3.9}$$

One thing to note here is that a local autoencoder cannot adapt to the data from other clients. But in Federated Learning local model parameters, updation occurs for E epochs in each communication round and global model learnings are done for T communication rounds. Furthermore, a major issue arises in Personalized Federated Learning, where each client should develop a local model from the globally trained model. From the challenges faced in the Personalized Federated Learning setting, we propose a model in which the encoding of data is done at the client, after which the encoded data is shared with the server. For learning a global decoder at the server, the objective function is mentioned below:

$$\boldsymbol{\phi}_k, \boldsymbol{\psi} \leftarrow \underset{\boldsymbol{\phi}_k, \boldsymbol{\psi}}{\arg\min} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathcal{L}(\mathbf{x}_{k,i}, h(\mathbf{z}_{k,i}; \boldsymbol{\psi})), \tag{3.10}$$

where $\mathbf{z}_{k,i} = g_k(\mathbf{x}_{k,i}; \boldsymbol{\phi}_k)$ and $h(\mathbf{z}_{k,i}; \boldsymbol{\psi}) \in \mathbb{R}^M$. One thing to note herein (3.10), the encoder is locally present at each client, and the decoder is globally present at the server. Therefore, a global loss function $\mathcal{L}(\cdot, \cdot)$ is used to learn a common set of decoder parameters $\boldsymbol{\psi}$. Even though this objective function satisfies the requirements of a Federated Learning framework to obtain a generalized global model, computing reconstruction loss $\mathcal{L}(\cdot, \cdot)$ in (3.10) is practically impossible because it requires data points $\mathbf{x}_{k,i} \in \mathcal{X}$ which due to privacy constraints are only available at the k clients and are not at the server. To solve this issue, we have proposed a solution, using a global *surrogate* encoder $g^s(\cdot; \boldsymbol{\phi}^s)$ at the server along with local surrogate decoders $h_k^s(\cdot; \boldsymbol{\psi}_k^s)$ at K clients. The global *surrogate* encoder $g^s(\cdot; \boldsymbol{\phi}^s)$ present at the server transforms the decoded vector given by the decoder present at the server to a latent space. Therefore, the global decoder-encoder pair $\{h(\cdot; \boldsymbol{\psi}), g^s(\cdot; \boldsymbol{\phi}^s)\}$ is such that $h(\cdot; \boldsymbol{\psi})$ maps latent space $\mathcal{Z}$ to a representation space $\mathcal{Y}$ ($h(\cdot; \boldsymbol{\psi}) : \mathcal{Z} \rightarrow \mathcal{Y}$) and the global surrogate encoder $g^s(\cdot; \boldsymbol{\phi}^s)$ maps $\mathcal{Y}$ to $\mathcal{Z}$ ($g^s(\cdot; \boldsymbol{\phi}^s) : \mathcal{Y} \rightarrow \mathcal{Z}$). The server's objective function for learning the

parameters of the decoder-surrogate encoder pair is expressed by

$$\boldsymbol{\psi}, \boldsymbol{\phi}^s \leftarrow \underset{\boldsymbol{\psi}, \boldsymbol{\phi}^s}{\arg\min} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathcal{L}(\mathbf{z}_{k,i}, g^s(\mathbf{y}_{k,i}; \boldsymbol{\phi}^s)), \tag{3.11}$$

where $\mathbf{y}_{k,i} = h(\mathbf{z}_{k,i}; \boldsymbol{\psi})$. $\mathbf{z}^s$, which are latent space vectors, are returned to the clients. For training of local encoder, the client makes use of a local surrogate decoder which is given by $h_k^s(\cdot; \boldsymbol{\psi}_k^s) : \mathcal{Z} \rightarrow \mathcal{X}$. The surrogate decoder provides the reconstructed output at the client, which allows the reconstruction loss to be computed and the local encoder-surrogate decoder pair to be trained. The overall local-objective function as a result of this is given by,

$$\boldsymbol{\phi}_k, \boldsymbol{\psi}_k^s \leftarrow \underset{\boldsymbol{\phi}_k, \boldsymbol{\psi}_k^s}{\arg\min} \sum_{i=1}^{N_k} \mathcal{L}_k(\mathbf{x}_{k,i}, h_k^s(\mathbf{z}_{k,i}^s; \boldsymbol{\psi}_k^s)), \tag{3.12}$$

where $\mathbf{z}_{k,i}^s = g^s(h(g_k(\mathbf{x}_{k,i}; \boldsymbol{\phi}_k); \boldsymbol{\psi}); \boldsymbol{\phi}^s)$, where $\mathbf{x}_{k,i} \in \mathcal{D}_k$ for all $k \in \mathcal{K}$. Compared to the single-client optimization framework, which is given in (3.9), we infer from (3.12) that global information is included in $\mathbf{z}_{k,i}^s$, because of which parameters of local encoder-surrogate decoder model are generalized. Client-specific parameters, specifically $\boldsymbol{\phi}_k, \boldsymbol{\psi}_k^s$, are obtained at the end of training in pFedAE for each client k, resulting in personalisation. In a nutshell, including a surrogate encoder at the server and K surrogate decoders at clients, we partition the centralised autoencoder optimisation framework given in (3.3) into an FL-based client-server optimisation framework. As a result, in pFedAE, the distributed local and global objective functions in the presence of unlabeled data are given by

$$\begin{aligned} \textbf{Server} : \min_{\boldsymbol{\psi}, \boldsymbol{\phi}^s} F(\boldsymbol{\psi}, \boldsymbol{\phi}^s) &= \sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathcal{L}(\mathbf{z}_{k,i}, g^s(\mathbf{y}_{k,i}; \boldsymbol{\phi}^s)), \\ \textbf{Client} : \min_{\boldsymbol{\phi}_k, \boldsymbol{\psi}_k^s} f_k(\boldsymbol{\phi}_k, \boldsymbol{\psi}_k^s) &= \sum_{i=1}^{N_k} \mathcal{L}_k(\mathbf{x}_{k,i}, h_k^s(\mathbf{z}_{k,i}^s; \boldsymbol{\psi}_k^s)). \end{aligned} \tag{3.13}$$

One thing to note in the above optimization framework is that the latent space representation $\mathbf{z}$ should be similar to $\mathbf{z}^s$ as far as possible. As it is already mentioned that training for a FL setup takes place over several communication rounds; therefore, the difference between $\mathbf{z}$ and $\mathbf{z}^s$ before convergence will carry essential global information to the clients because of which model generalizes in the initial few rounds. After the ap-

Algorithm 1: Personalized Federated Autoencoder

Input: Dataset $\mathcal{D}_k$ at the k-th client, T-number of rounds, E-number of local epochs, η-learning rate ;
Initialize parameters at all clients;
for $t \in [T]$ **do**
  for E *epochs* **do**
    $\forall k \in \mathcal{K}$, sample mini-batch $B_k \subset \mathcal{D}_k$
    $\{\phi_k, \psi_k^s\} \leftarrow \{\phi_k, \psi_k^s\} - \eta \nabla f_k(\phi_k, \psi_k^s)$
  end
  At Server: $\{\phi, \boldsymbol{\psi}^s\} \leftarrow \{\phi, \psi^s\} - \eta \nabla F(\phi, \psi^s)$
end
Output: Personalized client encoders (parameters ϕ_k for $k = 1, \ldots, K$) after T communication rounds.

proach method convergences, the model's personalisation aspect comes into the picture, and the objective function for minimization is given in (3.13) can be achieved.

One of the most important aspects of pFedAE is that the size of the latent variable, a user-specified parameter, determines the amount of data transferred during the communication round. As a result, the server and client autoencoders can be arbitrarily large without compromising communication efficiency.The summary of our method pFedAE is given in Alg. 1.

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, we have discussed the experimental setup used to evaluate and compare pFedAE with several Unsupervised Federated Learning baselines. These experiments aim to assess the performance and effectiveness of pFedAE compared to state-of-the-art baselines. The baselines used for comparison are Orchestra Lubana *et al.* (2022) and FedAvg-AE McMahan *et al.* (2017), federated versions of discriminative self-supervised learning methods such as SimCLR Chen *et al.* (2020), SpecLoss HaoChen *et al.* (2021), SimSiam Chen and He (2021), and BYOL Grill *et al.* (2020), and unsupervised personalized single-client training technique. The main motive of this chapter is to provide a clear understanding of the evaluation strategies employed to compare pFedAE with the above-mentioned baselines in the field of Unsupervised Federated Learning.

At the end of the chapter, we will provide solutions along with appropriate reasons to the following questions: (a) Is personalization helpful in the field of Federated Learning in providing improvised feature representation? (b) Will hyperparameter tuning be helpful to get better results while using a small set of labelled data? (c) What will be the local and global impact of statistical heterogeneity, number of epochs, and number of clients on unsupervised FL?

4.1 Datasets and Partitioning

To compare the performance of pFedAE with other state-of-the-art baselines, we have used CIFAR10 and CIFAR100 datasets Krizhevsky *et al.* (2009). Both the datasets mentioned above comprise 50,000 training images and 10,000 testing images. The difference in both these datasets is that CIFAR10 consists of 10 classes whereas CIFAR100 consists of 100 classes; both datasets have an equal number of images per class. Both CIFAR10 and CIFAR100 consist of RGB images of size 32x32.

For the purpose of Federated Learning-based experiments, we divide the dataset into K partitions for K clients. We can partition the Non-IID data for statistical heterogeneity in two following ways:

- **Dirichlet Distribution :** Dirichlet Distribution is the best possible Non-IID solution representing the real-world problem faced by Federated Learning methods because of data heterogeneity. Dirichlet Distribution is a probability distribution which is defined for probability vectors. These probability vectors comprise data for each client in the Federated Learning network from different classes. Considering the parameters of Dirichlet Distribution, we use α for each client's class data distribution. All the clients in the network will have different classes of data, i.e. Non-IID nature of the data can be seen when $\alpha \to 0$ whereas, all the clients in the network will have a similar type of data, i.e. IID nature of the data can be seen when $\alpha \to \infty$. For the experiments for the evaluation and comparison of pFedAE we have chosen four values of α 0.001 (highly heterogeneous), $0.1, 1$ (moderately heterogeneous) and 10^5 (homogeneous).

- **Pathological Setting :** The class pathological setting is where each client in the Federated Learning network is provided with data samples from specific classes. In case of fewer clients, it is also considered that no two classes can have the same set of classes. For our experiments, we provide each client two classes of data samples.

4.2 Implementation details

The implementation of pFedAE is done using the PyTorch framework. Using A100 NVIDIA GPUs, K clients are simulated during training. With the help of Pytorch backend communication, the clients and server can communicate easily. For the encoder, ResNet-18 is used as the backbone. We used CNN for the decoder, and the classifier for implementing pFedAE is a two-layer multilayer perceptron. For our experiments, we have set the number of clients to $K = 10$, α (Dirichlet parameter) to 0.1, the number of local epochs to $E = 1$ and the number of communication rounds to $C = 100$ unless otherwise mentioned. For all our experiments, we have kept batch size $B = 256$. Adam is the optimizer for the autoencoder training with the learning rate $\eta = 0.01$, whereas SGD with the learning rate $\eta = 0.1$ is used for the classifier training.

4.3 Baselines

We have used the baselines listed below for the comparison and evaluation of pFedAE:

- Unsupervised FL methods based on generating generalized models, such as Orchestra Lubana *et al.* (2022) and FedAvg-AE. FedAvg-AE trains the backbone autoencoder unsupervised using federated averaging McMahan *et al.* (2017).
- The next set of baselines which we chose for comparison with pFedAE are SimCLR, SpecLoss, SimSiam, and BYOL Lubana *et al.* (2022), which are federated versions of discriminative self-supervised learning algorithms.
- Another naive baseline we chose is single client training based on unsupervised personalized Federated Learning where feature representation is learned using local data using (3.9).
- The last baseline is based on a centralized supervised machine learning technique (AE-CenS). An autoencoder is used to learn the feature representation with all the data present at a single location. After that, a classifier is trained and the backbone is fine-tuned using the data available.

4.4 Linear and Semi-supervised Evaluation and Results

We have used two methods for the evaluation of the feature representation learned by pFedAE:

- Linear Evaluation Protocol where the backbone of the pFedAE model and baseline model is trained for 100 rounds. After the backbone training is done, we freeze the parameters of the backbone model, and a classifier is trained using the global data. The input of the classifier is the features encoded from the backbone model. The classifier training takes place for another 100 rounds.

- The second evaluation protocol is Semi-Supervised Evaluation. Here, we have a condition that only a small subset of the available data is labelled. We have evaluated Semi-Supervised evaluation for two settings: In one set, we have 1% global labelled data, and in the other, we have 10% of the global labelled data. For this evaluation protocol, we have not retrained the backbone model. Instead, we have used the same trained backbone model used in Linear Evaluation Protocol, and a classifier is fine-tuned for the model using 1% or 10% global labelled data for 100 rounds.

The evaluation and comparison of pFedAE with other baselines is reported in terms of global and local accuracy in the Table . 4.1. Global test accuracy evaluated on the global test set is the average accuracy for all the clients in the network. To make pFedAE a personalised model, we have also measured the local accuracy on the local test set, which is the average accuracy for all the clients in the network.

From the table, the following can be inferred:

Table 4.1: Top-1 Global and local accuracy (%) comparison under the Linear and Semi-supervised evaluation protocol for statistically heterogeneous ($\alpha = 0.1$) setting and the pathological (2-class) non-IID for CIFAR10 and CIFAR100 datasets.

Methods	CIFAR10			CIFAR100		
(Global)	Linear	10%	1%	Linear	10%	1%
pFedAE	**82.38**	**68.50**	**65.12**	**53.27**	**46.95**	**45.90**
Orchestra	70.64	68.29	54.25	35.64	27.65	8.02
f-BYOL	66.18	64.15	50.02.	38.88	29.47	10.67
f-SpecLoss	64.53	59.78	45.94	35.99	25.56	8.92
f-SimSiam	61.95	58.44	40.71	36.92	27.59	9.74
f-SimCLR	58.15	54.97	42.84	33.49	24.35	8.40
Single Client	68.57	35.78	20.83	33.14	33.42	30.37
FedAvg-AE	37.04	50.12	50.75	20.09	10.137	8.57
AE-CenS	86.33			55.64		
Methods	CIFAR10			CIFAR100		
(Local)	Linear	10%	1%	Linear	10%	1%
pFedAE	**83.16**	**71.24**	**65.87**	**53.69**	**47.34**	**46.06**
Single Client	76.98	45.22	30.35	35.802	36.62	29.73
FedAvg-AE	40.95	50.169	50.82	20.72	10.22	8.72
Method	CIFAR10			CIFAR100		
(2-class)	Linear	10%	1%	Linear	10%	1%
pFedAE	82.29	67.34	65.03	53.25	45.67	44.23

- From the Linear protocol evaluation technique, it can be seen that pFedAE is better than all the other unsupervised Federated Learning baselines. pFedAE performs better than Orchestra, which has reported the best accuracy under the unsupervised Federated Learning technique.
- In the semi-supervised setting, it can be seen that even though there is a drop in the performance of pFedAE it still outperforms other baselines. One more thing to note here is that pFedAE is robust to the availability of global labelled data for both CIFAR10 and CIFAR100. pFedAE still outperforms all the baselines.
- pFedAE outperforms single client training, which is a personalized baseline, meaning that global information is also an important factor for personalization.
- It can also be seen that pFedAE is at par with AE-CenS, which is a supervised centralized autoencoder-classifier baseline. For both CIFAR10 and CIFAR100, AE-CenS is $3 - 5\%$ better than pFedAE.

4.5 Results on Attributes of Federated Learning

In this section, we have evaluated the performance of pFedAE on various attributes like statistical heterogeneity, number of epochs and scalability of clients (number of clients). All the results are evaluated under the linear probe evaluation scheme.

4.5.1 Sensitivity to Statistical heterogeneity

We have used four different variations of α to analyse the performance of pFedAE compared to other baselines on the basis of global and local accuracy. The results of statistical heterogeneity can be seen in Fig. 4.2 and Fig. 4.3. While measuring global accuracy, we observe that pFedAE demonstrates robustness to statistical heterogeneity. It consistently outperforms all the baselines across different values of α. Regardless of the degree of statistical heterogeneity in the dataset, pFedAE consistently achieves higher accuracy than the baseline. This indicates that the personalized approach of pFedAE is very effective in varying data distribution across all the clients in the network. While measuring local accuracy, it can be observed that for the CIFAR10 dataset across the communication rounds, the accuracy variation is higher than CIFAR100. In Fig. 4.4, we plot the clients' mean and variation of local and global accuracy. We can observe a noticeable variation in accuracy, the reason being higher class level diversity per client in CIFAR100 as compared to that of CIFAR10. Since CIAFAR100 has access

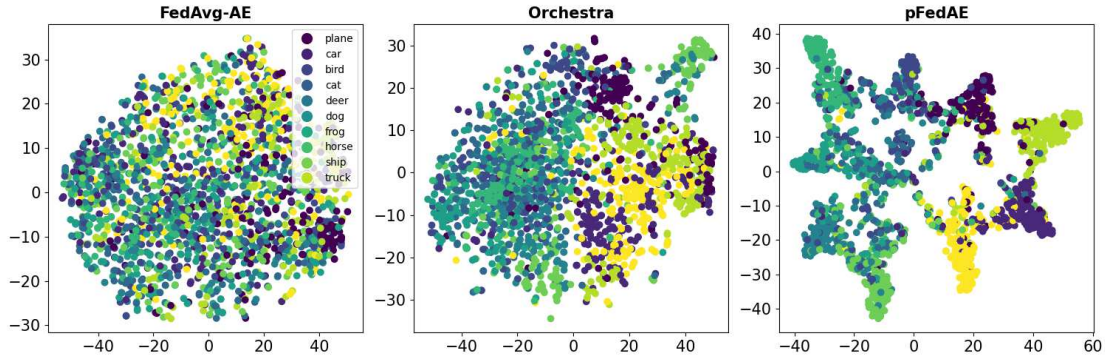

Figure 4.1: t-SNE projections of the latent representation from vanilla FL McMahan *et al.* (2017) where the underlying ML model is an autoencoder (FedAvg-AE), Orchestra Lubana *et al.* (2022) and the proposed pFedAE algorithm. PFedAE provides meaningful latent representations for different classes, albeit being an unsupervised technique.

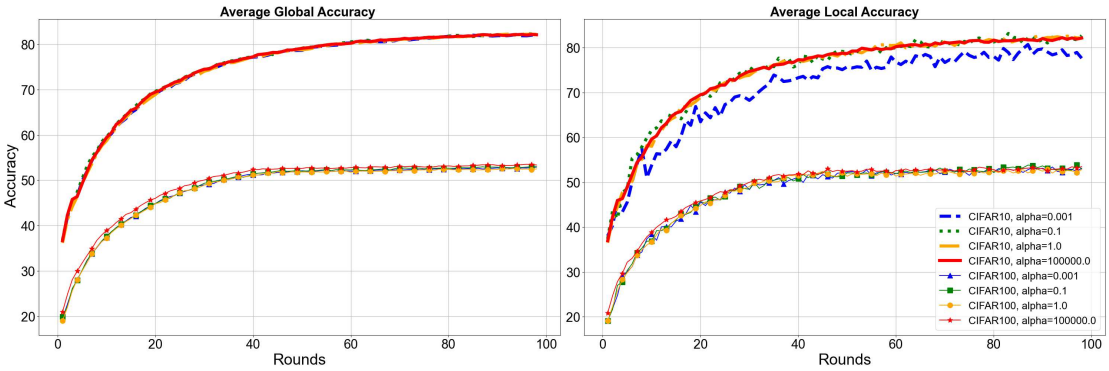

Figure 4.2: Global and local accuracy convergence plots for different values of α for CIFAR10 and CIFAR100 datasets.

to more classes as compared to CIFAR10 the model trained for CIFAR100 has a better generalized and personalized model.

In Table. 4.1, pathological Non-IID performance of pFedAE has been demonstrated. In the pathological experimental setup, we consider two classes for each client in the network. pFedAE outperforms existing Unsupervised Federated Learning approaches.

4.5.2 Robustness to Local Epochs

Robustness to the local number of epochs is one of the most important and crucial attributes of Federated Learning in two situations:

- **High Communication Cost :** When the number of local epochs is increased, the federated Learning technique is likely to perform well in situations where high communication costs are a problem. Because having more local epochs allows each client to perform more iterations of local training before communicating

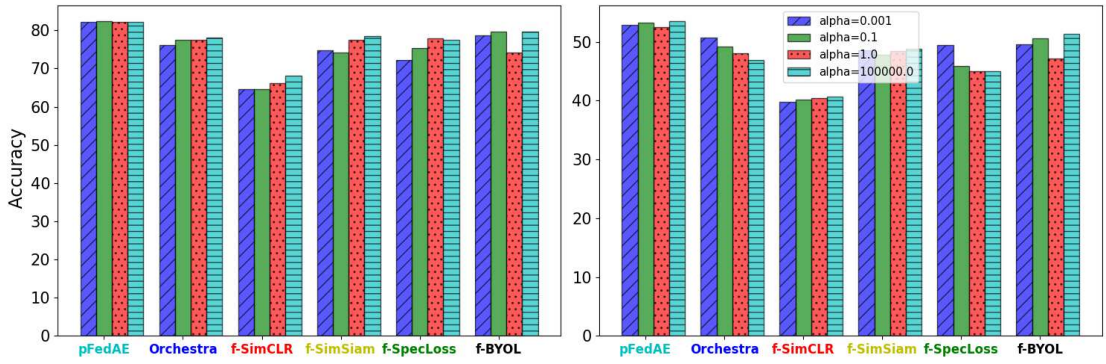

Figure 4.3: Sensitivity to statistical heterogeneity (varying values of α for CIFAR10 (left) and CIFAR100 (right) datasets.

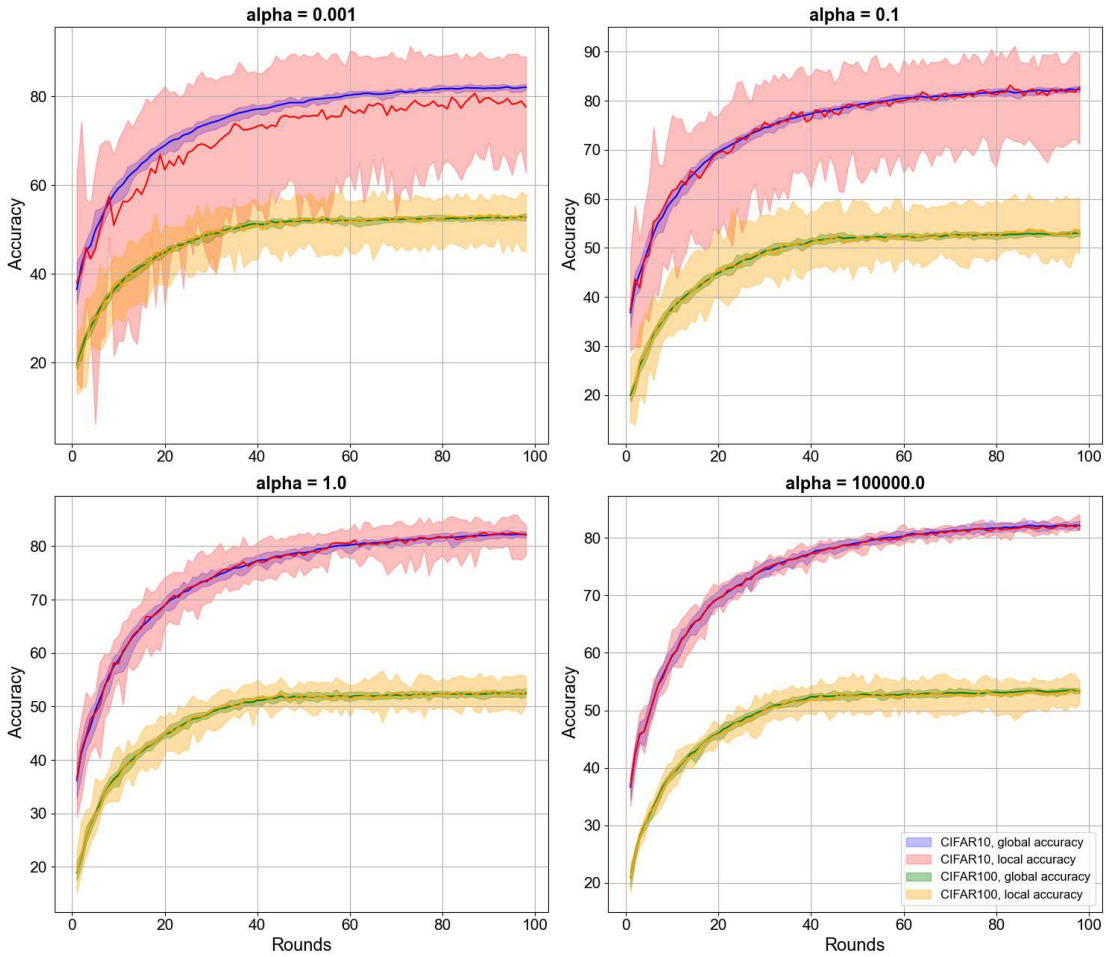

Figure 4.4: Standard deviation plots for local and global accuracy across clients, for different values of α on CIFAR10 and CIFAR100 datasets.

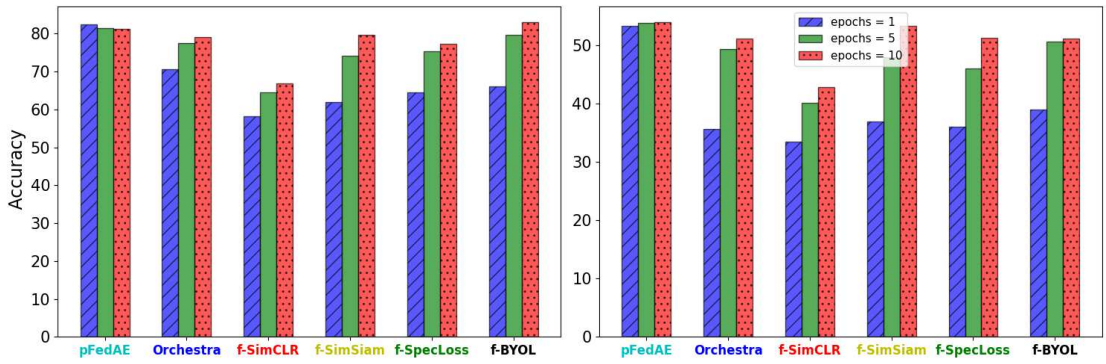

Figure 4.5: Robustness to local epochs for CIFAR10 (left) and CIFAR100 (right).

with the central server, it is more efficient. As a result, more local epochs and fewer communication rounds are required to achieve convergence or good performance.

- **Computational constraints :**By ensuring robustness to a limited number of local epochs, the Federated Learning technique may handle clients with varying computational constraints while yet providing a fair learning experience for all clients in the network. When clients face computational constraints in real-life scenarios, this capability is useful.

In Fig. 4.5 and Fig. 4.6, we observe the robustness of the pFedAE framework. If we reduce the number of local epochs, the global and local accuracy of pFedAE is robust, whereas the baselines perform poorly. Therefore, it can be said that pFedAE is a useful Federated Learning framework wherever there are high communication costs or computational constraints.

4.5.3 Scalability

The pFedAE framework's scalability is demonstrated in a cross-silo setting by altering the number of clients. For varied amounts of client settings, the number of local epochs is linearly scaled so that the overall number of iterations remains constant across all settings. According to Fig. 4.6 and Fig. 4.7, pFedAE achieves consistent performance across all settings, unlike other baselines where performance decreases as the number of clients increases. The decline in baseline performance can also be attributed to a lack of robustness to local epochs. pFedAE, as described in the previous section, is resistant to local epochs. Even with a small amount of data, it can still outperform other baselines.

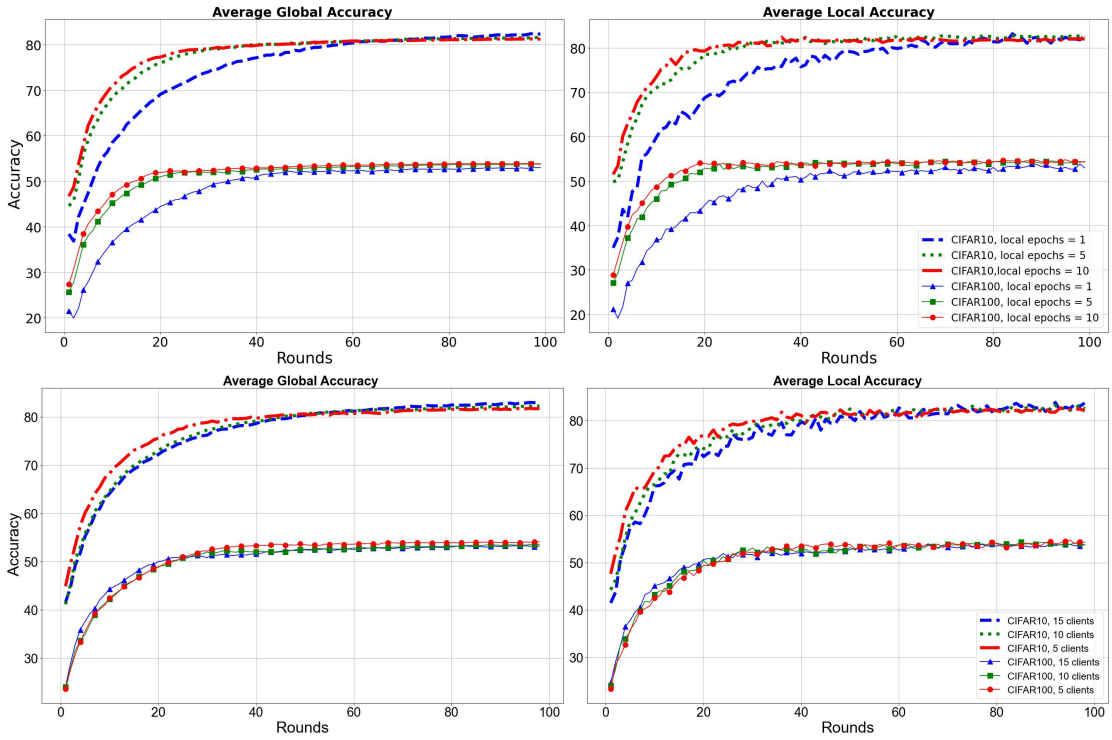

Figure 4.6: Global and local accuracy convergence plots for different numbers of local epochs (first and second from the left) and different numbers of clients (third and fourth from the left) in a cross-silo setting.

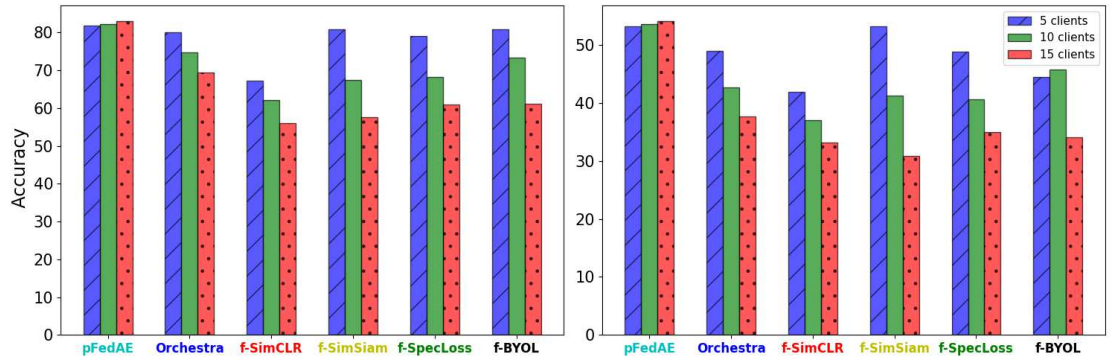

Figure 4.7: Varying number of clients (scalability) for CIFAR10 (left) and CIFAR100 (right).

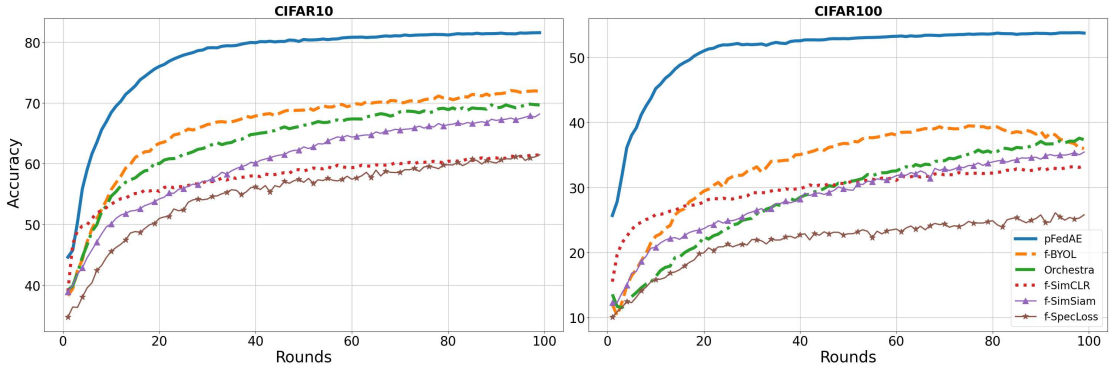

Figure 4.8: Convergence of pFedAE as compared to baseline methods using $E = 5$ local epochs and $C = 100$ rounds.

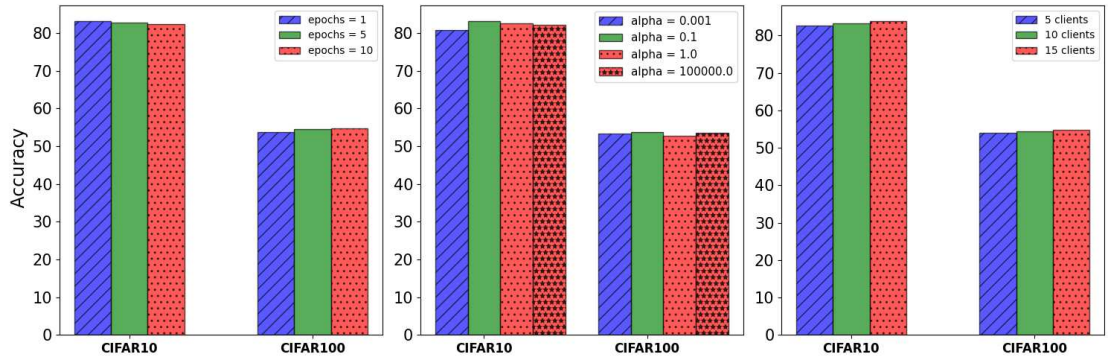

Figure 4.9: Local accuracy behaviour of FL attributes.

4.5.4 Convergence

As stated in Chapter 1, communication efficiency and complexity are critical features of Federated Learning. As a result, if a Federated Learning technique converges within a few communication rounds, it is considered optimum. The accuracy against communication rounds figure in Fig. 4.8 can be used to compare pFedAE to different baselines. It can be observed that pFedAE achieves great accuracy in just a few communication rounds.

4.5.5 Local Accuracy

We have demonstrated the average local accuracy on test data partitioned using the Dirichlet distribution to evaluate the personalising aspect of pFedAE. From Fig. 4.9, the local accuracy trend with varying numbers of local epochs, different values of α, and scalability with clients. Like global accuracy, local accuracy is resistant to all of the above-mentioned variances.

4.6 Assessment of Latent Space and Discussions

This section assesses the latent space quality generated in pFedAE.

4.6.1 Qualitative assessment via t-SNE

To assess latent space transformation by pFedAE, 2-dimensional t-SNE visualization on CIFAR10 test data is carried out. We have shown the t-SNE visualisation of latent space learned from pFedAE, FedAvg-AE, and Orchestra in Fig. 4.1. Even though our proposed method pFedAE is an unsupervised technique, the representation formed is well-separated in terms of target classes. As shown in Fig. 4.10, the latent space visualisation is also carried out with a variable number of local epochs. It can be observed from the t-sne plots that the visualization is consistent with the accuracy stated in the previous section, i.e. it can be clearly seen that by using pFedAE we can obtain distinction between latent space even with 1 local epoch, whereas in orchestra lower, distinction, can be observed even though more number of local epochs was used. Furthermore, for $alpha = 0.1$, Fig. 4.11 shows the influence of pFedAE on latent space learning and compares it with a single client training baseline. While comparing pFedAE and single-client training baseline, it can be seen that the global data projections at clients are poor in the case of single-client training. In Fig. 4.12, using a trained encoder of pFedAE and single client training baseline, we have shown t-sne projections of both global and local test data. It can be observed that both the global and local projections obtained by pFedAE are more informative as compared to that of single-client training. With this, we can say that accuracy is maintained at both global and local levels using pFedAE.

4.6.2 Quantitative Assessment via Inter-class difference

The difference of angle between the average embedding belonging to various classes in the latent space is used to measure the inter-class difference of latent space vectors, as given below

$$\mathbf{z}_c = \frac{1}{|\mathcal{D}_c^{\text{test}}|} \sum_{\mathbf{x} \in D_c^{\text{test}}} g(\mathbf{x}), \tag{4.1}$$

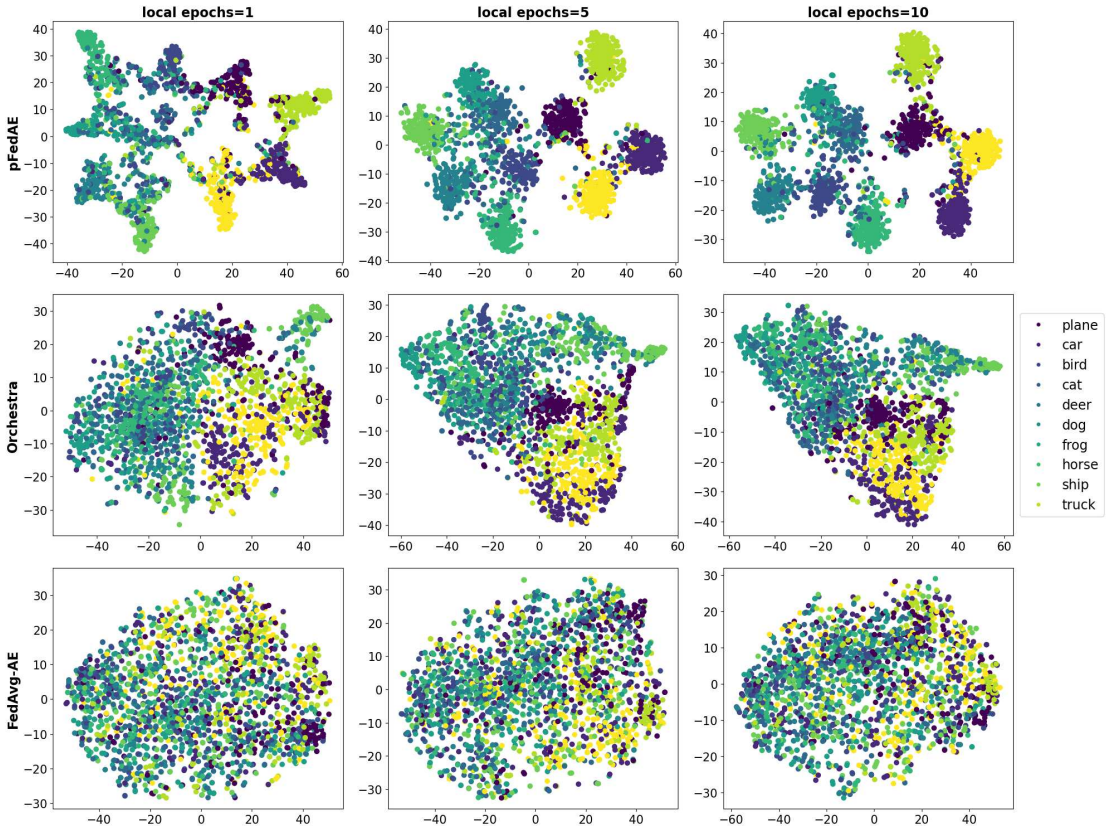

Figure 4.10: t-SNE projections of encoder models trained using pFedAE, Orchestra and FedAvg-AE using 1, 5 and 10 local epochs.

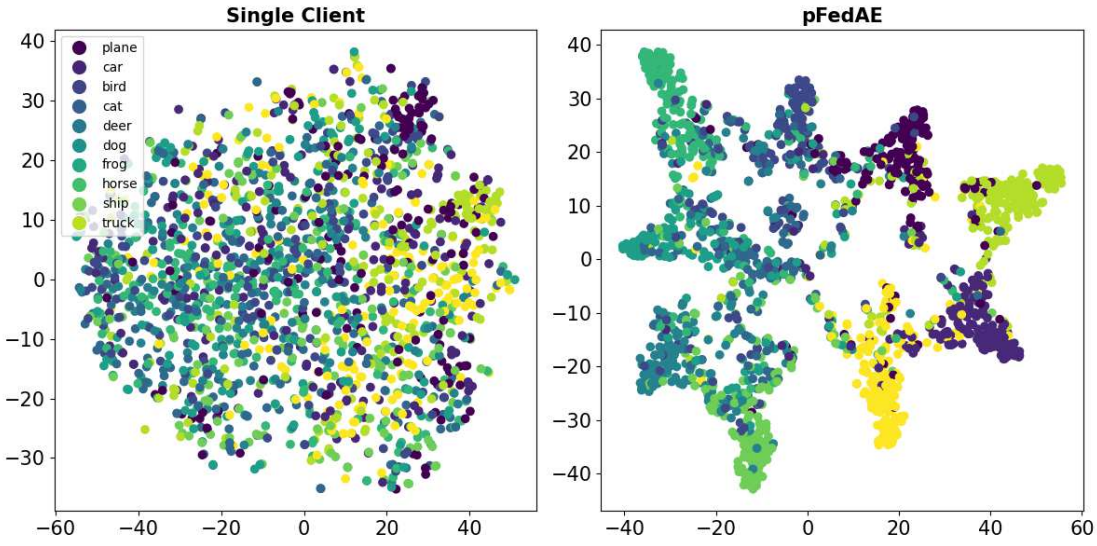

Figure 4.11: t-SNE projections of encoder models trained using pFedAE and single client training.

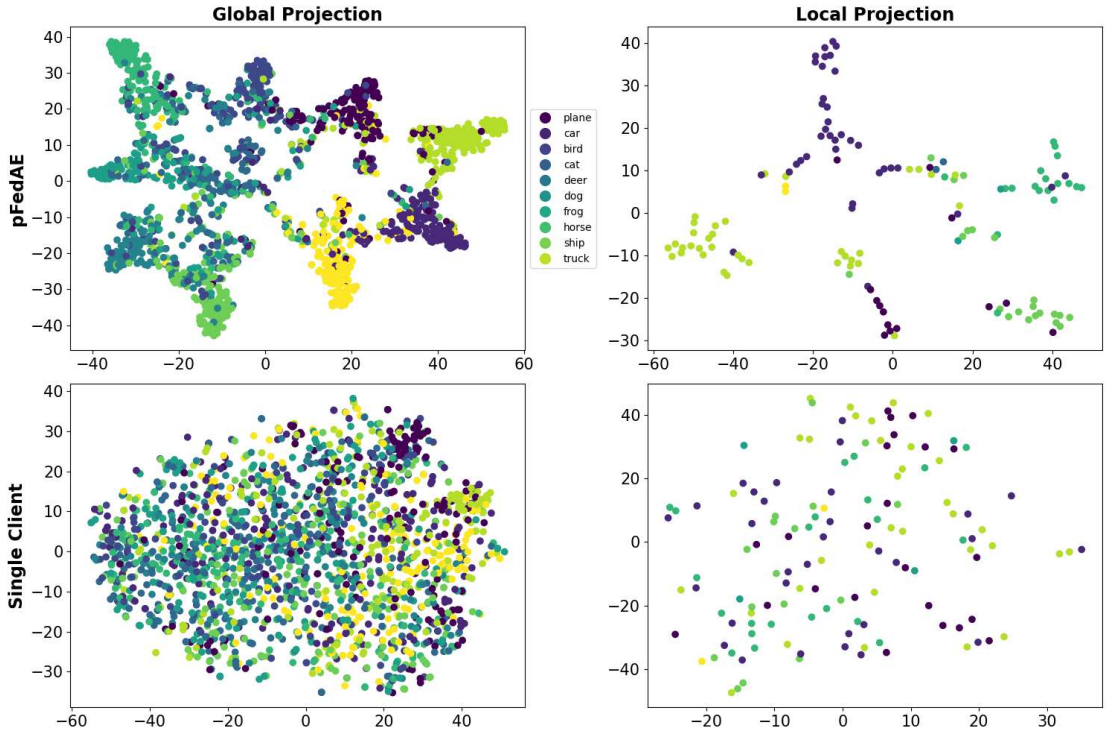

Figure 4.12: t-SNE projections of encoder models trained using pFedAE and single client training on the global test dataset and local test dataset with $\alpha = 0.1$

where,$\mathcal{D}_c^{\text{test}}$ is the test dataset for the class c.This average embedding is used as a general embedding for that specific class and is calculated using Eq. (4.1). The greater the inter-class difference, the better the embedding represents class information. We compute the angles between all pairs of class-wise general embedding:

$$\text{Angle}(c_i, c_j) = \arccos(\text{sim}(\mathbf{z}_{c_i}, \mathbf{z}_{c_j})) \tag{4.2}$$

We next create a histogram of the angle differences discovered between each pair. According to Fig. 4.13, pFedAE has higher inter-class angles ($60circ$ to $80circ$) than FedAvg-AE and Orchestra, showing improved class discrimination. With the help of these studies, we can conclude that pFedAE produces higher-quality latent space than any other baseline. These trials reveal that pFedAE findings have a higher quality of latent space than baselines.

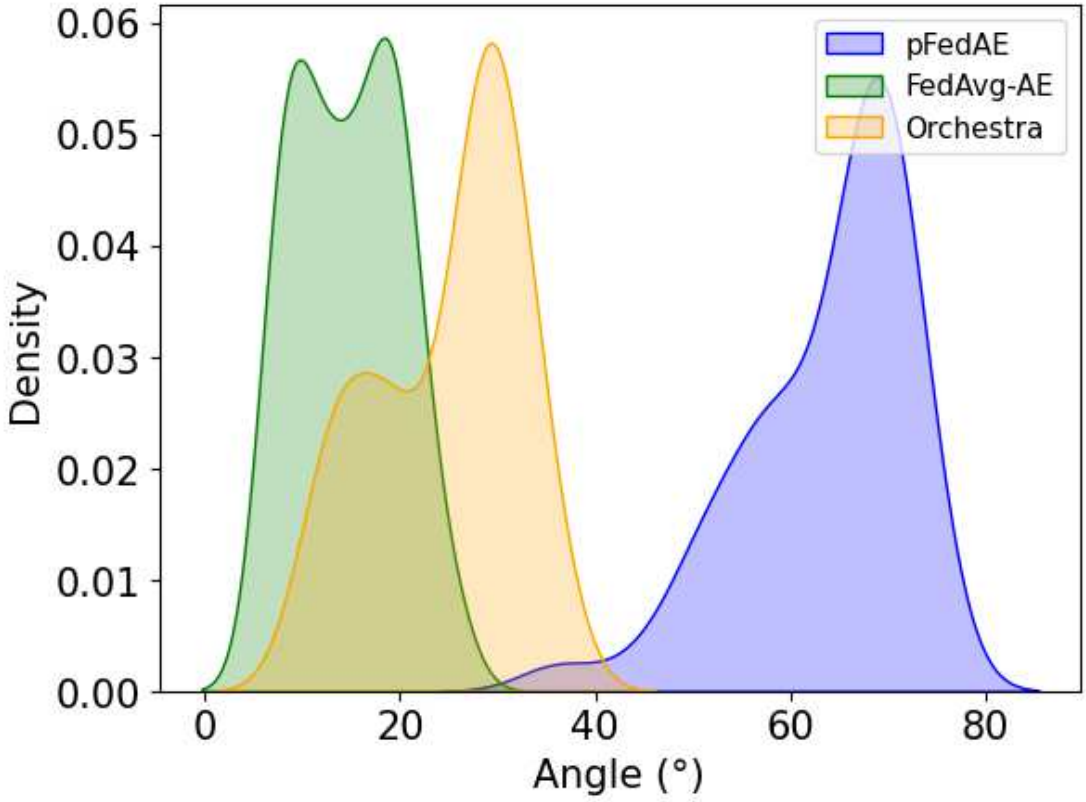

Figure 4.13: Histogram on inter-class difference.

CHAPTER 5

CONCLUSION

In this paper, we addressed an important but often overlooked problem of representation learning using a personalised federated learning framework known as pFedAE. The autoencoder's optimisation architecture is divided into global and local objective functions. The latent space representation is used at the server, while locally available data is used for the client. We employed a surrogate encoder at the server and K surrogate decoders at the clients for optimisation. With the help of various outcomes, we demonstrated that our method is both generalised and personalised. As a result, pFedAE outperforms all baselines, including an orchestra, FL-based autoencoder, and personalised single-client training. We are optimistic that pFedAE will be helpful in situations where communication efficiency precedes privacy.

REFERENCES

1. **Arivazhagan, M. G.**, **V. Aggarwal**, **A. K. Singh**, and **S. Choudhary** (2019). Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*.

2. **Baldi, P.**, Autoencoders, unsupervised learning, and deep architectures. *In Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings, 2012.

3. **Bengio, Y.**, **A. Courville**, and **P. Vincent** (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, **35**(8), 1798–1828.

4. **Brodie, M.**, **E. Pliner**, **A. Ho**, **K. Li**, **Z. Chen**, **S. Gandevia**, and **S. Lord** (2018). Big data vs accurate data in health research: large-scale physical activity monitoring, smartphones, wearable devices and risk of unconscious bias. *Medical hypotheses*, **119**, 32–36.

5. **Chen, T.**, **S. Kornblith**, **M. Norouzi**, and **G. Hinton**, A simple framework for contrastive learning of visual representations. *In International conference on machine learning*. PMLR, 2020.

6. **Chen, X.** and **K. He**, Exploring simple siamese representation learning. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.

7. **Dash, S.**, **S. K. Shakyawar**, **M. Sharma**, and **S. Kaushik** (2019). Big data in healthcare: management, analysis and future prospects. *Journal of Big Data*, **6**(1), 1–25.

8. **Fallah, A.**, **A. Mokhtari**, and **A. Ozdaglar** (2020). Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*.

9. **Girshick, R.**, **J. Donahue**, **T. Darrell**, and **J. Malik**, Rich feature hierarchies for accurate object detection and semantic segmentation. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

10. **Grill, J.-B.**, **F. Strub**, **F. Altché**, **C. Tallec**, **P. Richemond**, **E. Buchatskaya**, **C. Doersch**, **B. Avila Pires**, **Z. Guo**, **M. Gheshlaghi Azar**, *et al.* (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, **33**, 21271–21284.

11. **Hajihassani, O.**, **O. Ardakanian**, and **H. Khazaei**, Latent representation learning and manipulation for privacy-preserving sensor data analytics. *In 2020 IEEE Second Workshop on Machine Learning on Edge in Sensor Systems (SenSys-ML)*. 2020.

12. **Han, S.**, **S. Park**, **F. Wu**, **S. Kim**, **C. Wu**, **X. Xie**, and **M. Cha**, Fedx: Unsupervised federated learning with cross knowledge distillation. *In European Conference on Computer Vision*. Springer, 2022.

13. **HaoChen, J. Z.**, **C. Wei**, **A. Gaidon**, and **T. Ma** (2021). Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, **34**, 5000–5011.

14. **Hinton, G. E.** and **R. R. Salakhutdinov** (2006). Reducing the dimensionality of data with neural networks. *science*, **313**(5786), 504–507.

15. **Krizhevsky, A.**, **G. Hinton**, *et al.* (2009). Learning multiple layers of features from tiny images.

16. **Kulkarni, V.**, **M. Kulkarni**, and **A. Pant**, Survey of personalization techniques for federated learning. *In 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. 2020.

17. **Li, T.**, **A. K. Sahu**, **M. Zaheer**, **M. Sanjabi**, **A. Talwalkar**, and **V. Smith** (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, **2**, 429–450.

18. **Lu, N.**, **Z. Wang**, **X. Li**, **G. Niu**, **Q. Dou**, and **M. Sugiyama** (2022). Federated learning from only unlabeled data with class-conditional-sharing clients. *arXiv preprint arXiv:2204.03304*.

19. **Lubana, E. S.**, **C. I. Tang**, **F. Kawsar**, **R. P. Dick**, and **A. Mathur** (2022). Orchestra: Unsupervised federated learning via globally consistent clustering. *arXiv preprint arXiv:2205.11506*.

20. **Malekzadeh, M.**, **R. Clegg**, and **H. Haddadi**, Replacement autoencoder: a privacy-preserving algorithm for sensory data analysis. *In Proceedings-ACM/IEEE International Conference on Internet of Things Design and Implementation, IoTDI 2018*. 2018.

21. **Malekzadeh, M.**, **R. G. Clegg**, **A. Cavallaro**, and **H. Haddadi**, Mobile sensor data anonymization. *In Proceedings of the international conference on internet of things design and implementation*. 2019.

22. **McMahan, B.**, **E. Moore**, **D. Ramage**, **S. Hampson**, and **B. A. y Arcas**, Communication-efficient learning of deep networks from decentralized data. *In Artificial intelligence and statistics*. PMLR, 2017.

23. **Sattler, F.**, **T. Korjakow**, **R. Rischke**, and **W. Samek** (2021). Fedaux: Leveraging unlabeled auxiliary data in federated learning. *IEEE Transactions on Neural Networks and Learning Systems*.

24. **Shamsian, A.**, **A. Navon**, **E. Fetaya**, and **G. Chechik**, Personalized federated learning using hypernetworks. *In International Conference on Machine Learning*. PMLR, 2021.

25. **Smith, V.**, **C.-K. Chiang**, **M. Sanjabi**, and **A. S. Talwalkar** (2017). Federated multi-task learning. *Advances in neural information processing systems*, **30**.

26. **Tan, A. Z.**, **H. Yu**, **L. Cui**, and **Q. Yang** (2022). Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*.

27. **van Berlo, B.**, **A. Saeed**, and **T. Ozcelebi**, Towards federated unsupervised representation learning. *In Proceedings of the third ACM international workshop on edge systems, analytics and networking*. 2020.

28. **Wood, E.**, **T. Baltrušaitis**, **C. Hewitt**, **S. Dziadzio**, **T. J. Cashman**, and **J. Shotton**, Fake it till you make it: face analysis in the wild using synthetic data alone. *In Proceedings of the IEEE/CVF international conference on computer vision*. 2021.

29. **Zhao, Y.**, **P. Barnaghi**, and **H. Haddadi** (2022). Multimodal federated learning on iot data.

30. **Zhao, Y.**, **H. Liu**, **H. Li**, **P. Barnaghi**, and **H. Haddadi** (2020). Semi-supervised federated learning for activity recognition. *arXiv*.

31. **Zhuang, W.**, **X. Gan**, **Y. Wen**, **S. Zhang**, and **S. Yi**, Collaborative unsupervised visual representation learning from decentralized data. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021.