**People's Democratic Republic of Algeria**

**Ministry of Higher Education and Scientific Research**

**Mohamed Boudiaf University of M'Sila**

**Faculty of Mathematics and Computer Science**

**Department of Computer Science**

**N° :……..**

**FIELD : Mathematics and Computer Science**

**BRANCH : Computer Science**

**OPTION : ARTIFICIAL INTELLIGENCE**

**Thesis presented for obtaining the Academic Master's degree**

**by**:

Thameur Bouberra and Yassin Seghiri

**Entitled:**

# Studying The Effectiveness of Federated Learning in The Healthcare Domain

**Defended in front of the jury composed of:**

| | | |
|---|---|---|
| Dr. Hichem Debbi | University of M'sila | President |
| Dr. Noureddine Amraoui | University of M'sila | Reporter |
| Dr. Abdessattar Ghemougui | University of M'sila | Examiner |

**Academic Year: 2022 / 2023**

*This dissertation is dedicated to everyone who shares his talents and potential with others, to everyone who contributes to the development of society and the service of humanity, and to everyone with special needs.*

# Acknowledgements

We are grateful to Almighty Allah for giving us the strength and ability to achieve this goal.

We first would like to thank our supervisor, Dr. Noureddine Amraoui, for guiding and supporting us throughout the thesis completion process.

We would like also to thank the jury members for reviewing and evaluating my thesis. Their expertise and involvement are greatly appreciated.

We are deeply grateful to our families and friends for their unwavering support during this important phase of my life. Special thanks go to Zakaria, Ali, Hicham, and Marouan for their constant encouragement and assistance.

We would like also to express our gratitude to Isam and Habib for providing us with meals during this challenging journey.

May our accomplishments serve as inspiration to others embarking on their own journeys of discovery and innovation. We extend our heartfelt wishes for success and fulfillment to every individual pursuing their dreams and making positive contributions to society.

# Table of contents

# List of figures

# List of tables

# Nomenclature

## Acronyms / Abbreviations

**AI** Artificial intelligence

**CXR** Chest X Ray

**CPU** Central Processing Unit

**DL** Deep Learning

**FedAdam** Federated Adaptive Moment Estimation

**FedAdagrad** Federated Adaptive Gradient

**FedAvg** Federated Averaging

**FedProx** Federated Proximal

**FL** federated learning

**FLOWER** Friendly Federated Learning Framework

**FTL** Federated Transfer Learning

**GPU** processeur graphique

**GUI** Graphical user interface

**HFL** Horizontal Federated Learning

**Iid** Independent and Identically Distributed

**IoHT** Internet of Health Things

**JPEG** Joint Photographic Experts Group

**ML** Machine Learning

**MRI** Magnetic Resonance Imaging

**SHSs** Smart Healthcare Systems

**SMPC** secure multi party computation

**VFL** Vertical Federated Learning

**X-ray** X-radiation

# Chapter 1

# Introduction

## 1.1 Context and Problem Statement

Machine Learning (ML) is gradually becoming a valuable tool that augments research and discovery in many industries, including Smart Healthcare Systems (SHSs) [36]. The application of ML in SHSs has the potential to improve diagnostics, treatment planning, and personalized medicine, ultimately leading to better patient outcomes.

Unfortunately, in SHSs, data is distributed across different silos (i.e., hospitals) and not available on a centralized server to train a good ML model [14]. Indeed, there are many reasons why the classic centralized ML approach does not work for SHSs including: Data Protection Regulations (e.g., Europe The General Data Protection Regulation), User preferences, Data volume , User preferences, Data volume [36].

As a result, the healthcare industry faces significant barriers in leveraging this valuable resource and suffer ML models from insufficient training data, leading to sub-optimal performance in healthcare applications.

## 1.2 Objectives and Methodology

In this work, we aim to address the challenges facing SHSs to unleash the power of ML in such critical sector. Our goals are as follows:

- To analyze the challenges and limitations of traditional data-sharing methods in healthcare.
- To implement and compare different strategies that allows ML models to be trained on distributed data sources while preserving data privacy and adhering to relevant regulations.

To achieve our objectives , we propose to leverage Federated Learning (FL) approach. FL is a new distributed interactive ML concept that enables numerous clients to take part in ML training while maintaining data locally [27].

## 1.3 Motivation

There are several motivations for using Federated Learning in Smart Healthcare System. Here are some key reasons:

1. **Privacy-Preserving:** FL enables the analysis of sensitive health data from multiple sources without the need for data to be centralized in one location.
2. **Increased Data Availability**: By pooling data from multiple sources, FL can provide access to a larger and more diverse dataset than would be available in a single location.
3. **Improved Model Accuracy**: FL can allow for the development of more robust and accurate models by training them on a larger and more diverse set of data.

4. **Better Data Governance**: FL provides a framework for sharing data while still maintaining control over who has access to it. Hospitals and clinics can remain in control of their own data

5. **Lowered Cost**: FL can reduce the cost of data management and analysis by leveraging existing infrastructure and resources.

## 1.4 Contribution and Results

This project aims to investigate the effectiveness of FL in the context of Smart Healthcare Systems by comparing it with the traditional data-centralized approach.

Specifically, four FL strategies are implemented using the Flower framework, with each strategy trained on four different types of datasets: Brain Tumor, Pneumonia, Alzheimer's disease, and Covid-19, the performance of these FL strategies is then compared to that of a data-centralized model trained on the same datasets.

Our obtained experimental results reveal that the accuracy of the FL models is less than the data-centralized model for all four strategies, However, such a small extent can be tolerated considering the advantages of FL brought to SHSs.

The disparity in accuracy suggests that the challenges associated with decentralized FL learning in SHS settings, such as data heterogeneity, communication limitations, and variability in local datasets, may hinder the performance of FL models. Thus, we may conclude that there should be a trade-off between high accuracy and FL advantages.

## 1.5 Dissertation Structure

The rest of this dissertation is organized as follows:

- **Chapter 2**: presents our review of the state-of-the-art related Smart Healthcare Systems by providing a background on SHSs and discussing the limitations of applying classical ML in current SHSs.
- **Chapter 3**: introduces the Federated Learning (FL) approach leveraged in this work by its process, and types, and discusses it in the context of Smart Healthcare Systems.
- **Chapter 4**: presents our implementation and experimental study of a federated learning system on healthcare data to show the efficiency of employing such an approach in Smart Healthcare Systems.

Finally, we conclude this work by drawing conclusions and suggesting some future works.

# Chapter 2

# Smart Healthcare Systems: Background and Limitations

This chapter presents our review of the state-of-the-art related Smart Healthcare Systems (SHSs) ,we first provide a background on SHSs, then, we discuss the limitations of applying classical ML in current SHSs.

## 2.1 Smart Healthcare Systems

Machine learning (ML) involves the use of algorithms that can learn patterns and relationships data, these algorithms are typically trained on large datasets, where the input data is labeled with the correct output, the algorithm uses this labeled data to learn how to predict the output for new, unseen data [36].

ML has become a significant tool in the arsenal of artificial intelligence techniques used in healthcare, an overall system architecture for disease diagnosis using machine learning algorithms is shown in Figure 2.1.

The data used in these types of systems, such as Internet of Health Things (IoHT) environments, can come from benchmark datasets or real-time sensor data sent to the fog/edge/cloud for processing.

Afterward, the data are preprocessed, and necessary features are extracted to fit in the machine learning techniques.

Finally, the decision is transferred to the concerned person to take proper action.

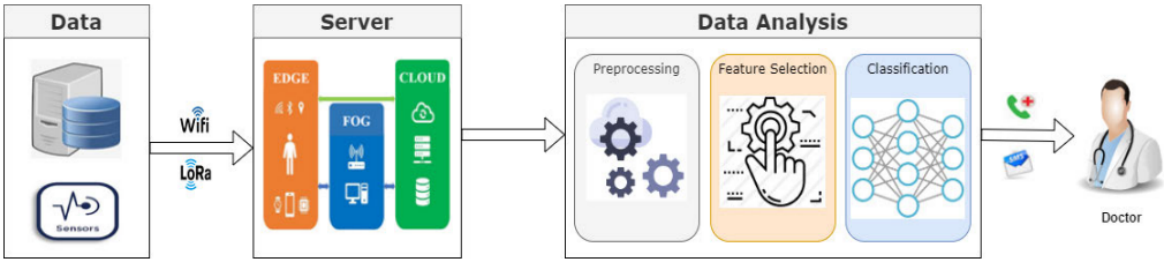



Fig. 2.1 An overall system architecture of machine learning-based framework for disease diagnosis in IoHT environment [36].

In the following sections, we present some major ML-based disease solutions that are becoming significant threats to human-being in recent times

### 2.1.1 Novel Coronavirus (Covid-19)

Novel coronavirus (Covid-19) has become a public health emergency, this pandemic is still going strong, and people all over the world are working to recover from it , according to the statistic, 98 million cases have already been identified, and 2 million people have died as a result [8].

Several studies have been carried out using contemporary technologies to lessen the severity of this disease [35].

Despite the fact that many works have already been done, we described the frameworks for diagnosing COVID-19 in IoT environments using machine learning algorithms.

| Authors | Year | Data | Techniques | Accuracy | Comments |
|---|---|---|---|---|---|
| Le et al. [60] | 2021 | CXR dataset and sensor data | CNN, SVM | 99.06% | No monitoring system from the doctor's end is developed. |
| Ramallo-González et al. [44] | 2021 | Sensor data | CNN | 66.67% | The outcome is relatively low for real-time use. |
| Ahmed et al. [9] | 2020 | Public datasets | Faster-RCNN, ResNet-101 | 98% | No usability study is mentioned. |
| Otoom et al. [38] | 2020 | CORD-19 dataset and sensor data | Eight ML techniques | 92.89% | The performance is relatively low for practical uses. |
| El-Rashidy et al. [19] | 2020 | Datasets and sensor data | ResNet-50 | 97.95% | Energy consumption and storage are challenges for this system. |

Table 2.1 Summary of Covid-19 Detection Frameworks in IoHT Environment [36]

### 2.1.2 Heart Disease

Heart disease has emerged as one of the most serious and acute illnesses affecting all elderly people, especially adults, according to estimates, heart disease causes about 18 million of deaths [28].

In order to lessen the severity of heart disease, researchers are concentrating on the development of decision-support systems in the smart healthcare environment.

Table 2.2 provides an overview of systems for detecting heart disease in the IoHT environment, taking into account features as the datasets used, the detection algorithms used, the accuracy as an evaluation metric, and comments of each developed system.

| Authors | Year | Data | Techniques | Accuracy | Comments |
|---|---|---|---|---|---|
| Sarmah [54] | 2020 | Hungarian HD dataset and real-time sensor data | DLMNN | 96.8 | The developed scheme obtained comparatively low performance in the case of small data size. |
| Ali et al. [10] | 2020 | Cleveland dataset and real-time sensor data | Feature fusion and ensemble learning (Logit-Boost) | 98.5% | The developed system used traditional techniques for feature selection, reduction, and classification. |
| Deperlioglu et al. [15] | 2020 | PASCAL and PhysioNet dataset and real-time sensor data | Autoencoder neural network | 96.03% | No voice command facility is available in this study to ensure less physical interaction. |
| Khan and Algarni [30] | 2020 | Hungarian and Framingham dataset and real-time sensor data | MSSO-ANFIS | 99.45% | No real-time study is shown in this system. |
| Khan [29] | 2020 | Cleveland database and real-time sensor data | MDCNN | 98.02% | No wearable prototype is mentioned. |

Table 2.2 Summary of Heart Disease Detection Systems in IoHT Environment[36]

### 2.1.3 Diabetes

Another deadly condition that affects humankind, diabetes claims many lives each year, according to an estimate, there were approximately 463 million diabetics worldwide in 2019, and by 2030 and 2045, those numbers are projected to rise to 578 million and 700 million, respectively [52].

Diabetes must be detected early because it is a condition that is spreading quickly, With the help of artificial intelligence, IoT, and big data, numerous studies are being conducted to detect diabetes early [51].

This section provides illustrations of recent works on diabetes detection. Table 2.3 provides a brief overview of diabetes detection frameworks, highlighting certain characteristics

like the datasets used, the detection algorithms used, the accuracy as a performance metric, and comments of each system reviewed in the IoHT environment.

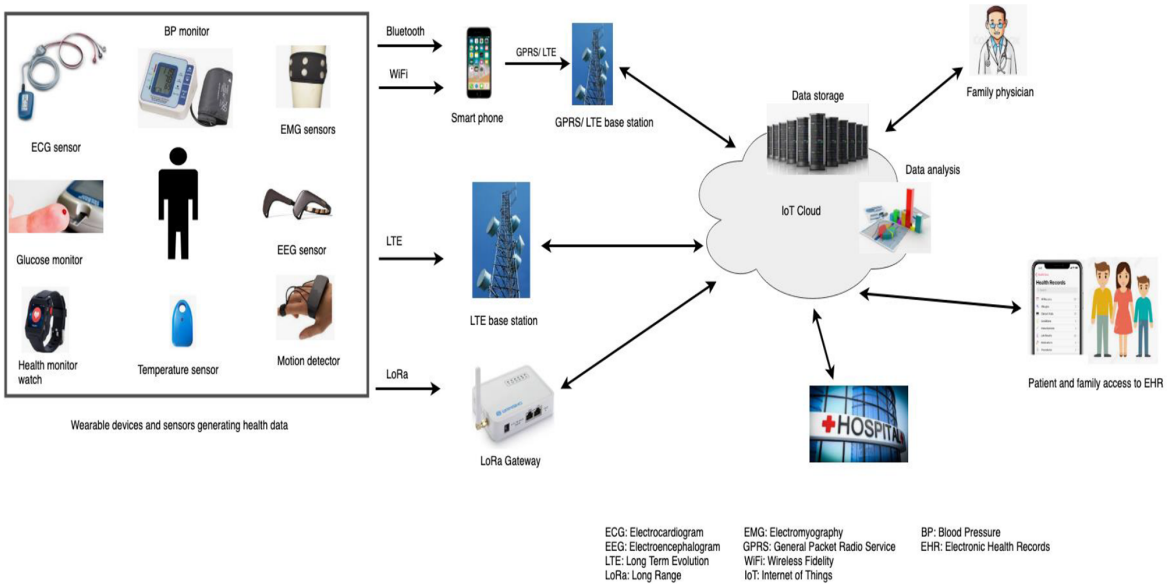
| Authors | Year | Data | Techniques | Accuracy | Comments |
|---|---|---|---|---|---|
| Rghioui et al. [47] | 2021 | Real-time sensor data | Naive Bayes, Random Forest, OneR, SMO | 96.05% | No user prototype is shown in this system. |
| Allugunti et al. [11] | 2020 | Real-time sensor data | Decision Tree | 96.43% | The detailed data collection procedure is not mentioned. |
| Efat et al. [18] | 2020 | Real-time sensor data | Neural network | 84.29% | The performance is relatively low for practical use. |
| Rghioui et al. [48] | 2020 | Real-time sensor data | Six machine learning algorithms | 97.87% | The glucose sensor and Arduino could not be operated at the same time. |
| Rghioui et al. [46] | 2020 | Real-time sensor data | Six classification algorithms | 99.66% | The latency is comparatively high in this system. |

Table 2.3 Summary of Diabetes Detection Frameworks in IoHT Environment [36]

## 2.2 Limitations of Current Smart Healthcare Systems

Smart Healthcare Systems have the potential to revolutionize the healthcare industry by improving patient outcomes, reducing costs, and increasing efficiency, however, current SHSs are not without their limitations.

In this response, we will discuss four major limitations of current smart healthcare systems.




**Sources:** Gopi and Hwang (2016), Zagan *et al.* (2017), Yang *et al.* (2016)

Fig. 2.2 A smart health-care system[26]

### 2.2.1 Privacy Concerns

One of the major limitations of current smart healthcare systems is privacy concerns, health-care data is highly sensitive and contains personal information, which makes it vulnerable to privacy breaches and patients have the right to keep their medical information private and secure, but there have been instances where healthcare data breaches have occurred.

Federated learning has been proposed as a solution to address privacy concerns in healthcare data training, it is allows models to be trained on distributed data without transferring the data itself, which preserves data privacy and security [61].

### 2.2.2 Lack of Datasets at Medical Sites

Another limitation of current smart healthcare systems is the lack of datasets at medical sites. Medical datasets are essential for developing machine learning models that can improve patient outcomes.

However, not all medical sites have access to large datasets, which limits the performance of machine learning models. Federated learning has been proposed as a solution to address this limitation by allowing machine learning models to be trained on data from multiple sites without transferring the data itself [55].

### 2.2.3 Limited Health Data Training Performance

Current smart healthcare systems also suffer from limited health data training performance, the performance of machine learning models depends on the quality and quantity of data used for training., and limited data can result in poor model performance, which can lead to incorrect diagnoses and treatments[49].

Federated learning has been proposed as a solution to address this limitation by allowing machine learning models to be trained on data from multiple sites, which can improve the quality and quantity of data used for training [12].

### 2.2.4 High Costs in Health Data Training

The high costs associated with health data training is another limitation of current smart healthcare systems, and training machine learning models requires significant resources, including computational power and human expertise[33].

Federated learning has been proposed as a solution to address this limitation by reducing the computational resources required for training machine learning models. Federated learning allows models to be trained on distributed data, which reduces the need for centralized resources [50].

## 2.3 Conclusion

Machine Learning-based Smart Healthcare Systems based have shown great potential in improving disease diagnosis and patient outcomes, but they are not without limitations.

Privacy concerns regarding healthcare data, the lack of datasets at medical sites, limited health data training performance, and high costs in health data training are some of the major limitations of current smart healthcare systems.,and addressing these limitations is crucial to ensure the widespread adoption and success of smart healthcare systems.

Future research and technological advancements should focus on developing robust and secure frameworks that leverage the power of machine learning while ensuring the privacy and security of healthcare data.

# Chapter 3

# Federated Learning for Smart Healthcare Systems

This chapter introduces the Federated Learning (FL) approach leveraged in this work.

We first present the FL process, types, and different model aggregation strategies in detail, then, we discuss the benefits of employing FL in Smart Healthcare Systems.

## 3.1 Overview

As a term first used by McMahan in 2016 [34], Federated Learning (FL) is a promising distributed Machine Learning (ML) method that enables multiple nodes or clients to create a joint learning model without exchanging data as depicted in Figure 3.1 [25].

In this way, it deals with important issues like data access rights, privacy, security, and accessibility to a variety of data types.

In FL, the model is trained repeatedly at various sites [27] and it differs from traditional centralized ML techniques that assume that local datasets are dispersed uniformly and from other conventional ML techniques where the datasets are transferred to a single server.

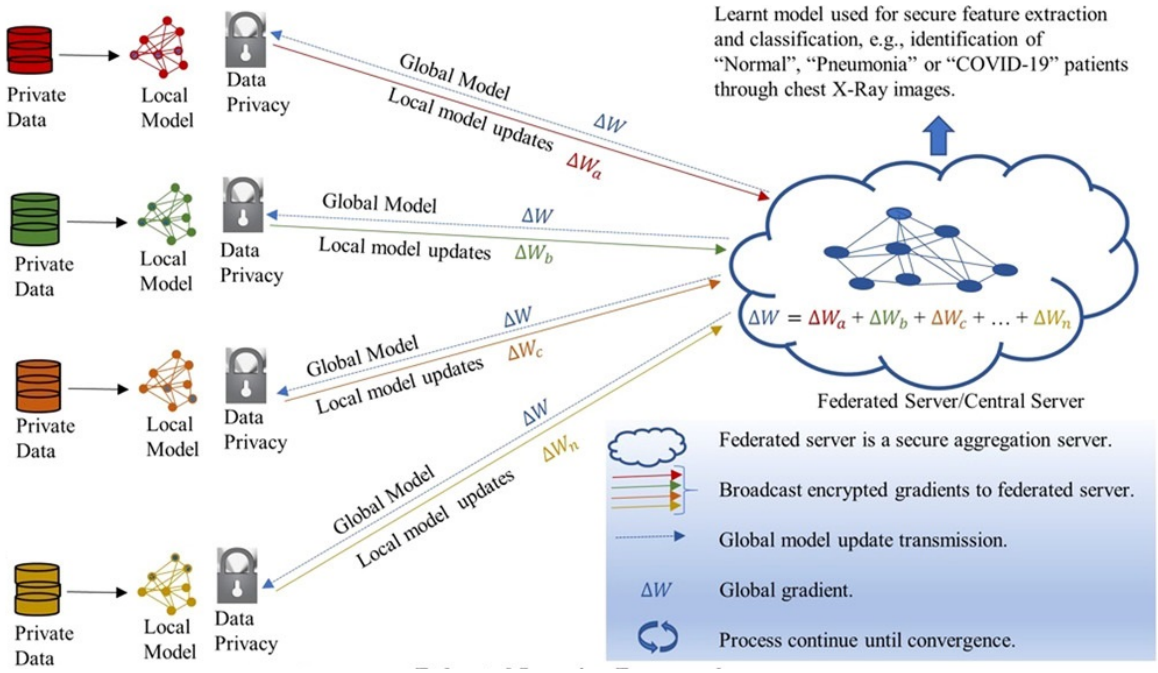



Fig. 3.1 Overview of FL Approach

As depicted in Figure 3.1, there are many industries, including autonomous vehicles, traffic monitoring and prediction, healthcare, the Internet of things, pharmaceutics, and industrial management are said to be applicable [41].

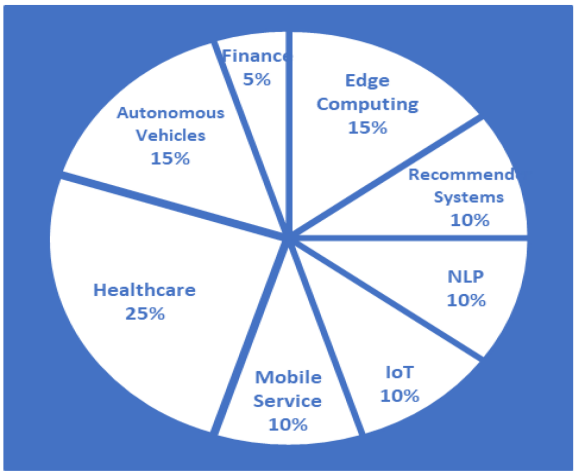



Fig. 3.2 Percentage of FL applications in different domains

### 3.1.1 Federated Learning Process

the FL process can be summarized in the six following steps [6]:

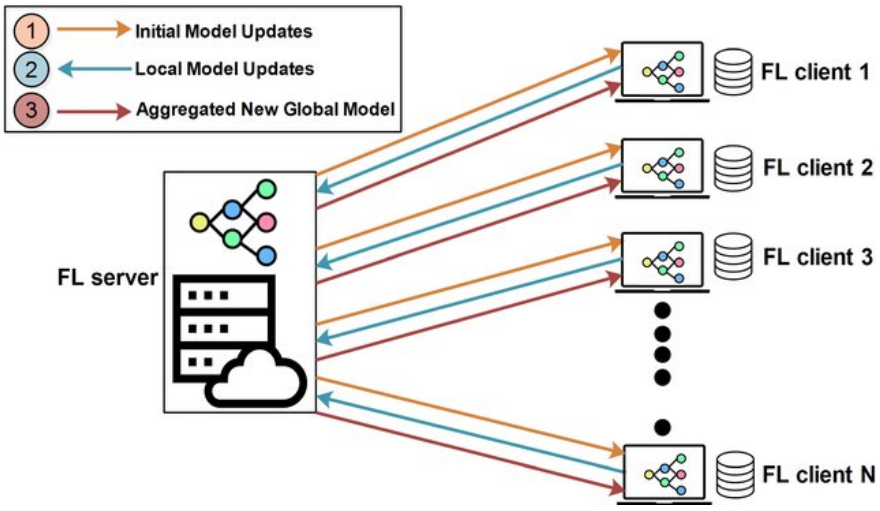



Fig. 3.3 General working process of federated learning [43]

**Step 1: Initialize the global model**

The process of federated learning begins with the initialization of a global model.

This latter is a starting point for the training process and is usually pre-trained on a large, representative dataset, the global model is initialized with a set of weights and biases that define its initial state, the goal of FL is to refine this global model using the data from the different nodes in the network.

The training process happens in a sequence of rounds, and in each round, the global model is sent to the nodes, At each node, the global model is used to make predictions on the local data, and the local model's weights and biases are updated using a local optimization algorithm, such as stochastic gradient descent. The updated weights and biases are then sent back to the server, where they are aggregated to update the global model.

The initialization of the global model is an important step in federated learning because it provides a starting point for the training process , the quality of the global model can have

a significant impact on the speed and accuracy of the Federated Learning process. A good initialization can lead to faster convergence and better results.

### Step 2: Send the global model to a number of connected organizations/devices (client nodes)

In this step, the central server sends the global model to multiple participating organizations or devices, referred to as client nodes, these latter can be various healthcare institutions, IoT devices, or even personal mobile devices, the distribution can be done in various ways, such as round-robin, random selection, or based on the availability of client nodes.

### Step 3: Train the model locally on the data of each organization/device (client node)

Each client node trains the received model on its local data, this local data remains on the device or within the organization, ensuring privacy and security, the training can be performed using standard machine learning algorithms like gradient descent, and the number of training epochs can vary depending on the size and complexity of the local dataset.

**Algorithm 1** Federated learning: client-side training at federated round $t$.

**Require:** local learning rate $\eta$ and loss function $\ell$
**Require:** num_local_epochs and local training data
**procedure** ClientUpdate($w^t$)
  $w \leftarrow w^t$ ▷ Initialize local model
  $\mathcal{B} \leftarrow$ Split $P_k$ into batches of size $B$
  **for** each local epoch $i$ from 1 to $E$ **do** ▷ With *SGD* optimizer
    **for** each batch $b$ in $\mathcal{B}$ **do**
      Compute gradient $g_i^b \leftarrow \nabla \ell(w; b)$
      Update local model $w \leftarrow w - \eta g_i^b$
    **end for**
  **end for**
  **return** $w$ ▷ Upload to server
**end procedure**

Fig. 3.4 client-side training at federated round t. [20]

### Step 4: Return model updates back to the server

Once the local training is completed, each client node sends its model updates back to the central server, these updates can include gradients, weights, or other model parameters that

have been adjusted during the local training, to maintain privacy, the updates can be encrypted or masked before transmission to the server.

**Algorithm 2** Federated learning: server-side aggregation procedure.

**Require:** T : *num_fedetared_rounds*
1: **procedure** AGGREGATING($C, K$)
2: Initialize global model $w^0$
3: **for** each round t = 1,2,... T **do**
4: $m \leftarrow \max(C \times K, 1)$
5: $S_t \leftarrow$ (random set of $m$ clients) ▷ Selected Clients for round $t$
6: **for** each client $k \in S_t$ **do** ▷ Run in parallel
7: Send $w^{t-1}$ to client $k$
8: $w_k^t \leftarrow$ CLIENTUPDATE ($k$, $w^{t-1}$)
9: **end for**
10: $w^t \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^t$ ▷ Aggregating clients updates
11: **end for**
12: **return** $w^T$
13: **end procedure**

Fig. 3.5 server-side aggregation procedure[20]

### Step 5: Aggregate model updates into a new global model

The central server collects the model updates from all participating client nodes and aggregates them to create an updated global model, this aggregation can be performed using various strategies, such as averaging the model updates or using more sophisticated techniques like secure multi-party computation (SMPC) or differential privacy, the goal of this step is to create a new global model that benefits from the collective knowledge of all participating nodes while preserving privacy.

### Step 6: Repeat steps 1 to 4 until the model converges

The federated learning process is iterative and steps 1 to 4 are repeated until the global model converges or reaches a predefined performance threshold, convergence can be monitored using various metrics like loss, accuracy, or other domain-specific measures, the number of iterations may depend on factors like the complexity of the problem, the size and diversity of the local datasets, and the desired level of model performance.

### 3.1.2 Types of Federated Learning

There are three types of FL setup,[17] as depicted in Figure 3.3.

These three types of federated learning setups offer flexibility and adaptability to different data distribution scenarios and privacy requirements. By enabling collaboration and learning from distributed data sources.

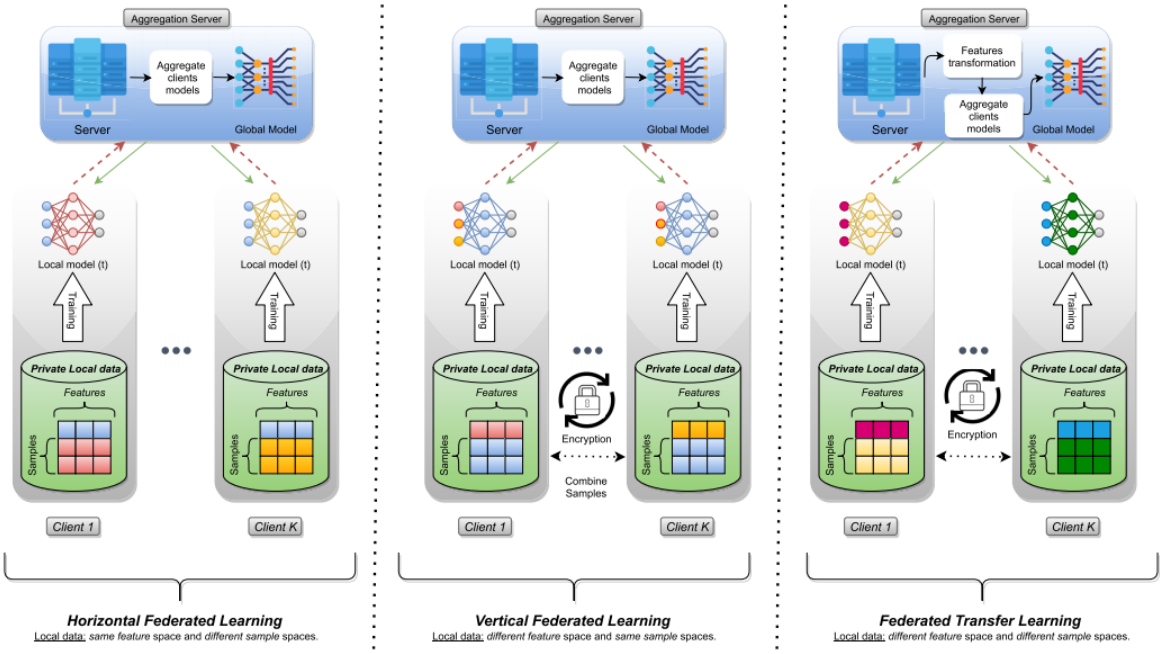



Fig. 3.6 Types of FL [21].

#### Horizontal FL (HFL)

In HFL, data is distributed across different devices or nodes with similar features, such as different users' mobile phones, the goal is to train a machine-learning model across all these devices without centralizing the data. HFL is particularly useful for applications that require privacy preservation, such as personalized recommendations or health monitoring.[6]

#### Vertical FL (VFL)

In VFL, data is distributed across different devices or nodes with different features, such as different hospitals in a healthcare network, the goal is to train a machine learning model across all these devices without centralizing the data, VFL is particularly useful for applications that require cross-institutional collaborations, such as drug discovery or patient diagnosis.[1]

#### Federated Transfer Learning (FTL)

FTL combines federated learning with transfer learning, allowing a machine learning model to be transferred between different devices or nodes while retaining the learned knowledge from the previous task, FTL is particularly useful for applications that require domain adaptation, such as predicting user behavior across different platforms or predicting patient outcomes across different hospitals [53].

### 3.1.3 Models Aggregation Strategies

In federated learning, model aggregation strategies are used to combine the models trained by different clients, frameworks such as Flower have implemented several such strategies, in the healthcare domain, federated learning has shown potential for smart healthcare applications, and several studies have investigated its use.

The choice of aggregation strategy depends on the specific application requirements and constraints here are some common models of aggregation strategies used in federated learning with Flower in healthcare.

#### FedAvg Strategy

Federated Averaging is a popular model aggregation strategy in federated learning, where the clients train their local models on their respective datasets and then send the updated model parameters to a central server, the server aggregates these model updates by computing the

average of the parameters and sends the updated global model back to the clients, this process iterates until convergence [34].

**Algorithm 1** Federated Averaging (`FedAvg`)

**Input:** $K, T, \eta, E, w^0, N, p_k, k = 1, \cdots, N$
**for** $t = 0, \cdots, T-1$ **do**
  Server selects a subset $S_t$ of $K$ devices at random (each device $k$ is chosen with probability $p_k$)
  Server sends $w^t$ to all chosen devices
  Each device $k \in S_t$ updates $w^t$ for $E$ epochs of SGD on $F_k$ with step-size $\eta$ to obtain $w_k^{t+1}$
  Each device $k \in S_t$ sends $w_k^{t+1}$ back to the server
  Server aggregates the $w$'s as $w^{t+1} = \frac{1}{K} \sum_{k \in S_t} w_k^{t+1}$
**end for**

Fig. 3.7 Algorithm FEDAVG [32]

**FedProx Strategy**

Federated Proximal extends the concept of FedAvg by introducing a proximal term to the optimization objective, it aims to enforce similarity between the local models and the global model by penalizing large parameter updates, this approach helps in mitigating the effect of data heterogeneity across clients and stabilizes the learning process in federated settings [31].

**FedAdagrad Strategy**

Federated Adaptive Gradient is an adaptation of the Adagrad optimizer for federated learning. It incorporates adaptive learning rates for each parameter in the global model, considering the frequency and magnitude of updates received from the clients, by assigning different learning rates to different parameters, FedAdagrad can effectively handle non-iid data distribution across clients in federated learning scenarios [45].

**Algorithm 1:** `FedProx` (Proposed Framework)

**Input:** $K, T, \mu, \gamma, w^0, N, p_k, k = 1, \cdots, N$
**for** $t = 0, \cdots, T-1$ **do**
Server selects a subset $S_t$ of $K$ devices at random (each device $k$ is chosen with probability $p_k$)
Server sends $w^t$ to all chosen devices
Each chosen device $k \in S_t$ finds a $w_k^{t+1}$ which is a $\gamma_k^t$-inexact minimizer of: $w_k^{t+1} \approx \arg\min_w h_k(w; w^t) = F_k(w) + \frac{\mu}{2}\|w - w^t\|^2$
Each device $k \in S_t$ sends $w_k^{t+1}$ back to the server
Server aggregates the $w$'s as $w^{t+1} = \frac{1}{K}\sum_{k \in S_t} w_k^{t+1}$
**end for**

Fig. 3.8 Algorithm FEDPROX [32]

**FedAdam Strategy**

Federated Adaptive Moment Estimation extends the popular Adam optimizer for the federated learning setting, it combines the benefits of adaptive learning rates from Adam with the averaging process in federated learning. FedAdam adjusts the learning rates of each parameter based on their past gradients, allowing clients to learn at different speeds while preserving the privacy of their data [23].

## 3.2 Benefits of FL for Smart Healthcare Systems

Federated Learning (FL) has emerged as a promising approach to address some of the limitations of current Smart Healthcare Systems, by allowing multiple institutions to collaboratively train AI models while keeping data locally stored, FL can significantly improve Smart Healthcare Systems in various ways.

Fig. 3.9 Applications Of Federated Learning In Healthcare[16]

### 3.2.1 Data Privacy Improvement

By keeping sensitive patient data within local institutions and only sharing model updates during the training process, FL ensures that patient information remains protected from unauthorized access or misuse, this approach can help alleviate privacy concerns while still enabling the development of effective AI models for healthcare applications [14].

### 3.2.2 Reasonable Trade-off between Accuracy and Utility

FL allows multiple institutions to collaborate on model training, providing access to diverse and large-scale datasets without compromising data privacy[17], this collaboration can lead to more accurate and generalizable models that can better diagnose, predict, and treat various medical conditions.

As a result, FL provides a reasonable trade-off between the accuracy of AI models and the utility of the data, maximizing the benefits for both patients and healthcare providers.

### 3.2.3 Low-cost Health Data Training

By distributing the training process across multiple institutions, FL can reduce the costs associated with health data training [4], each participating institution only needs to contribute a portion of the computational resources and storage, resulting in lower overall costs for each participant, this cost reduction can make advanced AI models more accessible to smaller institutions or those with limited budgets, ultimately contributing to a more equitable distribution of healthcare technology.

## 3.3 Conclusion

FL offers promising opportunities for smart healthcare systems, addressing challenges and limitations will be crucial to fully realize its potential and ensure the effective and ethical deployment of AI in healthcare settings.

# Chapter 4

# Implementation and Experimental Study

This Chapter presents our implementation and experimental study of a federated learning system on healthcare data to show the efficiency of employing such an approach in SHSs.

We first present implementation technologies,then, we introduce our conducted experimental study including evaluation datasets, protocol, and obtained results discussion.

## 4.1 Implementation Technologies

Following is the list of used technologies to implement the FL system and conduct the experimental study.

### 4.1.1 Numpy, matplotlib, ipython.display

These libraries play key roles in scientific computing, data analysis, and visualization within the Python ecosystem, making them valuable tools for researchers and practitioners in various fields.

### 4.1.2 Python

Python is a popular high-level programming language used for various purposes such as web development, scientific computing, data analysis, artificial intelligence, and more, it is an interpreted language, which means that it does not require compilation before execution, Python is known for its readability, simplicity, and versatility, making it an excellent choice for beginners and experienced programmers alike, it also has a vast and active community that contributes to its development and offers support to users [42].

### 4.1.3 PyTorch

PyTorch is an open-source machine learning library based on the Torch library, it is widely used for developing neural networks and deep learning models, it provides a dynamic computational graph, which allows users to define and modify the neural network's architecture at runtime, this feature makes it highly flexible and ideal for experimenting with various deep learning models, it is also provides a wide range of tools for data loading, pre-processing, and model training, making it an all-in-one platform for deep learning development [39].

### 4.1.4 Flower

Flower (Federated Learning Orchestrator) is an open-source framework for FL, it provides a high-level API for developers to implement FL algorithms easily, it abstracts the complexities of FL, such as communication, model aggregation, and security, making it easier for developers to focus on building the actual ML models. Flower also supports various deep learning frameworks such as PyTorch and TensorFlow, making it highly versatile.[6]

By combining PyTorch with the Flower framework, developers can leverage the capabilities of PyTorch for building and training deep learning models while utilizing Flower's functionalities for federated learning.

This combination allows for distributed and privacy-preserving machine learning, where the training data remains on the devices, ensuring data privacy and reducing the need for centralized data storage.

The design of Flower is based on a few guiding principles [6]:

- **Customizable**:FL systems vary wildly from one use case to another, Flower allows for a wide range of different configurations depending on the needs of each individual use case.
- **Extendable**: Flower originated from a research project at the University of Oxford, so it was built with AI research in mind, many components can be extended and overridden to build new state-of-the-art systems.
- **Framework-agnostic**: Different machine learning frameworks have different strengths, Flower can be used with any machine learning framework, for example, PyTorch, TensorFlow, Hugging Face Transformers, PyTorch Lightning, MXNet, scikit-learn, JAX, TFLite, fastai, Pandas for federated analytics, or even raw NumPy for users who enjoy computing gradients by hand.
- **Understandable**: Flower is written with maintainability in mind. and the community is encouraged to both read and contribute to the codebase.

### 4.1.5 Google Colab

Practicing on projects becomes a constraint since you need high-end PCs for such workloads, the answer to this issue is Google Colab, or Collaboratory is a cloud-hosted version of Jupyter Notebook, to use Colab, you do not need to install and runtime or upgrade your computer hardware to meet Python's CPU/GPU intensive workload requirements [22].

Colab allows you to write and execute Python in your browser, with:

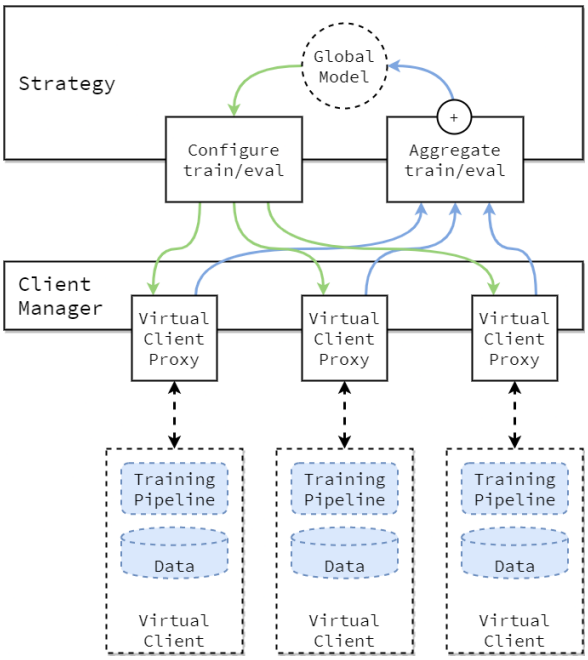



Fig. 4.1 FLOWER-architecture-VCE[6].

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code, all you need is a browser, Google Colab makes data science, deep learning, and neural network, and machine learning accessible to individual researchers who can not afford costly computational infrastructure [22].

### 4.1.6 CNN model for Image Classification

We leverage CNN model to do image classification tasks. The model consists of several convolutional layers, in order to improve the performance of the classification task, as depicted in Table 4.1.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| Conv2d | [batch, 32, 28, 28] | 320 |
| MaxPool2d | [batch, 32, 14, 14] | 0 |
| Conv2d | [batch, 64, 14, 14] | 18,496 |
| MaxPool2d | [batch, 64, 7, 7] | 0 |
| Conv2d | [batch, 128, 7, 7] | 73,856 |
| MaxPool2d | [batch, 128, 3, 3] | 0 |
| Dropout | [batch, 128, 3, 3] | 0 |
| Flatten | [batch, 1152] | 0 |
| Linear | [batch, 264] | 304,392 |
| Linear | [batch, 4] | 1,060 |

Table 4.1 Architecture of the Proposed Model for CNN

## 4.2 Evaluation Datasets

In this section, we present and describe the datasets used in our experimental study.

### 4.2.1 Covid-19 Radiography

A database of chest X-ray images for Covid-19-positive cases, along with normal and viral pneumonitis images, has been created by a group of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh, as well as their collaborators from Pakistan and Malaysia, they worked with medical professionals, this dataset for Covid-19, normal, and other lung infections is made available in phases [4].

- CXR images has been collected from padchest dataset [2].
- CXR images has been collected from a Germany medical school [3].
- CXR image has been collected from SIRM, Github, Kaggle and Tweeter [5].
- CXR images has been collected from another Github source [7].

Figure 4.2 present some samples from Covid-19 Radiography Dataset.

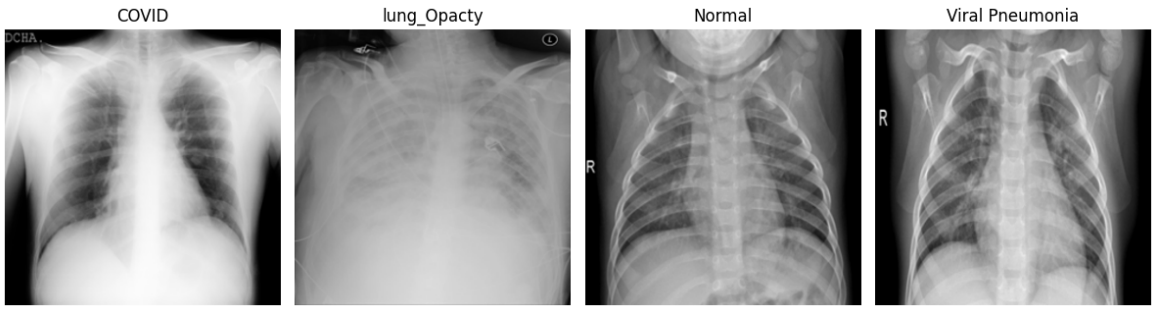



Fig. 4.2 Covid-19 Radiography

### 4.2.2 Pneumonia Chest X-Ray

This dataset consists of a set of X-ray JPEG images categorized into two classes (Pneumonia/Normal).

From retrospective cohorts of pediatric patients aged one to five at the Guangzhou Women and Children's Medical Centre in Guangzhou, chest X-ray images (anterior-posterior) were chosen,all chest X-ray imaging was done as part of the regular clinical care provided to patients.

All chest radiographs were initially screened for quality control before being removed from the analysis of the chest x-ray images, before the diagnoses for the images could be used to train the AI system, they were graded by two experienced doctors, the evaluation set was additionally examined by to account for any grading errors.[1]

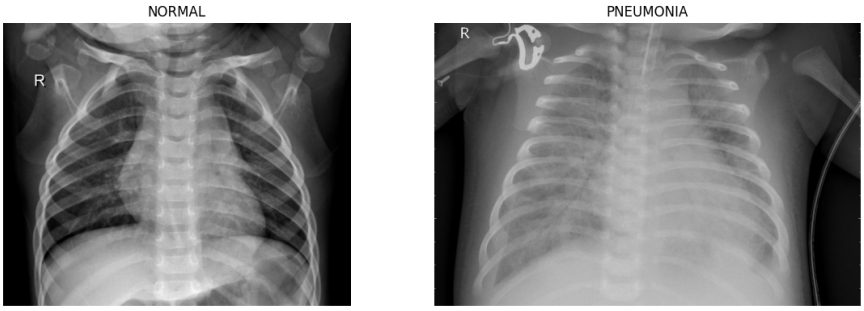



Fig. 4.3 Pneumonia Chest X-Ray

### 4.2.3 Brain Tumor

This dataset consists of the scanned images of brain of patient diagnosed of brain tumour [59], and the data has 2 classes of images namely: Brain tumer and Healthy Fig. 4.4.

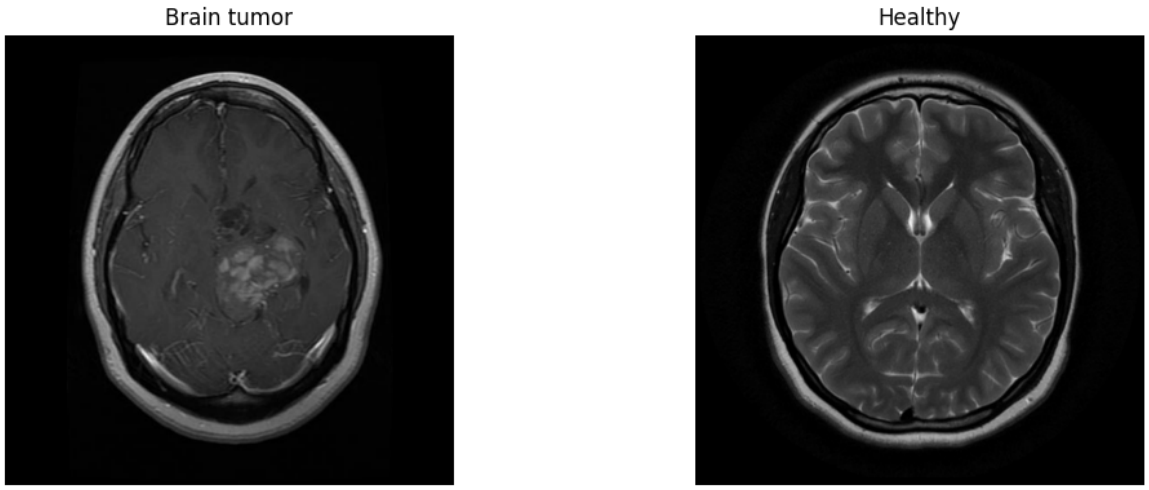



Fig. 4.4 Brain Tumor Radiography

### 4.2.4 Alzheimer's

This dataset is hand-collected from various websites with each and every label verified, it is data consisting of MRI images, and the data has four classes of images both in training as well as a testing set, namely: Mild Demented, Moderate Demented, Non Demented, Very Mild Demented.

The main goal behind sharing this dataset is to make a very highly accurate model to predict the stage of Alzheimer's [58].

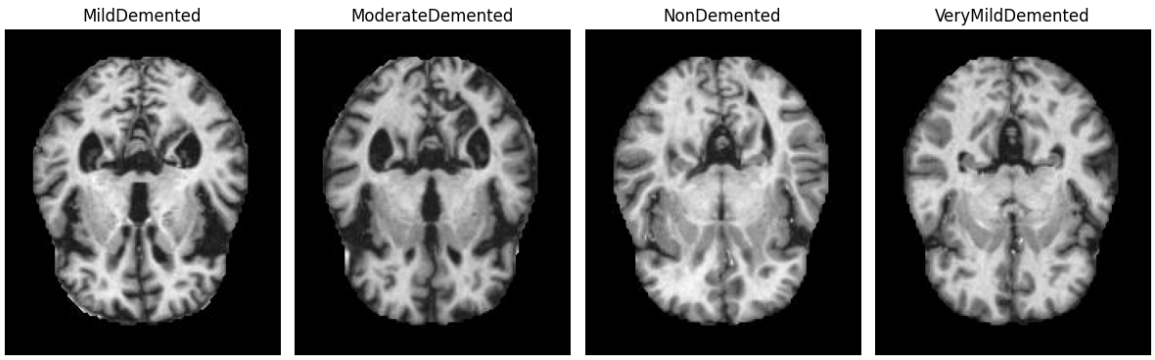



Fig. 4.5 Alzheimer's Radiography

## 4.3 Experimental Protocol

We conducted an experimental study employing two distinct modes for training CNN models: data-centralized and Federated, in the subsequent sections, we provide comprehensive insights into each mode.

### 4.3.1 Data-centralized Training

As depicted in Figure 4.6, data-centralized training refers to an ML training approach where all the training data is collected and stored in a central location or server, the process of data-centralized training involves the following steps:

- **Data Collection**: Data is collected from various sources, which can include user devices, sensors, databases, or other data repositories, this data is then transmitted to a central server.
- **Data Transfer**: The collected data is transferred to the central server for further processing, this transfer can occur through various means, such as network connections, cloud storage, or manual upload.

- **Data Aggregation**: Once the data is transferred to the central server, it is aggregated into a single dataset, this aggregation step combines the individual data samples from different sources into one cohesive training set.
- **Model Training**: Using the aggregated dataset, the machine learning model is trained on the central server, the training process typically involves running iterative algorithms that optimize the model parameters based on the provided data.
- **Model Evaluation**: After training, the model's performance is evaluated using separate validation or testing datasets, this step helps assess how well the model generalizes to unseen data and guides further model refinement.

**Data-centralized training** Centralized learning refers to the traditional approach where all the data is collected and stored in a central server or data center, in this setting, the training process takes place on the central server, where the model is trained using the entire dataset.

We provided is plotting a bar chart to visualize the train and test set sizes for different datasets, here's a breakdown of the plot:

### 4.3.2 Federated Learning

In federated learning, the dataset is typically split among multiple clients to simulate a decentralized setting where each client has its own data, this splitting of the dataset to clients is done to mimic the real-world scenario where different hospitals or devices hold their own data and participate in the federated learning process.

The dataset splitting is done to ensure that each client has a unique subset of the data, which represents the data it owns or has access to, this partitioning is important because federated learning aims to train models on decentralized data without sharing the data itself.

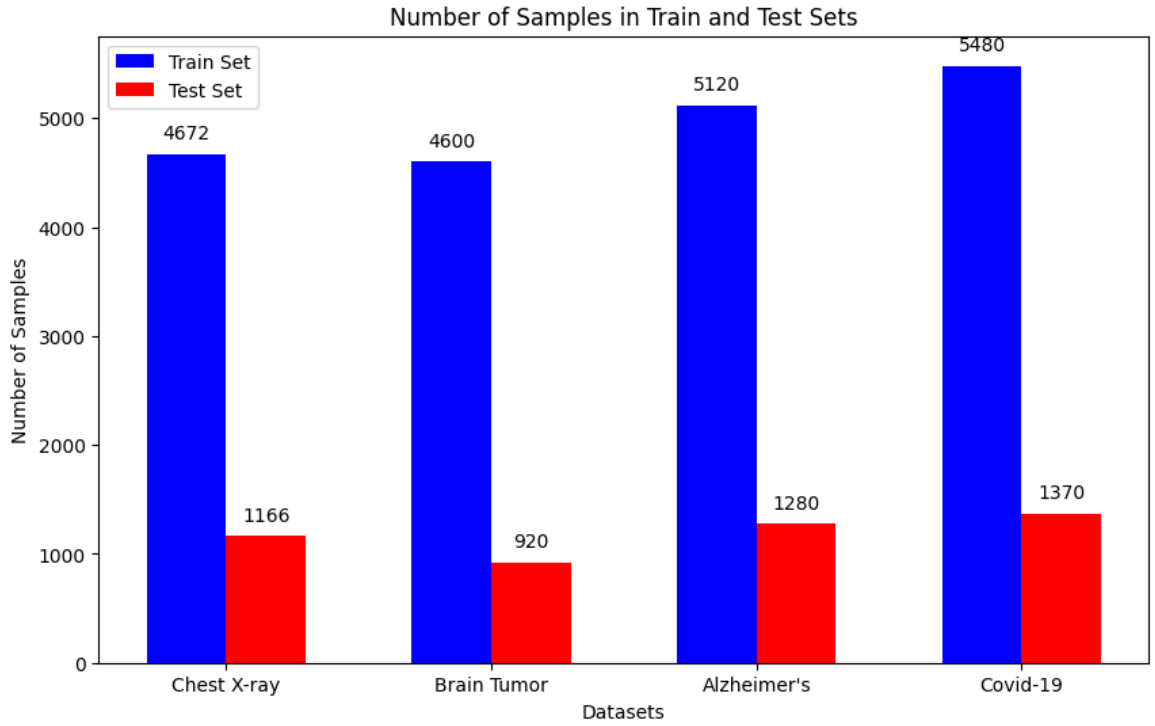



Fig. 4.6 flower-architecture [6].

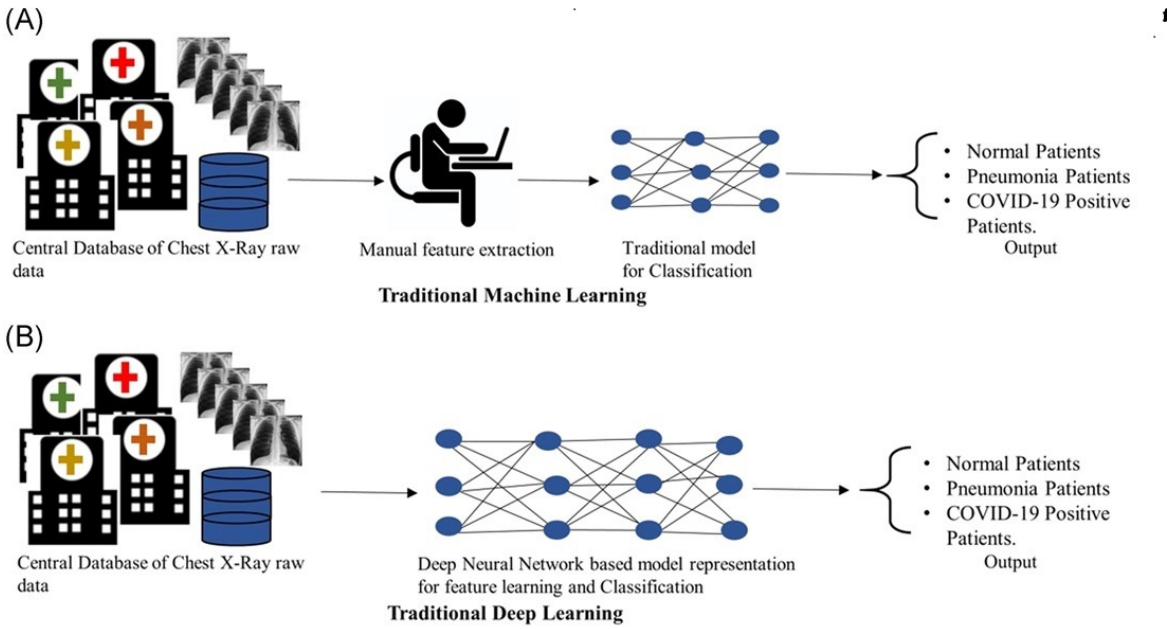



Fig. 4.7 Data-centralized Training

Instead, the model is trained locally on each client's data, and only the model updates (such as gradients or model parameters) are exchanged between the clients and the server.

To implement federated learning using Flower, we would follow the following steps:

#### 4.3.2.1 Data Preparation

We would start by preparing the data for training, this would involve pre-processing the MRI scans, such as normalizing the pixel values and resizing the images,and then split the data into multiple datasets, with each dataset representing a different hospital or clinic.

#### 4.3.2.2 Model Definition

We would define the machine learning model that we want to train, this would involve selecting the appropriate neural network architecture and defining the loss function and optimization algorithm.

#### 4.3.2.3 Client Implementation

We would implement the client-side of the federated learning system using Flower, this would involve writing code that would be run on each hospital or clinic's server, the client-side code would load the data for the hospital or clinic, train the machine learning model on the data, and then send the updated model parameters to the server.

#### 4.3.2.4 Server Implementation

We would implement the server-side of the federated learning system using Flower, this would involve writing code that would be run on a central server, the server-side code would receive the updated model parameters from the client-side code and aggregate them to produce a new global model, the server-side code would then send the updated global model parameters back to the client-side code for further training.

#### 4.3.2.5 Model Evaluation

We would evaluate the performance of the federated learning system by testing the global model on a held-out test dataset, this would allow us to assess the accuracy of the model and compare it to other machine learning models trained using traditional methods.

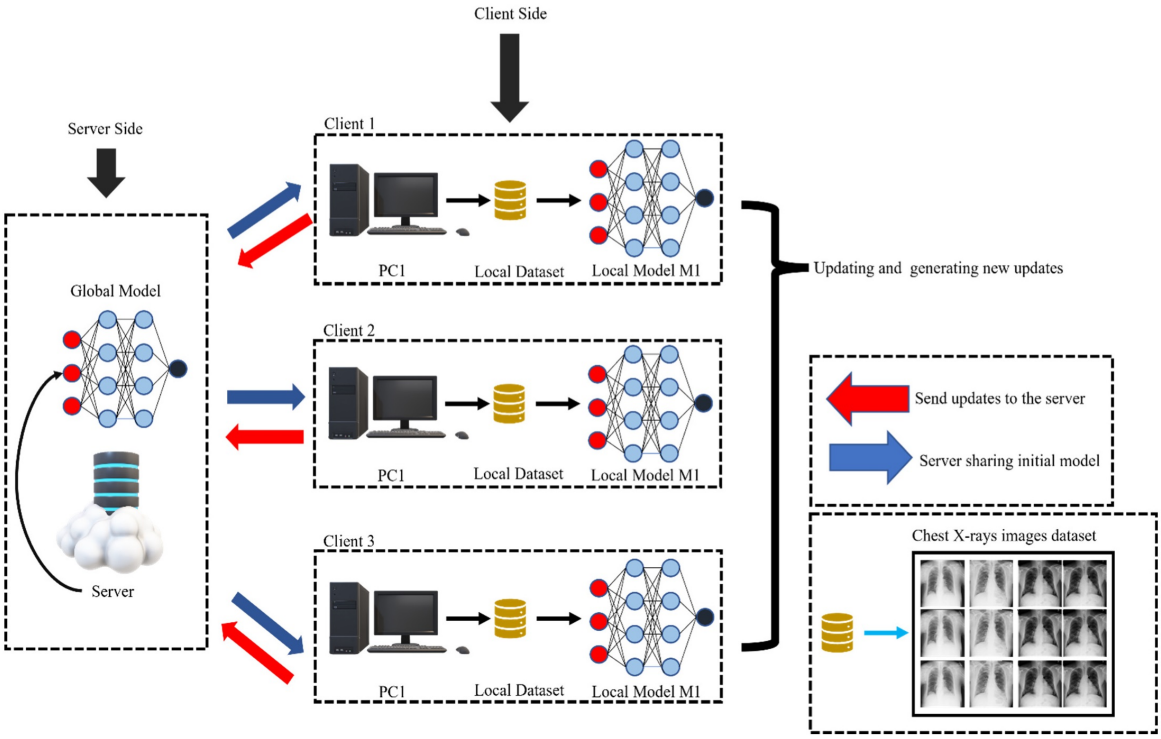



Fig. 4.8 Framework for FL in CXR image processing for multiple chest diseases.

Here, we simulate having multiple datasets from multiple hospitals organizations (also called the "cross-silo" setting in federated learning) by splitting the original dataset into multiple partitions, each partition will represent the data from a single organization, we're doing this purely for experimentation purposes, in the real world there's no need for data splitting because each organization already has their own data (so the data is naturally partitioned).

Each hospital will act as a client in the federated learning system, here example in Figure 4.9, we simulated five organizations participating in a federation means having five clients connected to the federated learning server and splitting the dataset into 5 clients.

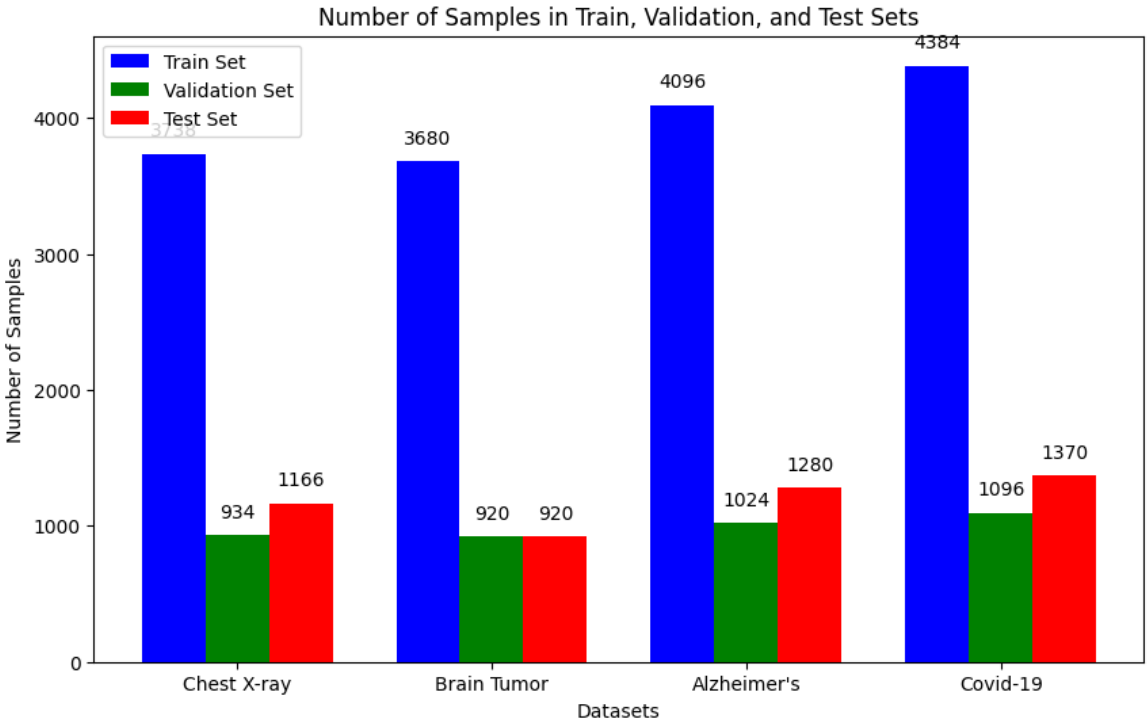



Fig. 4.9 datasets for 5 client

## 4.4 Colab GUI

Figure 4.10 shows the Colab GUI, this interface allows users to select the number of clients and dataset choice, load the datasets, train the model centrally, and perform federated learning using the Flower library, it provides an interactive way to experiment with federated learning in the healthcare domain.

It creates widgets using the ‘ipywidgets‘ library to create interactive elements in the interface, such as dropdowns and buttons.

The interface consists of several widgets and buttons that allow the user to interact with the system. Here’s an overview of the buttons and their functionality:

**Load Datasets Button**

- This button is used to load the datasets for federated learning.

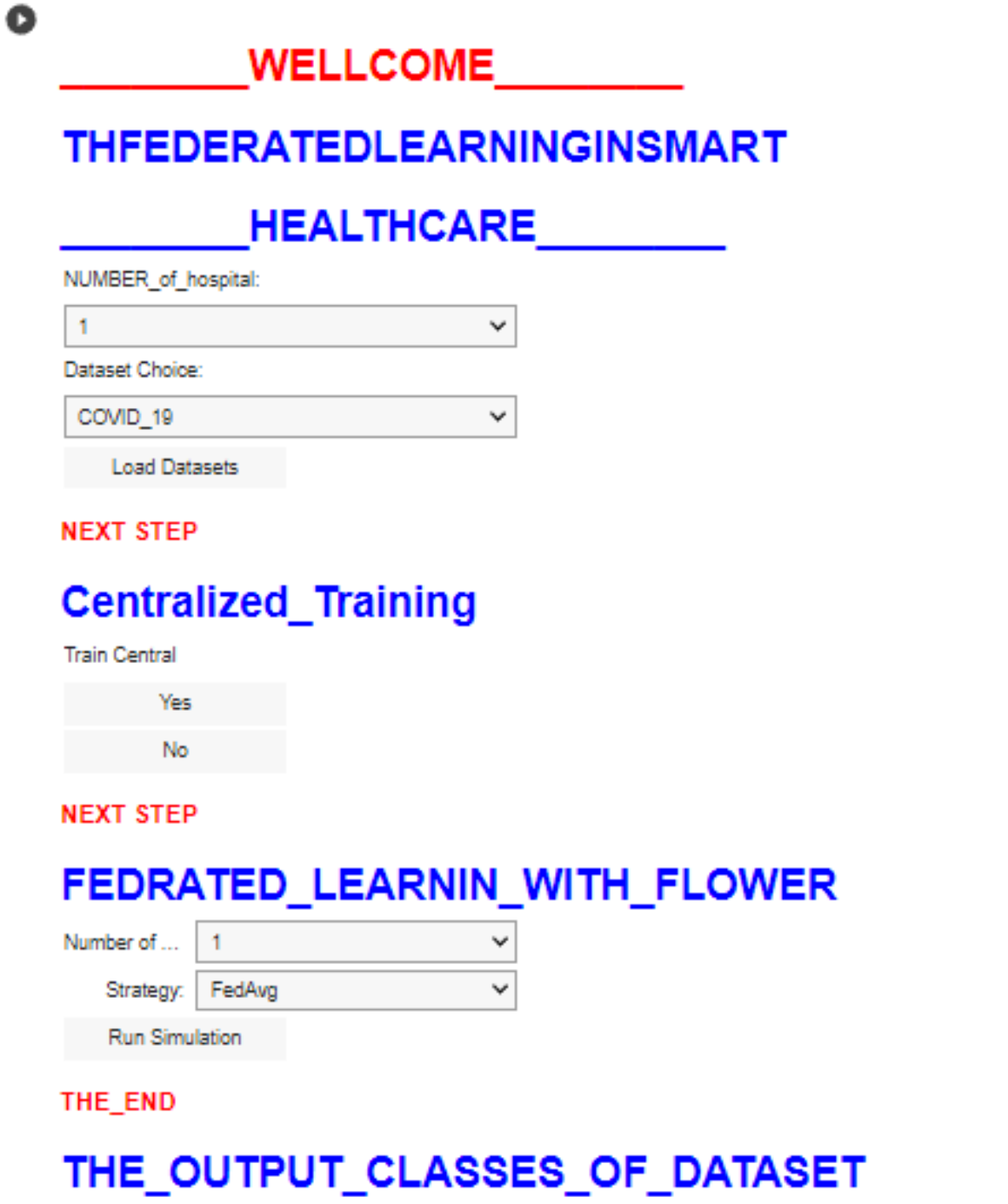



Fig. 4.10 Colab GUI

- When clicked, it triggers the ‘button clicked‘ function, which loads the datasets based on the selected number of clients and dataset choice.
- The datasets are split into training and validation sets, and DataLoader objects are created.
- The function also creates a neural network model based on the number of classes in the dataset.
- A sample of selected MRI datasets is displayed in the output widget.

**Train Central Yes/No Buttons**

- These buttons are used to choose whether to perform centralized training or not.
- Clicking the "Yes" button reveals an input field to enter the number of epochs and a "Train" button.
- Clicking the "No" button does not reveal any additional input fields.

**Train Button**

- This button is used to start the training process for the selected number of epochs.
- When clicked, it triggers the ‘on button train clicked’ function, which trains the neural network model for the specified number of epochs using the selected trainloade .
- The training progress is displayed in the output widget.

**Next Step Button**

- This button is used to proceed to the next step, which is federated learning with the FLOWER framework.
- Clicking the button displays a message indicating the next step.

**Federated Learning with Flower**

- The code for federated learning with FLOWER is provided and integrated with the interface.
- The code defines a ‘FlowerClient‘ class that serves as a client for federated learning using the FLOWER framework.
- It also includes functions for aggregating metrics and updating model parameters during federated learning.

## 4.5 Obtained Results

In this section, we present our obtained results for all four datasets following the two modes, data-centralized and federated with different aggregation strategies.

### 4.5.1 Data-Centralized Training

Table 4.2 shows the obtained accuracy and loss values of CNN models trained on the four dataset following a data-centralized training setting, the higher the accuracy, the better the model’s performance in correctly predicting or classifying the data.

| | Alzheimer | Chest X-ray | Covid-19 | Brain Tumor |
|---|---|---|---|---|
| Accuracy | 0.988 | 0.9962 | 0.9876 | 0.9994 |
| Loss | 0.0464 | 0.0116 | 0.0312 | 0.0033 |

Table 4.2 Centralized training

### 4.5.2 Federated Learning

In the following, we present the obtained accuracy and loss results after tuning the number of clients and rounds.

#### 4.5.2.1 Tuning Number of Clients

**Brain Tumor Dataset** Figure 4.11 and Figure 4.12 shows the obtained Accuracy value for different aggregation strategies with different numbers of clients applied on the Brain Tumor Dataset.

**Pneumonia Chest X-Ray Dataset** Figure 4.13 and Figure 4.14 shows the obtained Accuracy value for different aggregation strategies with different numbers of clients applied on the Pneumonia Chest X-Ray Dataset.

**Covid-19 Radiography Dataset** Figure 4.15 and Figure 4.16 shows the obtained Accuracy value for different aggregation strategies with different numbers of clients applied on the Covid-19 Radiography Dataset.

**Alzheimer's Dataset** Figure 4.17 and Figure 4.18 shows the obtained Accuracy value for different aggregation strategies with different numbers of clients applied on the Alzheimer's Dataset.

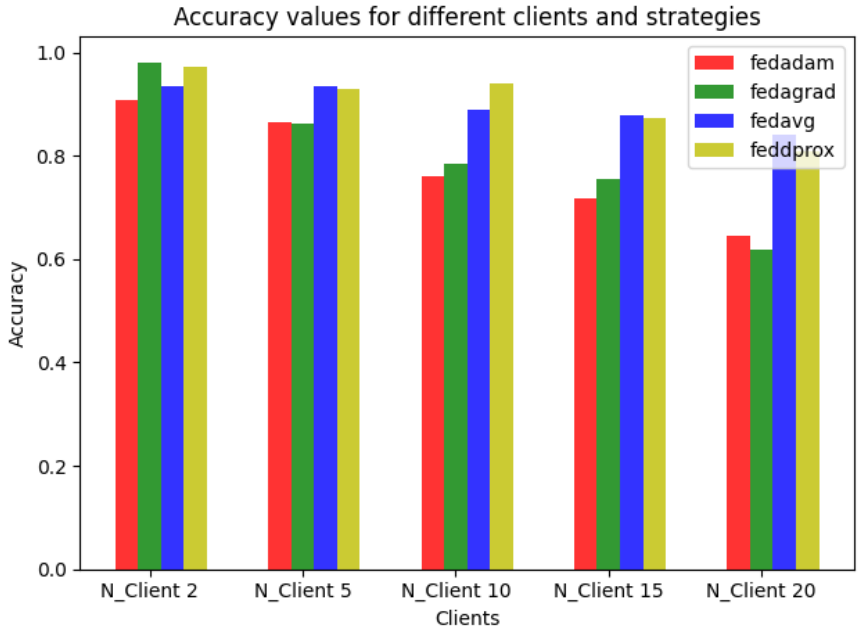



Fig. 4.11 Brain Tumor Dataset's accuracy

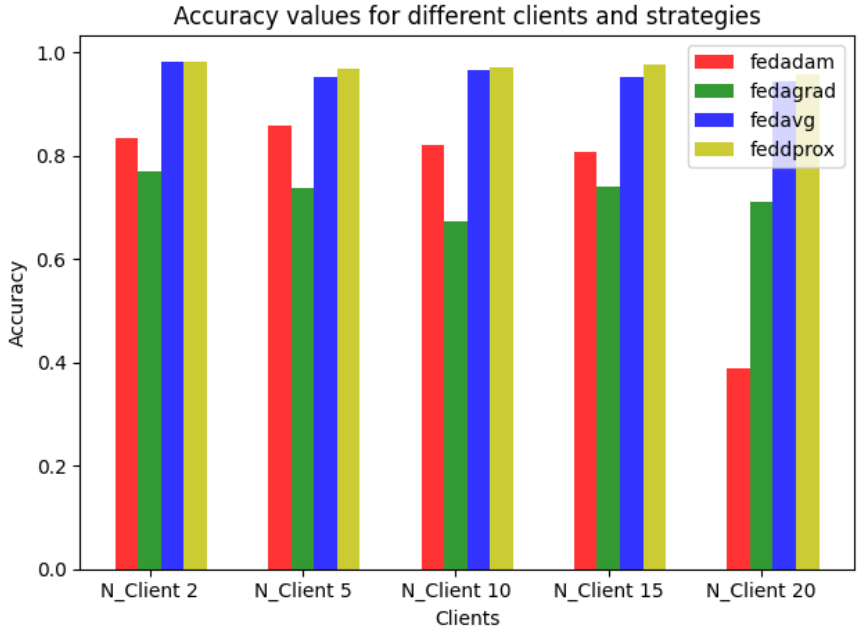



Fig. 4.12 Pneumonia Chest X-Ray Dataset's accuracy

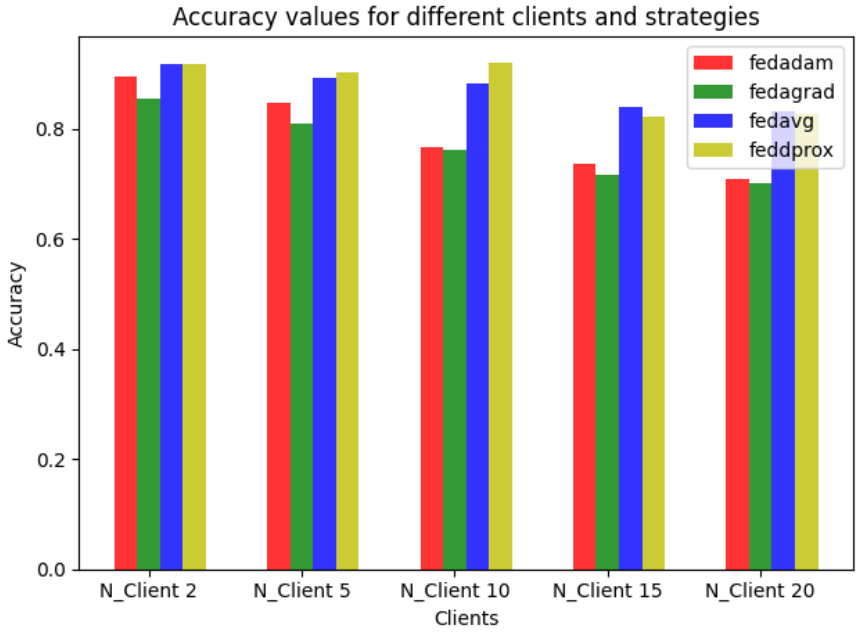



Fig. 4.13 Covid-19 Radiography Dataset's accuracy

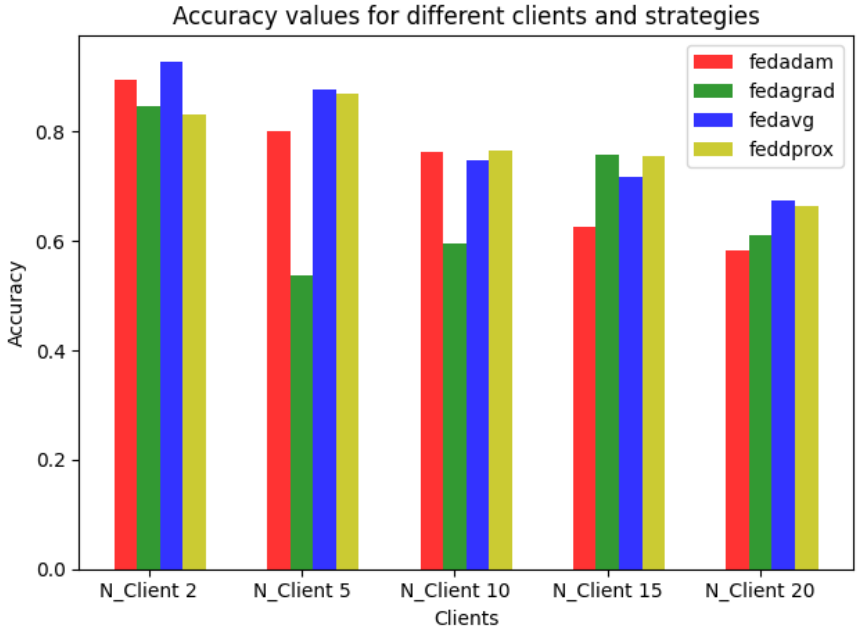
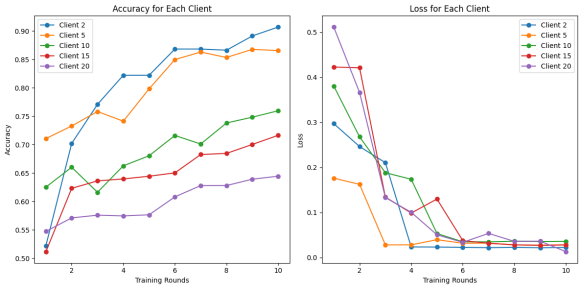
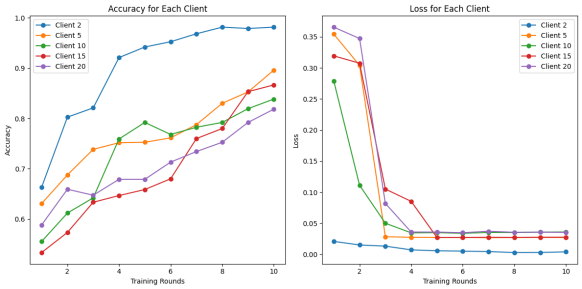



Fig. 4.14 Alzheimer's Dataset's accuracy

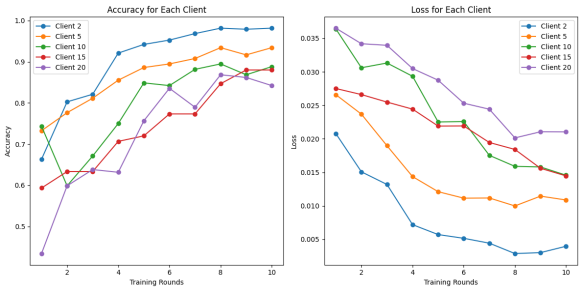
#### 4.5.2.2 Tuning Number of Rounds



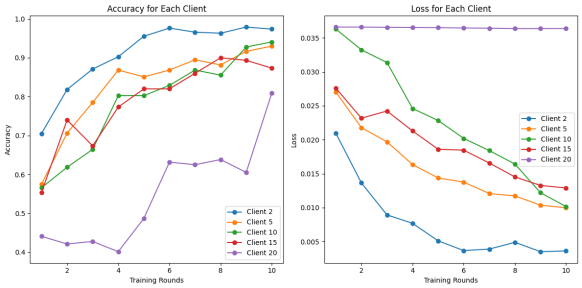

(a) fedadam strategy




(b) FedAdagrad strategy

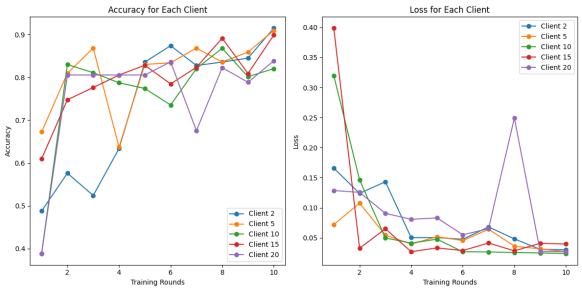



(c) fedavg strategy

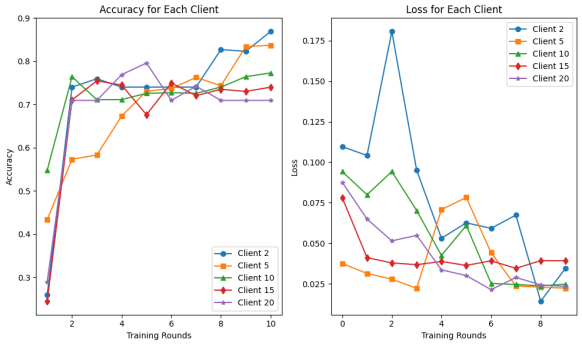



(d) FedProx strategy

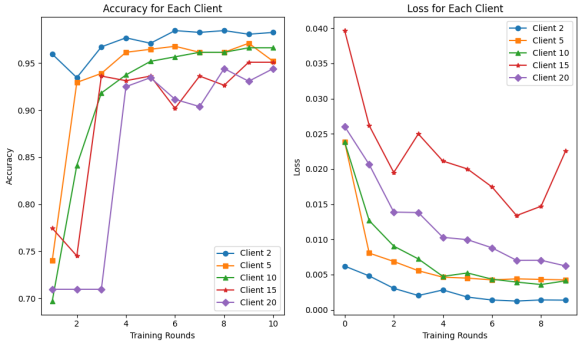
Fig. 4.15 Brain Tumor Dataset by the strategies



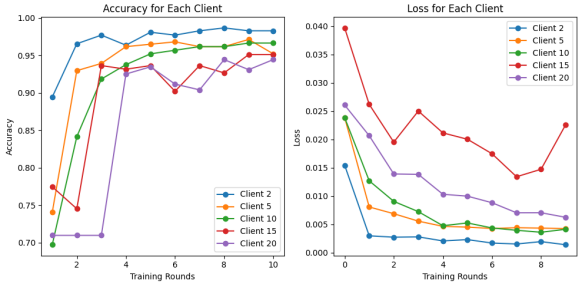

(a) fedadam strategy




(b) FedAdagrad strategy

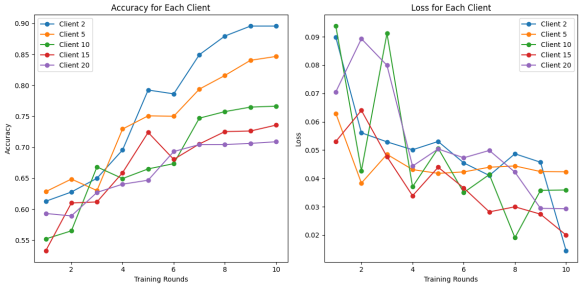



(c) fedavg strategy

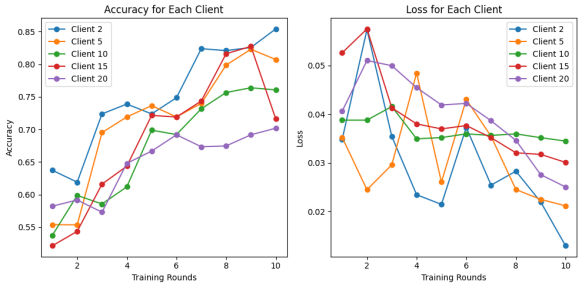



(d) FedProx strategy

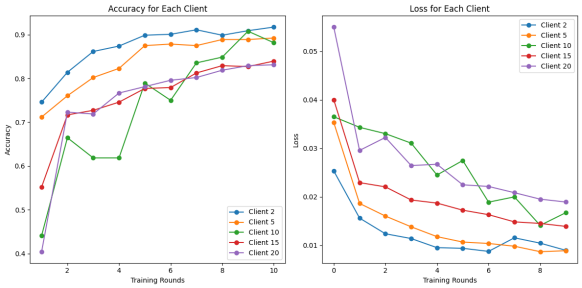
Fig. 4.16 Pneumonia Chest X-Ray Dataset by the strategies



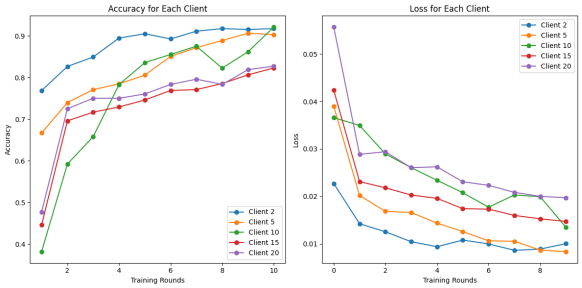

(a) fedadam strategy




(b) FedAdagrad strategy

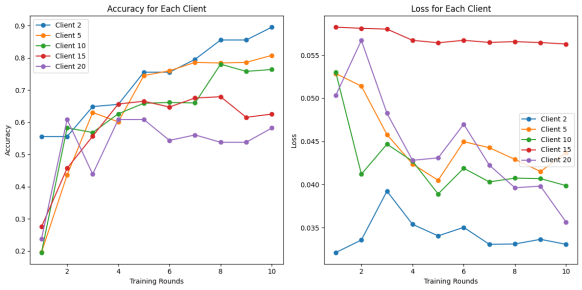



(c) fedavg strategy

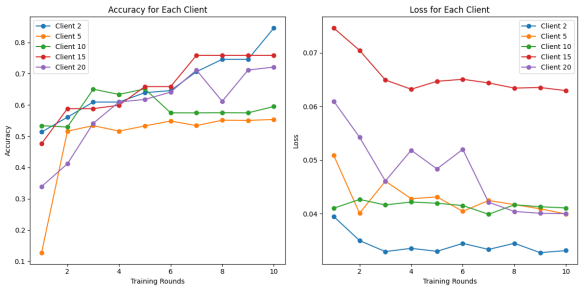



(d) FedProx strategy

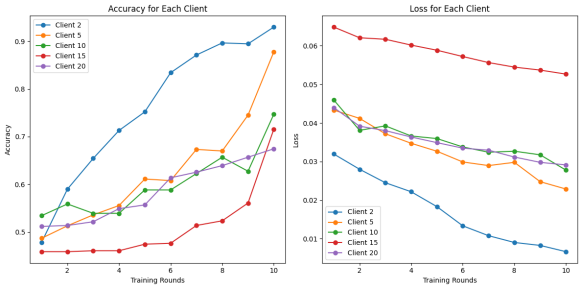
Fig. 4.17 Covid-19 Radiography Dataset by the strategies



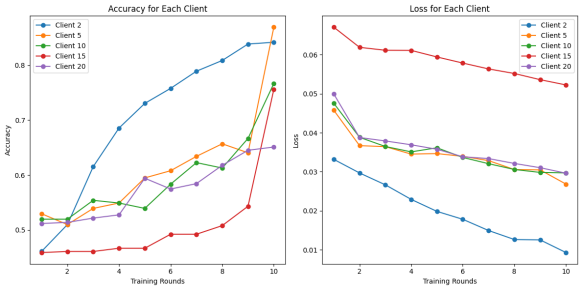

(a) fedadam strategy




(b) FedAdagrad strategy




(c) fedavg strategy




(d) FedProx strategy

Fig. 4.18 Alzheimer's Dataset by the strategies

**Observations** Based on the provided plotted results, here are some observations about the strategies and training rounds:

1. **Brain dataset**:
    - The FedProx strategy consistently achieves the highest accuracy values among all strategies across different client numbers.
    - The FedAvg strategy also performs well, maintaining relatively high accuracy values across different client scenarios.
    - The FedAdam and FedAdagrad strategies show more fluctuating results, with decreasing accuracy as the number of clients increases.
2. **Chest dataset**:
    - The FedProx strategy consistently outperforms the other strategies in terms of accuracy.
    - The FedAvg strategy also demonstrates stable accuracy values across different client numbers, but not as high as FedProx.
    - The FedAdam and FedAdagrad strategies exhibit decreasing accuracy as the number of clients increases.
3. **Covid-19 dataset**:
    - The FedProx strategy remains the highest-performing strategy in terms of accuracy.
    - The FedAvg strategy shows consistent and relatively high accuracy values across different client scenarios.
    - The FedAdam and FedAdagrad strategies exhibit decreasing accuracy with increasing numbers of clients.

4. **Alzheimer's dataset**:

    - The FedProx strategy consistently achieves the highest accuracy values among all strategies.
    - The FedAvg strategy maintains stable accuracy values across different client numbers.
    - The FedAdam and FedAdagrad strategies show decreasing accuracy as the number of clients increases.

Based on the provided information, the best-fixed strategy would be FedProx for all the datasets: Brain, Chest, Covid-19, and Alzheimer's.

In all cases, FedProx consistently achieved the highest accuracy among all the strategies considered, and its accuracy remained stable across different numbers of clients, this indicates that FedProx is a robust and reliable strategy for federated learning.

FedAvg also performed well and maintained relatively stable accuracy across different client numbers, it can be considered as a good alternative if the implementation or computational requirements of FedProx pose any challenges.

On the other hand, FedAdam and FedAdagrad showed fluctuating accuracy values and decreasing trends as the number of clients increased, these strategies may have limitations in handling larger client populations and may not provide as consistent results as FedProx or FedAvg.

In summary, based on the provided accuracy values, the best-fixed strategy would be FedProx, followed by FedAvg, for achieving high accuracy in data-centralized training on the given datasets.

## 4.6 Discussion

### 4.6.1 Influence of Number of Clients

Based on the analysis of the influence of the number of clients in the strategys across different datasets (Covid-19, Brain Tumor, Pneumonia), we can draw the following global conclusions:

**1. Loss Improvement**: Increasing the number of clients generally leads to a decrease in the loss values, this suggests that aggregating data from more clients can help improve the overall model performance.

**2. Accuracy Improvement**: The accuracy values also show improvement as the number of clients increases, however, the rate of improvement tends to diminish as the number of clients increases further, after a certain point, the accuracy values stabilize, indicating that adding more clients may not result in significant accuracy gains.

**3. Diminishing Returns**: The results suggest that there are diminishing returns in performance improvement when increasing the number of clients, while adding more clients initially leads to substantial improvements, the rate of improvement slows down after a certain threshold, this implies that there is an optimal number of clients that strikes a balance between model performance and computational efficiency.

**4. Dataset Sensitivity**: It's important to note that the conclusions drawn are specific to the analyzed datasets, the performance impact of the number of clients can vary depending on the nature of the data and the learning task, different datasets may exhibit different patterns in terms of loss and accuracy improvements with an increasing number of clients.

Overall, the findings suggest that the number of clients plays a crucial role in the strategy's performance, increasing the number of clients can generally lead to better loss and accuracy values, but the benefits may diminish beyond a certain point, it is essential to strike a balance between the number of clients, computational resources, and desired model performance when designing a federated learning strategy.

### 4.6.2 Influence of Number of Rounds

1. **Accuracy**: The accuracy of all clients generally improves with an increasing number of training rounds, however, there are variations in the rate of improvement and the final accuracy achieved by each client.

2. **Client Performance**: Among the clients, Client 20 shows the lowest accuracy throughout the training rounds, while Client 2 demonstrates the highest accuracy, this suggests that different clients may have varying capabilities or data distributions, which can impact their model performance.

3. **Training Rounds**: The training was conducted for 10 rounds for all clients, it's worth noting that the accuracy and loss values continue to change across these rounds, indicating ongoing model updates and improvements with more training.

4. **Loss**: The loss values for all clients generally decrease as the training progresses, this indicates that the model is learning and making better predictions over time.

5. **Convergence**: Some clients, such as Client 5 and Client 10, seem to converge relatively quickly, with their accuracy and loss stabilizing after a few rounds, on the other hand, clients like Client 2 and Client 20 continue to show significant changes in accuracy and loss even after 10 rounds.

6. **Comparative Performance**: Comparing the clients, Client 2 achieves the highest accuracy and has relatively lower loss values compared to the other clients, in contrast, Client 20 consistently exhibits lower accuracy and higher loss throughout the training rounds.

in summary, the influence of the number of rounds in the strategies is that increasing the number of rounds generally leads to improvements in loss reduction and accuracy for

the participating clients, however, there is a point of diminishing returns, after which the improvements become marginal, and the loss and accuracy values tend to stabilize.

The specific number of rounds required to reach stability may vary depending on the dataset and the number of clients involved in the federated learning process, it is important to monitor the loss and accuracy trends for each dataset and the client to determine the optimal number of rounds for training.

Overall, the results suggest that a sufficient number of rounds is necessary to allow the model to converge and capture the patterns and features of the data, however, blindly increasing the number of rounds beyond the point of diminishing returns may not yield significant improvements and could lead to unnecessary computational costs, therefore, it is crucial to strike a balance between the number of rounds and the achieved performance to optimize the federated learning process using the strategies.

### 4.6.3 Comparison of Strategies

Here is a summary of the comparison of the four federated learning strategies: FedAdam, FedAvg, FedAdagrad, and FedProx:

1. **FedAdam**:
    - Achieves the highest accuracy for 2 clients and gradually decreases as the number of clients increases.
    - Shows a decreasing trend in accuracy as the number of clients increases.
    - Performs relatively well with fewer clients but may struggle with larger client populations.
2. **FedAdagrad**:
    - Demonstrates fluctuating accuracy values with different numbers of clients.

- Achieves the highest accuracy with 2 clients and shows a slight decrease as the number of clients increases.
- Accuracy values fluctuate considerably between different client numbers.

3. **FedAvg**:
   - Exhibits relatively consistent accuracy values across different numbers of clients.
   - Achieves high accuracy with all client numbers, with a slight decrease as the number of clients increases.
   - Performs reasonably well but not as consistently high as FedProx.
4. **FedProx**:
   - Shows consistently high accuracy values among all the strategies considered.
   - Achieves consistently high accuracy across different numbers of clients, with a slight decrease as the number of clients increases.
   - Outperforms other strategies in terms of accuracy.

Based on these observations, FedProx stands out as the best strategy in terms of accuracy across all datasets, it consistently achieves high accuracy values and outperforms the other strategies. FedAvg also performs well, with relatively stable accuracy values, but it does not reach the same level of accuracy as FedProx. FedAdam and FedAdagrad show more fluctuating accuracy values and may struggle with larger client populations.

In conclusion, the comparison of strategies indicates that FedProx is the most effective strategy for data-centralized training, followed by FedAvg.

### 4.6.4 Comparison of Centralized vs Federated

After comparing the results of the data-centralized training in table with the previously plotted accuracy values for federated learning, we can make some observations:

1. **Accuracy**: the accuracy values obtained from data-centralized training are higher than those obtained from federated learning, for example, in the brain tumor and Pneumonia Chest X-Ray dataset, the accuracy achieved through data-centralized training is 0.99, while the highest accuracy achieved through federated learning was 0.98 Similarly, in Covid-19 dataset, the accuracy of 0.9962 in data-centralized training outperforms the highest accuracy of 0.91735 achieved through federated learning.

   while Alzheimer's dataset through data-centralized training is 0.988 and highest accuracy achieved through federated learning was 0.92.

2. **Loss**: The loss values obtained from data-centralized training are generally lower compared to federated learning, lower loss values indicate better model convergence and fit to the data.

   For example, in the Brain Tumor dataset, the loss of 0.0033 in data-centralized training is significantly lower than the highest loss of achieved through federated learning.

3. **Centralized vs Federated**: Data-centralized training often benefits from having access to the entire dataset during model training, resulting in higher accuracy and lower loss, however, it comes with the drawback of requiring centralized data, which may raise privacy concerns or practical limitations in certain scenarios.

   Federated learning, on the other hand, allows training models collaboratively on distributed data without the need for data centralization, although federated learning may result in slightly lower accuracy and higher loss due to the limitations of local data availability and communication constraints, it offers privacy preservation and decentralized benefits.

Our obtained experimental results reveal that the accuracy of the FL models is less than the data-centralized model for all four strategies.

Overall, the choice between data-centralized training and federated learning depends on the specific requirements and constraints of the application, including privacy concerns, data distribution, communication limitations, and the desired trade-off between accuracy and decentralization.

## 4.7 Related Work

Table 4.2 provides a summary of related work investigating the effectiveness of FL with respect to traditional data-centralized approaches.

These studies employed various machine learning algorithms, smart wearables, recurrent neural networks, and NLP techniques, the datasets used included electronic health records, distributed biomedical data, hospital data, and collaborative research datasets, the researchers addressed challenges related to data privacy, model accuracy, data heterogeneity, and the need for large-scale implementation, these works provide valuable insights into the potential applications and limitations of federated learning in healthcare.

Our project considerably contributes to this growing body of research by implementing FL strategies using the Flower framework, we compare the performance of these strategies on four different datasets and provide insights into the impact of FL on accuracy and data privacy in the context of smart healthcare.

| Study | Ref | Year | Content |
|---|---|---|---|
| Huang et al. | [24] | 2022 | FL-based machine learning algorithm and distributed data clustering while maintaining data privacy. Application in predicting mortality and duration of hospital stay using electronic health records (EHR). |
| Chen et al. | [13] | 2020 | Data aggregation using FL and smart wearables. Deployment of FL framework in smartphones for activity recognition and data collection while maintaining privacy concerns. Novel framework to introduce FL with smart wearable devices. |
| Silva et al. | [56] | 2019 | Analysis of distributed biomedical data using a federated framework. FL framework to gain secured access and meta-analysis of medical datasets concealing patient information. Successful application towards the study of subcortical brain changes. |
| Federated Learning for Smart Healthcare: A Survey | [37] | 2021 | This survey article reviews and surveys the application of FL in smart healthcare and IoT devices, which aligns with your investigation of FL in the context of Smart Healthcare Systems (SHSs). |
| Federated Learning in a Medical Context: A Systematic Literature Review | [40] | 2021 | This survey article examines federated learning and its relevance to sensitive healthcare data, which aligns with your investigation of FL in the context of SHSs and the preservation of patients' privacy. |
| Our Project | | 2023 | FL with Flower framework. Brain tumor, Pneumonia, Alzheimer's disease, Covid-19, the accuracy of the federated learning models was consistently lower than that of the data-centralized model for all four strategies, this suggests that challenges associated with decentralized learning in smart healthcare settings, such as data heterogeneity and variability in local datasets, may hinder the performance of federated learning models. |

Table 4.3 Studies on Federated Learning in Smart Healthcare

## 4.8 Conclusion

The comparison of centralized training and federated learning strategies reveals several key insights:

- The analysis of the influence of the number of rounds in the strategies indicates that increasing the number of rounds generally leads to improvements in loss reduction and accuracy for the participating clients, however, there is a point of diminishing returns, after which the improvements become marginal, and the loss and accuracy values tend to stabilize, it is important to monitor the loss and accuracy trends for each dataset and client to determine the optimal number of rounds for training.

- Regarding the influence of the number of clients, increasing the number of clients generally leads to a decrease in the loss values and improvement in accuracy, however, the rate of improvement diminishes as the number of clients increases further, there are diminishing returns in performance improvement when increasing the number of clients, and there is an optimal number of clients that balances model performance and computational efficiency.

- In the comparison of federated learning strategies, FedProx stands out as the most effective strategy in terms of accuracy, consistently achieving high accuracy values. FedAvg also performs well but not as consistently high as FedProx

  FedAdam and FedAdagrad show more fluctuating accuracy values and may struggle with larger client populations.

- Comparing data-centralized training with federated learning, data-centralized training often achieves higher accuracy and lower loss values, however, federated learning offers privacy preservation and decentralized benefits, making it a suitable approach in scenarios with privacy concerns or limitations on centralized data.

In summary, the choice between data-centralized training and federated learning depends on the specific requirements and constraints of the application, including privacy concerns, data distribution, communication limitations, and the desired trade-off between accuracy and decentralization.

# Chapter 5

# Conclusion

## 5.1 Concluding Remarks

The evaluation of different federated learning strategies, including FedAdagrad, FedProx, FedAdam, and FedAvg, reveals important insights based on both the state of the art and the obtained results.

Considering the state of the art, federated learning approaches have shown promise in enabling collaborative training across decentralized devices or servers while preserving data privacy, however, the performance of these strategies can be influenced by various factors, including the nature of the tasks and the optimization algorithms employed.

Based on the obtained results, it is evident that the performance of the federated learning strategies falls short compared to centralized training, the FedAdagrad strategy struggles to capture complex patterns and nuances in the data, leading to lower performance.

Similarly, the FedAdam strategy, which utilizes the Adam optimizer, proves to be less effective than centralized training in achieving high accuracy for the given tasks.

FedProx emerges as the superior strategy when considering accuracy, consistently delivering exceptional results. FedAvg also performs commendably, although not with the same consistent high accuracy as FedProx.

Our obtained experimental results reveal that the accuracy of the federated learning models is less in general than the data-centralized model for all four strategies, however, such a small extent can be tolerated considering the advantages of federated learning brought to Smart Healthcare Systems (SHSs).

However, it is crucial to note that these conclusions are specific to the results obtained in the evaluation and are subject to the limitations of the datasets and experimental setup, to draw more definitive conclusions, further analysis and evaluation on larger and more diverse datasets are required.

Overall, while federated learning holds promise for collaborative training in decentralized environments, optimizing the strategies and algorithms to achieve comparable performance with centralized training remains a challenge.

Further research and experimentation are necessary to overcome these limitations and fully harness the potential of federated learning in various applications.

## 5.2 Future Work

Our study has shed light on FL limitations and potential for improvement, thus, several directions for future work can be explored to advance the field of federated learning:

- **Algorithmic Optimization**: Future research could focus on developing novel optimization algorithms specifically tailored for federated learning, these algorithms should address the challenges of capturing complex patterns and nuances in decentralized data, ultimately improving the performance of federated learning strategies.

- **Data Heterogeneity**: Federated learning often deals with heterogeneous data across different devices or servers, future work could investigate techniques to effectively handle data heterogeneity, such as adaptive weighting schemes or data augmentation strategies, to enhance the performance of federated learning models.

- **Communication Efficiency**: Communication during the federated learning process can be a significant bottleneck, especially when dealing with a large number of devices or servers, exploring techniques to reduce communication overhead, such as compression algorithms or selective updates would improve the scalability and efficiency of federated learning.

- **Privacy and Security**: Privacy and security are critical concerns in federated learning, future work could focus on developing robust privacy-preserving mechanisms, such as secure aggregation or differential privacy, to ensure data privacy while maintaining the effectiveness of federated learning.

- **Task-Specific Optimization**: Different tasks may require tailored optimization approaches in federated learning,future research could investigate task-specific optimization strategies to improve the performance of federated learning models for specific domains or applications.

- **Real-World Deployment**: Evaluating federated learning strategies in real-world scenarios with large-scale and diverse datasets would provide valuable insights into their practicality and effectiveness, future work could focus on deploying federated learning systems in various domains, such as healthcare or finance, and evaluate their performance in real-world settings.

By addressing these future research directions, the field of federated learning can continue to evolve and overcome current limitations, leading to more robust and efficient approaches for collaborative and privacy-preserving machine learning in decentralized environments.

Furthermore, in an effort to promote knowledge sharing and facilitate further research in the field of federated learning, the code for this project has been made publicly available on GitHub [57].

By sharing the code, we hope that students and researchers can leverage it as a valuable resource for their own studies and experiments. This open availability can foster collaboration, accelerate the development of new techniques, and encourage the exploration of novel ideas in federated learning.

# References


[1] (2018). Structure of the ebola virus glycoprotein bound to an antibody from a human survivor. Accessed on May 5, 2023.

[2] (2023). Bimcv-covid19 dataset. Accessed on May 5, 2023.

[3] (2023). Covid-19 image repository. Accessed on May 5, 2023.

[4] (2023). Covid-19 radiography database. Accessed on May 5, 2023.

[5] (2023). Covid-cxnet. Accessed on May 5, 2023.

[6] (2023). Flower documentation. Accessed on June 1, 2023.

[7] (2023). Society of medical radiology. Accessed on May 5, 2023.

[8] (Year). Worldometer. Accessed on May 31, 2023.

[9] Ahmed, I., Ahmad, A., and Jeon, G. (2020). An iot-based deep learning framework for early assessment of covid-19. *IEEE Internet of Things Journal*, 8(21):15855–15862.

[10] Ali, F., El-Sappagh, S., Islam, S. R., Kwak, D., Ali, A., Imran, M., and Kwak, K.-S. (2020). A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion*, 63:208–222.

[11] Allugunti, V. R., Kishor Kumar Reddy, C., Elango, N., and Anisha, P. (2021). Prediction of diabetes using internet of things (iot) and decision trees: Sldps. In *Intelligent Data Engineering and Analytics: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), Volume 2*, pages 453–461. Springer.

[12] Baig, M. M. and Gholamhosseini, H. (2013). Smart health monitoring systems: an overview of design and modeling. *Journal of medical systems*, 37:1–14.

[13] Chen, Y., Qin, X., Wang, J., Yu, C., and Gao, W. (2020). Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93.

[14] De Aguiar, E. J., Faiçal, B. S., Krishnamachari, B., and Ueyama, J. (2020). A survey of blockchain-based strategies for healthcare. *ACM Computing Surveys (CSUR)*, 53(2):1–27.

[15] Deperlioglu, O., Kose, U., Gupta, D., Khanna, A., and Sangaiah, A. K. (2020). Diagnosis of heart diseases by a secure internet of health things system based on autoencoder deep neural network. *Computer Communications*, 162:31–50.

[16] DevFi (2023). Applications of federated learning in healthcare. https://www.devfi.com/applications-of-federated-learning-in-healthcare/. Accessed on June 5, 2023.

[17] Dinh C, N., Quoc-Viet, P., Pathirana, P. N., Ming, D., Seneviratne, A., Zihuai, L., Dobre, O., Won-Joo, H., et al. (2023). Federated learning for smart healthcare: A survey. *ACM Computing Surveys*, pages 1937–01.

[18] Efat, M. I. A., Rahman, S., and Rahman, T. (2020). Iot based smart health monitoring system for diabetes patients using neural network. In *Cyber Security and Computer Science: Second EAI International Conference, ICONCS 2020, Dhaka, Bangladesh, February 15-16, 2020, Proceedings 2*, pages 593–606. Springer.

[19] El-Rashidy, N., El-Sappagh, S., Islam, S. R., El-Bakry, H. M., and Abdelrazek, S. (2020). End-to-end deep learning framework for coronavirus (covid-19) detection and monitoring. *Electronics*, 9(9):1439.

[20] Feki, I., Ammar, S., Kessentini, Y., and Muhammad, K. (2021). Federated learning for covid-19 screening from chest x-ray images. *Applied Soft Computing*, 106:107330.

[21] Ferrag, M. A., Friha, O., Maglaras, L., Janicke, H., and Shu, L. (2021). Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis. *IEEE Access*, 9:138509–138542.

[22] Geekflare (2023). Google colab: Free cloud-based jupyter notebooks. Website. Accessed on May 25, 2023.

[23] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. (2018). Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

[24] Huang, L., Shea, A. L., Qian, H., Masurkar, A., Deng, H., and Liu, D. (2019). Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *Journal of biomedical informatics*, 99:103291.

[25] Ji, S., Saravirta, T., Pan, S., Long, G., and Walid, A. (2021). Emerging trends in federated learning: From model fusion to federated x learning. *arXiv preprint arXiv:2102.12920*.

[26] Johri, P., Arvindhan, M., and Daniel, A. (2021). Enabling technologies: A transforming action on healthcare with iot a possible revolutionizing. *Artificial Intelligence for a Sustainable Industry 4.0*, pages 265–279.

[27] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210.

[28] Kaptoge, S., Pennells, L., De Bacquer, D., Cooney, M. T., Kavousi, M., Stevens, G., Riley, L. M., Savin, S., Khan, T., Altay, S., et al. (2019). World health organization cardiovascular disease risk charts: revised models to estimate risk in 21 global regions. *The Lancet Global Health*, 7(10):e1332–e1345.

[29] Khan, M. A. (2020). An iot framework for heart disease prediction based on mdcnn classifier. *IEEE Access*, 8:34717–34727.

[30] Khan, M. A. and Algarni, F. (2020). A healthcare monitoring system for the diagnosis of heart disease in the iomt cloud environment using msso-anfis. *IEEE Access*, 8:122259–122269.

[31] Konečnỳ, J., McMahan, H. B., Ramage, D., and Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*.

[32] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450.

[33] Mazumdar, M., Lin, J.-Y. J., Zhang, W., Li, L., Liu, M., Dharmarajan, K., Sanderson, M., Isola, L., and Hu, L. (2020). Comparison of statistical and machine learning models for healthcare cost data: a simulation study motivated by oncology care model (ocm) data. *BMC health services research*, 20:1–12.

[34] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

[35] Mei, X., Lee, H.-C., Diao, K.-y., Huang, M., Lin, B., Liu, C., Xie, Z., Ma, Y., Robson, P. M., Chung, M., et al. (2020). Artificial intelligence–enabled rapid diagnosis of patients with covid-19. *Nature medicine*, 26(8):1224–1228.

[36] Nasr, M., Islam, M. M., Shehata, S., Karray, F., and Quintana, Y. (2021). Smart healthcare in the age of ai: recent advances, challenges, and future prospects. *IEEE Access*, 9:145248–145270.

[37] Nguyen, D. C., Pham, Q.-V., Pathirana, P. N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O., and Hwang, W.-J. (2022). Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(3):1–37.

[38] Otoom, M., Otoum, N., Alzubaidi, M. A., Etoom, Y., and Banihani, R. (2020). An iot-based framework for early identification and monitoring of covid-19 cases. *Biomedical signal processing and control*, 62:102149.

[39] Paszke, A., Gross, S., Chintala, S., and Chanan, G. (2017). Automatic differentiation in pytorch.

[40] Pfitzner, B., Steckhan, N., and Arnrich, B. (2021). Federated learning in a medical context: A systematic literature review. *ACM Transactions on Internet Technology (TOIT)*, 21(2):1–31.

[41] Pham, Q.-V., Dev, K., Maddikunta, P. K. R., Gadekallu, T. R., Huynh-The, T., et al. (2021). Fusion of federated learning and industrial internet of things: a survey. *arXiv preprint arXiv:2101.00798*.

[42] Python Software Foundation (2020). What is python. https://www.python.org/doc/essays/blurb/. Accessed on May 1, 2023.

[43] Qammar, A., Ding, J., and Ning, H. (2022). Federated learning attack surface: taxonomy, cyber defences, challenges, and future directions. *Artificial Intelligence Review*, pages 1–38.

[44] Ramallo-González, A. P., González-Vidal, A., and Skarmeta, A. F. (2021). Ciotvid: Towards an open iot-platform for infective pandemic diseases such as covid-19. *Sensors*, 21(2):484.

[45] Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. (2020). Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

[46] Rghioui, A., Lloret, J., Sendra, S., and Oumnad, A. (2020a). A smart architecture for diabetic patient monitoring using machine learning algorithms. In *Healthcare*, volume 8, page 348. MDPI.

[47] Rghioui, A., Naja, A., Mauri, J. L., and Oumnad, A. (2021). An iot based diabetic patient monitoring system using machine learning and node mcu. In *Journal of Physics: Conference Series*, volume 1743, page 012035. IOP Publishing.

[48] Rghioui, A., Naja, A., and Oumnad, A. (2020b). Diabetic patients monitoring and data classification using iot application. In *2020 International Conference on Electrical and Information Technologies (ICEIT)*, pages 1–6. IEEE.

[49] Richens, J. G., Lee, C. M., and Johri, S. (2020). Improving the accuracy of medical diagnosis with causal machine learning. *Nature communications*, 11(1):3923.

[50] Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., et al. (2020). The future of digital health with federated learning. *NPJ digital medicine*, 3(1):119.

[51] Rodriguez, I. (2021). On the management of type 1 diabetes mellitus with iot devices and ml techniques. *arXiv preprint arXiv:2101.02409*.

[52] Saeedi, P., Petersohn, I., Salpea, P., Malanda, B., Karuranga, S., Unwin, N., Colagiuri, S., Guariguata, L., Motala, A. A., Ogurtsova, K., et al. (2019). Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the international diabetes federation diabetes atlas. *Diabetes research and clinical practice*, 157:107843.

[53] Saha, S. and Ahmad, T. (2021). Federated transfer learning: concept and applications. *Intelligenza Artificiale*, 15(1):35–44.

[54] Sarmah, S. S. (2020). An efficient iot-based patient monitoring and heart disease prediction system using deep learning modified neural network. *Ieee access*, 8:135784–135797.

[55] Selvaraj, S. and Sundaravaradhan, S. (2020). Challenges and opportunities in iot healthcare systems: a systematic review. *SN Applied Sciences*, 2(1):139.

[56] Silva, S., Gutman, B. A., Romero, E., Thompson, P. M., Altmann, A., and Lorenzi, M. (2019). Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. In *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*, pages 270–274. IEEE.

[57] Thameur, B. (2023). Fl-for-smart-healthcare. https://github.com/BThameur/FL-for-Smart-Healthcare. Accessed on June 5, 2023.

[58] Tourist55 (2023). Alzheimer's dataset: 4 class of images. Accessed on May 23, 2023.

[59] Viradiya, P. (2023). Brain tumor dataset. https://www.kaggle.com/datasets/preetviradiya/brian-tumor-dataset. Accessed on May 5, 2023.

[60] Wang, X.-Z. (2010). International journal of machine learning and cybernetics. *International Journal of Machine Learning and Cybernetics*, 1(1-4):1–2.

[61] Zeadally, S., Siddiqui, F., Baig, Z., and Ibrahim, A. (2020). Smart healthcare: Challenges and potential solutions using internet of things (iot) and big data analytics. *PSU research review*, 4(2):149–168.

## Abstract

Federated Learning (FL) has emerged as a promising approach for training Machine Learning (ML) models in decentralized settings, such as Smart Healthcare Systems (SHSs). Adoption of FL in SHSs provides many advantages especially the preservation of patient's privacy.

This project aims to investigate the effectiveness of FL in the context of SHSs by comparing it with the traditional data-centralized approach. Specifically, four FL strategies are implemented using the Flower framework, with each strategy trained on four different types of datasets: brain tumor, pneumonia, Alzheimer's disease, and COVID-19. The performance of these FL strategies is then compared to that of a data-centralized model trained on the same datasets.

Our obtained experimental results reveal that the accuracy of the FL models is in general less than the data-centralized model for all four strategies. Such disparity in accuracy is explained by the challenges associated with decentralized FL learning in SHS settings.

Nevertheless, such a small extent of difference can be tolerated considering the benefits of FL brought to SHSs. Thus, we may conclude that there should be a trade-off between high accuracy and FL advantages.

**Keywords:** Machine Learning (ML), Federated Learning (FL), Smart Healthcare Systems (SHSs), Data Privacy, Decentralized Learning, Models Aggregation Methods.

---

## ملخص

برز التعلم الفيدرالي (FL) كنهج واعد لتدريب نماذج التعلم الآلي (ML) في البيئات اللامركزية، مثل أنظمة الرعاية الصحية الذكية (SHS). يوفر اعتماد FL في SHS العديد من المزايا خاصة الحفاظ على خصوصية المرضى.

يهدف هذا المشروع إلى التحقيق في فعالية FL في سياق SHS من خلال مقارنتها بالنهج التقليدي المتمحور حول البيانات. على وجه التحديد، يتم تنفيذ أربع استراتيجيات FL باستخدام إطار Flower، مع تدريب كل إستراتيجية على أربعة أنواع مختلفة من مجموعات البيانات: ورم المخ، والالتهاب الرئوي، ومرض الزهايمر، و COVID-19. ثم تتم مقارنة أداء استراتيجيات FL هذه بأداء نموذج مركزية البيانات تم تدريبه على نفس مجموعات البيانات.

تكشف النتائج التجريبية التي حصلنا عليها أن دقة نماذج FL بشكل عام أقل من نموذج البيانات المركزي لجميع الاستراتيجيات الأربع. يتم تفسير هذا التباين في الدقة من خلال التحديات المرتبطة بالتعلم اللامركزي لـ FL في إعدادات SHS، مثل عدم تجانس البيانات، وقيود الاتصال، والتنوع في مجموعات البيانات المحلية.

ومع ذلك، يمكن تحمل مثل هذا النطاق الضئيل من الاختلاف مع الأخذ في الاعتبار فوائد FL التي يتم جلبها إلى SHSs. وبالتالي، قد نستنتج أنه يجب أن يكون هناك مفاضلة بين الدقة العالية ومزايا FL.

**كلمات مفتاحية:** التعلم الآلي (ML)، التعلم الموحد (FL)، أنظمة الرعاية الصحية الذكية (SHS) ، خصوصية البيانات ، التعلم اللامركزي ، طرق التجميع.