Multi-Center Federated Learning to Cluster Clients with non-IID data

A thesis submitted in fulfilment of the requirements
for the degree of

Doctor of Philosophy
in
Analytics

by

Ming Xie

to

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

August 2020

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Ming Xie* declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science*, *Faculty of Engg. & IT* at the University of Technology Sydney, Australia.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note: – Signature removed prior to publication.

[Ming Xie]

DATE: 30th August, 2022

PLACE: Sydney, Australia

Abstract

Federated learning (FL) is a new machine learning paradigm to collaboratively learn an intelligent model across many clients without uploading local data to the server. Non-IID data across clients is a significant challenge for the FL system because its inherited distributed machine learning framework is designed for the scenario of IID data across clients. Clustered FL is a type of FL method to solve non-IID challenges using a client clustering method in the FL context. However, even adopts a client clustering FL method still facing minor problems such as unstable against client-wise outliers and the drop of model performance with model poisoning attack.

To face the aforementioned challenges, the main research objective of the thesis is to study that how to make FL effectively, seamlessly solved non-IID data across clients in horizontal clients partition settings.

The main research objective has been studied from four coherently linked perspectives: (i) how to make FL to address the non-IID distribution of data across different clients in a effective and scalable manner so that they can be applied to real world cases which consists of thousands of client and varies type of devices, (ii) how to make cluster FL methods more robust to client-wise outliers, (iii) how to make better balance between the performance of global models and the extent of personalisation of local models, (iv) how to make FL training more robust to model poisoning attack by density methods.

This thesis proposes a novel FL framework with robust clustering algorithm and secure the models to tackle client-wise outliers as well as model poisoning in the FL system. Specifically, we will develop a robust federated aggregation operator using a bootstrap median-of-means mechanism that can produce a higher breakdown point to tolerate a larger proportion of outliers. All work experiments on three benchmark datasets have demonstrated the effectiveness of the proposed method that outperforms other baseline methods in terms of evaluation criteria.

In short, we develop a original, effective clustered FL baseline algorithm which can improve FL performance in horizontal clients partition settings. We compared proposed work against several state-of-the-art FL algorithms using both synthetic and real-world data.

Dedication

dedicate my thesis to my mother and my wife for their love

ACKNOWLEDGMENTS

I, Ming Xie, acknowledge Dr. Chandranath Adak for providing this amazing thesis template.I am truly grateful to my supervisor, Prof. Guodong Long, for his great guidance through every facet of the research world. In the past few years, now only I acquired critical technical knowledge in Federated Learning research topic, but also how to clearly write and communicate those my ideas formally and accurately. In addition, I realize the importance of prepare presentations and engage audience plays a vital role in research. Likewise, interact with other academics and build research collaborations. All of this was crucial to my future work and life.

A very special gratitude goes to A/Prof. Lu Qin, who has provided many life changing ideas and advice not only on academic but also on personal life. Words cannot express my emotions in this regard.

Thanks to all those who shared endless discussion and conversation in UTS building 2. It was great spending time in the building and in the laboratory with all of you.

I am also grateful to all the following people, who have helped and supported me along the way: Prof Chengqi Zhang, Dr. Jing Jiang, Dr. Xueping Peng, Mr. Wensi Tang, Mr. Jie ma, Dr. Shen Tao, Dr. Xubo Wang, Dr. Wentao Li, Mr. Yang Wang.

I am indebted to Faulty of Engineering & IT of the University of Technology Sydney for the wonderful, inclusive and productive work environment, to LINKAGE scholarship for initially funding my PhD. Thank you all the staff, admins and proof readers of my Faulty, they are very professional. I am extremely grateful for living in Sydney, even though I moved out from time to time, it is a city full of culture diversity, beautiful beach and I love to work here.

Finally, I would like to thank my family, for all the obstacles they quietly removed so I could realize my dreams.

List of Publications

Related to the Thesis :

1. Ming Xie, Jie Ma, Guodong Long, "Personalized Federated Learning with Robust Clustering against Model Poisoning", Advanced Data Mining and Applications, 2022 [CORE rank B] (Accepted by 10 Aug 2022)

2. Ming Xie, Jie Ma, Guodong Long, Chengqi Zhang, "Robust Clustered Federated Learning", The Asia Pacific Web (APWeb), 1-15, 2022 [CORE rank B] (Accepted by 7 June 2022)

3. Ming Xie, Jing Jiang, Tao Shen, Yang Wang, Leah Gerrard, Allison Clarke, "A Green Pipeline for Out-of-Domain Public Sentiment Analysis", International Conference on Advanced Data Mining and Applications, 190-202, 2021, DOI: https://doi.org/10.1007/978-3-030-95405-5_14, [CORE rank B] (**Best Student Paper Award**)

4. Guodong Long*, Ming Xie*, Tao Shen, Tianyi Zhou, Xianzhi Wang, Jing Jiang, "Multi-center Federated Learning: Client Clustering for Better Personalization", World Wide Web Journal, June 2022, DOI: https://doi.org/10.1007/s11280-022-01046-x [**CORE rank A**]

   Notes: * indicates equal contributions.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Motivation

1.1.1 Background

From the day of existence of computers, people have been dreaming about artificial intelligence (AI) - machines that can think, behave, recognize or act that only humans can do. The question that whether humans able to make or create computers into such intelligent machines has been controversial and many influential people have been raised views or provide answers to this question. Getting a solution to this question is very optimistic in this era because of the advance in machine learning, specifically Deep Learning, which is the study of computer algorithms that can learn automatically through experience and by the use of data. Breakthrough in computer technology [116] such as microprocessors enables humans to build powerful computers. Moreover, machine learning models powered by big data are also becoming ever more present in our everyday lives, with voice assistants using speech recognition on mobile phones and autoresponders and online customer support in the business, and self-driving cars employing computer vision to guide us to anywhere we want.

Computers today are Turing-complete, in other words, even human mind can be replicate by any computable algorithm, some work of Alan Turing did prove that any problem can be computed by Universal Turing Machine [102], thus, assuming that the human mind can be described by some algorithm, Turing Machine is powerful enough

to represent it. The difficult part of building artificial intelligence has become how to build a an algorithm that can produce desired behavior that human consider intelligent. Motivation for learning a pattern or a skill is to improve the success of humna in the real world. From this logic, even the whole learning problem seems too difficult to be solved at once. The way of human solves hard problem is to start from the basic. The goal for machine learning is to build a machine that can learn a pattern or a skill to a successful extent automatically by learning from experience(or data).

As an example, let's consider one of the most basic supervised learning problems, recognizing handwritten digits from images. Traditional machine learning requires first captured a training set, i.e. lots of data kept in this dataset, and the bigger the data, the better the machine can improve itself and can achieve better learning performance. A very well-known model of using data to achieve decision making of human-level success is in 2016, an AI program called AlphaGo became the champion winner of the game of Go. The unprecedented success has many contributing factors, though on one important reason is: that AlphaGo [96] used 30 million moves from 160,000 actual games as training data to achieve the excellent results.

With AlphaGo's and other deep learning (DL)-based research have seen great achievements, we have truly witnessed the huge potential in artificial intelligence, and have began to expect more mature, sophisticated AI technology, or even self-aware robots in many applications. Although being too ambitions to think of, people naturally hope that the emerging AI research can stay getting more fruits like AlphaGo and industry can harness data-enabled technologies to drive business growth. However, the current trend of machine learning popularity is a bit disappointing, with the exception of a few fields, the increasing volume of data being generated is making it difficult for traditional systems to keep up with. New big data solutions are needed to store, manage, and analyze all this data. [86, 118], rendering the realization of techniques such as collecting, compiling and making data available to scientists more challenging.

1.1.2 Federated Learning

In order to leverage proprietary data or data sets that under stricter data regulation, and break through the barriers between data sources, there is a need to introduce new machine learning technology. However, not all suitable cases have considerable properly labeled and complete data available in a centralized location (e.g., doctors' diagnoses from medical image analysis). Curating such large, high-quality datasets can be time-consuming and tedious and often requires domain experts. Efforts from

individual organizations result in data silos, with each containing high-quality but small datasets. For these type of domains, very few organizations manage to gather high-quality, complete, fully labeled, and sufficiently large datasets, which are required for these DL applications to be effective. Traditionally, data were gathered in a centralized location to build ML models. However, due to concerns related to data ownership and confidentiality, user privacy, and new laws over data management and data usage, such as the General Data Protection Regulation, private, secure, efficient, and fair distributed model training is required.

Hence, the question on how we want to build machine learning models learning from data that scatter across organizations, geographic locations, and potentially in massive number of users, in a secure, efficient and fair learning settings. Later, A new machine learning technology has been reported in 2016, with the first publications on federated averaging in telecommunication settings, which is known as Federated Learning (FL). The idea of FL is: instead of training on centralized data, separate models can be trained locally where the data reside in a distributed manner. Then, the respective local model updates can be communicated to obtain a global model, in which the communication process is carefully designed such that the data of an individual organization or device remain private.

In short, the goal of FL is to build *single*, *global* prediction model using data stored in different site and the key of this procedure under the constraint which the training data do not leave their premises. First, this work formally defines FL as a learning procedure mathematically. In particular, the goal is typically to minimize the following objective function:

$$\underset{w}{\text{minimize}} = F(w) \quad \text{,where} \quad F(w) = \sum_{k=1}^{m} p_k F_k(w) \tag{1.1}$$

Above equation lists a number of model-based parameters. The meaning of the parameters are:

- m is the total number of users
- p_k specifys the impact of each user, normally set to $p_k = \frac{1}{n}$ or $p_k = \frac{n_k}{n}$, where $n = \sum_k n_k$
- F_k often define as empirical risk of the local data

– n_k is the number of sample of local data

The remainder of the thesis will all reference this function as it is the vanilla algorithm. However, the other framework may alter this function to their application of interest.

Even though, computing, storage simple queries across distributed nodes, or user devices is not a new one. Many different techniques have been studied and applications are well established in past decades. However, there are various improvement over distributed computing of FL. First, FL trains a local model by directly using the computing power on remote devices. Second, in some settings and systems, those remote devices communicate with each other as to collaboratively train a global model while traditional distributed computing often work independently. Three, FL typically involves massive number of devices and the assumption made on the properties of local datasets are not complicated than traditional distributed computing.

Reference	**Focus Point**
[46]	Run all models and find the minimal value before assign cluster id
[90]	Hierarchy framework
[17]	Objective function different
[40]	Static clustering and Euclidean distance
[81]	Hypo cluster
[91]	Byzantine settings

Table 1.1: Close Existing Research

FL concept was first introduced by researchers at Google to update language models, in Google's keyboard system for word auto-completion [53]. FL builds a joint model using the data located at different sites, where each party contributes some data to train the model. The devices can be owned by different individuals or organizations, and can be of different types (e.g., smartphones, sensors, vehicles, etc.). The data is never centralized or shared with any third party; instead, the training takes place locally on the devices and the model is aggregated across the devices. Global model is then encrypted and shared among the participants so that no participant can reverse-engineer others' data. This resulting joint model performance is an approximation of the ideal model trained with centralized data. In practice, this added security and privacy results in certain accuracy loss, but it is often worth for specific application domains provided the fact data is hard to collect together for those application domains. In addition to the privacy and security benefits, collaborative training in FL can yield better models than those trained by individual organizations or devices. This aspect is also very import for

applications of IOT domain, health care. Please note, FL is a machine learning technology for decentralised data sets, this principle applicable to both traditional machine learning based models as well as DL based models. Many FL proposed method can work well with normal models, such linear models, trees and logistic regression, as well as DL models, such as CNN and RNN.

No doubt FL has promising potential, however, a number of work and related researcher maintain that a serious challenge assoicate with FL: the non-IID. or heterogeneity to improve the statistical and computational effectiveness of FL. Such challenges arise in Federated Learning, due to the highly decentralized system architecture. In FL, since the data source and computing nodes are end users' personal devices, the issue of data heterogeneity, also known as non-i.i.d. data, naturally arises. Exploiting data heterogeneity is particularly crucial in applications such as recommendation systems and personalized advertisement placement, and it benefits both the users' and the enterprises. For example, mobile phone users who read news articles may be interested in different categories of news like politics, sports or fashion; advertisement platforms might need to send different categories of ads to different groups of customers. FL has been employed in a variety of applications, ranging from medical to IoT, finance, transportation, defense, and mobile apps. Its applicability makes FL highly reliable, with several highly successful experiments having been conducted already.

1.1.3 Research Problems

client with non-IID data Despite the recent successes of Federated Learning in the past years, there are still many challenging problems to be solved. For instance, security of training process, data heterogeneity, Federated Learning incentive. The thesis is aiming to solve one of these problems, namely data heterogeneity. One major challenge is how to make Federated Learning to address the non-IID distribution of data across different clients in a effective and scalable manner so that they can be applied to large-scale network which consists of thousands of client and varies type of devices.

An important characteristics of multi-center FL is to group population into clusters, find their cluster identities then conduct normal local gradient descent over local data sets. However, often some data sets in FL setting are drift far away from majority of the population. This thesis claims that a bottle neck is the assumption upon which vanilla federated learning in non-IID data, that is one global model can not fit all clients [85].

To address this major problem, Chapter 3 proposed a clustered Federated Learning framework, however, other research problems also emerge from the proposed framework. Another challenge is how to enhance the clustering algorithm that our framework applied to be more robust to different types of client-wise outliers, i.e. client that behave differently from the majority of population. Chapter 4 focus on this specific problem by proposed a robust version of bootstrap of median-of-means clustering algorithm.

Therefore, a possible way to address the non-IID challenge and data heterogeneity is to divide the participants into many groups based on their learned gradients, this also known as clustering approach given a subset of the client population. We build FeSEM algorithm for this purpose. Several similar work have been published around or after the date of our FeSEM paper published in this sub area of Federated Learning. Their themes presented in Table 1.1 are summarized as follows. Their respective focuses will be explained in details in 2. The major theme of this thesis is to describe a different FL framework that have been developed to overcome the data heterogeneity challenges. As the best of our knowledge, our FeSEM algorithm is the only one that keeps multiple global models on the central server. To prove the usefulness of this algorithm, empirical results on three standard benchmark data sets will be extensively described. Finally, approaches that can possibly lead building of more personalization model by using regularisation and more robust and secure model aggregation by the use a K-means clustering variant will be discussed. In short, our goal is to find a solution that not only inherits the communication efficiency of the federated SGD but also retains the capability of handling non-IID data on heterogeneous datasets.

Robust clustering for multi-center On top of clients with non-IID data, there is a need to our proposed clustered Federated Learning for better clients clustering performance by the use of a Bootstrap of Median of Means clustering algorithm to overcome the assumption that all clients required to online. Chapter 4 focus on how to make multi-center FL more robust and practical for real world case and boost performance of the global model.

Consider a scenario where each client tries to train a model on customers' sentiments on food in a country. In a international information system. different countries collects their own client's data. Obviously, customers' reviews on food are likely to be related to their cultures, life-styles, and environments. Unlikely there exists a global model universally fitting all countries. Instead, pairwise collaborations among countries that share similarity in culture, life-styles, environments and other factors may be the key

to achieve reasonable inference performance in personalized federated learning with non-IID data. Also we would like to enhance our clustering algorithm to be more robust for possible client-wise outliers. Above problem is the focus of Chapter 4

Model Poisoning Attacks On top of clients with non-IID data, Model poisoning is another challenge in realistic FL case. Poisoning attacks in machine learning are a relatively new research area, and there are many current challenges associate with model poisoning. One challenge is to develop better methods for detecting and defending against poisoning attacks. Additionally, researchers are working to develop new and more robust machine learning models that are secure to poisoning attacks. In a distributed system of FL, some malicious agents may upload fake or dirty gradients to the server in the aggregation step, and then the aggregated model to distribute is poisoned. It is naive to adopt anomaly detection techniques to find these malicious agents or outliers. Local outlier factor (LOF) [16] is an efficient method based on the density of data points. Chapter 5 foucs how to make better balance between the performance of global models and the extent of personalisation of local models.

We formulate the problem of multi-center FL tried to solve as the joint clustering of users with penalized outliers, and then optimizing of the global model for users in each cluster, this is the main idea of 4. In particular, (i) each user's local model is assigned to its closest global model in terms of shared layers only, and (ii) each global model leads to the smallest loss over all the users in the associated cluster while the outliers will be penalized on updating global models. The optimization algorithm, which we use to solve the aforementioned problem, can be described as an EM algorithm. The proposed multi-center FL with bi-level personalized components not only inherits the communication efficiency of the federated SGD but also retains the capability of handling non-IID data on both individual and group levels. The import edge of this idea is to build good quality global models as well as local models.

An outlier detection method in machine learning is a technique used to identify unusual data points that do not conform to the general data distribution. These data points are typically considered to be noise or errors in the data set. Outlier detection methods can be used to pre-process data sets to remove outliers, or to identify unusual data points for further analysis.

There are a variety of outlier detection methods, each with its own advantages and disadvantages. Some common methods include:

- Density-based methods: These methods identify outliers based on the local density of data points. Data points that are isolated from the rest of the data are considered to be outliers.

- Distance-based methods: These methods identify outliers based on their distance from the rest of the data. Data points that are far from the center of the data distribution are considered to be outliers.

- Statistical methods: These methods identify outliers based on their deviation from the mean or median of the data. Data points that are significantly different from the rest of the data are considered to be outliers.

- Machine learning methods: These methods use a variety of techniques, such as support vector machines, to identify outliers.

Some examples of outliers detection in the theme of Federated Learning are those systematic mislabelling data or Byzantine failures [10, 11]. However, even a small proportion of outliers can render clustering unreliable, cluster centers and model parameter estimators can be severely biased, and thus reduce the performance of that model trained by the data from that cluster in multi-center framework. This motivates the need for a robust EM algorithm against outliers in distributed setting which aimed to be robust to local models send by outlier/adversarial remote devices. Robust statistical learning and its sub-field robust machine learning have been investigated throughout many years, such as classical MOM, Trimmed-Mean of K-medians, and some others via kernel methods. Those methods are not directly applicable to our distributed settings here as our loss functions are more specific to clustering. In this work, we propose a extended multi-center Federated Learning framework, secure against corrupted or adversary outliers among normal nodes. In particular, we adopt a bootstrap sampling method and a robust approach based on a median-of-means estimator. The reason for our proposed idea is the use of these techniques and approaches inspired by the recent development of robust machine learning and the use of median-of-means from robust statistics. Moreover, the EM algorithm with such initialization strategy and replaced enough blocks and iterative estimate the mean of random variables by median-of-means effectively avoid the disappearance of clusters in some blocks. Close work has been done in clustering applications also shows acceptable results.

1.2 Structure of the Thesis

This thesis we cover a very specific subset of this current research problems related to Federated Learning. This report we first introduces the motivation of multi-center federated learning framework which trains multiple global model in multiple communication rounds. We concede that this framework has a few pitfalls and limitations, however, note that framework is fully extensible. So we then study a few techniques and regularization based off this framework. We also organize this report to follow this order. I will also analyze the importance of robustness and security in modem massively distributed system, in our case, applied to Federated Learning. Therefore, we will introduce two extensions to our novel solution -

Chapter 2 conducts a general review on federated learning in the context of state-of-art methods that focus on the non-i.i.d. data. First, we will briefly introduces other branches of distributed machine learning techniques which should give audience a better picture of the development of FL. In addition, we will highlight some of state-of-art clustered FL studies, what these clustering methods are and compare them with ours in the context of theory, non-i.i.d. setting or applications domains. Then, we introduces techniques for a more robust, central clustering algorithm that replaced mean-based EM, as well as introduces bootstrap sampling and median-of-means estimator, which are techniques used to get a more robust model aggregation for each communication round. The dataset as well as the experiment is hosted in a public repository for promote of reproducible research.

Chapter 3 introduces a clustered federated learning framework - multi center federated learning. First, we introduces a novel distance function - federated loss and report a few important baselines we will use to show the usefulness, then we mathematically prove the problem. The training algorithm is described in detail. We also introduce how we solve the first research problem, non-IID or data heterogeneity, effectively by the use of our framework. Then this work conducts extensive experiments on three benchmark dataset and describe convergence, model accuracy results after applying multi center federated learning framework.

Chapter 4 introduces how to make Federated Learning more robust and secure by the use of a K-means clustering variant. First, this thesis introduces a new emerged clustering technique called K-bMOM. Then I'll discuss how K-bmom works and how it enhance the breakdown point of data points. The objective function of this variant stay the same with original framework, in addition, the training algorithm of initial

stage and update stage is described in detail. We aslo introduce how we solve the second research problem by the use of MOM to mitigate the decrease of performance caused by outliers. This work conduct a portion of data pollution attack over three dataset, FEMNIST, CELEBA and Synthetic, to report the result of our proposed FL framework after applying K-bmom technique, as well as compare them with the result before applying K-bmom technique. Those datasets have about 9000 users will participate FL training and the result will be analyzed. The dataset as well as the experiment is hosted in a public repository for promote of reproducible research.

Chapter 5 explains the importance of defence against model poisoning attacks in FL settings. This chapter we introduce our solution to the third research problem, i.e. model poisoning attacks while keeping satisfying performance. To tackle this problem that data of each client is usually not independent nor identically distributed (non-IID), personalized FL (PFL) or clustered FL which can be seen as a cluster-wise PFL is to learn multiple models across clients or clusters. To detect anomalous clients (outliers), the Local outlier factor (LOF) is a popular method based on the density of data points. Thus, a nested bi-level optimization objective is constructed, and an algorithm of personalized FL with robust clustering is proposed to detect outliers and keep the state-of-the-art performance.

Finally, Chapter 6 summarized the all three published work and achievements and concludes the thesis with some discussion on future direction.

1.3 Claims of the Thesis

The important original contributions of this thesis are:

- Proposes a simple yet novel multi-center aggregation approach (Section 3) to address the non-IID challenge of Federated Learning.
- Design an objective function, namely multi-center federated loss (Section 3.3.2) for collaboratively training in Federated Learning.
- Proposed a named Federated Stochastic Expectation Maximization (FeSEM) (Section 3.3.3) to solve the optimization of the aforementioned objective function.
- Present the algorithm as an easy-to-implement and strong baseline for FL.

- Empirical comparison with other FL learning approaches and discuss the impact on benchmark datasets. (Section 3.5)
- Code is freely available, open sourced, audience can use it for reproduce the described experiments
- Extensions of the base multi-center learning framework
  - Designs updated objective function in FedPRC (Section 5.3) for multi-center personalized aggregation model
  - Adds a regularization to balance bteween priority of Federated Loss or similiarity among local models
  - Introduces a more robust version of clustering algorithm to extend previous multi-center Federated Learning approach
  - Robust Adaption of original framework by replacing bootstrap sampling, a median-of-means estimator
  - Proves aforementioned sampling technique and estimator enable faster convergence and robust to outliers and malicious users
  - Replaced FeSEM algorithm and extended the framework to address model poisoning issue
  - Present the algorithm as an easy-to-implement variant to FedSGD and FedAVG while maitaining strong performance
- We propose a novel PFL with robust clustering (FedPRC) algorithm to solve the complex optimization problem, and the algorithm can resist Byzantine workers.
- We formulate the PFL problem with robust clustering into a nested bi-level optimization framework. .
- both original framework and extensions effectiveness is evaluated on benchmark datasets. (Section 4 and Section 5)

CHAPTER 2

RELATED WORK

2.1 Federated Learning Preliminaries

This section will first describe some general concepts of FL, then compared FL with other ML-based deployment architectures then will discuss other existing studies of FL prior to our work, i.e. studies that focus on the non-IID challenges of FL, then followed by a short introduction to non-parametric Bayesian modelling and further reading. However, this thesis has no intention to become a comprehensive taxonomies covering various challenging aspects, contributions, and trends in the literature of FL. Furthermore, the focus of this section to discuss core challenges and open research directions towards robust clustered FL algorithm with non-IID data, I'll review and compare in-depth 9 of related work in this topic.

To understand how FL develop to its current state, we need to review the basic idea of optimization in machine learning. The reason for doing such is that our proposed approach and FL have some close connections to other domains of supervised machine learning, such as multi-task learning, meta-learning, transfer-learning, Bayesian, ensemble learning.

All these learning applications have the same underlying optimization theory, the stochastic gradient optimization (or the variants). However, SG optimization may need a minor modification to better suit a particular learning context. Relational diagram fig 2.1 shows Federated Learning and its structure to others. This introductory section briefly outlines the theoretical framework that gives rise to empirical risk minimization

problems from our approaches and clarifies non-convex functions that we are trying to optimize in the remainder of this thesis.

The optimization methods and its variants have several essential features for machine learning, such as their fast convergence rates and ability to exploit parallelism. However, a recent study shows that in DNNs methods that incorporate derivative information lead to some issues because most DNNs have some properties which make SG not the optimal methods for optimization. These properties are all represented in our system, namely highly nonlinear and non-convex. Admittedly, SG and its variants are still able to converge in the context of large-scale machine learning. Alternative classes of approaches should be investigated. We denote model parameters as w

Prediction and Loss functions:

$$\mathscr{H} := h(\cdot; w) \ : \ w \in \mathbb{R}^d \tag{2.1}$$

An empirical risk minization problems means we wish to minminize the expected risk:

$$R(w) = \int_{\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}} \ell(h(x;w),y) dP(x,y) = \mathbb{E}[\ell(h(x;w),y)].$$

The empirical risk we wish to minimize is:

$$R_n(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i;w),y_i)$$

Optimization methods for machine learning fall into two broad categories. We refer to them as stochastic and batch. The prototypical stochastic optimization method is the stochastic gradient method (SG) [], which, in the context of minimizing R_n and with $w_1 \in \mathbb{R}^d$ given, is defined by.

$$w_{k+1} \leftarrow w_k - \alpha_k \nabla f_{i_k}(w_k)$$

Here, for all $k \in \mathbb{N} := \{1,2,...\}$, the index i_k (corresponding to the seed $\zeta_{[i_k]}$, i.e., the sample pair (x_{i_k}, y_{i_k})) is chosen randomly from $1, \ldots, n$ and α_k is a positive stepsize. Each iteration of this method is thus very cheap, involving only the computation of the gradient $\nabla f_{i_k}(w_k)$ corresponding to one sample.

The distributed machine learning community appears to be very successful in the past decade [21, 22, 34, 61]. Numerous studies are relevant to linear and convex models via parallelization and distribution, where distributed gradient computation is the natural first step. Sequentially, in the context of deep learning region, suggestions for scaling

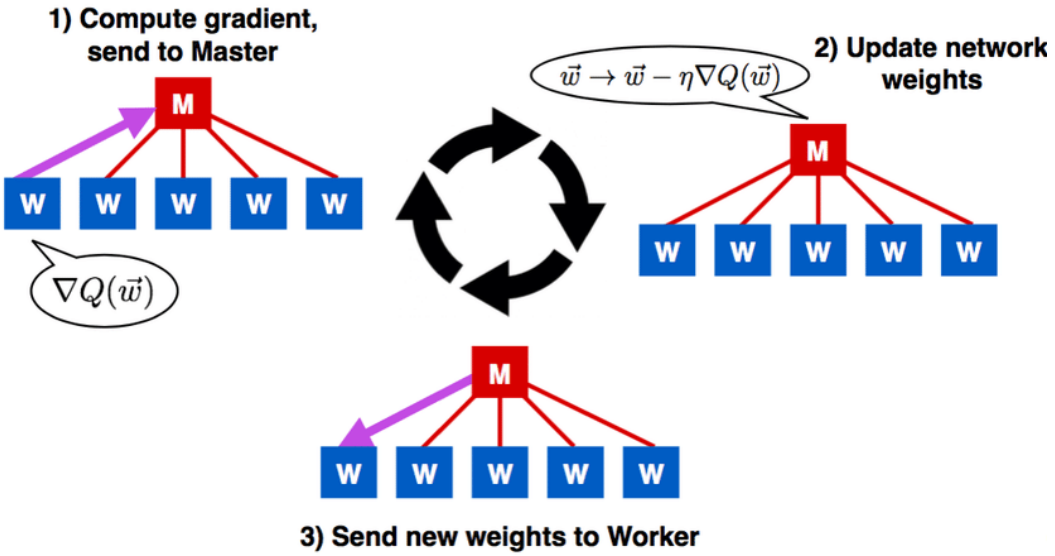


Figure 2.1: Downpour-SGD Algorithm [34]

up deep learning include the use of a farm of GPUs to train a collection of many small models and subsequently averaging their predictions [27]. This approach can be treated as the foundation of FL notion. Only FL are more widely applicable to the problem either convex or sparse.

2.2 Federated Learning Challenges

Federated learning (FL) enables data scientists and ML engineers to utilise rich user-generated data from mobile devices without sacrificing user privacy. This new technology has quickly attracted numerous research interest since 2016, and a recent survey shows a total of 7546 papers are published in the duration of 2016 to 2020 [93]. In addition, many studies investigate FL from several aspects, e.g., system perspective, personalised models, scalability, communication efficiency, and privacy. Most related work addresses a particular concern such as security or privacy. It has been applied to various industry applications, such as banking, smart healthcare, and mobile internet applications. Many survey of the applications for Federated Learning have done [1, 93].

ML community in general is received much attention as we are becoming more fascinated by AI decision making. A number of centralised server deployed DL applications are ranging from as simple as Netflix is following in Google and Facebook footsteps to

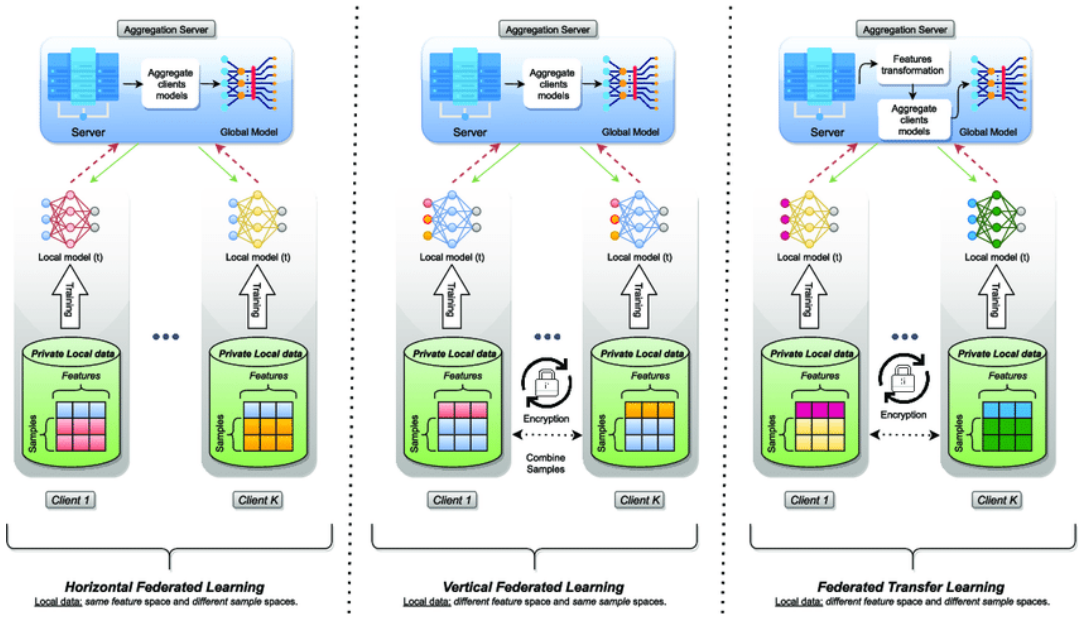


Figure 2.2: Horizontal Federated Learning, Vertical Federated Learning, Federated Transfer Learning [42]

improve its services, to as sophisticated as self-driving cars [66], smart healthcare [88], customer service and employee retention [6], advertising [64], and many more coming to realise soon. The key factor behind the success of DL-based applications is a huge volume of data generated by end-users on mobile devices, flexible software and hardware architecture and comprehensive accessibility. In typical cases, the paradigm of DL is to continuously stream generated data into the cloud, where it is analysed, more features are extracted, and we can train better models on high-performance servers. Often those server are deployed in the cloud. As soon as user interactions occured with available ML services in the cloud, more training data are gathered, and more intelligent ML-based applications are therefore produced. However, the privacy of available data used for training and for the astounding success of DL is becoming a rising concern for the users.

FL can be categorised into vertical FL and horizontal FL. Moreover, FL has many distinct challenges compared to traditional centralised server-client networks. According to (Li, 2020) [69], four major challenges of FL are categorised. No consensus exist between what is the most important challenge for FL, but it's commonly believed these four challenges is crucial to FL. We are going to discuss four major FL challenges here briefly.

Expensive Communication. Communication is a significant challenge in federated networks [65]. It has commonly occurred that wireless and end-use devices operate on lower bandwidth than interlink between data centers and can be expensive and unstable. This has led to significant recent interest in solutions to communication cost reduction of federated learning. Two possible methods which address communication efficiency are: (i) reducing the total number of communication rounds, or (ii) reducing the size of the message sent between devices and the central server.

Systems Heterogeneity. The variance of devices in federated networks can be huge, as well as the geographic location of the device, the computing capacity, the storage, and the network bandwidth used (3G, 4G, 5G, 6G, optics) of each user may differ to a large degree. In addition, each device that decides to contribute to the federated network, sometimes their update may be polluted or received not in the current round due to power issues, or network connectivity. Each device may also be unreliable and not to mention in the real world applied learning in a system that consists of tens of millions devices that makes a significant challenges to FL system.

non-IID Data. In machine learning, the non-IID data problem is a challenge that can arise when training a model on data that is not independent and identically distributed. This can happen when the data is collected from different sources, or when it is partitioned in a non-random way. The assumption of IID, which stands for independent and identical distribution, is a must for centralised machine learning. In contrast, this basic assumption across client nodes does not hold for federated learning setups. The insight to FL is such: under these setups, the performances of the training process may vary significantly according to the degree of unbalanced local data samples, the particular probability distribution of the training examples (i.e., features and labels) which stored at the local nodes. This topic is particularly relevant to this thesis.

Privacy Concerns. Existing privacy preserving techniques can still put user data in risk. A number of studies have focusd on the privacy concerns of Federated Learning in the past. For example, Google proposed [15] an secure aggregation method. Ref [39, 105] proposed algorithms for secure multi-party decision tree for vertically partitioned data. Vaidya and Clifton proposed secure association mining rules [103], secure k-means [104]. Though recent methods and models aim to address privacy concerns of FL, which some tools used including secure multiparty computation or differential privacy, these

approaches often provide privacy at the cost of reduced model performance or system efficiency. This topic is not particularly relevant to this thesis and audience who like further information can see Ref [14, 55].

Apart from aforementioned four things, the motivation of FL is also relevant to a number of broader research areas. Edge computing, meta-learning and neural architecture search to name a few,. It's exacting to see so FL full of potential. The focus of this is thesis is only non-IID data due to time constraint. If time allow, final section outline several directions of future work that are relevant to a wide range of research communities and practitioners.

2.3 Federated Learning with Non-IID Data

This section discuss FL unique characteristics and challenges, and reviews some effective approaches of FL with non-IID data. This section provides their propose definitions, categorizations, experimental results and some comparison among them as non-IID Data is the most relevant challenge to this thesis.

It is noted in some research work based on assumption that the data in the k-th device in distributed environment are i.i.d. sampled from the distribution $\mathscr{D}$. Then the overall distribution is a mixture of all local data distributions: $\mathscr{D} = \sum_{k=1}^{N} p_k \mathscr{D}$. Under such assumption, FedAvg is almost euqal to local SGD [73], and the latter assumes the data are IID (independent, identically distributed) generated by an unknown function and then sliced among the N devices, that is $\mathscr{D} = \mathscr{D}$, for all $k \in [N]$.

Traditionally, the data distribution over different devices in decentralised setting is IID, which is a natural assumption of real-world applications. However, early research work (McMahan, 2017) proposed [83] only one global model as a single-center to aggregate the information of all users. The stochastic gradient descent (SGD) for single-center aggregation is designed for IID data, and therefore, conflicts with the non-IID setting in FL. Federated clustering approaches can be divided into two types: model clustering and data clustering.

Title	Main Contribution	Trained model	Aggregation	Dataset

Agnostic federated learning	domain agnostic	L-REG CNN LSTM	FedAvg	Adult, Fasion MNIST, PTB
An efficient framework for clustered federated learning	cluster id and model	L-REG CNN	IFCA	Synethic, Rotated MNIST and CIFAR, FEMNIST
Client adaptation improves federated learning with simulated non-iid clients	adaptation through conditional gated	CNN	CGAU	Freesound, CIFAR-10
Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints	Iterative bi-partitioning	CNN	FedAvg	MNIST, CIFAR-10
Federated learning with hierarchical clustering of local updates to improve training on non-IID data	FL + HC	CNN	FedAvg	FEMNIST

Federated learning with matched averaging	construct global model from matching hidden elements	CNN MLP LSTM	Matched Averaging	CIFAR-10, Shakespear
Adaptive Personalized Federated Learning	generalization bound of mixture of local and global models	CNN	APFL	MNIST, CIFAR-10, FEMNIST, Synthetic
Fedgroup: Ternary cosine similarity-based clustered federated learning framework toward high accuracy in heterogeneous data	novel data-driven measure named EDC	MLP MCLR LSTM	FedGroup	FEMNIST, MNIST, Synethic, Sent140
Heterogeneous Federated Learning	explicit feature information alignment	CNN	Aligned Model Averging	FEMNIST, CIFAR
On the byzantine robustness of clustered federated learning	CFL to byzantine settings	CNN	FedAvg	MNIST Fashion CIFAR10
On the convergence of fedavg on non-iid data	Convergence analysis of FedAVG	CNN	FedAvg	MNIST

Personalized federated learning with moreau envelopes	Moreau envelopes regularization	CNN	pFedMe	MNIST Synthetic
Personalized federated learning: a meta-learning approach	multi-task, learned an initial	CNN	Per-FedAvg	MNIST CIFAR-10
Robust and communication-efficient federated learning from non-iid data	A compression, communication efficient FL framework	CNN LSTM L-REG		CIFAR-10 KSW MNIST
robust federated learning in a heterogeneous environment	robust heterogeneous Federated optimization	REG	FedAvg	Synthetic
Tackling the objective inconsistency problem in heterogeneous federated optimization	solution to slowdown due to objective inconsistency	CNN	Normalized Averaging	CIFAR-10 Synthetic

Above Table 2.1 describes a general overview of some recent developed methods to the non-IID problem in FL.

We review model clustering type. Clustered Federated Learning (CFL) [90] and Robust FL in heterogeneous network [47], which claims a novel framework involves three-steps module process, both are model clustering methods for FL. CFL is a method that extends existing FedAvg with iterative clustering. Key idea of CFL is cluster the weight of each client model by the use of gradient $\nabla_\theta r_i(\theta^*)$ cosine similarity. Simply speak-

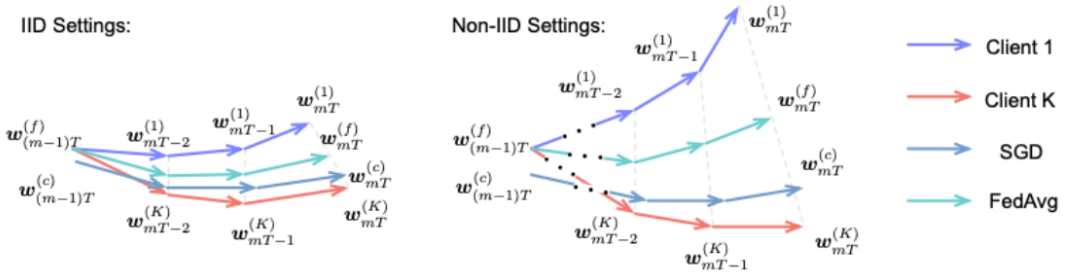


Figure 3: Illustration of the weight divergence for federated learning with IID and non-IID data.

Figure 2.3: Non-IID data learning in Decentralized ML [56]

ing, different clusters' gradient direction should have significant divergences. Moreover, to compute the cos similarity, it use the local empirical risk value calculated between two models, and then carries on FedAvg training and clustering again in each divided cluster unit. These steps are repeated until all models no longer meet the clustering conditions. This method can be viewed as a modification version of FedAvg, which requires more computational and communication resources to the centralised server and verification is stronger than FedAvg. Moreover, the greater the degree of data heterogeneity is, the more generated clusters. This brings one unique benefits compared to model-based clustering method, no need to specify the number of clusters in advance. Therefore, this method is not suitable for the case of limited resources and complex data distribution. Analogous to CFL, the second method, Robust FL in heterogeneous network, also clusters models in FL based on the empirical risk function value, but the difference is that these clustering models are not based on FedAvg and only need to complete local independent training. Moreover, this method incorporate three different modular steps, each of those clients has not been fully theoretically work well as a system and thus not suitable to sensitive applications.

Next, we review next two recent effective clustering FL based solutions, three approaches of personalized model [81], which present a systematic learning-theoretic study of personalization, and IFCA. (Ghosh, 2020) [46] propose a similar algorithm named IFCA, for which a convergence bound is established under the assumption of good initialization and all clients having the same amount of data. Moreover, IFCA holds a number of K global parameters in the central server. To begin with, workers send their loss to centralised server, center machine estimates the cluster identities of each worker machine by running k-means on the collection of workers local models. Then With the cluster identity estimations, the center machine runs any federated learning algorithm

such as FedAVG or Second order optimization. These steps repeated until convergence condition is met. Both IFCA and our approach use EM-based algorithm to serve the central clustering procedure, they also discussed that convergence analyses in the finite sample setting, both EM and alternating minimization are known to be hard. Analogous to IFCA, the method of three approaches of personalized model each client is greedily assigned to the cluster whose model yields the lowest loss on its local data. There are two more components in the method, fine-tuned, Dapper and Mapper.

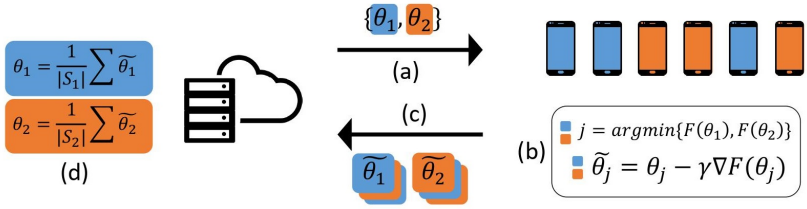


Figure 2.4: Overview of IFCA architecture [46]

Recently, (Marfoq, 2021) [82] propose a multi-task learning framework. Analogously, this framework common to soft clustered FL that allows individual stored data to follow a mixture of distributions. This framework has recently emerged as an efficient way to learn mixture of models in the federated setting and exhibits tolerance for more implicit relations across learned models. In summary, the key of their method is the EM algorithm is used and the mixing coefficients are calculated based on the training losses, while it is similar to FeSEM, they named this algorithm FedEM. However, FedEM requires a parameters local update from each worker in each round in each cluster, which entails significantly more for computing and storage requirements than conventional FedAvg. However, our FeSEM each round only collect model updates from a small set of workers. For generic data distributions and loss functions. Typical algorithms include FedAMP [58], which adds an attention-inducing function to the local objective, and pFedMe [38], which formulates the regularization as Moreau envelopes. A highly influential work of multi-task learning in FL is [97]. The main contribution based on general MTL (multitask learning) but they also discuss distributed MTL, though do not adequately address the systems challenges associated with Federated Learning. However, there is no discussion the relation with Federated Learning with non iid data [119]. Their contributions is about naturally fitting separate yet related models simultaneously. Moreover, a novel method called MOCHA is proposed, which aims handle systems challenges that arise in federated learning, including high communication cost,

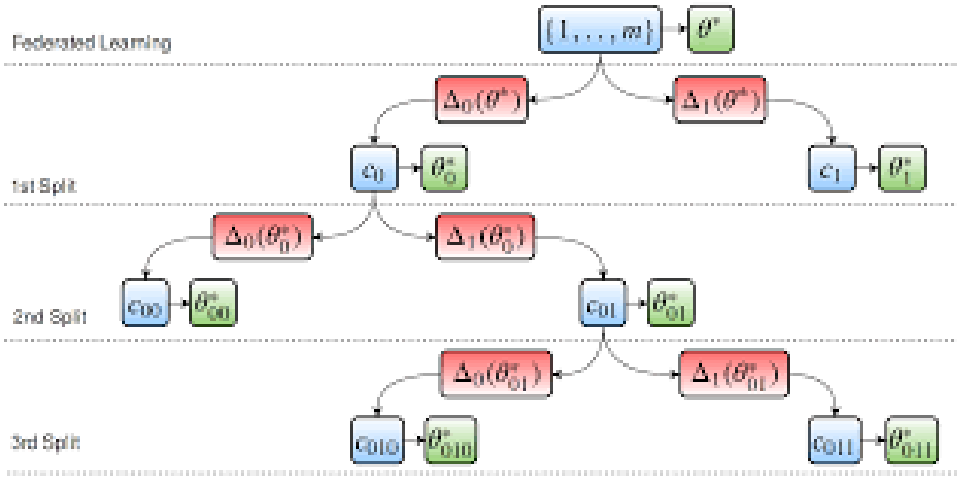


Figure 2.5: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints [90]

stragglers, and fault tolerance in this paper. They identify two types of MTL framework and only choose one of them which is structure of the related task is unknown before training, but leave the other one which structure of task is unknown. Their justification is not good enough by say in reality relationship of task is not always known.

2.4 Clustering Methods for Federated Learning

This section discuss the benefits of using clustering method, K-means in particular, in mutli-center federated learning framework. We review and discuss current research interest of K-means in the setting of centralized or decentralized architecture.

Before we go into details of our multi-center FL method, let's to first introduce the general concept of clustering. Clustering is a method in order to divide a set of data into subsets, called clusters, in a way that data assigned to the same cluster are similar in some sense. How to handle data without label and under minimal assumptions makes clustering a challenging task. Yet, it is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including pattern recognition, image analysis, DNA microarray analysis, information retrieval, bio-informatics and machine learning. Moreover, clustering can save massive amount of time in prepossessing for supervised applications. Multiple intrepretations across disciplines of defining a cluster, have led to an abundance of application-specific algorithms, such as distance-based, hierarchical, squared error-Based and so on [110].

Among the algorithms which takes data in a vector format, K-means and Gaussian

mixture model (GMM) based clustering are two popular schemes. The K-means algorithm relies on a measure of the variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares to partition data into clusters. This algorithm requires the number of clusters to be specified. The k-means algorithm divides a set of N samples X into K disjoint cluster C, each described by the mean μ_j of the samples in the cluster. The means are commonly called the cluster "centroids"; note that they are not, in general, points from X, although they live in the same space. K-means algorithm aims to choose centroids that minimize the inertia, which can be viewed as total Eulcidean distance to the centroids in the data samples. In addition, soft K-means is well suited for overlapping clusters by allowing each datum to belong to multiple clusters. GMM-based clustering considers data drawn from a probability density function (pdf), where each pdf is also a function to be estimated given the samples and assignments of the samples to their corresponding pdf. This GMM-based clustering then can be viewed as finding a estimate of maximum likelihood (ML) for the GMM parameters. Typically, the estimated can be obtained by the use of expectation-maximization algorithms. Kernel methods have been developed for the non-linearly separable situation for clustering. If interested, further reading of EM, K-means, GMM are included.

In Chapter 3, a novel method is proposed defining the problem of learning of multiple optimal models from distributed manner across network via clustering framework, therefore, our goal is to identify these clusters which leads to the use of stochastic of EM algoirthms, which is a extension to EM. The reason of selecting EM is that it has been a well established technique for mixture model based clustering. We will look into more details of EM in Chapter 3

Challenges arise when training federated models from data that is not identically distributed across devices, namely,statistical heterogeneity. HADDADPOUR and MAHDAVI [51] conducted theoretical convergence analysis for FL with heterogeneous data. Hsu et al. [57] measured the effects of non-IID data for federated visual classification. Yang et al. [111] proposed a heterogeneity-aware platform design for FL. Liang et al. [74] discussed the local representations that enable data to be processed on new devices in different ways according to their source modalities instead of using a single global model. A number of author modeled data heterogeneity and statistical heterogeneity via methods such as multi-task learning [3, 30, 97] or meta-learning [24, 41, 60, 63], variants of personalised federated learning [38, 38, 58]. Meta-learning approaches [92] show a similarity with our approaches, but the final outcome is somewhat different, i.e., our approaches are finding a optimal balance between single global model with high

accuracy over all clients and finding personalised model for every device, which the number of shared models from latter method is massively larger than the respective ones from our approaches.

Li et al. [71] proposed FedDANE by adapting the DANE [94] to a federated setting. In particular, FedDANE is a federated Newton-type optimization method. Li et al. [70] proposed FedProx for the generalization and re-parameterization of FedAvg [83]. It adds a proximal term to the objective function of each device's supervised learning task, and the proximal term is to measure the parameter-based distance between the server and the local model.

Arivazhagan et al. [4] added a personalized layer for each local model, i.e., FedPer, to tackle heterogeneous data. Similarly to us, Sattler et al. [90], Mansour et al. [81], Briggs et al. [17] and Yurochkin et al. [117] overcome challenge of heterogeneous data in FL via clustering methods, but at a different level to us.

2.5 Robust Methods for Federated Learning

In this section, we adopted most similar theoretical background of robust estimation where study of a mixture of K cluster from distrubtion of non-IID sample when a fraction of data is adversarially corrupted. Large body of work in clustering has been studied [8, 31, 45].

This proposed method is designed to be an enhanced version of FeSEM [108], which is aiming to modify the server clustering process. The basic of FL is that many nodes in a network collaboratively train a classifier on their own local dataset. The dataset kept on a local device is only a shard of a much larger dataset. Specifically, the problem of FL can be usually denote as minimizing this formula 1.1 is the local objective function which describes how good are the trained classifier, and the local objective function is different for different classifiers (e.g., logistic regression, neural network). In particular, a master node needs to keep a global model whose parameters w normally is the weighted average of model parameters w_i of all worker nodes. During typical FedAvG learning process, three steps will be performed at each iteration:

- step i: a master node sends the global model parameter to all worker nodes
- step ii: the worker nodes evaluate the gradient of $\nabla F(w)$ with respect to the global model parameter using local objective function and own training data then sends

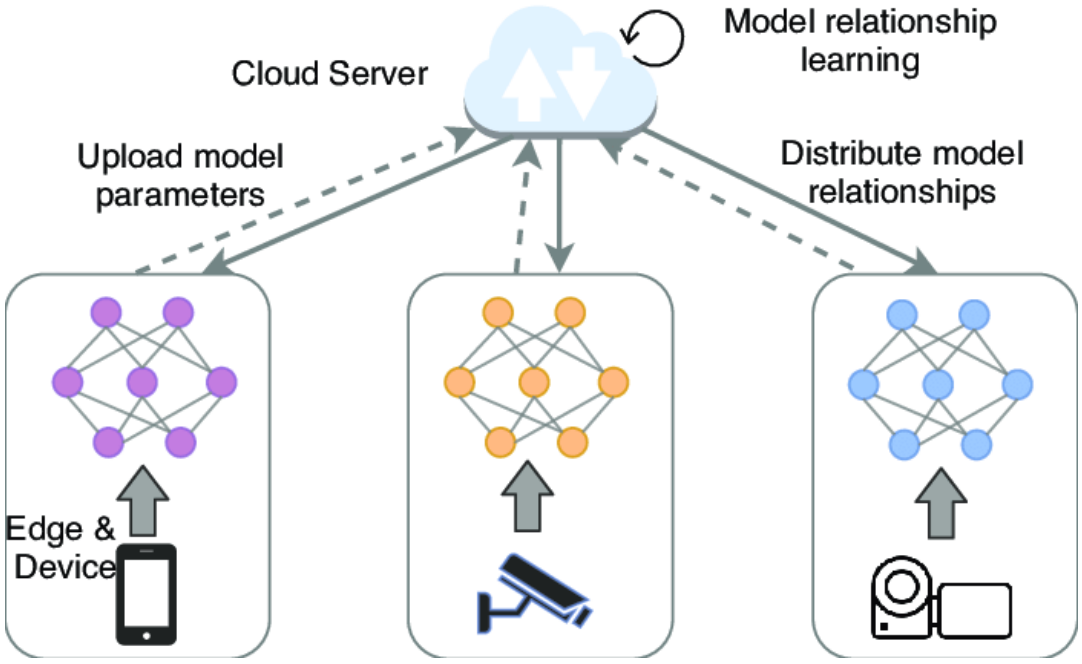


Figure 2.6: A multi-task learning approach included for Federated Learning [97]

back the update

- step iii: a master node aggregates all local updates to obtain a new global model using a certain aggregation rule. In the case of Federated Averaging, the rule is the weighted average. Formally noted as : $w = \sum_{k=1}^{K} \frac{n_k}{n} w_k$, where n_k is the number of samples kept in kth node and n is the total number of samples on all nodes.

In addition, the idea of FeSEM is to train multiple global models w^l instead of one. The principle is to estimate a worker node cluster identity via finding a center with minimal distance. Here the distance-based clustering method is used, and the metric is often squared Euclidean distance. Throughout the learning process of FeSEM, the worker nodes will have a cluster identity and run local updates within the same cluster. Then each cluster obtains a new global model using a weighted average that is identical to Federated Averaging. FeSEM can be viewed as one of the important techniques to learn an unknown mixture from samples. Though, one drawback of existing EM is poorly performed with dimensions. Given that the local model size can have millions of parameters in practice, the need to have the robustness of high dimensional SEM is naturally obvious. This work focus on the property of robustness to a small number of outliers and improved convergence. In the future, our goal is create secure and

resilient FeSEM algorithm which against malicious clients or adversarial threatss such as Byzantine nodes. Next, this thesis will introduce another type of robust clustering.

FL is designed for specific scenarios that can be further expanded to a standard framework to preserve data privacy in large-scale machine learning systems or mobile edge networks [75] [78] [77]. For example, [114] expanded FL by introducing a comprehensive, secure FL framework that includes horizontal FL, vertical FL, and federated transfer learning. [62] discussed the advances and open problems in FL. [19] proposed LEAF, a benchmark for federated settings with multiple datasets. [79] proposed an object detection-based dataset for FL.

Traditionally, the data distribution over different workers in decentralised setting is non-IID, which is a natural assumption of real-world applications. However, early FL approaches [83] use only one global model as a single-center to aggregate the information of all users. The stochastic gradient descent (SGD) for single-center aggregation is designed for IID data, and therefore, conflicts with the non-IID setting in FL. Some research work is done which are popular approaches to this problem, clustering, multi-task learning, local adaption, ensemble learning. Federated clustering approaches can be divided into two types: model clustering and data clustering.

CFL [90] and Robust FL in heterogeneous network [47], which claims a novel framework involves three-steps module process, both are identified as FL model clustering methods. CFL is a method that extends existing FedAvg with iterative clustering. Analogous to CFL, Robust FL in heterogeneous network, also identified as FL clusters models, which performs clustering on local empirical risk minimizers, but the difference is that these clustering models are not based on FedAvg and only need to complete local independent training. Moreover, this method incorporate three different modular steps, each of those has not been fully theoretically work well as a system and thus not suitable to sensitive applications. Research [46] proposes a similar algorithm named IFCA, for which a convergence bound is established under the assumption of good initialization and all clients having the same amount of data. The work [80] proposes a unified bi-level optimization framework for CFL and prove the convergence. Typical algorithms include FedAMP [58], which adds an attention-inducing function to the local objective, and pFedMe [38], which formulates the regularization as Moreau envelopes. A highly influential work of multi-task learning in FL is [97]. Multi-task learning has recently emerged as an alternative approach to learn personalized models in the federated setting and allows for more nuanced relations among clients' models. A number of other robust study in the field distributed or Federated Learning [67, 115], but they do not have a

clustering structure of the nodes. Some other techniques including prototype [99] and graph [23] also be applied into FL to improve its privacy or performance.

In addition, the idea of Clustered FL is to train multiple global statistical models w^k instead of one. The principle is to estimate a worker node cluster identity via finding a center with minimal distance. Here the distance-based clustering method is used, and the metric is often squared Euclidean distance. Throughout the learning process, the worker nodes will have a cluster identity and run local updates within the same cluster. Then each cluster obtains a new global model using a weighted average that is identical to Federated Averaging. FeSEM [108] can be viewed as one of the important techniques to learn an unknown mixture from samples. Though, one drawback of existing EM is poorly performed with dimensions. Given that the local model size can have millions of parameters in practice, the need to have the robustness of high dimensional SEM is naturally obvious. This work focus on the property of robustness to a small number of outliers and improved convergence. In the future, we will robustify the SEM against resistance to malicious data or model attacks such as adversarial federated nodes.

PFL PFL is the most popular technique to address non-IID challenge in FL, as vanilla FL [83] delivers only one globally shared model which cannot fit all clients' data. Based on granularity, PFL can be categorized into cluster-wise PFL and client-wise PFL. For the cluster-wise PFL, also called clustered FL, clients are grouped in to several clusters, and then identical number of models are trained based on these clusters. There are mainly two variants in cluster-wise PFL methods, representation of a client and the clustering method. The work [108] use model parameters to represent clients and K-means to do clustering. CFL [90] use hierarchy clustering to divide clients into two clusters based on the cosine similarity of gradients iteratively. The loss of models is also used to cluster clients by HypCluster [81] and IFCA [46]. The unified formulation and convergence of cluster-wise PFL is studied by [80].

For the client-wise PFL, each client has its personalized model, either in model structure or model parameters, even in the loss function. A simple but effective method is to fine-tune the trained global model [26, 41]. Ditto [68] proposed a bi-level optimization framework using a penalty term to constrain the distance between the local model and global model. FedRep [29] divides the network into the backbone and the head, and learns shared parameters for the backbone and unique parameters for the head. FedProto [98] adopts prototypes instead of gradients to communicate and is more privacy-protective

and communication-effective. Researches by [24, 95] aim to train a global hyper-network or meta-learner instead of a global model before sending it to clients for local optimization. Meta learning and multi-task learning are also applied into PFL including [41, 97].

Model Poisoning Attacks and detection The way malicious agent generates an arbitrary update vector by merely shuffle data labels sounds very similar to the standard dirty-label poisoning in [25]. However, in Federated Learning setting, the possibility of a an adversary controlling a small number of malicious agents, perform a model poisoning attack to manipulate the learning process so that the jointly trained global model which turn into misclassifiation over some data is much higher. FL is apparently vulnerable to model poisoning attacks due to its decentralized nature. A line of work has been done already [7] [10] [32]. In contrast to previous work, this work focus to detect these malicious agents during central clustering phase by applying density method then reduce the impact of those agents' updates to the aggregation of the cluster center.

Anomaly detection can be described the problem of finding patterns in data that do not confirm to expected behavior. Of these, anomalies and outliers are two terms used most commonly in the context of anomaly detection. Clustering can be used as a technique for training of the normality model, where similar data points are grouped together into clusters using a distance function, for example [84]. While LOF [16] is a widely-used density-based anomaly detection method. However, in the case of our method, we already know malicious agents are the anomalies that we tried to identify. The outcome after preclude those identified outliers would be benign agents, then only the benign agents weight matrix feed into our clustering algorithm. The identifying outliers stage has no inherit relation to next clustering phase.

In this section, we will introduce our adversarial settings. There are two common threats models in Federated Learning; the first case is data is mislabeled or maliciously injected some wrong data, which is called noise labelling and has been addressed by some work including [107]; the other threat is the attackers aim to manipulate the learning process such that the learned model has a higher testing error rate. Normally, attackers can only inject data into training datasets with the aim to make data poisoning attacks when they have full control of those worker nodes, while the learning process is often deemed as protected. This work will focus on model poisoning attacks and is based on the assumption that attackers have knowledge about the aggregation rule. The most basic aggregation rule uses mean estimator and weighted mean in Federated Learning.

Several model poisoning attack and their variants emerged in recent literature [7]. In

this work, we do not claim bootstrap sampling nor robust clustering is a comprehensive defense measure against heavy model poisoning attacks. Hence, we select the idea of Krum [10] which is simply boosting each iteration of the learned model in some worker nodes to manipulate the learning process such that the learned model achieved label misclassifications. The way of explicit boosting works is to mimic the benign worker nodes during the learning process; the node tries to perform the same number of epochs on the local dataset via the same training objectives to obtain an initial gradient update. Since the malicious node wants to ensure the outcome deviates from the true label, it will have to overcome the scaling effect of gradient updates collected from other nodes. In other words, the final gradient updates the malicious nodes send back are then scaled a factor λ by which the malicious nodes boost the initial update. This attack has proved to negate the combined effect of normal worker nodes.

Other Robust Clustering The robust clustering algorithm is a principal approach to enhance the robustness against the presence of outliers [44]. A significant number of work has been reported in this context, such as [36], [87]. Typical robust clustering methods include mixture modeling [112], trimming approach [43]. A number of works in robust clustering have been studied by [112] [43] [33] [2] [113] [50] from recent literature. Our proposed solution is based on FeSEM, which proposed a clustered federated learning method to cluster worker nodes using stochastic expectation-maximization methods (SEM). Recent works of using bootstrap of classical MOM with K-means are emerging [18]. The median-of-means (MOM) estimator of the mean in dimension one consists in taking the median of some arithmetic means derived from a collection of samples, as in our case, derived from a collection of local model parameters. The bootstrap of median-of-means is thus a collection block b_1^B, which are generated through a random process, that sample drawn from a collection without replacements (disjoint blocks) and according to the uniform distribution on the remaining data at each step. It is apparently that bootstrap MOM is a randomized estimator. Though, for any static sample data size n, one can decide what is the right block size n_B and the number of blocks B to define a bootstrap MOM estimator. To the classical MOM, in contrast, where the product of the block size with the number of blocks should be equal to the data size defined previously.

The objective of clustering is to group similar objects together, and dissimilar objects into different clusters. And robust clustering is to enhance the robustness of clustering results against outliers [44]. Many works have been done in this area including [36], [87]. Vanilla robust clustering methods include mixture modeling [112], trimming approach

[43]. Recently a number of works in robust clustering have been studied by [112] [43] [33] [2] [113] [50]. The work [18] researches K-means with bootstrap of median-of-means (MOM). The MOM estimator can mitigate the influence of outliers, which estimator of mean is not good at addressing outliers. The bootstrap of MOM (bMOM) enhance the robustness against outliers thus can achieve better breakdown point, which is a measure to quantify the toleration of outliers.

Chinese Restaurant Process Next we will further examine Bayesian methods, Chinese Restaurant Process in particular. Non-parametric Bayesian clustering methods such as the infinite Gaussian mixture model [89] or Dirichlet process [100] are a class of algorithms that can be used to cluster data points without making any assumptions about the underlying distribution of the data. These tools are particularly well-suited to our theme, which is clustering clients in Federated Learning, as Federated Learning data cannot be easily summarized by parametric model, or the number of clusters from data is unknown before hand. However, they may then lead two limitations. The first cause from the requirement to generate each observation from a well-defined distribution. For example, in latent Dirichlet allocation [12, 13, 48], each word of a text document is sampled from a multi-nominal distribution of a corresponding topic. If we want to incorporate features such as the editor or publisher, then the model has to be changed. The second one comes from the assumption of that population are exchangeable. Exchangeability refers the no changes to the random variable when indices changed.

A small number of studies which focused on Bayesian of FL are proposed. Yurochkin propose a Bayesian non-parametric methodology based on the India Buffet Process. We will introduce an akin to India Buffet Process - Chinese Restaurant Process(CRP). CRP is a process that puts a distribution over partitions formed over devices. Image a Chinese restaurant with an infinite number of tables, each of which can seat an infinite number of customers sits at a new table with probability proportional to alpha. Notice we can view these sitting table decisions as either an observation joins a existing component c or generates a new component for this observation. The labelling of table is independent to the order of observations. No doubt this non-parametric process resembles some properties of Dirichlet process.

The Chinese restaurant process can be defined for the following procedure. Assume an infinite large restaurant is serving customers. The table is a partition of customers. The first customer walks in the restaurant and sits at the first table. The second customer enters and decides either to sit with the first customer, or by himself at a new table. In

general, the nth customer sits down at a table with a probability that is proportional to the number of people already sitting at that table, the other option is to select a new table to sit with a probability proportional to the hyper-parameter α. What this mean is that the more customer sitting at a table, the more likely the next customer sitting at that table. Because of exchangeability of this procedure, the order where customers sit down is not determined by the table indices and we can draw each customer's table assignment z_n by pretending they are the last person to sit down. Let K be the number of tables and let n_k be the number of people sitting at each table. For the nth customer, we define Bernoulli distribution over table assignments conditioned on $\mathbf{z}_{-n}$, that means all other table assignments except the nth:

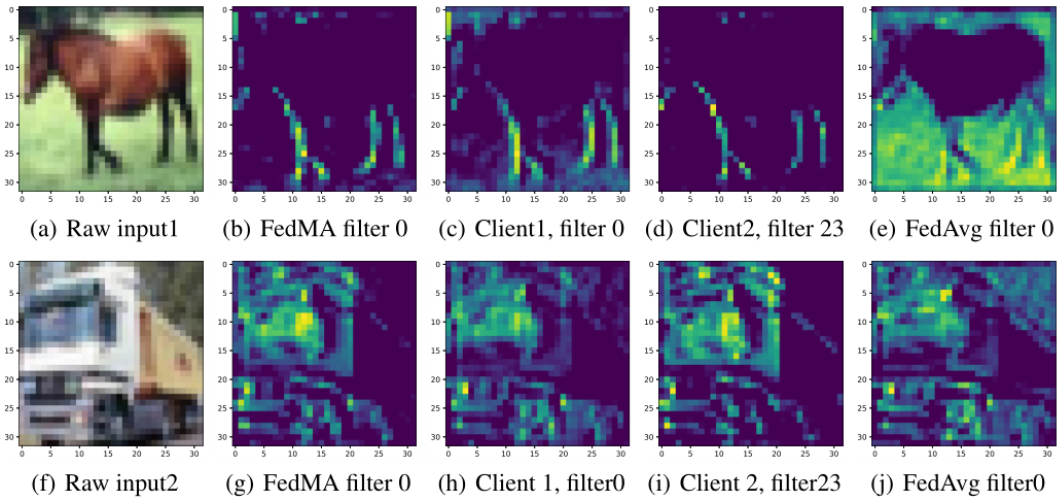
$$p(z_n = k|\mathbf{z}_{-n}, \alpha) \propto \begin{cases} n_k, & if\ k \leq K \\ \alpha, & if\ k = K+1 \end{cases} \tag{2.2}$$

When all N customers have been seated, their table assignments provide a random partition. Though the process is described sequentially, the CRP is exchangeable. The probability of a particular partition of N customers is invariant to the order in which they sat down.

Neuron Matching This subsection to discussed an idea which originally proposed by Mikhail Yurochkin [117]. This technique is suitable for our algorithm FeSEM and may be adopted as an extension. In FL, each device-i has a private dataset $\mathscr{D}_i = \{\mathscr{X}_i, \mathscr{Y}_i\}$, where $\mathscr{X}_i$ and $\mathscr{Y}_i$ denote the input features and corresponding gold labels respectively. Each dataset $\mathscr{D}_i$ will be used to train a local supervised learning model $\mathscr{M}_i : \mathscr{X}_i \rightarrow \mathscr{Y}_i$. $\mathscr{M}$ denotes a deep neural model parameterized by weights W. It is built to solve a specific task, and all devices share the same model architecture.

We now present one key building block of neuron matching in the context of multi-center of FL. The underlying concept comes from a Beta Bernoulli Process [49, 101], which has been studied extensively. Our model assumes the following generative process. First, draw a collection of elements (channels in the case of CNN neural network) from a Beta process prior with has a base measure D and mass parameter γ_0. Moreover, their empirical evaluation choose to be a Gamma process. Each element θ_n is a vector of formed from the feature extractor weight-bias pairs with the corresponding weights of the softmax regression.

Next, from the set $\{\mathscr{D}_i\}$, for each node of this set, we perform local training process, which is a data generative process can be equivalently described as a set of sequence of

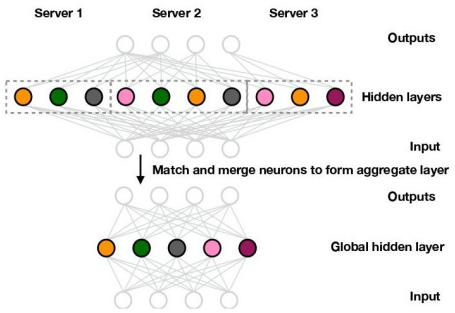


(a) Raw input1 (b) FedMA filter 0 (c) Client1, filter 0 (d) Client2, filter 23 (e) FedAvg filter 0

(f) Raw input2 (g) FedMA filter 0 (h) Client 1, filter0 (i) Client 2, filter23 (j) FedAvg filter0

Figure 2.7: Case study of impact of Neuron matching

Bernoulli trails. In each of these trials, they take value of 1 for including a global atom and 0 for otherwise. The can be formulated as following: That is from a form of pairs, which represent the set of elements used by node i. Finally, assume the observed local atoms are samples drawn from the a distribution as following. Under this model, quantity of interest is the collection of random variables that match observed atoms at any given nodes. It denote the random variable as $\{\mathbf{B}^j\}_{j=1}^J$, where a value of $\mathbf{B}_{i,l}^j = 1$ implies that $\mathcal{T}_{jl} = \theta_i$ and 0 as otherwise, (there is a one-to-one correspondance between observed atoms and global atoms). The $\{\mathbf{B}^j\}$ is to be inferred by cast the posterior distribution into a linear sum assignment. Then they adopt a Hungarian algorithm to solve this sub problem. The illustration of PFNM is shown as in Figure 2.8.



Algorithm 1 Single Layer Neural Matching

1: Collect hidden layers from the J servers and form $\mathbf{v}_{jl}$.
2: Form assignment cost matrix per (9).
3: Compute matching assignments B^j using the Hungarian algorithm (Supplement 1).
4: Enumerate all resulting unique global neurons and use (5) to infer the associated global weight vectors from all instances of the global neurons across the J servers.
5: Concatenate the global neurons and the inferred weights and biases to form the new global hidden layer.

Figure 2.8: Neuron Matching Steps

The PFNM algorithm proposed by Yurochkin et al. [117] only focus on model aggregating and final result is a single model which is not as effective to capturing the differences of data distributions among devices as multi-center approach. Our work also takes non-parametric Bayesian approach which allow us to model growing number of data distributions and estimate associated parameters of theirs.

CHAPTER 3

MULTI-CENTER FEDERATED LEARNING

The widespread of mobile phones and Internet-of-Things has witnessed a huge volume of data generated by end-users on mobile devices. Generally, a service provider on the server side collect users' data and train a global machine learning model such as deep neural networks. Such a centralized machine learning approach causes severe practical issues, e.g., communication costs, consumption of device batteries, and the risk of violating the privacy and of user data.

Federated learning (FL) [83] is a decentralized machine learning framework that learns models collaboratively using the training data distributed on remote devices to boost communication efficiency. Basically, it learns a shared pre-trained model by aggregating the locally-computed updates, and each update is derived from learning the data in the corresponding local device. Therefore, a straightforward aggregation algorithm is responsible for averaging the many local models' parameters, weighted by the size of the training data on each device. Compared with conventional distributed machine learning, FL is robust against unbalanced and non-IID data distributions, which is the defining characteristic of modern AI products for mobile devices.

The vanilla FL addresses a practical setting of distributed learning, where 1) the central server is not allowed to access any user data which protects users' privacy, and 2) the data distribution over different users is non-IID, which is a natural assumption of real-world applications. However, early FL approaches [83, 114] use only one global model as a single-center to aggregate the information of all users. The stochastic gradient descent (SGD) for single-center aggregation is designed for IID data, and therefore,

conflicts with the non-IID setting in FL.

Recently, the non-IID or heterogeneity challenge of FL has been studied to improve the robustness of global models against outlier/adversarial users and devices [47, 70, 71]. Moreover, Sattler et al. proposed an idea of clustered FL (FedCluster) that addresses the non-IID issue by dividing the users into multiple clusters. However, the hierarchical clustering in FedCluster is achieved by multiple rounds of bipartite separation, each requiring the federated SGD algorithm to run until convergence. Hence, its computational and communication efficiency will become bottlenecks when applied to a large-scale FL system. More recently, Mansour et al. [81] and Ghosh et al. [46] proposed to cluster the local models according to the loss of hypothesis. In particular, each user will try all K global models representing K clusters, and then select the best global model as the cluster ID by considering the lowest loss of running the global model on local data. However, this posts high communication and computation overheads because the selected nodes will spend more resources for receiving and running multiple global models.

In this thesis, we propose a novel multi-center FL framework that updates multiple global models by aggregating information from multiple user groups. In particular, the datasets of the users in the same group are likely to be generated or derived from the same or similar distribution. We formulate the problem of the multi-center FL as the joint clustering of users, and then optimizing of the global model for users in each cluster. In particular, (1) each user's local model is assigned to its closest global model, and (2) the global model in each cluster leads to the smallest loss over all the associated users. The proposed multi-center FL not only inherits the communication efficiency of the federated SGD but also retains the capability of handling non-IID data on heterogeneous datasets. Lastly, we propose a new optimization method in line with EM algorithm to train our model.

We summarise our main contributions as:

- We propose a novel multi-center aggregation approach (Section 3.3.1) to address the non-IID challenge of FL.
- We design an objective function, namely multi-center federated loss (Section 3.3.2), for user clustering in FL.
- We propose Federated Stochastic Expectation Maximization (FeSEM) (Section 3.3.3) to solve the optimization of the proposed objective function.
- We present the algorithm as an easy-to-implement and strong baseline for FL. Its effectiveness is evaluated on benchmark datasets. (Section 3.5)

3.1 Related work

Decision making is the process of making choices by identifying a decision, gathering information, and assessing alternative resolutions. In most of the scenarios, each individual person usually makes a personal choice given the collected information. To model the personalized decision-making process, a general solution is to collect the user's personal characteristics, e.g. demographics, behavior history [51], and social networks, as part of the input to be considered by a centralized intelligent model. This solution usually train a large-scale machine learning or recommendation models at cloud server using the collected personal data from users, thus it will cause privacy concerns. In recent, a new service architecture has been proposed to provide service based on a standalone on-device intelligent in each smart device. In particular, a unique intelligent model customized for each user will be deployed to the user' smart device, so as to provide service independently while not relying on the decision from the cloud server. The user's personal data will be stored locally to train the intelligent model, thus no personal data will be uploaded to the server.

To solve the problem caused by non-IID data in a federated setting proposed clustered FL (FedCluster) by integrating FL and bi-partitioning-based clustering into an overall framework, and proposed a hypothesis-based federated clustering that assigns the cluster by considering the loss of running the global model on local data. Ghosh proposed a robust FL comprising three steps: 1) learning a local model on each device, 2) clustering model parameters to multiple groups, each being a homogeneous dataset, and 3) running a robust distributed optimization in each cluster. propose a general form to model the clustered FL problem into a bi-level optimization framework, and then conduct theoretical analysis on the convergence.

3.2 Background

3.2.1 Problem Setting

In FL, each device-i has a private dataset $\mathscr{D}_i = \{\mathscr{X}_i, \mathscr{Y}_i\}$, where $\mathscr{X}_i$ and $\mathscr{Y}_i$ denote the input features and corresponding gold labels respectively. Each dataset $\mathscr{D}_i$ will be used to train a local supervised learning model $\mathscr{M}_i : \mathscr{X}_i \rightarrow \mathscr{Y}_i$. $\mathscr{M}$ denotes a deep neural model parameterized by weights W. It is built to solve a specific task, and all devices share the same model architecture.

For the i-th device, given a private training set $\mathscr{D}_i$, the training procedure of $\mathscr{M}_i$ is represented in brief as

$$\min_{W_i} L_s(\mathscr{M}_i, \mathscr{D}_i, W_i), \tag{3.1}$$

where $L_s(\cdot)$ is a general definition of the loss function for any supervised learning task, and its arguments are model structure, training data and learnable parameters respectively, and W' denotes the parameters after training. In general, the data from one device is insufficient to train a data-driven neural network with satisfactory performance. An FL framework optimizes the local models in a distributed manner and minimizes the loss of the local data on each device.

Hence, the optimization in vanilla FL over all the local models can be written as

$$\min_{\{W_i\}_{i=1}^m} \sum_{i=1}^{m} \frac{|D_i|}{\sum_j |\mathscr{D}_j|} L_s(\mathscr{M}_i, \mathscr{D}_i, W_i), \tag{3.2}$$

where m denotes the number of devices.

On the server side, the vanilla FL aggregates all local models into a global one $\mathscr{M}_{global}$ which is parameterized by $\tilde{W}^g$. In particular, it adopts a weighted average of the local model parameters $[W_i]_{i=1}^m$, i.e.,

$$\tilde{W}^g = \sum_{i=1}^{m} \frac{|\mathscr{D}_i|}{\sum_j |\mathscr{D}_j|} W_i, \tag{3.3}$$

which is the nearest center for all $\{W_i\}_{i=1}^m$ in terms of a weighted L2 distance:

$$\tilde{W}^g \in \arg\min_{\tilde{W}} \sum_{i=1}^{m} \frac{|D_i|}{\sum_j |\mathscr{D}_j|} \|\tilde{W} - W_i\|_2^2. \tag{3.4}$$

More generally, we can replace the L2 distance in Eq. (3.4) by other distance metric $\mathrm{Dist}(\cdot,\cdot)$ and minimize the difference between the global model and all the local models, i.e.,

$$\min_{\tilde{W}} \frac{1}{m} \sum_{i=1}^{m} \mathrm{Dist}(W_i, \tilde{W}). \tag{3.5}$$

The above aims to find a consistent solution across global model and local models. Note that a direct macro average is used here regardless of the weight of each device, which treats every device equally. The weights used in Eq. (3.2) can easily be incorporated for a micro average.

The divergence $\mathrm{Dist}(\cdot,\cdot)$ between the global model and local models plays an essential role in the FL objective. The simple L2 distance for $\mathrm{Dist}(\cdot,\cdot)$ does not take into account the

fact that two models can be identical under the arbitrary permutation of neurons in each layer. Hence, the lack of neuron matching may cause misalignment in that two neurons with similar functions and different indexes cannot be aligned across models [117]. However, the index-based neuron matching in FL [94] is the most widely used method and works well in various real applications. One potential reason for this is that the index-based neuron matching can also slowly align the function of neurons by repeatedly initializing all local models with the same global model. To simplify the description, we will discuss our method for index-based neuron matching, and then discuss a possible extension by adding function-based neuron matching [106] (Section3.4.1).

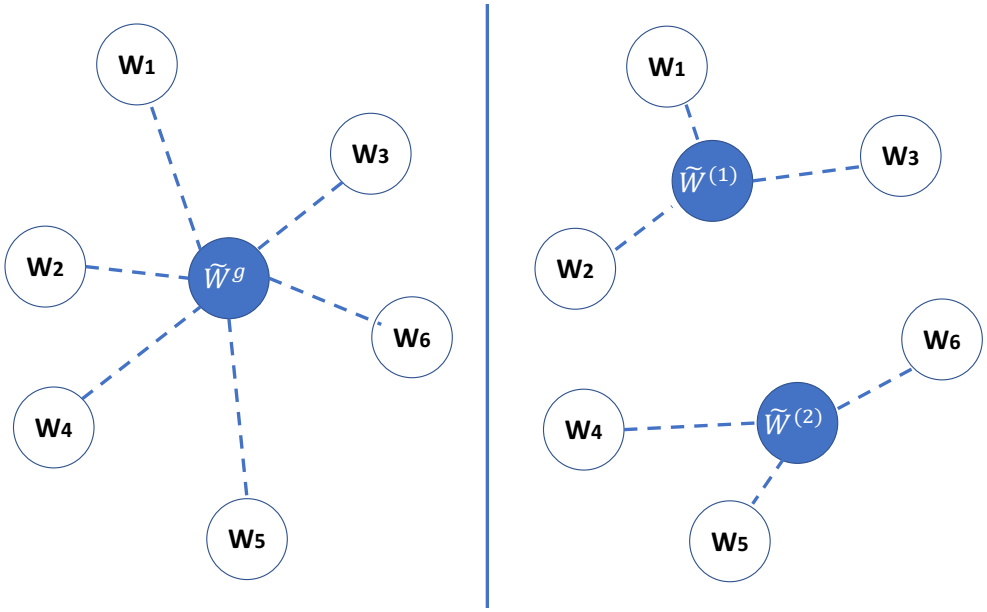


Figure 3.1: Comparison between single-center aggregation in vanilla FL (left) and multi-center aggregation in the proposed one (right). Each W_i represents the local model's parameters collected from the i-th device, which is denoted as a node in the space. $\tilde{W}$ represents the aggregation result of multiple local models.

3.2.2 Motivation

Federated learning (FL) usually aggregates all local models to a single global model. However, this single-center aggregation is fragile under heterogeneity. In contrast, we consider FL with multiple centers to better capture the heterogeneity by assigning nodes to different centers so only similar local models are aggregated. Consider two extreme cases for the number of centers, K: (1) when $K = 1$, it reduces to the FedAvg with a single global model, which cannot capture the heterogeneity and the global model might

perform poorly on specific nodes; (2) When $K = m$, the heterogeneity problem can be avoided by assigning each node to one global model. But the data on each device used to update each global model can be insufficient and thus we lose the main advantage of FL. Our goal is to find a sweet point between these two cases to balance the advantages of federated averaging and the degradation caused by underlying heterogeneity.

Learning one unique model for each node has been discussed in some recent FL studies for better personalized models. They focus on making a trade-off between shared knowledge and professionalisation. The personalising strategy either applies fine-tuning of the global model [119] for each node, or only updates a subset of personalised layers for each node [4, 74], or deploys a regularisation term in the objective [35, 38, 52]. In contrast, Multi-center FL in this thesis mainly focuses to address the heterogeneity challenge by assigning nodes to different global models during aggregation. But it can be easily incorporated in these peronalization strategies. In the following, we will start from the problem setting for the the vanilla FL, and then elucidate our motivation of improving FL's tolerance to heterogeneity by multi-center design.

3.3 Methodology

3.3.1 Multi-Center Model Aggregation

To overcome the challenges arising from the heterogeneity in FL, we propose a novel model aggregation method with multiple centers, each associating with a global model $\tilde{W}^{(k)}$ updated by aggregating a cluster of user's models with nearly IID data. In particular, all the local models will be grouped to K clusters, denoted as $C_1, \cdots, C_K$, each covering a subset of local models with parameters $\{W_j\}_{j=1}^{m_k}$.

An intuitive comparison between the vanilla FL and our multi-center FL is illustrated in Fig. 3.1. As shown in the left figure, there is only one center model in vanilla FL. In contrast, the multi-center FL shown in the right has two centers, $W^{(1)}$ and $W^{(2)}$, and each center represents a cluster of devices with similar data distributions and models. Obviously, the right one has a smaller intra-cluster distance than the left one. As discussed in the following Section 3.3, intra-cluster distance directly reflects the possible loss of the FL. Hence, a much smaller intra-cluster distance indicates our proposed approach potentially reduces the loss of FL.

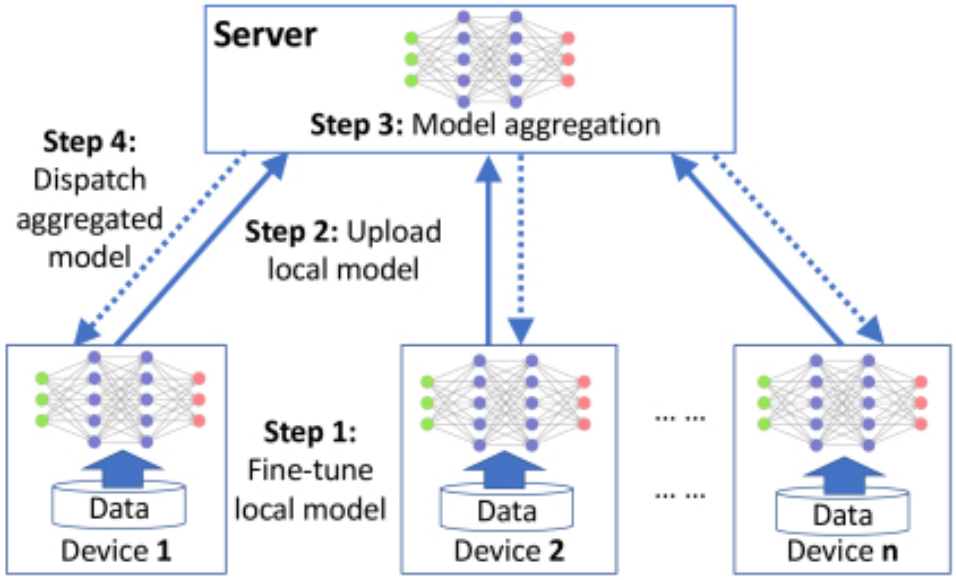


Figure 3.2: A high-level view of Federated Learning

3.3.2 Problem Formulation

Solving a joint optimization on a distributed network. The multi-center FL problem can be formulated as

$$\min_{\{W_i\},\{r_i^{(k)}\},\{\tilde{W}^{(k)}\}} \sum_{i=1}^{m} \frac{|D_i|}{\sum_j |\mathscr{D}_j|} L_s(\mathscr{M}_i, \mathscr{D}_i, W_i) + \frac{\lambda}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} r_i^{(k)} \mathrm{Dist}(W_i, \tilde{W}^{(k)}), \tag{3.6}$$

where λ controls the trade-off between supervised loss and distance. We solve it by applying an alternative optimization between server and user: (1) on each node-i, we optimize the above objective w.r.t. W_i while fixing all the other variables; and (2) on the server, we optimize $\{r_i^{(k)}\}, \{\tilde{W}^{(k)}\}$ for $i \in [m]$ and $k \in [K]$ while fixing all local models $\{W_i\}$.

Multi-center assignment at the server end. The second term in Eq. (3.6) aims to minimize the distance between each local model and its nearest global model. Under the non-IID assumption, the data located at different devices can be grouped into multiple clusters where the on-device data in the same cluster are likely to be generated from one

distribution. As illustrated on the right of Fig. 3.1, we optimizes the assignments and global models by minimizing the intra-cluster distance, i.e,

$$\min_{\{r_i^{(k)}\},\{\tilde{W}^{(k)}\}} \frac{1}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} r_i^{(k)} \text{Dist}(W_i, \tilde{W}^{(k)}), \tag{3.7}$$

where cluster assignment $r_i^{(k)}$, as defined in Eq. (3.9), indicates whether device-i belongs to cluster-k, and $\tilde{W}^{(k)}$ is the parameters of the aggregated model for cluster-k.

Distance-constrained loss for local model optimization. Because the distance between the local model and the global model are essential to our new loss, we don't expect the local model will be changed too much during the local updating stage. The new loss consists of a supervised learning loss and a regularization term to constrain the local model to ensure it is not too far from the global model. This kind of regularization term is also known as the proximal term in [70] that can effectively limit the impact of the variable local updates in FL. We minimize the loss below for each local model W_i as follows:

$$\min_{W_i} \frac{|D_i|}{\sum_j |\mathscr{D}_j|} \cdot L_s(\mathscr{M}_i, \mathscr{D}_i, W_i) + \frac{\lambda}{m} \sum_{k=1}^{K} r_i^{(k)} \text{Dist}(W_i, \tilde{W}^{(k)}) \tag{3.8}$$

3.3.3 Optimization Algorithm

In general, Expectation-Maximization (EM) [9] can be used to solve the distance-based objective function of clustering, e.g., K-Means. However, in contrast to the general objective of clustering, our proposed objective, as described in Eq. 3.7, has a dynamically changing W_i during optimization. Therefore, we adapt the Stochastic Expectation Maximization (SEM) [20] optimization framework by adding one step, i.e., updating W_i. In the modified SEM optimization framework, named federated SEM (FeSEM), we sequentially conduct: 1) E-step – updating cluster assignment $r_i^{(k)}$ with fixed W_i, 2) M-step – updating cluster centers $\tilde{W}^{(k)}$, and 3) updating local models by providing new initialization $\tilde{W}^{(k)}$.

Firstly, for the **E-Step**, we calculate the distance between the cluster center and nodes – each node is the model's parameters W_i, then update the cluster assignment $r_i^{(k)}$ by

$$r_i^{(k)} = \begin{cases} 1, & if\ k = \arg\min_j \text{Dist}(W_i, \tilde{W}^{(j)}) \\ 0, & otherwise. \end{cases} \tag{3.9}$$

Algorithm 1: FeSEM – Federated Stochastic EM

Initialize $K, \{W_i\}_{i=1}^{m}, \{\tilde{W}^{(k)}\}_{k=1}^{K}$
while *stop condition is not satisfied* **do**
  E-Step:
  Calculate distance $d_{ik} \leftarrow \text{Dist}(W_i, \tilde{W}^{(k)}) \ \ \forall i,k$
  Update $r_i^{(k)}$ using d_{ik} (Eq. 3.9)
  M-Step:
  Group devices into C_k using $r_k^{(k)}$
  Update $\tilde{W}^{(k)}$ using $r_i^{(k)}$ and W_i (Eq. 3.10)
  for *each cluster* $k = 1, \ldots K$ **do**
    for $i \in C_k$ **do**
      Send $\tilde{W}^{(k)}$ to device i
      $W_i \leftarrow$ **Local_update**$(i, \tilde{W}^{(k)})$
    end
  end
end

Secondly, for the **M-Step**, we update the cluster center $\tilde{W}^{(k)}$ according to the W_i and $r_i^{(k)}$, i.e.,

$$\tilde{W}^{(k)} = \frac{1}{\sum_{i=1}^{m} r_i^{(k)}} \sum_{i=1}^{m} r_i^{(k)} W_i. \tag{3.10}$$

Thirdly, to **update the local models**, the global model's parameters $\tilde{W}^{(k)}$ are sent to each device in cluster k to update its local model, and then we can fine-tune the local model's parameters W_i using a supervised learning algorithm on its own private training data while considering the new loss as described in Eq. 3.8.

The local training procedure is a supervised learning task by adding a distance-based regularization term. The local model is initialized by the global model $\tilde{W}^{(k)}$ which belong to the cluster associated with the node.

Lastly, we repeat the three stochastic updating steps above until convergence. The sequential executions of the three updates comprise the iterations in FeSEM's optimization procedure. In particular, we sequentially update three variables $r_i^{(k)}$, $\tilde{W}^{(k)}$, and W_i while fixing the other factors. These three variables are jointly used to calculate the objective of our proposed multi-center FL in Eq. 3.7.

We implement FeSEM in Algorithm 1 which is an iterative procedure. As elaborated in Section 3.3.2, each iteration comprises of three steps to update the cluster assignment, the cluster center, and the local models, respectively. In the third step to update the local model, we need to fine-tune the local model by implementing Algorithm 2.

Algorithm 2: Local_update

Input: i – device index
$\tilde{W}^{(k)}$ – the model parameters from server
Output: W_i – updated local model
Initialization: $W_i \leftarrow \tilde{W}^{(k)}$
for *N local training steps* **do**
  Update W_i with training data $\mathscr{D}_i$ (Eq. 3.8)
end
Return W_i to server

Datasets	FEMNIST				FedCelebA			
Metrics(%)	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1
NoFed	79.0±2.0	67.6±0.6	81.3±1.9	51.0±1.2	83.8±1.4	66.0±0.4	83.9±1.6	67.2±0.6
FedSGD	70.1±2.2	61.2±3.4	71.5±1.8	46.7±1.2	75.7±2.3	60.7±2.4	75.6±2.0	55.6±2.6
FedAvg	84.9±2.0	67.9±0.4	84.9±1.6	45.4±1.9	86.9±0.5	**78.0**±1.0	86.1±0.4	54.2±0.6
FedDist	79.3±0.8	67.5±0.5	79.8±1.1	50.5±0.5	71.8±0.9	61.0±0.8	71.6±1.0	61.1±0.7
FedDWS	80.4±0.8	67.2±1.6	80.6±1.2	51.7±1.1	73.4±1.7	59.3±0.9	73.4±1.9	50.3±0.5
Robust(TKM)	78.4±1.0	53.1±0.5	77.6±0.7	53.6±0.7	90.1±1.3	68.0±0.7	90.1±1.3	68.3±1.1
FedCluster	84.1±1.1	64.3±1.3	84.2±1.0	**64.4**±1.6	86.7±0.7	67.8±0.9	87.0±0.9	67.8±1.3
HypoCluster(3)	82.5±1.7	61.3±0.6	82.2±1.3	61.6±0.9	76.1±1.5	53.5±1.0	72.7±1.8	53.8±1.9
FedDane	40.0±2.9	31.8±3.1	41.7±2.4	31.7±1.6	76.6±1.1	61.8±2.0	75.9±1.0	62.1±2.2
FedProx	72.6±1.8	62.8±1.6	74.3±2.1	50.6±1.2	83.8±2.0	60.9±1.2	84.9±1.8	65.7±1.2
FeSEM(2)	84.8±1.1	65.5±0.4	84.8±1.6	52.0±0.5	89.1±1.3	64.6±1.0	89.0 ±1.3	56.0±1.3
FeSEM(3)	87.0±1.2	68.5±2.0	86.9±1.2	41.7±1.5	88.1±1.9	64.3±0.8	87.5±2.0	55.9±0.8
FeSEM(4)	**90.3**±1.5	70.6±0.9	**91.0**±1.8	53.4±0.6	**93.6**±2.7	**74.8**±1.5	**94.1**±2.2	**69.5**±1.1
FeSEM-MA(3)	**90.4**±1.5	**71.4**±0.5	87.0±2.0	64.3±0.5	84.5±0.8	64.1±0.7	85.1±1.0	63.0±1.3

Table 3.1: Comparison of our proposed FeSEM(K) algorithm with the baselines on FEMNIST and FedCelebA datasets. Note the number in parenthesis following "FeSEM" denotes the number of clusters, K.

3.4 Some Possible Extensions

To further handle heterogeneous data in FL scenario, our multi-center FL approach can be easily extended with other packages. We discuss two beneficial techniques here.

3.4.1 Model Aggregation with Neuron Matching

The vanilla FL algorithm, FedAvg [83], uses model aggregation with index-based neuron matching which may cause the incorrect alignmentment. Neurons with similar functions are usually take different indexes in two models. Recently, a function-based neuron

DATASET	FEMNIST	FedCelebA
# of data points	805,263	200,288
# of device	3,550	9,343
# of Classes	62	2
Model architecture	CNN	CNN

Table 3.2: Statistics of datasets.

matching [106] in FL is proposed to align two models by matching the neurons with similar functions. In general, the index-based neuron matching can gradually align the neuron's functionality across nodes by repeatedly forcing each local model to be initialized using the same global model. However, the function-based neuron matching can speed up the convergence of neuron matching and preserve the unique functional neuron of the minority groups.

In this work, we integrate layer-wise matching and then averaging(MA) [106] into ours to increase the capacity to handle heterogeneous challenges. The distance between the local model and the global model is the neuron matching score that is calculated by estimating the maximal posterior probability of the j-th client neuron l generated from a Gaussian with mean W_i, and ϵ and $f(\cdot)$ are guided by the Indian Buffet Process prior [117].

3.4.2 Selection of K

The selection of K, the number of centers, is essential for a multi-center FL. In general, the K is defined based on the prior experience or knowledge of data. If there is no prior knowledge, the most straightforward solution is to run the algorithm using different K and then select the K with the best performance in terms of accuracy or intra-cluster distance. Selecting the best K in a large-scale FL system is time consuming, hence we simplify the process by running the algorithm on a small number of sampled nodes with several communication rounds. For example, we can randomly select 100 nodes and test K in FL with three communication rounds only, and then apply the K to the large-scale FL.

3.5 Experiments

As a proof-of-concept scenario to demonstrate the effectiveness of the proposed method, we experimentally evaluate and analyze FeSEM on two datasets.

3.6 Training Setups

Datasets. We employed two publicly-available federated benchmarks datasets introduced in LEAF [19]. LEAF is a benchmarking framework for learning in federated settings. The datasets used are Federeated Extended MNIST (FEMNIST)[1] [28] and Federated CelebA (FedCelebA)[2] [76]. We follow the setting of the benchmark data in LEAF. In FEMNIST, images is split according to the writers. For FedCelebA, images are extracted for each person and developed an on-device classifier to recognize whether the person smiles or not. A statistical description of the datasets is described in Table 3.2.

Local model. We use a CNN with the same architecture from [76]. Two data partition strategies are used: (a) an ideal IID data distribution using randomly shuffled data, (b) a non-IID partition by use a $\mathbf{p}_k \sim Dir_J(0.5)$. Part of the code is adopted from [106]. For FEMINST data, the local learning rate is 0.003 and epoch is 5. and for FedCelebA, 0.03 and 10 respectively.

Baselines. In the scenario of solving statistical heterogeneity, we choose FL methods as follows:

1. **NonFed**: We will conduct the supervised learning task at each device without the FL framework.
2. **FedSGD**: uses SGD to optimise the global model.
3. **FedAvg**: is an SGD-based FL with weighted averaging. [83] .
4. **FedCluster**: is to enclose FedAvg into a hierarchical clustering framework [90].
5. **HypoCluster(K)**: is a hypothesis-based clustered-FL algorithm with different K [81].
6. **Robust** our implementations based on the proposed method in [47], see this baseline settings in Appendix.
7. **FedDANE**: this is an FL framework with a Newton-type optimization method. [71].
8. **FedProx**: this is our our own implementations following [70]. We set scaler of proximal term to 0.1.
9. **FedDist**: we adapt a distance based-objective function in Reptile meta-learning [85] to a federated setting.

[1] http://www.nist.gov/itl/products-and-services/emnist-dataset
[2] `http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`

10. **FedDWS**: a variation of FedDist by changing the aggregation to weighted averaging where the weight depends on the data size of each device.
11. **FeSEM(K)**: our multi-center FL implemented on federated SEM with K clusters.
12. **FeSEM-MA(K)**: FeSEM integrates the matched averaging [106].

Training settings. We used 80% of each device's data for training and 20% for testing. For the initialization of the cluster centers in FeSEM, we conducted pure clustering 20 times with randomized initialization, and then the "best" initialization, which has the minimal intra-cluster distance, was selected as the initial centers for FeSEM. For the local update procedure of FeSEM, we set N to 1, meaning we only updated W_i once in each local update.

Evaluation metrics. Given numerous devices, we evaluated the overall performance of the FL methods. We used classification accuracy and F1 score as the metrics for the two benchmarks. In addition, due to the multiple devices involved, we explored two ways to calculate the metrics, i.e., micro and macro. The only difference is that when computing an overall metric, "micro" calculates a weighted average of the metrics from devices where the weight is proportional to the data amount, while "macro" directly calculates an average over the metrics from devices.

3.6.1 Experimental Study

Comparison study. As shown in Table 3.1, we compared our proposed FeSEM with the baselines and found that FeSEM achieves the best performance in most cases. But, it is observed that the proposed model achieves an inferior performance for Micro F1 score on the FedCelebA dataset. A possible reason for this is that our objective function defined in Eq. 3.7 does not take into account the device weights. Hence, our model is able to deliver a significant improvement in terms of "macro" metrics. Furthermore, as show in the last three columns in Table **??**, we found that FeSEM with a larger number of clusters empirically achieves a better performance, which verifies the correctness of the non-IID assumption of the data distribution.

Convergence analysis. To verify the convergence of the proposed approach, we conducted a convergence analysis by running FeSEM with different cluster numbers K (from 2 to 4) in 100 iterations. As shown in Fig. 3.3, FeSEM can efficiently converge on both datasets and it can achieve the best performance with the cluster number $K = 4$.

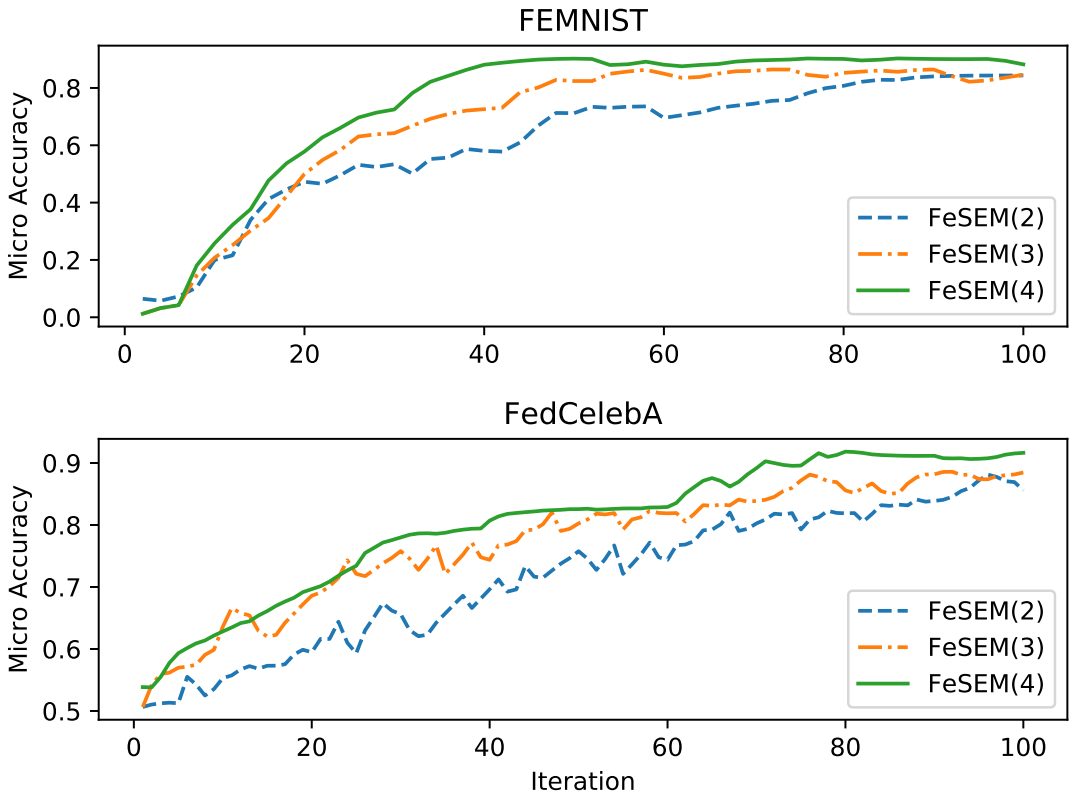


Figure 3.3: Convergence analysis for the proposed FeSEM with different cluster number (in parenthesis) in terms of micro-accuracy.

Clustering analysis. To check the effectiveness of our proposed optimization method and whether the devices grouped into one cluster have similar model, we conducted a clustering analysis via an illustration. We used two-dimensional figures to display the clustering results of the local models derived from FeSEM(4) on the FEMNIST dataset. In particular, we randomly chose 400 devices from the dataset and plotted each device's local model as one point in the 2D space after PCA dimension reduction. As shown in Fig. 3.4, the dataset suitable for four clusters that are distinguishable to each other.

Case study on clustering. To intuitively judge whether nodes grouped into the same cluster have a similar data distribution, we conducted case studies on a case of two clusters that are extracted from a trained FeSEM(2) model. For FMNIST, as shown on the top of Fig. 3.5, cluster on the right consists writers who are likely to recognize hand-writings with a smaller font, and on the left consists writers who are likely to recognize hand-writing with a bolder and darker font. For FedCelebA, see full face images in Appendix section 2, the face recognition task in cluster1 is likely to handle the smiling

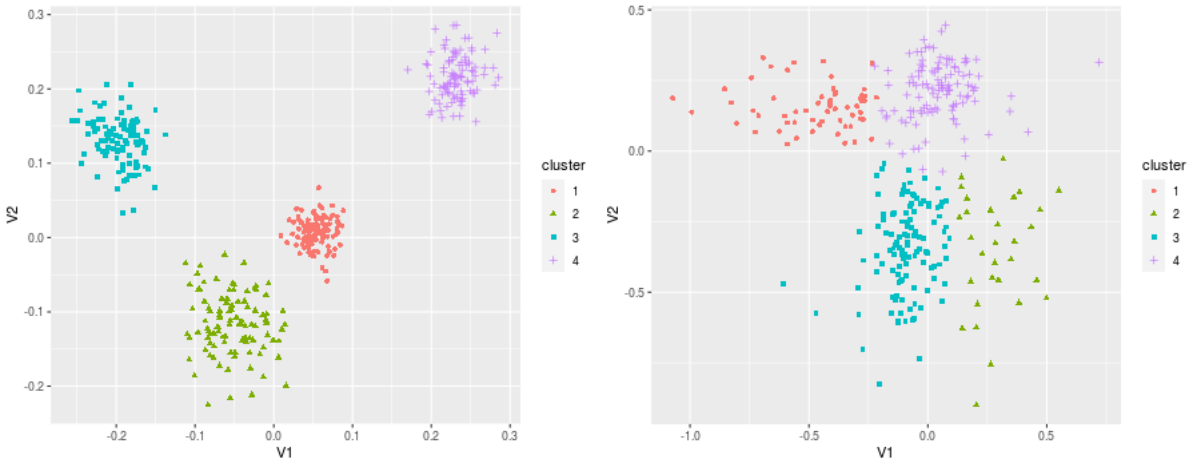


Figure 3.4: Clustering analysis for different local models (using PCA) derived from FeSEM(4) using FEMNIST and Celeba data.

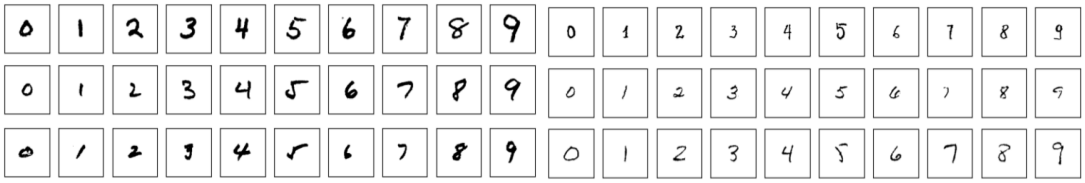


Figure 3.5: Figure shows the clustering effect of FeSEM on dataset FEMNIST by writers, on the left are three writers handwritten digits which are smaller and lighter than on the right ones

faces with a relatively simple background, also exhibits to be young people. While cluster on the right is likely to handle the faces with more diverse background and also seems to be more older people.

3.7 Conclusion and Remarks

In section 1 of this chapter, address the robustness of the FeSEM. We proposed a practical and robust version of the FeSEM algorithm to offset the adversarial worker nodes using K-bmom estimator. This algorithm has better properties and converges faster than the original FeSEM. In experiments, we show that on three datasets, the proposed algorithm has similar performance to other baselines while showing much superior clustering performance than FeSEM in all three datasets. We also discussed possible extensions of FedRobust for better distance computation if the local model is a neural network.

In section 2 of this chapter, we proposed a novel FL algorithm to tackle the non-IID challenge of FL. This proposed method can efficiently capture the multiple hidden

distributions of numerous devices or users. An optimization approach, federated SEM, is also proposed to solve the multi-center FL problem effectively. The experimental results show the effectiveness of our algorithm, and several analyses are further provided for an deeper insight into the proposed approach.

CHAPTER 4

ROBUST CLUSTERING FOR MUTLI-CENTER FEDERATED LEARNING

4.1 Introduction

Federated Learning (FL) is a new machine learning paradigm to enable many clients collaboratively learn intelligent models. The vanilla FL, namely FedAvg [83], was proposed to learn a server-side intelligent model using many distributed clients without direct access to their local dataset. This distributed machine learning framework with data locality can greatly mitigate the risk of privacy [62] in contrast to a traditional learning system with centralized data storage. Due to the heterogeneous nature of such a distributed system, a major challenge for FL is to tackle non-IID data across clients. For example, a smartphone typing tool, GBoard in an Android smartphone, needs to auto-fill the incomplete words by considering the typing context and user's language preference. The user's historical data are usually non-IID across clients, thus the learning system needs to tackle this non-IID challenge in the FL's distributed settings.

To solve the non-IID challenge in FL, one solution is to enhance the robustness of a single model at the server to tackle various distributions across clients. However, this kind of solution can only tackle the scenario with slight differences of data distributions across clients. A recent solution for tackling non-IID issues is personalized FL that aims to optimize each client-wise local model while using the global model as a regularize to exchange shared information and constraint the divergence across client-wise local

models. The personalized FL suffers the increased complexity of optimization problems that usually treat client-wise model learning as a joint optimization problem across clients. Moreover, it is impractical to find a proper trade-off between shared knowledge and personalization.

Clustered FL is a trade-off solution between single model FL and personalized FL. It aims to learn multiple global models on the server while each global model is a cluster-wise personalized model for the clients with similar data distribution. In particular, the clustering method is a tool to assign clients to different clusters. Therefore, clustered FL can gain better personalization capability than single model FL, and also can learn a model with better generalization than client-wise personalized FL methods. However, clustering among clients is very sensitive to outliers or adversarial attacks. In general, an Outlier is a data point that primarily differs from other observations. Some examples of outliers in the case of Federated Learning are those systematic mislabelling of data or Byzantine failures [10]. In practice, even a small proportion of outliers can render clustering unreliable, while cluster centres and model parameter estimators can be severely biased. Therefore, tackling client-wise outliers will be a new challenge for clustered FL systems.

To tackle the aforementioned challenge, this thesis proposes a novel robust clustered Federated Learning framework to tackle the client-wise outliers which could be a minority of users with abnormal behaviour patterns or could be from malicious clients equipped with Byzantine attack tools, i.e., arbitrarily corrupt the information using some adversarial attache mechanism. In particular, we enhance the federated aggregation mechanism by adopting a bootstrap sampling method and a robust approach based on a median-of-means estimator. We formulate the problem into a bi-level optimization framework for a general form and then use a stochastic EM method to solve the optimization problem in an alternative updating strategy.

The motivation for using bootstrap median-of-mean to implement robust clustered FL is quite straightforward. The clients clustering in the FL system is usually based on the client's local models that usually to be a high dimensional vector derived from deep neural networks, such as CNN, RNN and Transformers. The high dimensional data exaggerates the outlier problem in clustering, and also most distance-based regularization-based robust clustering is impractical in this scenario. Specifically, most clustering methods use a mean-based estimator to compute the centre of a cluster. However, computing barycenter or mean can be very sensitive to the presence of outliers. In contrast to the mean-based estimator with penalty term, a median point-based estimator will be a better

option to implement the robustness of the clustering algorithm.

The part of this thesis's contributions are summarized as below.

- We propose a simple yet effective approach, namely FedRoc, to implement robust clustered FL.
- We adopt bootstrap sampling during initialization together with a median-of-means estimator to solve the outlier problem in clustered FL contexts.
- We formulate the problem into a bi-level optimization problem.
- Compared with other FL methods on a few datasets, it shows that FedRoC is computationally competitive and more robust than any other baseline algorithms.

The remainders of this thesis is organized as below. Related work of this section has been introduced at Chapter 2 already, and then introduce the method at Section 4.2. The experiment results has been analyzed in Section 4.3. We make conclusion and discuss future work at Section 4.4.

4.2 Methodology

4.2.1 Problem Definition

The basic of FL is that many clients in a network collaboratively train a global model. The dataset kept on a local device is only a shard of a much larger dataset. To formuate the FL system, it is composed of m smart devices that has a private dataset $\mathscr{D}_i = \{< \mathscr{X}_i, \mathscr{Y}_i >\}$ for each, where $\mathscr{X}_i$ and $\mathscr{Y}_i$ denote input samples and corresponding labels respectively, and $i \in \{1,...,m\}$ is the index of a client. Each dataset $\mathscr{D}_i$ on the device will be used to train a local supervised learning model $\mathscr{M}_i : \mathscr{X}_i \to \mathscr{Y}_i$. $\mathscr{M}$ usually denotes a deep neural model parameterized by weights ω. It is built to solve a specific task, and all devices share the same model architecture. Generally, the problem of FL can be usually denote as minimizing this formula below,

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{m} \frac{n_i}{n} \mathscr{L}(\mathscr{M}_i, \mathscr{D}_i, \omega_i) \tag{4.1}$$

where n_i is the number of samples kept in i-th node and n is the total number of samples on all nodes, $\mathscr{L}(\mathscr{M}_i, \mathscr{D}_i, \omega_i)$ is the local objective function which describes how good are the trained classifier, and the local objective function is different for different

classifiers (e.g., logistic regression, neural network). For the i-th device, given a private training set $\mathscr{D}_i$, the training procedure of $\mathscr{M}_i$ is briefly notes as

$$\omega_i^* = \underset{\omega_i}{\operatorname{argmin}} \mathscr{L}(\mathscr{M}_i, \mathscr{D}_i, \omega_i), \tag{4.2}$$

where $\mathscr{L}(\cdot)$ is a general definition of loss function for any supervised learning task, and its arguments include model structure $\mathscr{M}_i$, training data D_i and learnable parameters ω_i whose vector space is d dimensional.

In particular, a master node needs to keep a global model whose parameters w normally is the weighted average of model parameters ω_i of all worker nodes. During the FL learning process, three steps will be performed at each iteration:

- step I: a master node sends the global model parameter to all.
- step II: worker nodes compute update with respect to the global model parameter using local objective function and training data kept on devices then sends back the update.
- step III: a master node aggregates all updates to obtain a new global model using a certain aggregation rule. In the case of Federated Averaging, the rule is the weighted average. Formally noted as: $w = \sum_{i=1}^{m} \frac{n_i}{n} \omega_i$.

To tackle the non-IID challenge in FL, clustered FL is an important variant which can achieve good performance on this. And its formulation can be written as a bi-level objective as below,

$$\underset{C}{\text{minimize}} \ \frac{1}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} r_{i,k} \mathscr{L}(\mathscr{M}, D_i, c_k) \tag{4.3a}$$

$$\textit{subject to } r_{i,k}, C = \underset{r_{i,k}, C}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{i=1}^{m} r_{i,k} d(g_i, G_k) \tag{4.3b}$$

where all clients share the same model structure $\mathscr{M}$, K is the number of clusters, $r_{i,k}$ is the indication assignment matrix of the clustering problem to determine that i-th device belongs to cluster k, $C = \{c_1, \ldots, c_K\}$ represents centroids of K clusters, g_i and G_k are the measure of client i and cluster k, respectively, which can be model parameters or loss, d is the distance function. To simply the formulation, weight of each client is set to be $1/m$. The upper Equation 5.7a is the objective of FL, and the lower Equation 5.7b is the clustering objective, while a bi-level optimization structure is adopted to connect the FL with clustering.

4.2.2 Robust clustered FL with bMOM

While K-means is widely used for clustering, its robustness is limited. It shows a poor convergence rate or is not able to correctly group data when there are outliers and adversarial contamination. Several other robust versions of EM or K-means already existing, such as K-PDTM, trimmed-K-means, K-medians have also been proposed. Compared to the above robust variants of K-means, K-bMOM may be increased the computation complexity at each step of the learning process due to bootstrapping. Yet, K-bMOM holds a number of beneficial effects, such as better break down points. Also, K-bMOM in theory has a higher convergence rate when there is a certain level of outliers in the sample compared to the K-means method.

The breakdown point is a classical measure in the robust statistics literature to measure, which represents the maximum proportion of outliers that leaves the estimator bounded. In bMOM, it implies if the block size n_B is rightly chosen, then the probability that the bMOM remains stable under adversarial contamination tends to be one when the number of blocks tends to infinity [18]. This shows that when the number of blocks is big enough, then the bMOM can have a better breakdown point than empirical means. Overall, to address the outliers and non-IID in FL and make the performance of FL more robust, bMOM estimator is imported to combined with clustered FL. And the loss function of the clustering task can be defined as:

$$\mathscr{R} = \text{med}\{\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\|\omega_i - c_k^{(b)}\|_2^2 : b \in \{1,...,B\}, n_B > K\}, \tag{4.4}$$

where B blocks are bootstrapped from m devices' data with block size $n_B > K$, med is to find the median of B minimum losses, and $c_k^{(b)}$ is the center of cluster k based on b-th block of data.

Combined bMOM with the clustered FL problem, using model parameters to measure the client or cluster, and Euclidean distance to measure the distance of clients and clusters, and k-means as the clustering method, we can formulate the loss as a bi-level optimization problem.

$$\underset{C^{(b_{med})}}{\text{minimize}} \ \frac{1}{m}\sum_{k=1}^{K}\sum_{i=1}^{m} r_{i,k}\lambda_i \mathscr{L}(\mathscr{M}_k, \mathscr{D}_i, c_k^{(b_{med})}) \tag{4.5}$$

$$\textit{subject to } r_{i,k}, C^{(b_{med})} = \underset{r_{i,k}, c_1^{(b_{med})},...,c_K^{(b_{med})}}{\text{argmin}} \ \mathscr{R} \tag{4.6}$$

where $C^{(b_{med})} = \{c_1^{(b_{med})}, \ldots, c_K^{(b_{med})}\}$ represents centroids of the block b_{med}, which has the median loss in B blocks, and λ_i is the weight of i-th device.

Given the above formulation and analysis, we believe bMOM is robust against this setting due to the statistical nature of the classical median of means. As with every iteration in the learning process, we sample a list of blocks from the worker nodes. When the number of malicious worker nodes is sufficiently small than the number of blocks, this randomness will nullify the effect caused by malicious worker nodes. As we take the median of a list of empirical risk computed on each cluster, the block that malicious worker node is naturally either larger or small than the median empirical risk of normal block and will be discarded. In the experiment section, this feature of K-bMOM is proved empirically.

4.2.3 Algorithm

To address the bi-level optimization problem above, we proposed an Robust Clustered FL algorithm called FedRoC. FedRoC starts with K initial model parameters. To initialize FedRoC, firstly we do $Bootstrap(B, n_B)$ to sample $n_B > K$ devices with replacement randomly and uniformly for B times to get B blocks $1, \ldots, B$. Then k-means++ initialization [5] is proceeded for each block, and the empirical risks of B blocks are calculated. At last block with median risk and its centroids are got.

For the iterative round, four steps will be performed. At first we still need to do $Bootstrap(B, n_B)$, and then we perform the EM algorithm and calculate the empirical risk for each block. The next step is to select the block which has the median clustering loss, and its centroids. And the last step is to perform local update of FL for each cluster in the selected block, and get the updated centroids. Then we iterate these four steps until convergence.

The pseudo code of FedRoC is shown in Algorithm 3.

4.3 Experiment

4.3.1 Training Setups

As a proof-of-concept scenario to demonstrate the effectiveness of the proposed method, we experimentally evaluate and analyze the proposed FeRobust on federated benchmarks dataset(Caldas et al. 2018).

Dataset We employed three publicly-available federated benchmarks datasets introduced in LEAF [19], which is a benchmarking framework for learning in federated

Algorithm 3: FedRoC

Input: $\{D_1, D_2, \ldots, D_m\}, K$
Output: $r_{i,k}, C^{(b_{med})}$
Initialize:
$Bootstrap(B, n_B)$
for *blocks b from 1 to B:* **do**
  K-means++ initialization
  Calculate the empirical risk
end
Select the block b_{med} get initialized centroids $\{c_1^{(b_{med}),0}, \ldots, c_K^{(b_{med}),0}\}$.
Iterate:
while *stop condition is not satisfied* **do**
  $Bootstrap(B, n_B)$
  for *blocks b from 1 to B:* **do**
    E-Step:
    Assign each device in b to its closest centroid using updated centroids
    M-Step:
    Recompute the centroids
    Calculate the empirical risk
  end
  Select the block b_{med} and get centroids $\{c_1^{(b_{med}),t}, \ldots, c_K^{(b_{med}),t}\}$
  Federated Learning-Step:
  for *each cluster* $k = 1, \ldots K$ *in* b_{med} **do**
    Assign $c_k^{(b_{med})}$ to every device in Cluster k. **for** $i \in C_k$ **do**
      for E *local epochs* **do**
        $c_k^{(b_{med}),t+1} \leftarrow c_k^{(b_{med}),t} - \eta \nabla \mathscr{L}(c_k^{(b_{med}),t}, \mathscr{M}_k, D_i)$
      end
    end
  end
end

settings. The datasets used are Federeated Extended MNIST (FEMNIST)[1] [28] and Federated CelebA (FedCelebA)[2] [76], and finally Synthetic dataset which inspired by [72]. We follow the data processing instructions from its official repository. In FEMNIST, the handwritten images are splited according to the writers. For FedCelebA, the face images are extracted for each person and developed an on-device classifier to recognize whether the person smiles or not. For Synthetic, the dataset is generated with 1000 nodes and five classes. A statistical description of the datasets is described in Table 4.1.

[1] http://www.nist.gov/itl/products-and-services/emnist-dataset
[2] `http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`

Table 4.1: Statistics of datasets. "# of inst. per dev." represents the average number of instances per device.

DATASET	SYNTHETIC	FEMNIST	FEDCELEBA
# Data points	107553	805,263	200,288
Model	LOG-REG	CNN	CNN
Classes	5	62	2
# of device	1000	3,550	9,343
LR	0.01	0.003	0.1
Epochs	10	5	10

Local Model We use a CNN with the same architecture from [76] for two image classification datasets and multi-class logistic regression for a Synthetic dataset. Two data partition strategies are used: (a) an ideal IID data distribution using randomly shuffled data, (b) a non-IID partition by use a $\mathbf{p}_k \sim Dir_J(0.5)$. Part of the code is adopted from [106]. For FEMINST data, the local model's learning rate is 0.003, and the local epoch is 5. For FedCelebA, the learning rate is 0.1, and the local epochs are 10.

Baselines

1. **NonFed**: We will conduct the supervised learning task at each device without the FL framework.

2. **FeSEM**: A clustered FL method that clusters clients by considering the distance between their model parameters. It uses stochasti EM as the optimization algorithm. [108]

3. **FedAvg**: The vanilla FL method [83] proposed by Google in 2017. It is an SGD-based FL with weighted averaging.

4. **FedCluster**: A clustered FL method that is to enclose FedAvg into a hierarchical clustering framework [90].

5. **HypCluster(K)**: A clustered FL method that measure distance using the performance of each client's model, namely hypothesis-based clustered FL [81].

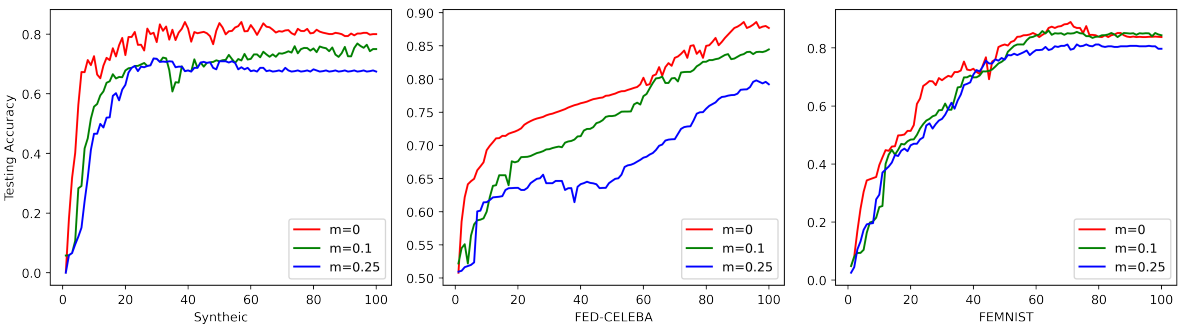
6. **FedRoC** Our proposed algorithm that is robust clustered FL algorithm using bootstrap median-of-mean to tackle outliers in clustering.

Training settings. We used 80% of each device's data for training and 20% for testing. For the initialization of the cluster centers in FeSEM, we conducted pure clustering 20 times with randomized initialization, and then the "best" initialization, which has the minimal intra-cluster distance, was selected as the initial centers for FeSEM. For the local update procedure of FeSEM, we set N to 1, meaning we only updated ω_i once in each local update.

Evaluation metrics. Given each global model of a cluster perform differently across numerous devices of a cluster, we evaluated the overall performance of the FL methods. We used classification accuracy and F1 score as the metrics for the two benchmarks. In addition, due to the multiple devices involved, we explored two ways to calculate the metrics, i.e., micro and macro. The only difference is that when computing an overall metric, "micro" calculates a weighted average of the metrics from devices where the weight is proportional to the data amount, while "macro" directly calculates an average over the metrics from devices.

Local Personalisation When multiple global models have been trained using multiple client groups, we can discuss the local personalization in three categories. 1) For the non-outlier clients who participated in the training, they don't need to conduct local personalization and use the global model directly. 2) For the outlier clients who participated in the training, they can conduct local updates. 3) For the unseen clients in the training stage, they need to download all global models to get the best performed one. Then, they can conduct the local update for a few steps and calculate the distance between the local model and the global model. If the distance is bigger than a threshold, then use the local model; otherwise, use the global model.

Figure 4.1: Convergence Analysis from Benchmarks with Model Poisoning Attack



4.3.2 Experiment Analysis

Convergence To verify the convergence of the proposed approach, we conducted a convergence analysis by running FedRoC with different cluster numbers K (from 2 to 5) in 100 iterations by the same set of other hyperparameters. As shown in Fig. 4.1, robust clustered Federated Learning can efficiently converge on all datasets, and results show that the best performance can be achieved with the cluster number $K = 5$. In this figure, we show the testing accuracy against the number of iterations on three datasets. The Red line shows the FeRobust with no model poisoning attack. The green line shows the FedRoC with 10% of workers are Byzantine nodes, and the Blue line shows the FeRobust with 25% of workers are Byzantine node. The figure display that the testing accuracy of FedRoC dropped by varied of 3.0-11.2 in Synthethic, Femnist and Celeba, while the testing accuracy on Celeba decreases the most.

Table 4.2: FeSEM v.s. FedRoC

Dataset	Approach	No Attak	m = 0.1	m= 0.25
Synthetic	FeSEM	2.8 ± 1.6	3.6 ± 2.2	3.9 ± 2.0
	FedRoC	1.5 ± 0.6	1.8 ± 0.8	1.0 ± 0.8
FEMNIST	FeSEM	3.0 ± 0.2	4.1±2.4	4.9 ± 2.0
	FedRoC	1.2 ± 0.2	1.1 ± 0.6	1.1 ± 0.8
Celeba	FeSEM	2.4 ± 0.2	3.2 ± 2.0	3.5 ± 2.1
	FedRoC	0.7 ± 0.5	0.7 ± 0.7	0.9 ± 0.7

Comparison Study We report the experiment of classification on three datasets and start training a global model with/without Byzantine nodes. There are m=0.05 of total clients as Byzantine nodes. Unsurprisingly, the average convergence rate without Byzantine nodes is faster than FedRoC, even without Byzantine nodes. However, figure on the right, we report the case training a global model with Byzantine nodes. Each Byzantine node estimates an update on their auxiliary datasets and before sending it to the server, scaled by a large factor (set to the number of total workers that have sampled to train at the current round), note that in FeSEM and Fedavg, each worker, including Byzantine worker is selected uniformly at the beginning of each round. The figure displays that those aggregation rules operated by other baseline methods do not tolerate any Byzantine nodes presence, while FedRoC only suffers an insignificant performance drop when there are 25% of Byzantine nodes. It also displays that with higher m Byzantine nodes, the further decrease of other algorithms but FedRoC stands the same performance. It is worth mentioning that according to the result of 4.3, FedRoC does

not achieve the state-of-the-art performance among other clustered federated learning methods. However, the better "mean operator" enables FedRoC to be an effective and resilient method against model poisoning attacks. Our empirical result is aligned with the property of FedRoC, and the average similarity measure of each cluster with its most similar cluster in 4.2 supports this hypothesis.

Table 4.3: Comparison of our proposed FedRoC(K) algorithm with the baselines on FEMNIST and FedCelebA datasets. Note the number in parenthesis denotes the number of clusters, K.

Datasets	FEMNIST				FedCelebA			
Metrics(%)	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1
NoFed	79.0	67.6	81.3	51.0	83.8	66.0	83.9	67.2
FeSEM	90.3	70.6	91.0	53.4	93	74.8	94.1	69.4
FedAvg	84.9	67.9	84.9	45.4	86.1	**78.0**	86.1	54.2
FedCluster	84.1	64.3	84.2	64.4	86.7	67.8	87.0	67.8
HypCluster	77.7	60.9	74.2	62.4	77.6	55.4	80.4	55.2
FedRoC(5)	88.6	69.3	86.3	62.2	87.2	72.7	90.1	68.3

Clustering case study As a case study, a figure displays that plots all nodes and their assignment in training. The highlight of this table is that, as expected, every iteration of the block, which is the median block among others, shows different risks to those blocks has Byzantine nodes. Those blocks show very different statistical properties, and their gradient updates will not take into the global model. It also shows a successful dense against model poisoning attacks.

4.4 Conclusion and Remarks

In this chapter, we propose a enhanced of existing multi-center FL approach to offset the adversarial worker nodes using the K-bMOM estimator. This algorithm has better breakdown point to address outliers and converges fast. In experiments based on three datasets, the proposed algorithm has similar performance to other baselines while showing much superior clustering performance than baseline methods in all three datasets. We also discussed possible extensions of FedRoC for better distance computation if the local model is a neural network.

CHAPTER 5

PERSONALIZED FEDERATED LEARNING WITH LOF AGAINST MODEL POISONING

5.1 Introduction

Federated Learning (FL) [83] first proposed in 2017 is widely-used to protect clients' data privacy in distributed applications recently, such as Google's Gboard on Android [83], Apple's Siri [37], Computer Visions [54, 59, 79], Smart Cities [120] and Healthcare [77, 88, 109]. The classical FL method, called FedAvg [83], is to train a global model across all clients using gradients to communicate efficiently and privately. Vanilla FL is apparently vulnerable to model poisoning attacks due to its decentralized nature. Thus, it is challenging to develop a Federated Learning application that has a good personalized decision-making ability while being robust against model poisoning attacks.

The non-IID challenge is also proposed that can lower the training performance in both accuracy and efficiency. It indicates that the data distribution of each client can be different due to unique attributes or behaviour, thus a globally shared model may not generalize well and fairly in all clients. Personalized FL (PFL) is the most popular method to address this challenge. Based on granularity, PFL can be categorized into cluster-wise PFL and client-wise PFL. PFL methods, such as Ditto [68] and WeCFL [80], train multiple models client-wisely or cluster-wisely to adapt to each client or cluster better, while knowledge is still shared to improve the performance.

Model poisoning is another challenge in realistic FL. In a distributed system of FL, some malicious agents may upload fake or dirty gradients to the server in the aggregation step, and then the aggregated model to distribute is poisoned. It is naive to adopt anomaly detection techniques to find these malicious agents or outliers. Local outlier factor (LOF) [16] is an efficient method based on the density of data points.

To tackle these two challenges above at the same time, it is difficult to embed the anomaly detection technique into the PFL. We constructed a nested bi-level optimization problem to combine client-wise PFL, cluster-wise PFL, and anomaly detection together. An algorithm of personalized FL with robust clustering (FedPRC) is proposed to detect outliers and keep the state-of-the-art performance. Our contributions are summarized as below.

- We formulate the PFL problem with robust clustering into a nested bi-level optimization framework.
- We propose a novel PFL with robust clustering (FedPRC) algorithm to solve the complex optimization problem, and the algorithm can resist Byzantine workers.
- The experimental analysis demonstrates the effectiveness and superior performance in comparison with baselines in multiple benchmark datasets.

The remaining sections of the thesis are organized as follows. We will formulate the problem of PFL with robust clustering in Section 3.3. Then the FedPRC algorithm is proposed in Section 5.4. Experimental settings and empirical study are discussed in Section 5.5.1 and 5.5.2, respectively.

5.2 Motivation

5.2.1 Model poisoning and anomaly detection

The way malicious agent generates an arbitrary update vector by merely shuffle data labels sounds very similar to the standard dirty-label poisoning in [25]. However, in Federated Learning setting, the possibility of a an adversary controlling a small number of malicious agents, perform a model poisoning attack to manipulate the learning process so that the jointly trained global model which turn into misclassifiation over some data is much higher. FL is apparently vulnerable to model poisoning attacks due to its decentralized nature. A line of work has been done already [7] [10] [32]. In contrast to

previous work, this work focus to detect these malicious agents during central clustering phase by applying density method then reduce the impact of those agents' updates to the aggregation of the cluster center.

Anomaly detection can be described the problem of finding patterns in data that do not confirm to expected behavior. Of these, anomalies and outliers are two terms used most commonly in the context of anomaly detection. Clustering can be used as a technique for training of the normality model, where similar data points are grouped together into clusters using a distance function, for example [84]. While LOF [16] is a widely-used density-based anomaly detection method. However, in the case of our method, we already know malicious agents are the anomalies that we tried to identify. The outcome after preclude those identified outliers would be benigh agents, then only the benign agents weight matrix feed into our clustering algorithm. The identifying outliers stage has no inherit relation to next clustering phase.

5.2.2 Problem

Federated learning (FL) usually aggregates all local models to a single global model. However, this single-center aggregation is fragile under heterogeneity. In contrast, we consider FL with multiple centers to better capture the heterogeneity by assigning nodes to different centers so only similar local models are aggregated. Consider two extreme cases for the number of centers, K: (i) when $K = 1$, it reduces to the FedAvg with a single global model, which cannot capture the heterogeneity and the global model might perform poorly on specific nodes; (ii) When $K = m$, the heterogeneity problem can be avoided by assigning each node to one global model. But the data on each device used to update each global model can be insufficient and thus we lose the main advantage of FL. Our goal is to find a sweet point between these two cases to balance the advantages of federated averaging and the degradation caused by underlying heterogeneity.

Learning one unique model for each node has been discussed in some recent FL studies for better individual level personalized models. They focus on making a trade-off between shared knowledge and professionalisation. The personalising strategy either applies fine-tuning of the global model [119] for each node, or only updates a subset of personalized layers for each node [4, 74], or deploys a regularisation term in the objective [35, 38, 52]. These personalization is tightly integrated with the model aggregation procedure. In contrast, we propose a light-weight personalization solution by simply conduct a limite4d number of local updating.

5.3 Methodology

Before the methodology, the notations are list below, which can be separated into three parts, FL, clustering and LOF.

Table 5.1: Table of Notations

Components	Notation	Definition
FL	m	Number of clients in FL system
	$D_i, \lvert D_i \rvert$	The dataset and its size on Client i
	$\mathscr{M}_i$	Model function or structure of Client i
	ω_i	Model parameters of Client i
	$\mathscr{L}_i$	Loss function of Client i
	λ_i	The importance weight of Client i, usually measured by its dataset size
	E	Number of local update steps
Clustering	K	Number of clusters
	$r_{i,k} \in \mathbb{R}^{m*K}$	The assignment matrix, $r_{i,k} = 1$ if $i \in k$ else $r_{i,k} = 0$
	$i \in k$	Client i belongs to Cluster k
	g_i	General form to represent Client i depending on h_i, l_i, D_i or something else, e.g. model parameters or loss
	G_k	General form to represent the centroid of Cluster k, and usually a linear combination of g_i with $i \in k$
	$d(g_i, G_k)$	The distance function of general representations between Client i and the center of Cluster k, e.g. Euclidean distance.
LOF	n	Number of neighbours
	c_i	Indicator. 1 if Client i belongs to inliers else 0.

5.3.1 PFL

For the classical FL problem, the objective can be formulated as below,

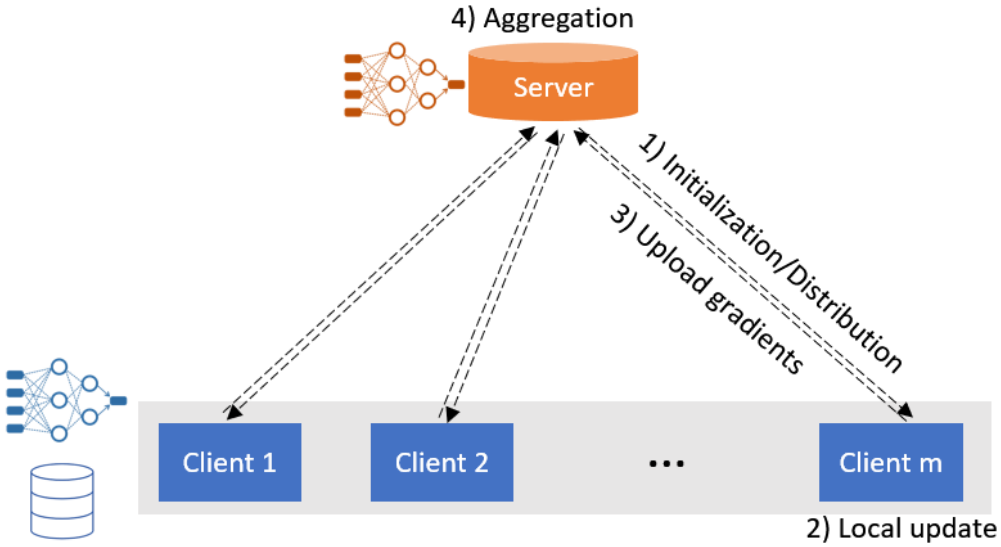
$$\underset{\omega}{\text{minimize}} \sum_{i=1}^{m} \lambda_i \mathscr{L}(\mathscr{M}, D_i, \omega) \tag{5.1}$$

And the framework is shown in Figure 5.1. The algorithm FedAvg [83] is also implied in this figure, which can be summarized as four steps, model initialization or distribution from server to clients, local update on clients, gradients upload from clients to the server and model aggregation on the sever.

For the client-wise PFL problem, its objective can be formulated as below,

$$\underset{\{\omega_i\}}{\text{minimize}} \sum_{i=1}^{m} \lambda_i \mathscr{L}_i(\mathscr{M}_i, D_i, \omega_i) \tag{5.2}$$

Figure 5.1: Framework of classical FL



which means an arbitrary client i may have its importance λ_i, unique dataset D_i, model structure $\mathscr{M}_i$, model parameters ω_i, and loss function $\mathscr{L}_i$.

5.3.2 LOF

To understand the LOF [16], a density-based anomaly detection method, there are five key definitions step by step. Firstly n-d of an object o is defined as the distance $d(o,p)$ between o and $p \in D$ which satisfies:

- There are at least n objects $o' \in D|\{o\}$, which holds $d(o,o') \leq d(o,p)$, and
- There are at most $n-1$ objects $o' \in D|\{o\}$, which holds $d(o,o') < d(o,p)$.

Then the n-d neighborhood of an object o can be defined as:

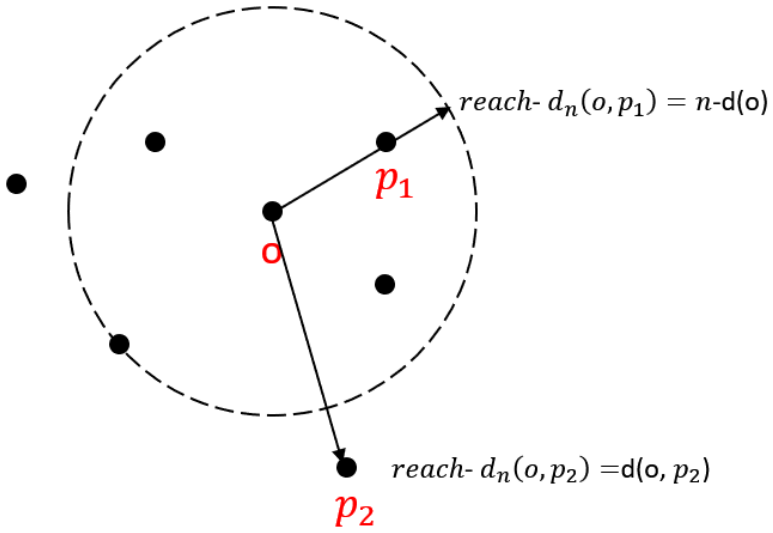
$$N_{n\text{-}d(o)}(o) = \{q \in D|\{o\} \mid d(o,q) \leq n\text{-}d(o)\} \tag{5.3}$$

Thirdly, the reachability distance of an object p w.r.t. object o is defined as:

$$reach\text{-}d_n(o,p) = max\{n\text{-}d(o), d(o,p)\} \tag{5.4}$$

As shown in Figure 5.2, the reachability distance of o,p_1 and o,p_2 equals n-$d(o)$ and $d(o,p_2)$, respectively.

Figure 5.2: Reachability distance of o, p_1 and o, p_2, respectively, for $n = 5$



Then the local reachability density (lrd) of object o is defined as:

$$lrd_n(o) = \frac{|N_n(o)|}{\sum_{p \in N_n(o)} reach\text{-}d_n(o,p)} \tag{5.5}$$

Finally the LOF of object o is defined as:

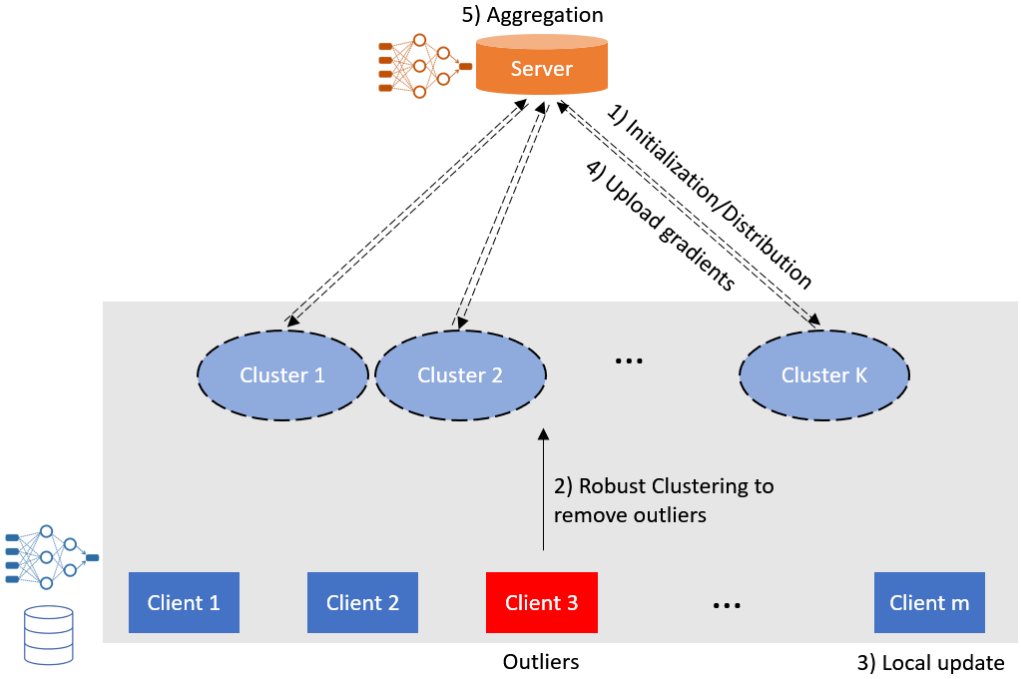
$$LOF_n(o) = \frac{\sum_{p \in N_n(o)} \frac{lrd_n(p)}{lrd_n(o)}}{|N_n(o)|} \tag{5.6}$$

To judge whether an object belongs to outliers, usually yes if its $LOF > 1$, which means it has lower density than neighbours, thus an outlier. With proper n chosen, the breakdown point for LOF can be at least 0.5, which means unless the malicious clients being the majority and behaviour similarly, LOF will always works.

5.3.3 Proposed method

For our proposed method of personalized FL with robust clustering structure to attack the model poisoning, its framework is illustrated in Figure 5.3. And its optimization objective can be formulated into the below equation which is like a nested bi-level optimization problem.

Figure 5.3: Framework of proposed method



$$\underset{\{\omega_i\}}{\text{minimize}} \ \frac{1}{m} \sum_{k=1}^{K} \sum_{i=1}^{m} \lambda_i r_{i,k} c_i \mathscr{L}(\mathscr{M}, D_i, \omega_i) \tag{5.7a}$$

$$\textit{subject to } \{r_{i,k}\} = \underset{\{r_{i,k}\}}{\text{argmin}} \sum_{k=1}^{K} \sum_{i=1}^{m} \lambda_i r_{i,k} c_i d(g_i, G_k) \tag{5.7b}$$

$$\{c_i\} = I_{LOF_n(g_i)>1} \tag{5.7c}$$

5.4 Algorithm

To solve the complex Objective 5.7 above, which has three variables, Ω as the ultimate variable, and R and C as the hidden variables, we need to carefully design an algorithm to solve them step by step. Thus Algorithm 4 named Personalized FL with robust clustering (FedPRC) is proposed as below.

For the initialization, K-means++ [5] is used to set up a more robust initial for the clustering. For the iteration process, it can be merged by two modules, robust clustering and FL. The Robust clustering module is composed of three steps, the Expectation step (E step), the LOF step, and the Maximization step (M step). And the FL module is composed of three steps either, the Distribution step, Local update step and the Aggregation step.

Algorithm 4: Personalized FL with robust clustering (FedPRC)

Input: $\{D_1, D_2, \ldots, D_m\}, K, n$
Output: $\{r_{i,k}\}, \{c_i\}, \{\omega_i\}$
Initialize:
K-means++ initialization
Iterate:
while *stop condition is not satisfied* **do**
  E-Step:
  Assign each device in b to its closest centroid using updated centroids
  LOF-Step:
  Use LOF_n to label outliers
  M/Aggregation-Step:
  Recompute the centroids with inliers.
  Local update-Step:
  for *each cluster* $k = 1, \ldots K$ **do**
    Assign centroids to every device in Cluster k.
    for $i \in C_k$ **do**
      for E *local epochs* **do**
        $\omega_i^{t+1} \leftarrow \omega_i^t - \eta \nabla \mathscr{L}(\mathscr{M}, D_i, \omega_i^t)$
      end
    end
  end
end
End:
Fine tuning-Step
Fine tuning ω_i for E' epochs.

Due to the M step in robust clustering being the same as the Aggregation step in FL, these two modules can be merged together to form the iteration process. Until convergence or stop condition is satisfied, the output is K models for K clusters. To achieve better performance for each client, a simple but effective personalization technique called fine-tuning is imported as the optimum of one cluster is not the optimum of its clients. Finally, we can get m personalized models with robustness against model poisoning for every client.

5.5 Experiments

As a proof-of-concept scenario to demonstrate the effectiveness of the proposed method, we experimentally evaluate and analyze the proposed FedPRC based on the LEAF framework, a FL benchmark [19].

DATASET	FEMNIST	CelebA
# of Data	805,263	200,288
Classes	62	2
# of device	3,550	9,343
Model	CNN	CNN
LR	0.003	0.1
Local Epochs	5	10

Table 5.2: Statistics of datasets. "#" represents the number of instances.

5.5.1 Experimental settings

Datasets. We employed two publicly-available federated benchmarks datasets introduced in LEAF [19]. LEAF is a benchmarking framework for learning in federated settings. The datasets used are Federeated Extended MNIST (FEMNIST)[1] [28] and Federated CelebA (FedCelebA)[2] [76]. We follow the setting of the benchmark data in LEAF. In FEMNIST, the handwritten images is split according to the writers. For FedCelebA, the face images are extracted for each person and developed an on-device classifier to recognize whether the person smiles or not. A statistical description of the datasets is described in Table 5.2.

Local model We use a CNN with the same architecture from [76]. Two data partition strategies are used: (a) an ideal IID data distribution using randomly shuffled data, (b) a non-IID partition by use a $\mathbf{p}_k \sim Dir_J(0.5)$. Part of the code is adopted from [106]. For FEMINST data, the local model's learning rate is 0.003 and epoch is 5. For FedCelebA, the learning rate is 0.1 and the epochs is 10.

outliers In this work, we evaluate the proposed method using the outliers generated from a poisoning attack tool. The idea of model poisoning adopts from Krum [10] which is simply boosting each iteration of the learned model in some worker node. Malicious clients assign wrong labels to each samples in local dataset. In other words, explicit boosting works is to mimic the benign worker clients during the learning process; the client tries to perform the same number of epochs on the local dataset via the same training objectives to obtain an initial gradient update. Since the malicious client wants to ensure the outcome deviates from the true label, it will have to overcome the scaling

[1] http://www.nist.gov/itl/products-and-services/emnist-dataset

[2] `http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`

effect of gradient updates collected from other nodes. In other words, the final gradient updates the malicious nodes send back are then scaled a factor Λ by which the malicious nodes boost the initial update. The Λ here is a hyper-parameter which is a multiplier for malicious clients which used to force the trained global model to close its direction. Here we use the number of clients of a subset each iteration then times two as Λ.

Baselines In the scenario of solving statistical heterogeneity, we choose FL methods as follows:

1. **NonFed**: We will conduct the supervised learning task at each device without the FL framework.

2. **FedSGD**: uses SGD to optimise the global model.

3. **FedAvg**: is an SGD-based FL with weighted averaging. [83] .

4. **FedCluster**: is to enclose FedAvg into a hierarchical clustering framework [90].

5. **HypoCluster(K)**: is a hypothesis-based clustered-FL algorithm with different K [81].

6. **Robust** design a framework run in a modular manner, namely, a robust clustering model, and a communication efficient, distributed, robust optimization over each cluster separately [47].

7. **FedDANE**: is an FL framework with a Newton-type optimization method. [71].

8. **FedProx**: adds a proximal term onto an objective function of the learning task on the device [70].

9. **FedDist**: we adapt a distance based-objective function in Reptile meta-learning [85] to a federated setting.

10. **FedDWS**: a variation of FedDist by changing the aggregation to weighted averaging where the weight depends on the data size of each device.

11. **FedPRC**(K): our proposed algorithm FedPRC with different numbers of clusters K.

Training settings We used 80% of each device's data for training and 20% for testing. For the initialization of the cluster centers in FedPRC, we conducted pure clustering 10 times with randomized initialization over the gradients matrix where is computed by each client conduct 1 epoch local training, and then the "best" initialization, which has the minimal intra-cluster distance, was selected as the initial centers for FedPRC. For the local update procedure of FedPRC, we set N to 1, meaning we only updated W_i once in each local update.

Evaluation metrics. Given numerous devices, we evaluated the overall performance of the FL methods. We used classification accuracy and F1 score as the metrics for the two benchmarks. In addition, due to the multiple devices involved, we explored two ways to calculate the metrics, i.e., micro and macro. The only difference is that when computing an overall metric, "micro" calculates a weighted average of the metrics from devices where the weight is proportional to the data amount, while "macro" directly calculates an average over the metrics from devices.

5.5.2 Experimental study

Comparison study As report in Table 5.3, we compared our proposed FedPRC with the baselines and found that our proposed FL framework achieves the best performance in most cases. We can see our proposed FedPRC outperforms all baselins in all metrics, which shows the effectiveness and significance of FedPRC. Furthermore, as report in the last three columns in Table 5.3, we found that FedPRC with a larger number of clusters empirically achieves a better performance, which verifies the correctness of the non-IID assumption of the data distribution. Due to the experiment on both dataset is very consuming, we use grid search technique for the number of clusters and only run full experiment with those values, i.e. two-four.

Convergence analysis To verify the convergence of the proposed approach, we conducted a convergence analysis by running FedPRC with different cluster numbers K (from 2 to 4) in 100 iterations. As shown in Fig. 5.4, FedPRC can efficiently converge on both datasets and it can achieve the best performance with the cluster number $K = 4$. The last step is fine-tuning.

Datasets	FEMNIST				CelebA			
Metrics(%)	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1	Micro-Acc	Micro-F1	Macro-Acc	Macro-F1
NoFed	79.4	67.6	81.3	51.0	83.8	66.0	83.9	67.2
FedSGD	70.1	61.2	71.5	46.7	75.7	60.7	75.6	55.6
FedAvg	84.9	67.9	84.9	45.4	86.1	78.0	86.1	54.2
FedDist	79.3	67.5	79.8	50.5	71.8	61.0	71.6	61.1
FedDWS	80.4	67.2	80.6	51.7	73.4	59.3	73.4	50.3
Robust(TKM)	78.4	53.1	77.6	53.6	90.1	68.0	90.1	68.3
FedCluster	84.1	64.3	84.2	64.4	86.7	67.8	87.0	67.8
HypoCluster(3)	82.5	61.3	82.2	61.6	76.1	53.5	72.7	53.8
FedDane	40.0	31.8	41.7	31.7	76.6	61.8	75.9	62.1
FedProx	51.8	34.2	52.3	34.4	83.4	60.9	84.3	65.2
FedPRC(2)	91.3	64.9	91.7	64.1	93.8	77.2	94.1	71.5
FedPRC(3)	91.1	63.1	91.0	62.6	93.6	77.8	93.3	70.6
FedPRC(4)	**92.7**	**66.4**	**92.4**	**65.7**	**94.4**	**80.4**	**94.6**	**72.7**

Table 5.3: Comparison of our proposed FedPRC(K) algorithm with the baselines on FEMNIST and FedCelebA datasets. Note the number in parenthesis following "FedPRC" denotes the number of clusters, K.

5.6 Conclusion and Remarks

This chapter proposed a personalized FL method with robust clustered structure to tackle model poisoning attack in FL while still keep the state-of-the-art performance. It is novel to combine client-wise, cluster-wise PFL and robust clustering together to tackle the non-IID and model poisoning challenges in FL.

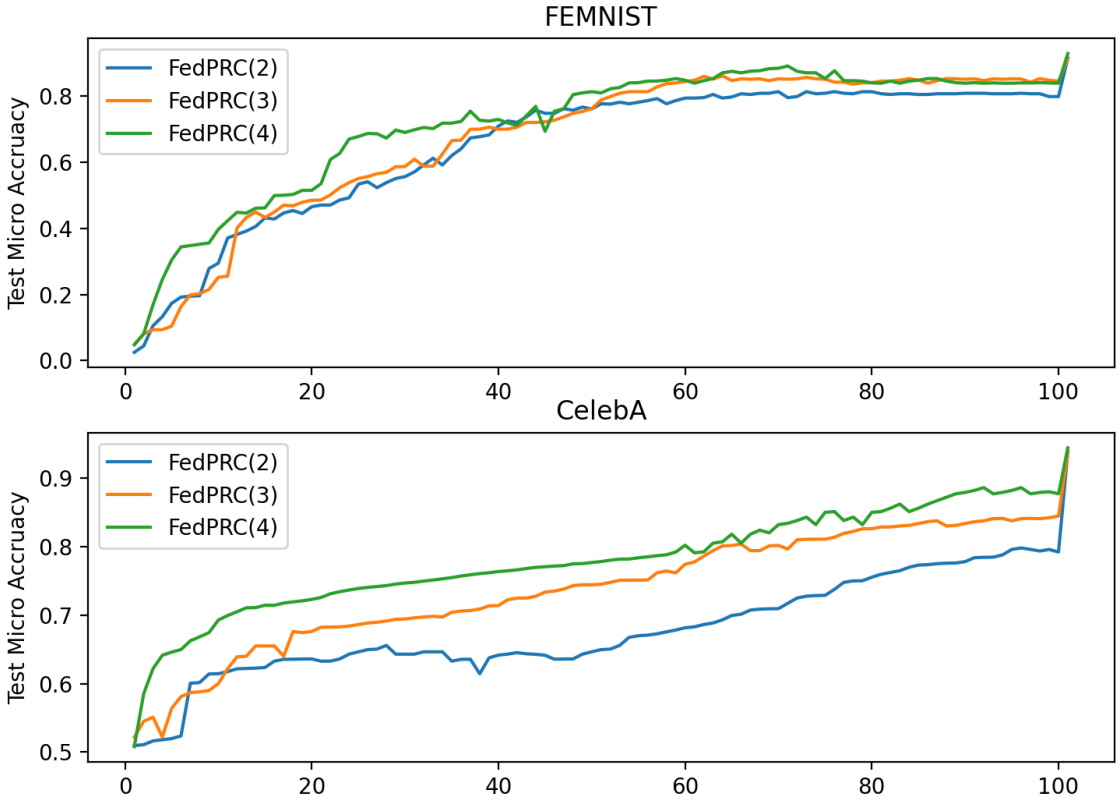


Figure 5.4: Convergence analysis for the proposed FedPRC with different cluster number (in parenthesis) in terms of micro-accuracy.

CHAPTER 6

CONCLUSION

I surely hope this work will help to make the clustering approaches, also know as multi-center framework over the whole Federated Learning network specifically, as well as the personlization models, secure models that against model poisoning attack in Federated learning more attractive for future research. The achieved results show that whole population could be partitioned into different clusters or groups in which the users have similar preferences. Therefore, learning an intelligent model customised for each group with non-IID data can be considered as competitive edge over vanilla FL. The main conclusion of this work are:

- propose a novel multi-center aggregation approach to address the non-IID challenge of personalized decision-making system.
- design an objective function, namely multi-center federated loss, for user clustering in FL.
- propose Federated Stochastic Expectation Maximization (FeSEM) to solve the optimization of the proposed objective function.
- present the algorithm as an easy-to-implement and strong baseline for FL. Its effectiveness is evaluated on benchmark datasets.
- introduce a robust version of K-means to extend previous multi-center Federated Learning approach.

- adopt bootstrap sampling during initialization together with a median-of-means estimator, which we prove theoretically enable faster convergence and robust to outliers and malicious node.

- present an algorithm as an easy-to-implement and strong baseline for FL, and present two different expansions for improved robust properties which target outliers/malicious nodes, namely FeROC and FedPRC. we compared and analyzed proposed approaches on a few datasets, the result shows its computationally competitive than FeSEM and more robust than any other clustering algorithms.

- we create a open-source repository for multi-center federated learning to prompt reproducible research.

6.1 GreenSAP in Federated Learning

Since the day of Federated Learning invented, much effort have been put into natural language modelling related research. Google's first commercial usage that success employ Federated Machine Learning is Gboard which is a mobile keyboard prediction program. In addition, While Google has launched several applications on langauge modeling tasks, FL solutions are also used in many Apple products such as Siri and Doc.ai. FL is a technology not only required investigation from machine learning area, but also techniques from distributed optimization, statistics, cybersecurity, communication, systems, cryptography and many more other disciplines. We believe that a number of NLP tasks can be study in the FL context.

For our future reference and audience who shared similar background, we suggest two possible idea can applied sarcasm detection model in Federated Learning setting. First, during Federated Learning training phase, all device,regardless their language and their graphic zones, will include in the population and treated indistinctly. The outcome of learning process is a global sarcasm language model where the model just setup a foundation (also known as pre-trained) for the deployed one. Then when client request to deployed model first time, the server will start a fine-tuning procedure according to user's features, then the final model in use on client device will be most mobile context-aware one. Secondly, we borrow the idea of meta-learning, which is similar to multi-task learning. Since it also learns how to perform a tasks by using the experience from other tasks. In Federated Learning setting, each new user can be viewed as a new task and the algorithm will uses the learning of a client to personalized the global model

which lead to a more comprehensive sarcasm language model.

In short, many NLP tasks can be studied or included in Federated Learning setting, we hope to extend our sentiment analysis model and build a practical sentiment analysis application in Federated Learning setting.

There is of course still much space for future improvements even in such a small research topic. The Federated Learning technology aims to overcome data regulation challenges that impede machine learning from getting widely implemented. To help with Federated Learning to reach its expected potential, the following systematic issues need to be addressed, optimize local computing resource usage, asynchronous update and distributed optimization. Among these, future research that would help might be:

- Explore different loss function in detail in the context of Federated Learning
- Identify a few possible application scenario for Federated Learning
- Explore different representation as clustering features for similarity measure
- Communication efficiency and asynchronous architecture

BIBLIOGRAPHY

[1] S. ABDULRAHMAN, H. TOUT, H. OULD-SLIMANE, A. MOURAD, C. TALHI, AND M. GUIZANI, *A survey on federated learning: The journey from centralized to distributed on-site learning and beyond*, IEEE Internet of Things Journal, 8 (2020), pp. 5476–5497.

[2] L. F. ANA AND A. K. JAIN, *Robust data clustering*, in 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings., vol. 2, IEEE, 2003, pp. II–II.

[3] A. ARGYRIOU, T. EVGENIOU, AND M. PONTIL, *Convex multi-task feature learning*, Machine Learning, 73 (2008), pp. 243–272.

[4] M. G. ARIVAZHAGAN, V. AGGARWAL, A. K. SINGH, AND S. CHOUDHARY, *Federated learning with personalization layers*, arXiv preprint arXiv:1912.00818, (2019).

[5] D. ARTHUR AND S. VASSILVITSKII, *k-means++: The advantages of careful seeding*, tech. rep., Stanford, 2006.

[6] E. ASCARZA, *Retention futility: Targeting high-risk customers might be ineffective*, Journal of Marketing Research, 55 (2018), pp. 80–98.

[7] A. N. BHAGOJI, S. CHAKRABORTY, P. MITTAL, AND S. CALO, *Analyzing federated learning through an adversarial lens*, in International Conference on Machine Learning, PMLR, 2019, pp. 634–643.

[8] G. BIAU, L. DEVROYE, AND G. LUGOSI, *On the performance of clustering in hilbert spaces*, IEEE Transactions on Information Theory, 54 (2008), pp. 781–790.

[9] C. M. BISHOP, *Pattern recognition and machine learning*, springer, 2006.

[10] P. BLANCHARD, E. M. EL MHAMDI, R. GUERRAOUI, AND J. STAINER, *Machine learning with adversaries: Byzantine tolerant gradient descent*, Advances in Neural Information Processing Systems, 30 (2017).

[11] P. BLANCHARD, E. M. E. MHAMDI, R. GUERRAOUI, AND J. STAINER, *Byzantine-tolerant machine learning*, arXiv preprint arXiv:1703.02757, (2017).

[12] D. M. BLEI, A. KUCUKELBIR, AND J. D. MCAULIFFE, *Variational inference: A review for statisticians*, Journal of the American statistical Association, 112 (2017), pp. 859–877.

[13] D. M. BLEI, A. Y. NG, AND M. I. JORDAN, *Latent dirichlet allocation*, Journal of Machine Learning Research, 3 (2003), pp. 993–1022.

[14] K. BONAWITZ, V. IVANOV, B. KREUTER, A. MARCEDONE, H. BRENDAN MCMAHAN, S. PATEL, D. RAMAGE, A. SEGAL, AND K. SETH, *Practical secure aggregation for federated learning on user-held data*, arXiv e-prints, (2016), pp. arXiv–1611.

[15] K. BONAWITZ, V. IVANOV, B. KREUTER, A. MARCEDONE, H. B. MCMAHAN, S. PATEL, D. RAMAGE, A. SEGAL, AND K. SETH, *Practical secure aggregation for privacy-preserving machine learning*, in proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1175–1191.

[16] M. M. BREUNIG, H.-P. KRIEGEL, R. T. NG, AND J. SANDER, *Lof: identifying density-based local outliers*, in Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104.

[17] C. BRIGGS, Z. FAN, AND P. ANDRAS, *Federated learning with hierarchical clustering of local updates to improve training on non-iid data*, in 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–9.

[18] C. BRUNET-SAUMARD, E. GENETAY, AND A. SAUMARD, *K-bmom: A robust lloyd-type clustering algorithm based on bootstrap median-of-means*, Computational Statistics & Data Analysis, 167 (2022), p. 107370.

[19] S. CALDAS, P. WU, T. LI, J. KONEČNỲ, H. B. MCMAHAN, V. SMITH, AND A. TALWALKAR, *Leaf: A benchmark for federated settings*, arXiv preprint arXiv:1812.01097, (2018).

[20] O. CAPPÉ AND E. MOULINES, *On-line expectation–maximization algorithm for latent data models*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 71 (2009), pp. 593–613.

[21] K. S. CHAHAL, M. S. GROVER, K. DEY, AND R. R. SHAH, *A hitchhiker's guide on distributed training of deep neural networks*, Journal of Parallel and Distributed Computing, 137 (2020), pp. 65–76.

[22] P. CHAUDHARI, C. BALDASSI, R. ZECCHINA, S. SOATTO, A. TALWALKAR, AND A. OBERMAN, *Parle: parallelizing stochastic gradient descent*, arXiv preprint arXiv:1707.00424, (2017).

[23] F. CHEN, G. LONG, Z. WU, T. ZHOU, AND J. JIANG, *Personalized federated learning with graph*, arXiv preprint arXiv:2203.00829, (2022).

[24] F. CHEN, M. LUO, Z. DONG, Z. LI, AND X. HE, *Federated meta-learning with fast convergence and efficient communication*, arXiv preprint arXiv:1802.07876, (2018).

[25] X. CHEN, C. LIU, B. LI, K. LU, AND D. SONG, *Targeted backdoor attacks on deep learning systems using data poisoning*, arXiv preprint arXiv:1712.05526, (2017).

[26] G. CHENG, K. CHADHA, AND J. DUCHI, *Fine-tuning is fine in federated learning*, arXiv preprint arXiv:2108.07313, (2021).

[27] D. CIREGAN, U. MEIER, AND J. SCHMIDHUBER, *Multi-column deep neural networks for image classification*, in 2012 IEEE conference on computer vision and pattern recognition, IEEE, 2012, pp. 3642–3649.

[28] G. COHEN, S. AFSHAR, J. TAPSON, AND A. VAN SCHAIK, *Emnist: Extending mnist to handwritten letters*, in 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2921–2926.

[29] L. COLLINS, H. HASSANI, A. MOKHTARI, AND S. SHAKKOTTAI, *Exploiting shared representations for personalized federated learning*, in International Conference on Machine Learning, PMLR, 2021, pp. 2089–2099.

[30] L. CORINZIA AND J. M. BUHMANN, *Variational federated multi-task learning*, 2019.

[31] J. A. CUESTA-ALBERTOS, A. GORDALIZA, AND C. MATRÁN, *Trimmed k-means: an attempt to robustify quantizers*, The Annals of Statistics, 25 (1997), pp. 553–576.

[32] G. DAMASKINOS, E.-M. EL-MHAMDI, R. GUERRAOUI, A. GUIRGUIS, AND S. ROUAULT, *Aggregathor: Byzantine machine learning via robust gradient aggregation*, Proceedings of Machine Learning and Systems, 1 (2019), pp. 81–106.

[33] R. N. DAVÉ AND R. KRISHNAPURAM, *Robust clustering methods: a unified view*, IEEE Transactions on fuzzy systems, 5 (1997), pp. 270–293.

[34] J. DEAN, G. CORRADO, R. MONGA, K. CHEN, M. DEVIN, M. MAO, M. RANZATO, A. SENIOR, P. TUCKER, K. YANG, ET AL., *Large scale distributed deep networks*, in Advances in neural information processing systems, 2012, pp. 1223–1231.

[35] Y. DENG, M. M. KAMANI, AND M. MAHDAVI, *Adaptive personalized federated learning*, arXiv preprint arXiv:2003.13461, (2020).

[36] A. DESHPANDE, P. KACHAM, AND R. PRATAP, *Robust k-means++*, in Conference on Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 799–808.

[37] A. DIFFERENTIAL PRIVACY TEAM, *Learning with privacy at scale*, 2017.

[38] C. T. DINH, N. H. TRAN, AND T. D. NGUYEN, *Personalized federated learning with moreau envelopes*, arXiv preprint arXiv:2006.08848, (2020).

[39] W. DU AND Z. ZHAN, *Building decision tree classifier on private data*, (2002).

[40] M. DUAN, D. LIU, X. JI, R. LIU, L. LIANG, X. CHEN, AND Y. TAN, *Fedgroup: Ternary cosine similarity-based clustered federated learning framework toward high accuracy in heterogeneous data*, arXiv preprint arXiv:2010.06870, (2020).

[41] A. FALLAH, A. MOKHTARI, AND A. OZDAGLAR, *Personalized federated learning: A meta-learning approach*, arXiv preprint arXiv:2002.07948, (2020).

[42] M. A. FERRAG, O. FRIHA, L. MAGLARAS, H. JANICKE, AND L. SHU, *Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis*, IEEE Access, 9 (2021), pp. 138509–138542.

[43] L. A. GARCÍA-ESCUDERO, A. GORDALIZA, C. MATRÁN, AND A. MAYO-ISCAR, *A general trimming approach to robust cluster analysis*, The Annals of Statistics, 36 (2008), pp. 1324–1345.

[44] L. A. GARCÍA-ESCUDERO, A. GORDALIZA, C. MATRÁN, AND A. MAYO-ISCAR, *A review of robust clustering methods*, Advances in Data Analysis and Classification, 4 (2010), pp. 89–109.

[45] ——, *Exploring the number of groups in robust model-based clustering*, Statistics and Computing, 21 (2011), pp. 585–599.

[46] A. GHOSH, J. CHUNG, D. YIN, AND K. RAMCHANDRAN, *An efficient framework for clustered federated learning*, arXiv preprint arXiv:2006.04088, (2020).

[47] A. GHOSH, J. HONG, D. YIN, AND K. RAMCHANDRAN, *Robust federated learning in a heterogeneous environment*, arXiv preprint arXiv:1906.06629, (2019).

[48] T. GRIFFITHS, *Gibbs sampling in the generative model of latent dirichlet allocation*, (2002).

[49] T. L. GRIFFITHS AND Z. GHAHRAMANI, *The indian buffet process: An introduction and review.*, Journal of Machine Learning Research, 12 (2011).

[50] S. GUHA, R. RASTOGI, AND K. SHIM, *Rock: A robust clustering algorithm for categorical attributes*, Information systems, 25 (2000), pp. 345–366.

[51] F. HADDADPOUR AND M. MAHDAVI, *On the convergence of local descent methods in federated learning*, arXiv preprint arXiv:1910.14425, (2019).

[52] F. HANZELY AND P. RICHTÁRIK, *Federated learning of a mixture of global and local models*, arXiv preprint arXiv:2002.05516, (2020).

[53] A. HARD, K. RAO, R. MATHEWS, S. RAMASWAMY, F. BEAUFAYS, S. AUGENSTEIN, H. EICHNER, C. KIDDON, AND D. RAMAGE, *Federated learning for mobile keyboard prediction*, arXiv preprint arXiv:1811.03604, (2018).

[54] C. HE, A. D. SHAH, Z. TANG, D. F. N. SIVASHUNMUGAM, K. BHOGARAJU, M. SHIMPI, L. SHEN, X. CHU, M. SOLTANOLKOTABI, AND S. AVESTIMEHR, *Fedcv: A federated learning framework for diverse computer vision tasks*, arXiv preprint arXiv:2111.11066, (2021).

[55] L. HE, S. P. KARIMIREDDY, AND M. JAGGI, *Secure byzantine-robust machine learning*, arXiv preprint arXiv:2006.04747, (2020).

[56] K. HSIEH, A. PHANISHAYEE, O. MUTLU, AND P. GIBBONS, *The non-iid data quagmire of decentralized machine learning*, in International Conference on Machine Learning, PMLR, 2020, pp. 4387–4398.

[57] T.-M. H. HSU, H. QI, AND M. BROWN, *Measuring the effects of non-identical data distribution for federated visual classification*, arXiv preprint arXiv:1909.06335, (2019).

[58] Y. HUANG, L. CHU, Z. ZHOU, L. WANG, J. LIU, J. PEI, AND Y. ZHANG, *Personalized cross-silo federated learning on non-iid data*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 7865–7873.

[59] D. JALLEPALLI, N. C. RAVIKUMAR, P. V. BADARINATH, S. UCHIL, AND M. A. SURESH, *Federated learning for object detection in autonomous vehicles*, in 2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService), IEEE, 2021, pp. 107–114.

[60] Y. JIANG, J. KONEČNỲ, K. RUSH, AND S. KANNAN, *Improving federated learning personalization via model agnostic meta learning*, arXiv preprint arXiv:1909.12488, (2019).

[61] J. M. JOHNSON AND T. M. KHOSHGOFTAAR, *The effects of data sampling with deep learning and highly imbalanced big data*, Information Systems Frontiers, 22 (2020), pp. 1113–1131.

[62] P. KAIROUZ, H. B. MCMAHAN, B. AVENT, A. BELLET, M. BENNIS, A. N. BHAGOJI, K. BONAWITZ, Z. CHARLES, G. CORMODE, R. CUMMINGS, ET AL., *Advances and open problems in federated learning*, Foundations and Trends® in Machine Learning, 14 (2021), pp. 1–210.

[63] M. KHODAK, M.-F. F. BALCAN, AND A. S. TALWALKAR, *Adaptive gradient-based meta-learning methods*, Advances in Neural Information Processing Systems, 32 (2019).

[64] J. KIETZMANN, J. PASCHEN, AND E. TREEN, *Artificial intelligence in advertising: How marketers can leverage artificial intelligence along the consumer journey*, Journal of Advertising Research, 58 (2018), pp. 263–267.

[65] J. KONECNÝ, H. B. MCMAHAN, F. X. YU, P. RICHTÁRIK, A. T. SURESH, AND D. BACON, *Federated learning: Strategies for improving communication efficiency*, CoRR, abs/1610.05492 (2018).

[66] L. LI, K. OTA, AND M. DONG, *Humanlike driving: Empirical decision-making system for autonomous vehicles*, IEEE Transactions on Vehicular Technology, 67 (2018), pp. 6814–6823.

[67] L. LI, W. XU, T. CHEN, G. B. GIANNAKIS, AND Q. LING, *Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets*, in AAAI, vol. 33, 2019, pp. 1544–1551.

[68] T. LI, S. HU, A. BEIRAMI, AND V. SMITH, *Ditto: Fair and robust federated learning through personalization*, in International Conference on Machine Learning, PMLR, 2021, pp. 6357–6368.

[69] T. LI, A. K. SAHU, A. TALWALKAR, AND V. SMITH, *Federated learning: Challenges, methods, and future directions*, IEEE Signal Processing Magazine, 37 (2020), pp. 50–60.

[70] T. LI, A. K. SAHU, M. ZAHEER, M. SANJABI, A. TALWALKAR, AND V. SMITH, *Federated optimization in heterogeneous networks*, arXiv preprint arXiv:1812.06127, (2018).

[71] T. LI, A. K. SAHU, M. ZAHEER, M. SANJABI, A. TALWALKAR, AND V. SMITHY, *Feddane: A federated newton-type method*, in ACSSC, IEEE, 2019, pp. 1227–1231.

[72] T. LI, M. SANJABI, AND V. SMITH, *Fair resource allocation in federated learning*, CoRR, abs/1905.10497 (2019).

[73] X. LI, K. HUANG, W. YANG, S. WANG, AND Z. ZHANG, *On the convergence of fedavg on non-iid data*, arXiv preprint arXiv:1907.02189, (2019).

[74] P. P. LIANG, T. LIU, L. ZIYIN, R. SALAKHUTDINOV, AND L.-P. MORENCY, *Think locally, act globally: Federated learning with local and global representations*, arXiv preprint arXiv:2001.01523, (2020).

[75] W. Y. B. LIM, N. C. LUONG, D. T. HOANG, Y. JIAO, Y.-C. LIANG, Q. YANG, D. NIYATO, AND C. MIAO, *Federated learning in mobile edge networks: A comprehensive survey*, IEEE Communications Surveys & Tutorials, (2020).

[76] Z. LIU, P. LUO, X. WANG, AND X. TANG, *Deep learning face attributes in the wild*, in Proceedings of the IEEE ICCV, 2015, pp. 3730–3738.

[77] G. LONG, T. SHEN, Y. TAN, L. GERRARD, A. CLARKE, AND J. JIANG, *Federated learning for privacy-preserving open innovation future on digital health*, in Humanity Driven AI, Springer, 2022, pp. 113–133.

[78] G. LONG, Y. TAN, J. JIANG, AND C. ZHANG, *Federated learning for open banking*, in Federated learning, Springer, 2020, pp. 240–254.

[79] J. LUO, X. WU, Y. LUO, A. HUANG, Y. HUANG, Y. LIU, AND Q. YANG, *Real-world image datasets for federated learning*, arXiv preprint arXiv:1910.11089, (2019).

[80] J. MA, G. LONG, T. ZHOU, J. JIANG, AND C. ZHANG, *On the convergence of clustered federated learning*, arXiv preprint arXiv:2202.06187, (2022).

[81] Y. MANSOUR, M. MOHRI, J. RO, AND A. T. SURESH, *Three approaches for personalization with applications to federated learning*, arXiv preprint arXiv:2002.10619, (2020).

[82] O. MARFOQ, G. NEGLIA, A. BELLET, L. KAMENI, AND R. VIDAL, *Federated multi-task learning under a mixture of distributions*, Advances in Neural Information Processing Systems, 34 (2021).

[83] B. MCMAHAN, 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, pp. 1273–1282.

[84] A. P. MUNIYANDI, R. RAJESWARI, AND R. RAJARAM, *Network anomaly detection by cascading k-means clustering and c4. 5 decision tree algorithm*, Procedia Engineering, 30 (2012), pp. 174–182.

[85] A. NICHOL AND J. SCHULMAN, *Reptile: a scalable metalearning algorithm*, arXiv preprint arXiv:1803.02999, 2 (2018).

[86] S. PANZERI, R. SENATORE, M. A. MONTEMURRO, AND R. S. PETERSEN, *Correcting for the sampling bias problem in spike train information measures*, Journal of neurophysiology, 98 (2007), pp. 1064–1072.

[87] D. PAUL, S. CHAKRABORTY, AND S. DAS, *Robust principal component analysis: A median of means approach*, arXiv preprint arXiv:2102.03403, (2021).

[88] N. RIEKE, J. HANCOX, W. LI, F. MILLETARI, H. R. ROTH, S. ALBARQOUNI, S. BAKAS, M. N. GALTIER, B. A. LANDMAN, K. MAIER-HEIN, ET AL., *The future of digital health with federated learning*, NPJ digital medicine, 3 (2020), pp. 1–7.

[89] M. ROSEN-ZVI, T. GRIFFITHS, M. STEYVERS, AND P. SMYTH, *The author-topic model for authors and documents*, in Proceedings of the 20th conference on Uncertainty in artificial intelligence, 2004, pp. 487–494.

[90] F. SATTLER, K.-R. MÜLLER, AND W. SAMEK, *Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints*, arXiv preprint arXiv:1910.01991, (2019).

[91] F. SATTLER, K.-R. MÜLLER, T. WIEGAND, AND W. SAMEK, *On the byzantine robustness of clustered federated learning*, in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 8861–8865.

[92] J. SCHMIDHUBER, *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*, PhD thesis, Technische Universität München, 1987.

[93] M. SHAHEEN, M. FAROOQ, T. UMER, AND B. KIM, *Applications of federated learning; taxonomy, challenges, and research trends. electronics 2022, 11, 670*, 2022.

[94] O. SHAMIR, N. SREBRO, AND T. ZHANG, *Communication-efficient distributed optimization using an approximate newton-type method*, in ICML, 2014, pp. 1000–1008.

[95] A. SHAMSIAN, A. NAVON, E. FETAYA, AND G. CHECHIK, *Personalized federated learning using hypernetworks*, in International Conference on Machine Learning, PMLR, 2021, pp. 9489–9502.

[96] D. SILVER, J. SCHRITTWIESER, K. SIMONYAN, I. ANTONOGLOU, A. HUANG, A. GUEZ, T. HUBERT, L. BAKER, M. LAI, A. BOLTON, ET AL., *Mastering the game of go without human knowledge*, nature, 550 (2017), pp. 354–359.

[97] V. SMITH, C.-K. CHIANG, M. SANJABI, AND A. TALWALKAR, *Federated multi-task learning*, 2018.

[98] Y. TAN, G. LONG, L. LIU, T. ZHOU, Q. LU, J. JIANG, AND C. ZHANG, *Fedproto: Federated prototype learning over heterogeneous devices*, arXiv preprint arXiv:2105.00243, (2021).

[99] ——, *Fedproto: Federated prototype learning across heterogeneous clients*, in AAAI Conference on Artificial Intelligence, vol. 1, 2022, p. 3.

[100] Y. W. TEH, M. I. JORDAN, M. J. BEAL, AND D. M. BLEI, *Hierarchical dirichlet processes*, Journal of the american statistical association, 101 (2006), pp. 1566–1581.

[101] R. THIBAUX AND M. I. JORDAN, *Hierarchical beta processes and the indian buffet process*, in Artificial Intelligence and Statistics, 2007, pp. 564–571.

[102] A. TURING, *Intelligent machinery (1948)*, B. Jack Copeland, (2004), p. 395.

[103] J. VAIDYA AND C. CLIFTON, *Privacy preserving association rule mining in vertically partitioned data*, in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 2002, pp. 639–644.

[104] ——, *Privacy-preserving k-means clustering over vertically partitioned data*, in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003, pp. 206–215.

[105] J. VAIDYA, C. CLIFTON, M. KANTARCIOGLU, AND A. S. PATTERSON, *Privacy-preserving decision trees over vertically partitioned data*, ACM Transactions on Knowledge Discovery from Data (TKDD), 2 (2008), pp. 1–27.

[106] H. WANG, M. YUROCHKIN, Y. SUN, D. PAPAILIOPOULOS, AND Y. KHAZAENI, *Federated learning with matched averaging*, in International Conference on Learning Representations, 2020.

[107] Z. WANG, T. ZHOU, G. LONG, B. HAN, AND J. JIANG, *Fednoil: A simple two-level sampling method for federated learning with noisy labels*, arXiv preprint arXiv:2205.10110, (2022).

[108] M. XIE, G. LONG, T. SHEN, T. ZHOU, X. WANG, J. JIANG, AND C. ZHANG, *Multi-center federated learning*, arXiv preprint arXiv:2108.08647, (2021).

[109] J. XU, B. S. GLICKSBERG, C. SU, P. WALKER, J. BIAN, AND F. WANG, *Federated learning for healthcare informatics*, Journal of Healthcare Informatics Research, 5 (2021), pp. 1–19.

[110] R. XU AND D. WUNSCH, *Survey of clustering algorithms*, IEEE Transactions on neural networks, 16 (2005), pp. 645–678.

[111] C. YANG, Q. WANG, M. XU, S. WANG, K. BIAN, AND X. LIU, *Heterogeneity-aware federated learning*, arXiv preprint arXiv:2006.06983, (2020).

[112] M.-S. YANG, C.-Y. LAI, AND C.-Y. LIN, *A robust em clustering algorithm for gaussian mixture models*, Pattern Recognition, 45 (2012), pp. 3950–3961.

[113] M.-S. YANG AND K.-L. WU, *A similarity-based robust clustering method*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 434–448.

[114] Q. YANG, Y. LIU, T. CHEN, AND Y. TONG, *Federated machine learning: Concept and applications*, ACM Transactions on Intelligent Systems and Technology (TIST), 10 (2019), p. 12.

[115] D. YIN, Y. CHEN, R. KANNAN, AND P. BARTLETT, *Defending against saddle point attack in byzantine-robust distributed learning*, in International Conference on Machine Learning, PMLR, 2019, pp. 7074–7084.

[116] X. YUAN, *A quantum-computing advantage for chemistry*, Science, 369 (2020), pp. 1054–1055.

[117] M. YUROCHKIN, M. AGARWAL, S. GHOSH, K. GREENEWALD, T. N. HOANG, AND Y. KHAZAENI, *Bayesian nonparametric federated learning of neural networks*, arXiv preprint arXiv preprint arXiv:1905.12022, (2019).

[118] A. ZEWE, *Can machine-learning models overcome biased datasets?*

[119] 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).

[120] Z. ZHENG, Y. ZHOU, Y. SUN, Z. WANG, B. LIU, AND K. LI, *Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges*, Connection Science, 34 (2022), pp. 1–28.