FEDERATED LEARNING WITH HETEROGENEOUS DATA FORMAT

Under the guidance of

Prof Mrinal Kanti Das
Department of Data Science, IIT Palakkad

-Santhosh V(142302020)
-Siva Kumar(142302008)

Quick Outline

- What is Federated Learning
- Objective
- Applications
- Types Of Federated Learning (Data Partitioning)
- Types Of Federated Learning(Based On Client)
- Aggregation Algorithms
- Results & Observations
- References

Shift from centralized data to decentralized data

The standard ML considers a centralized dataset processed in tightly integrated system.

- Sending data to cloud for centralized ML is too costly

  - Self driving cars generate several TBs of data every day

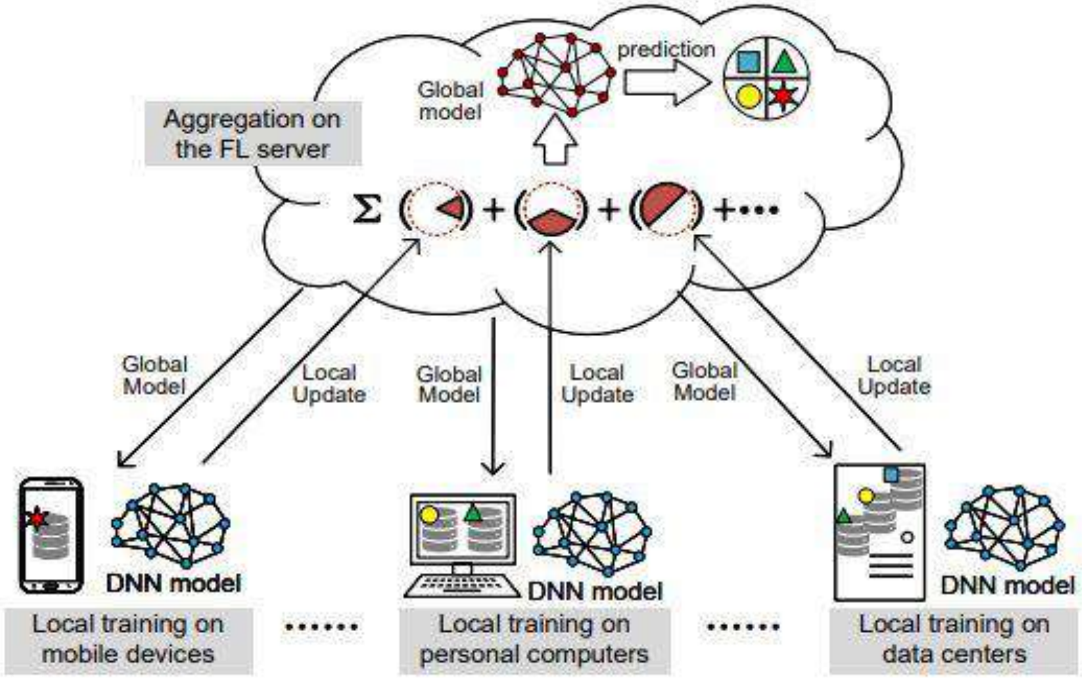
  - Some wireless networks have limited bandwidth

- Dats is too sensitive(medical reports)

  - Data privacy

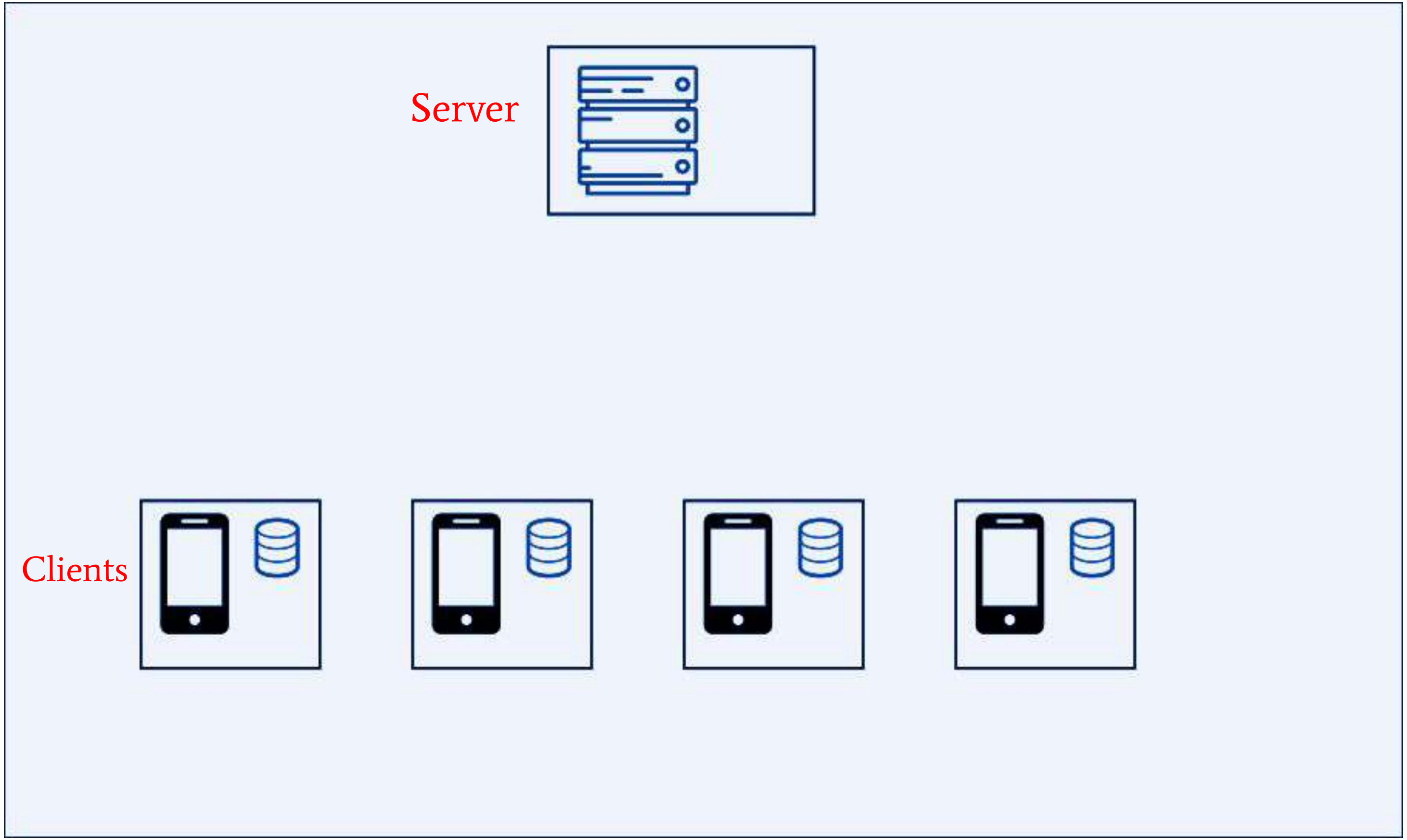
  - Keeping the control over data and give the competitive advantage in business and research

What is Federated Learning?

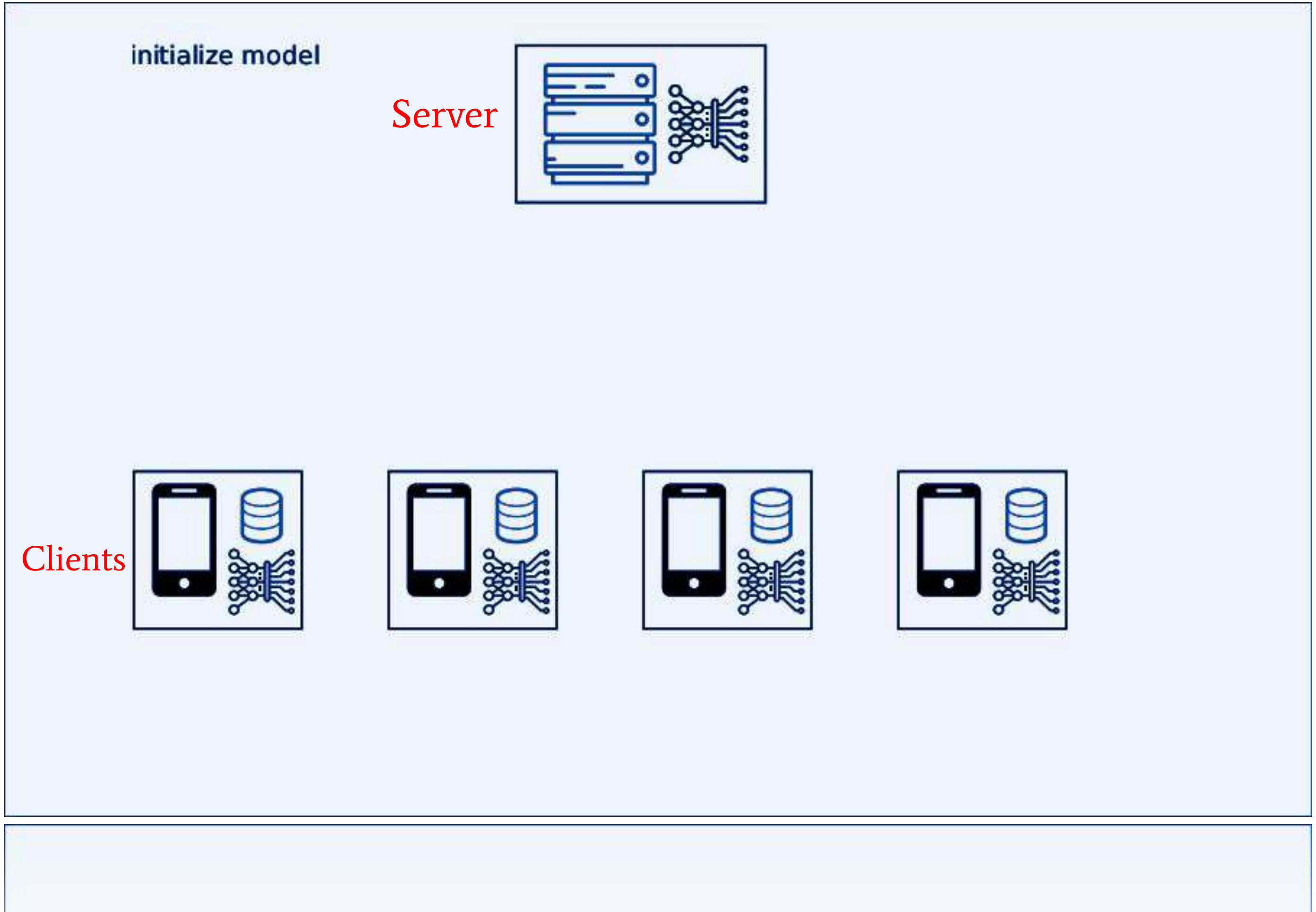

Aggregation on the FL server
Global model
prediction
Σ
Global Model
Local Update
Global Model
Local Update
Global Model
Local Update
DNN model
DNN model
DNN model
Local training on mobile devices
Local training on personal computers
Local training on data centers

Federated Learning Process

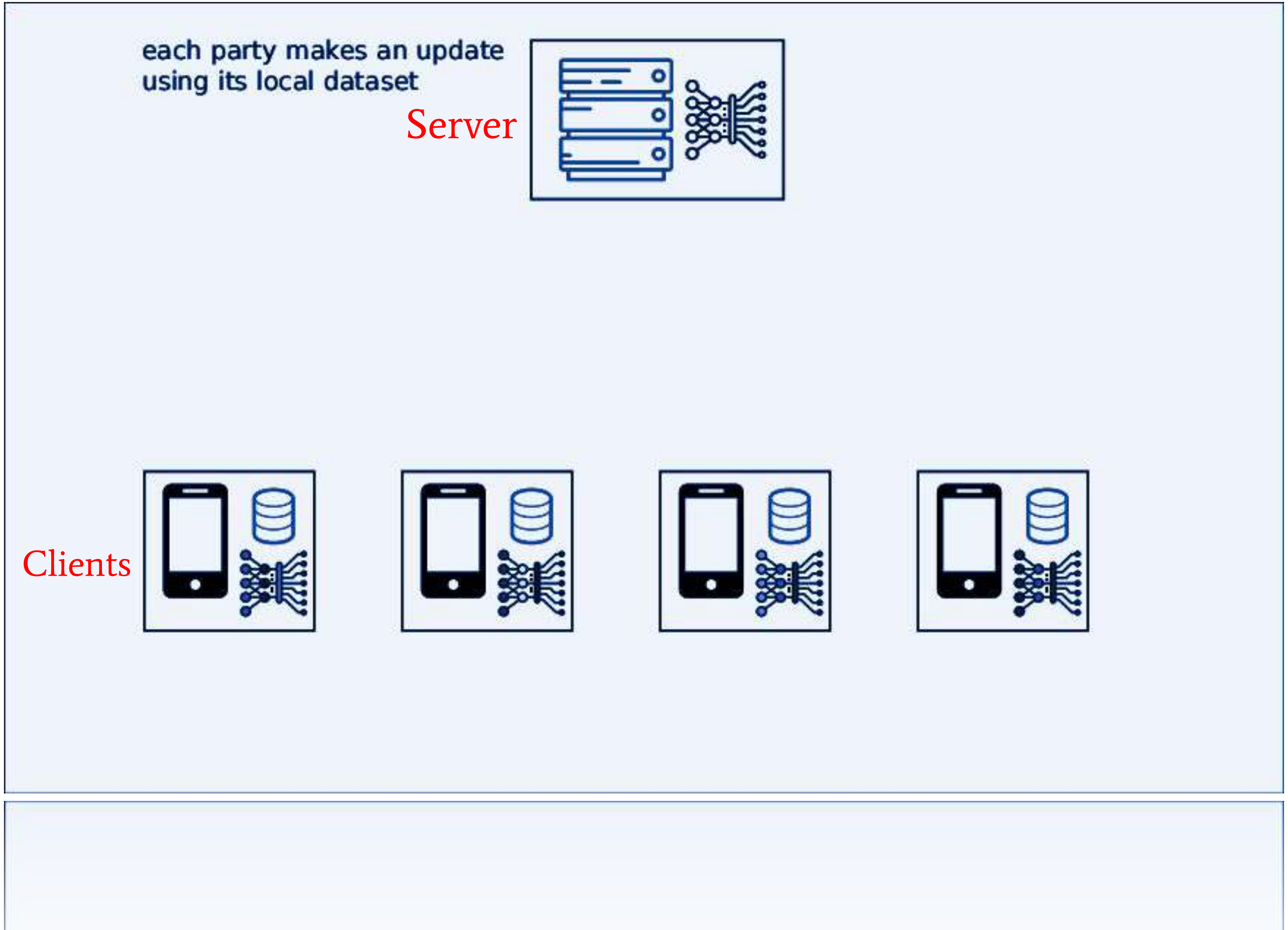
Client - Server Setup



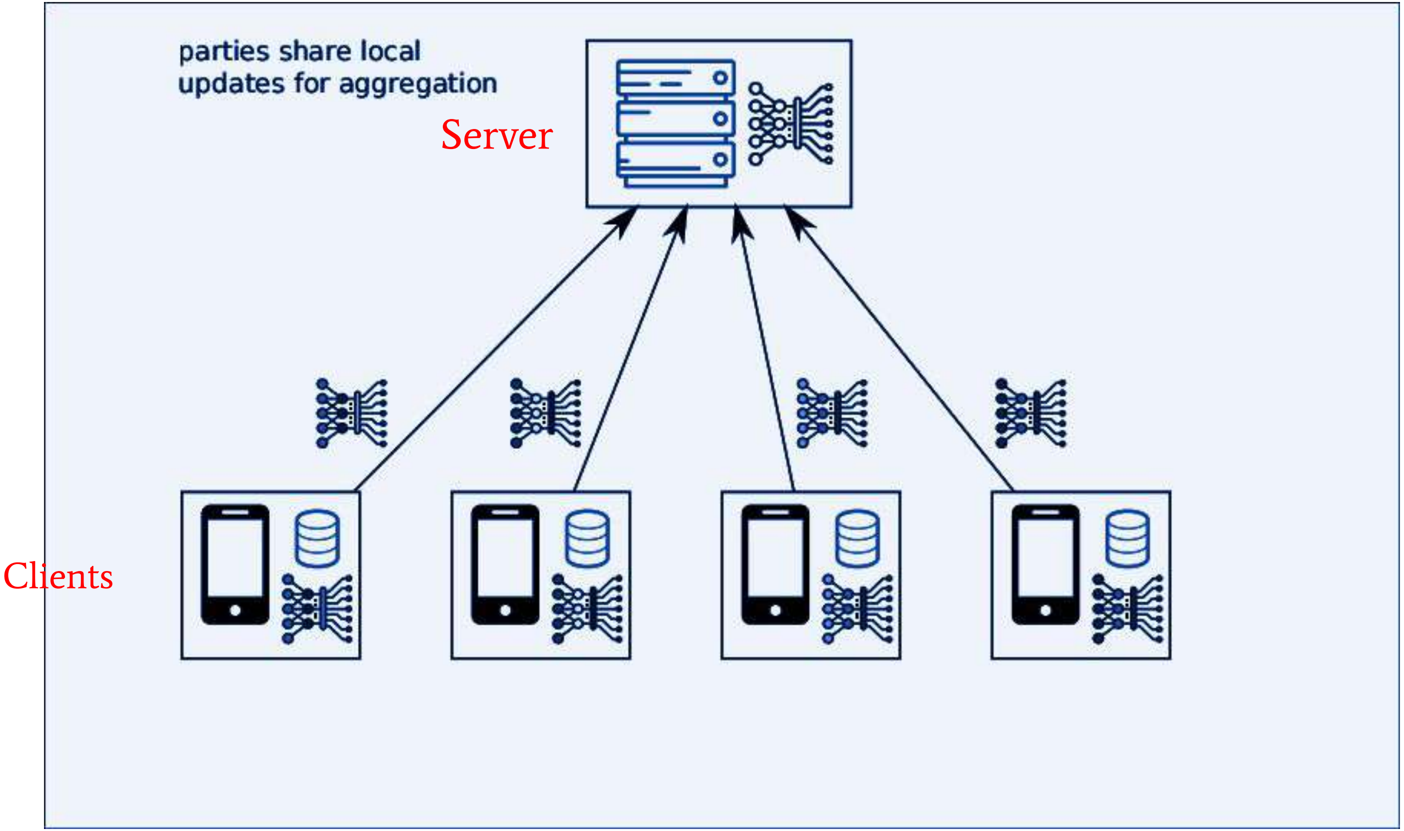
Step1: Initialization of Model



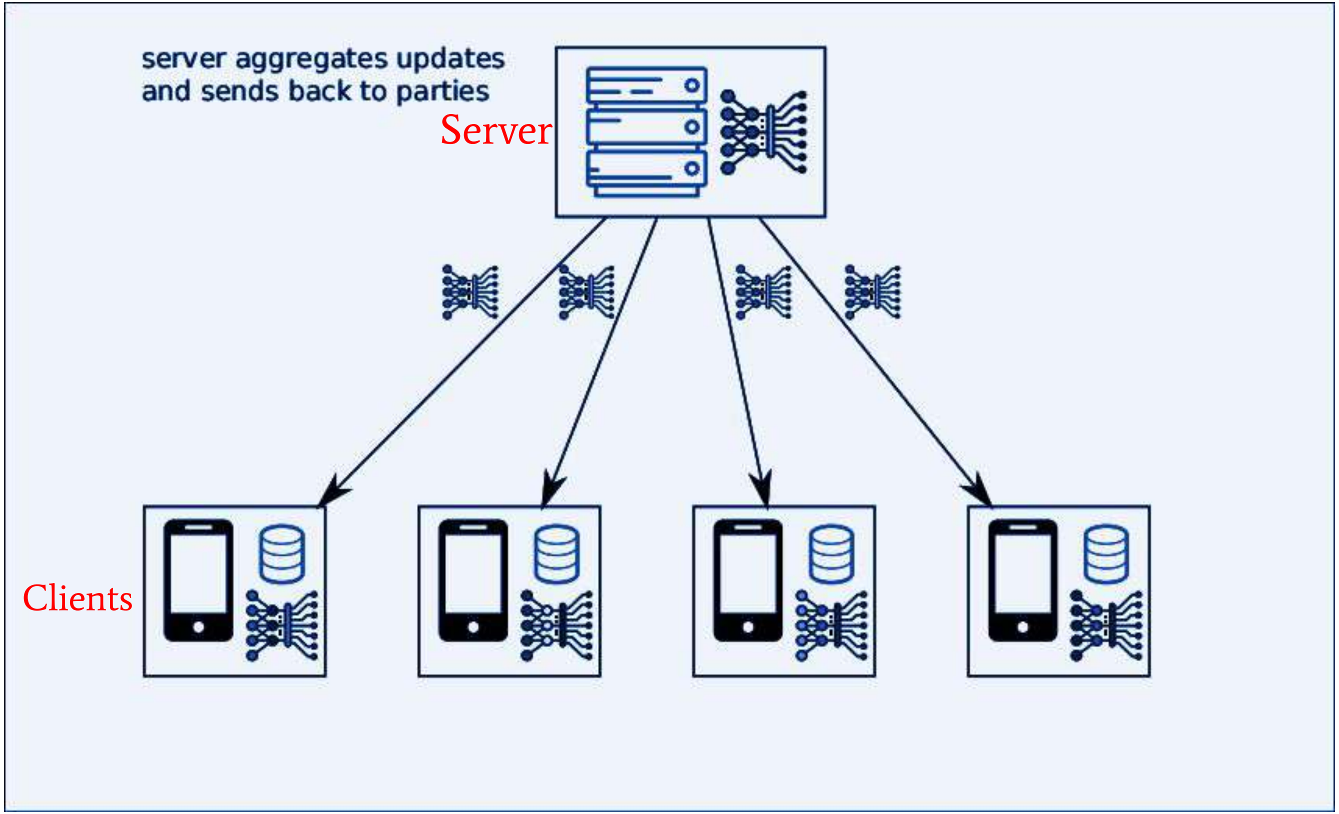
Step2: Training the model on client data



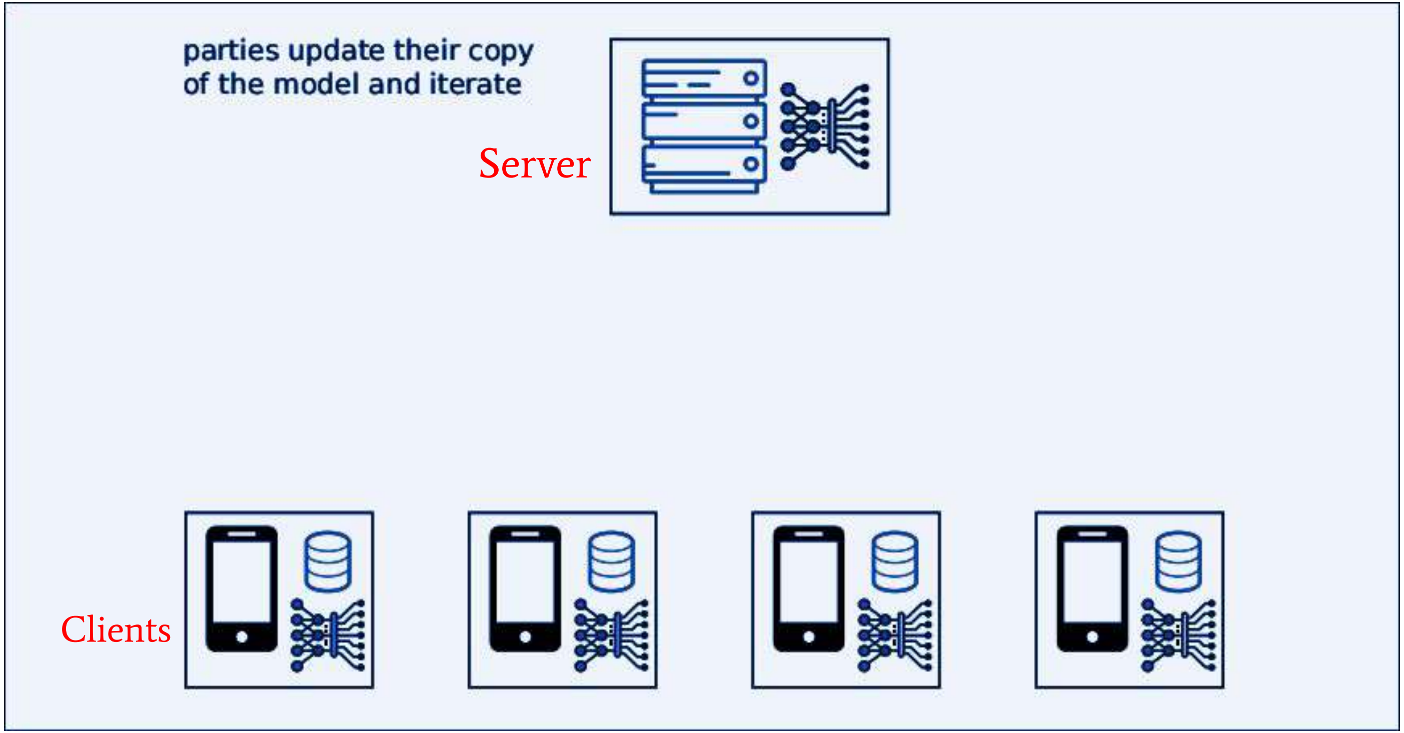
Step3: Sharing the parameters to Global model



Step4: Aggregation of parameters at server

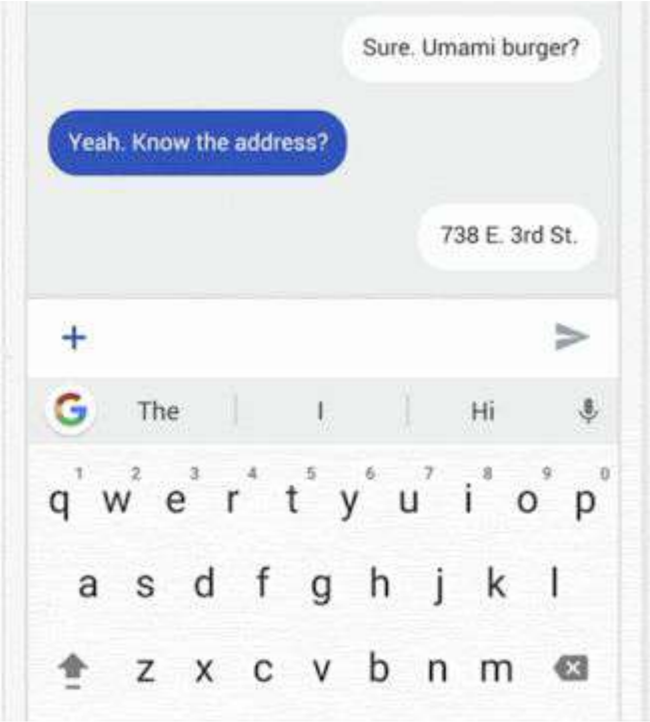


Step5: Sharing the updated parameters to clients



Applications

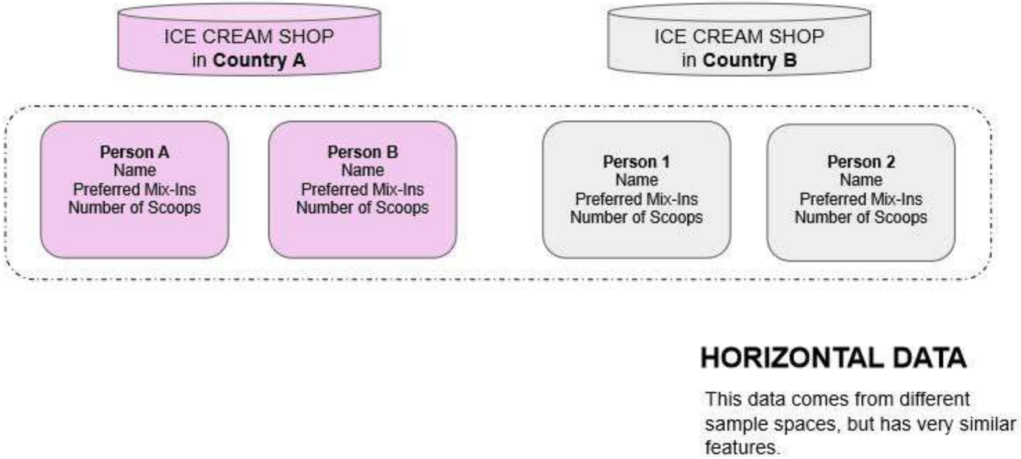
Google G-Board - Federated GRU(Gated Recurrent Unit)



Types of federated learning - Data partitioning

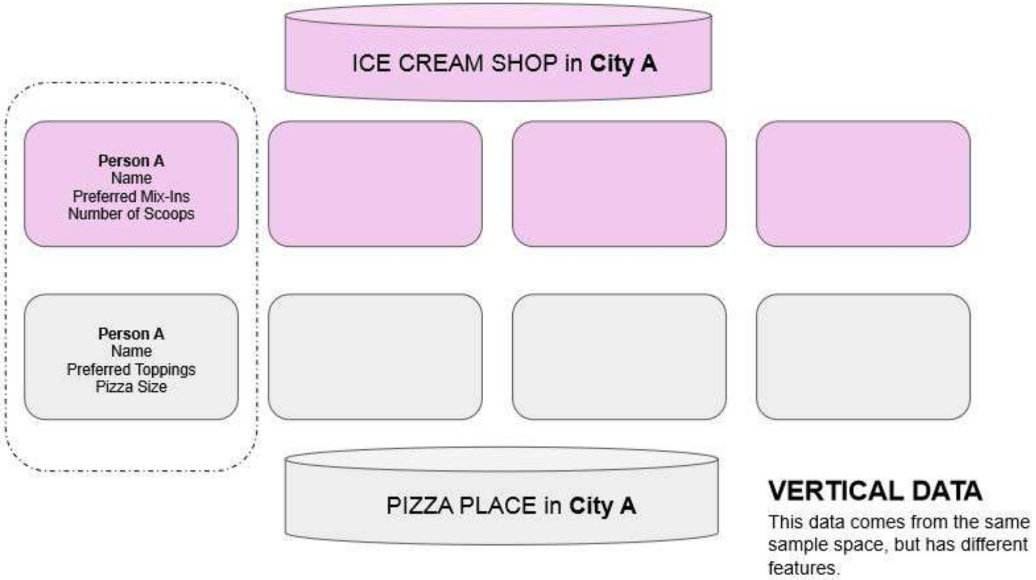
Horizontal Federated Learning

- Sample-based federated learning or homogenous federated learning
- Involves separating the data that has the same features but operates within a different sample space



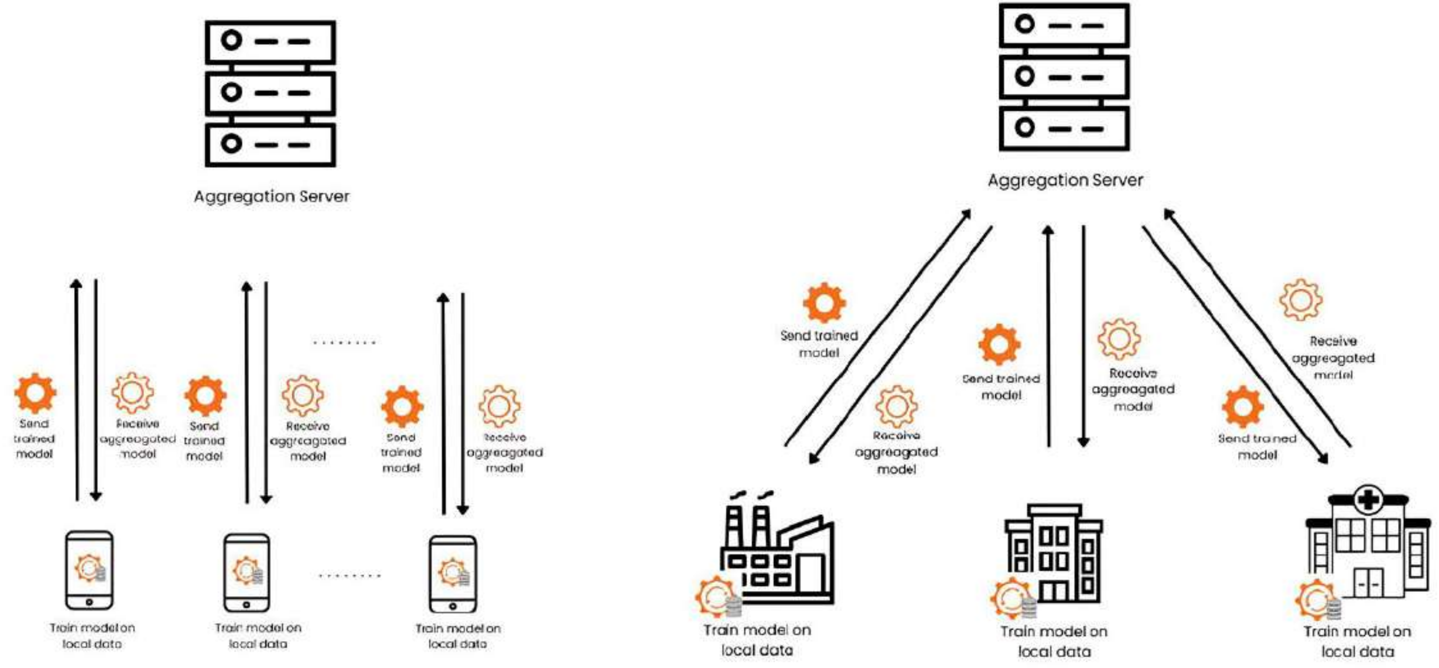
Vertical Federated Learning

- Feature-based federated learning or heterogeneous federated learning.
- Data shares the same sample space, but different feature space



Types of federated learning - Clients

Cross-Device vs Cross-Silo FL



- Small size node (smartphones, edge devices, etc)
- Might not be available at each iteration.

a) Cross-Device Federated Learning

- Large size node (companies organizations)
- Necessary to participate in each iteration

b) Cross-Silo Federated Learning

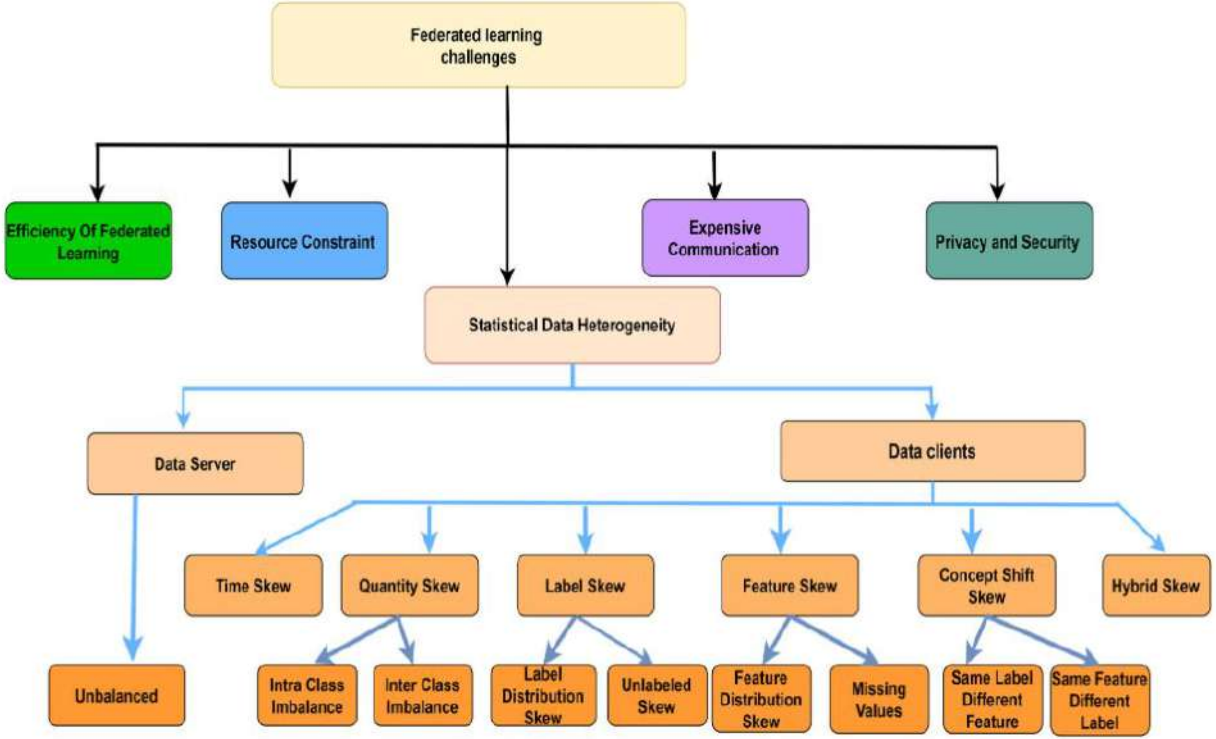
Cross-device FL

- Massive number of parties
- Small dataset per party (could be size 1)
- Limited availability and reliability
- Some parties may be malicious
- Communication is often the primary bottleneck.
- Partitioning by example (horizontal)

Cross-silo FL

- 2-100 parties
- Medium to large dataset per party
- Reliable parties, almost always available
- Parties are typically honest
- Might be computation or communication.
- Example-partitioned (horizontal) or feature-partitioned (vertical).

The main challenges in Federated learning ?



https://ieeexplore.ieee.org/document/10103619

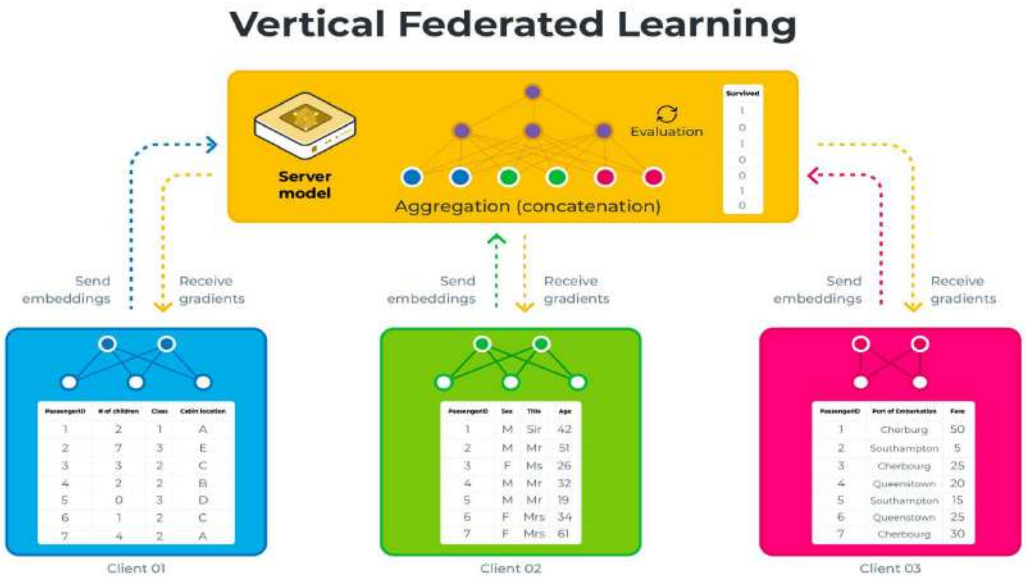
Problem Statement

FL systems gaining wider adoption for **privacy preserving machine learning**

- Heterogeneity is expected to cause **lowered performance of the trained models** with **longer convergence time**.
- Leading to **excessive energy consumption** for both the cloud infrastructure and battery powered devices.
- Find Out Innovative algorithms for FL task with lower convergence time with minimal impact on data privacy.

The main challenges in Vertical Federated learning

- One of the main challenges in vertical federated learning is aggregation of the weights



AGGREGATION ALGORITHMS

FEDAVG-（Communication-Efficient Learning of Deep Networks from Decentralized Data）

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
  initialize w_0
  for each round $t = 1, 2, \dots$ **do**
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of m clients)
    for each client $k \in S_t$ **in parallel do**
      $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
    $m_t \leftarrow \sum_{k \in S_t} n_k$
    $w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{m_t} w_{t+1}^k$ *// Erratum*[4]

ClientUpdate(k, w): *// Run on client k*
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

---

https://arxiv.org/abs/1602.05629

FEDPROX- (Federated Optimization in Heterogeneous Networks)

Algorithm 2 `FedProx` (Proposed Framework)

Input: K, T, μ, γ, 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 γ_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

---

https://arxiv.org/abs/1812.06127

QFEDAVG

Algorithm 2 q-`FedAvg`

Input: $K, E, T, q, 1/L, \eta, w^0, p_k, k = 1, \cdots, m$
for $t = 0, \cdots, T-1$ **do**
  Server selects a subset S_t of K devices at random (each device k is chosen with prob. p_k)
  Server sends w^t to all selected devices
  Each selected device k updates w^t for E epochs of SGD on F_k with step-size η to obtain $\bar{w}_k^{t+1}$
  Each selected device k computes:

$$\Delta w_k^t = L(w^t - \bar{w}_k^{t+1})$$
$$\Delta_k^t = F_k^q(w^t)\Delta w_k^t$$
$$h_k^t = qF_k^{q-1}(w^t)\|\Delta w_k^t\|^2 + LF_k^q(w^t)$$

  Each selected device k sends Δ_k^t and h_k^t back to the server
  Server updates w^{t+1} as:

$$w^{t+1} = w^t - \frac{\sum_{k \in S_t} \Delta_k^t}{\sum_{k \in S_t} h_k^t}$$

end for

https://arxiv.org/pdf/1905.10497

Open Source Federated Learning Frameworks

FATE

TensorFlow federated

NVIDIA FLARE

Flower

OpenFL intel

PySyft

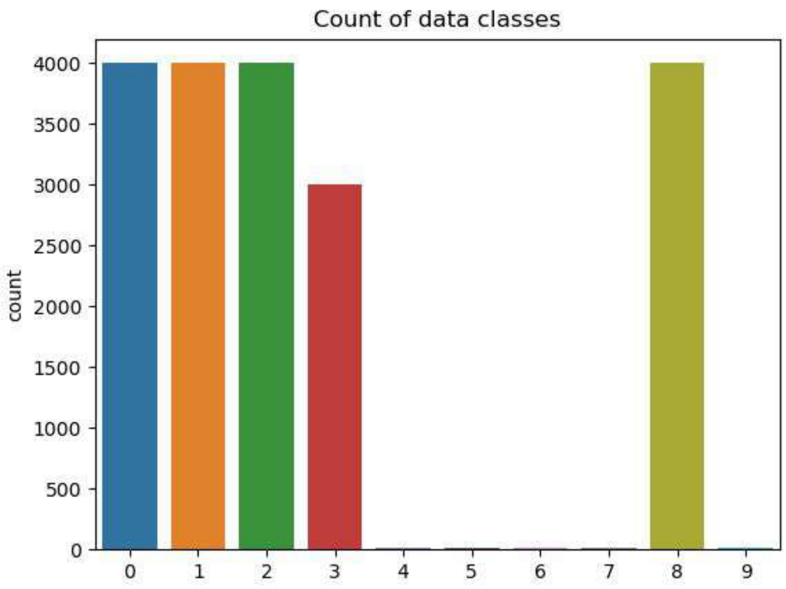
FedML

Federated Learning

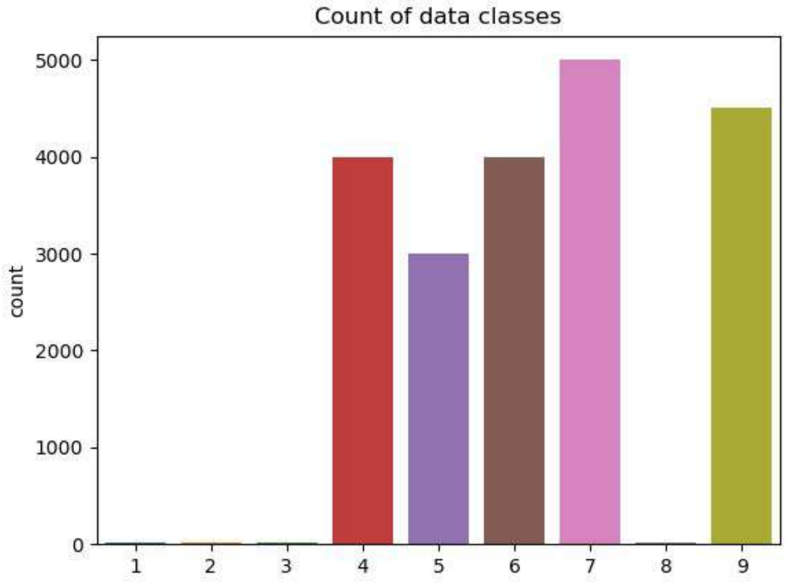
https://medium.com/elca-it/flower-pysyft-co-federated-learning-frameworks-in-python-b1a8eda68b0d

Baseline Models - Flower Framework

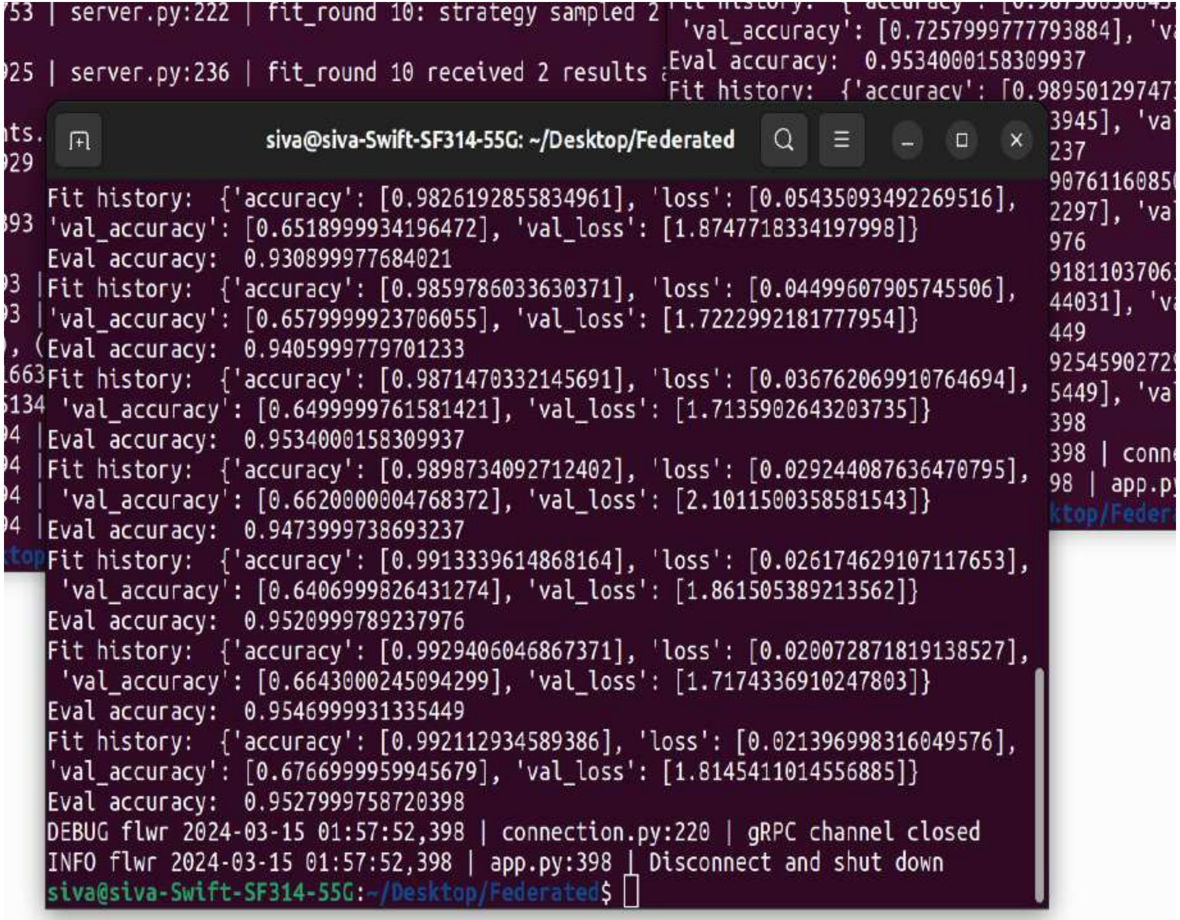
Horizontal Federated Learning - MNIST dataset(with different class distributions to each client)



Client 1

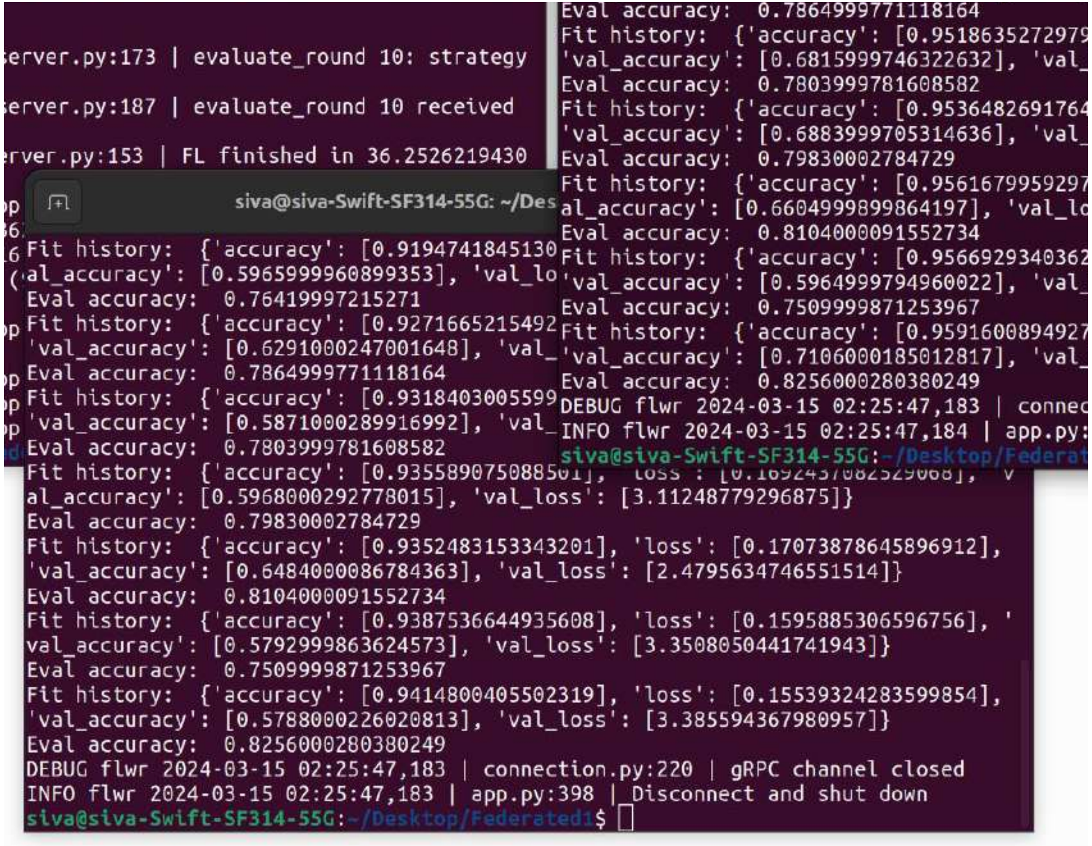


Client 2



Client1 accuracy- 0.6766
Client2 accuracy- 0.7257
Global accuracy- 0.9527

Horizontal Federated Learning - MNIST fashion dataset(with different class distributions to each client)



Client1 accuracy- 0.7106
Client2 accuracy- 0.5788
Global accuracy-0.8256

EXPERIMENT SETUP - 2

Centralized VS Decentralized

DATASET : TITANIC DATASET

```
Running centralised training...
Train accuracy: 84.248%
Test accuracy: 82.022%
Running decentralised training...
Iteration 1, loss = 0.64399825
Iteration 2, loss = 0.56226789
Iteration 3, loss = 0.49493030
Iteration 4, loss = 0.45459877
Iteration 5, loss = 0.43634872
Iteration 6, loss = 0.43703215
Iteration 7, loss = 0.43900626
Iteration 8, loss = 0.42954290
Iteration 9, loss = 0.42684795
Iteration 10, loss = 0.42327625
Iteration 11, loss = 0.42247988
Iteration 12, loss = 0.42048271
Iteration 13, loss = 0.41931750
Iteration 14, loss = 0.41798792
Iteration 15, loss = 0.41693441
```

```
Client 0 test accuracy: 73.034%
Client 1 test accuracy: 81.461%
Client 2 test accuracy: 71.348%
Combined test accuracy: 80.337%
```

EXPERIMENT SETUP - 2

Centralized VS Decentralized

DATASET : CANCER DATASET

```
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs.
Stopping.
Client 0 test accuracy: 96.512%
Client 1 test accuracy: 94.186%
Client 2 test accuracy: 94.186%
Combined Test accuracy: 96.512%
```

```
Running centralised training...
Train accuracy: 98.758%
Test accuracy: 95.349%
```

Implementation

EXPERIMENT SETUP -3

Experimenting with different aggregating algorithm

1.Fedavg

2.FedProx

3.QFedavg

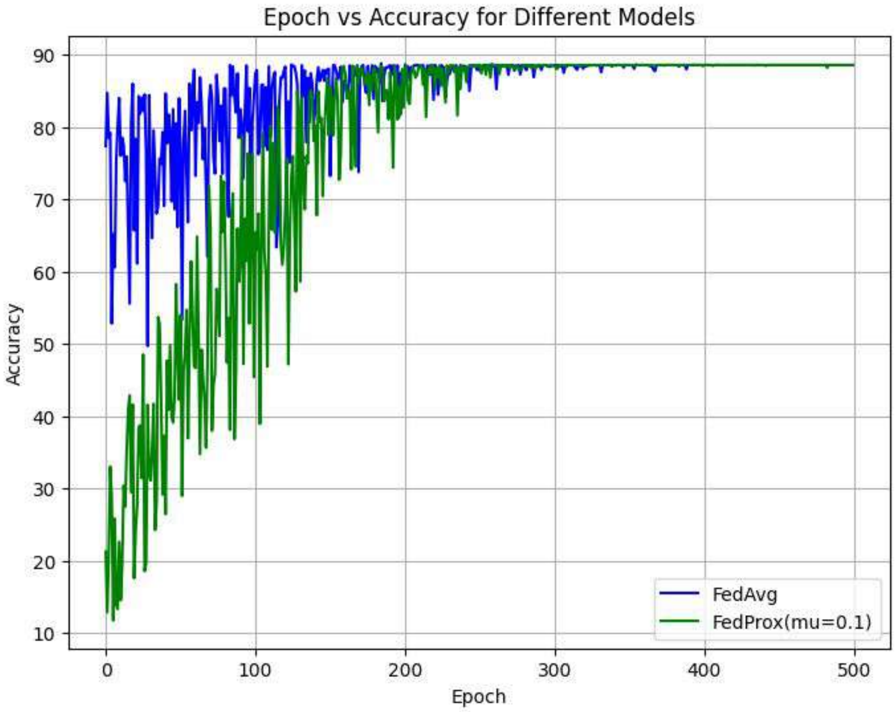
DATASET: TITANIC DATASET, WATER QUALITY DATASET

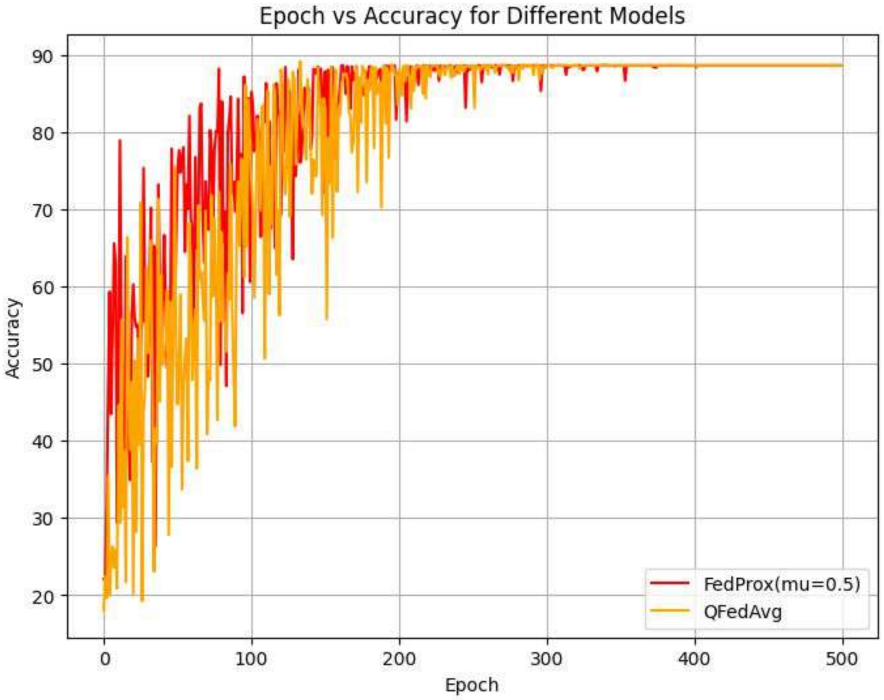
https://github.com/datasciencedojo/datasets/blob/master/titanic.csv

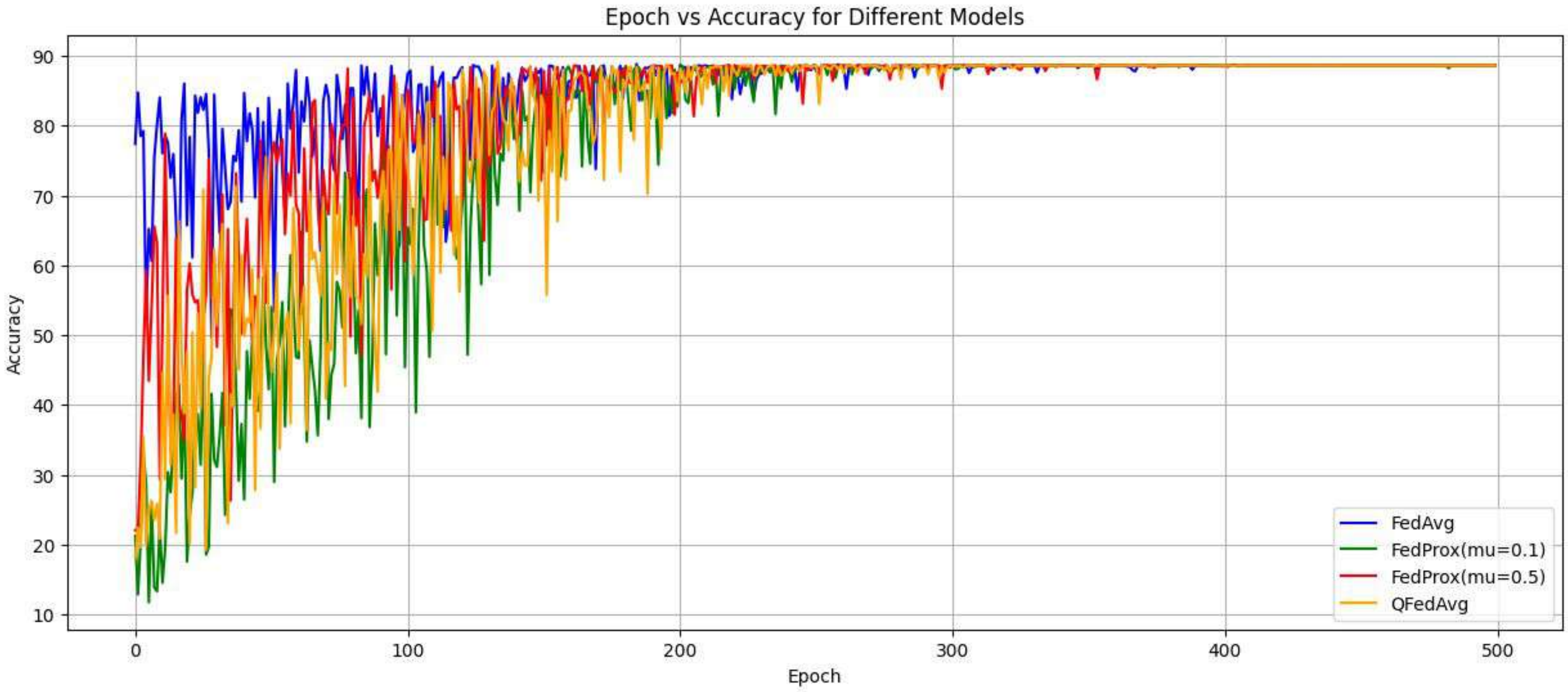
Water Quality Dataset

- Dataset - Classification Problem(Two class)(20 features)(7999 data points)
- Number of clients : 3
- Client 1 - Features {'aluminium', 'ammonia', 'arsenic', 'barium', 'cadmium', 'chloramine', 'chromium' , ‘is_safe’}
- Client 2 - Features {'copper', 'flouride', 'bacteria', 'viruses', 'lead', 'nitrates', 'nitrites',’is_safe’}
- Client 3 -Features{‘mercury’ ‘perchlorate’ ‘radium’ ‘selenium’ ‘silver’ ‘uranium’,’is_safe’}

Results & Observations





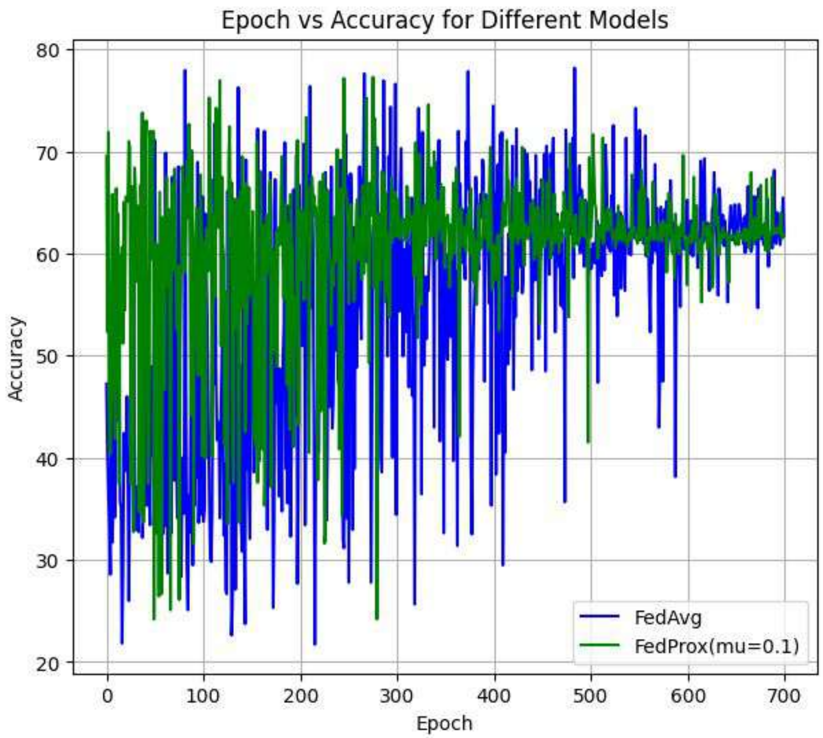

Epoch vs Accuracy for Different Models
Accuracy
Epoch
90
80
70
60
50
40
30
20
10
0
100
200
300
400
500
FedAvg
FedProx(mu=0.1)
FedProx(mu=0.5)
QFedAvg

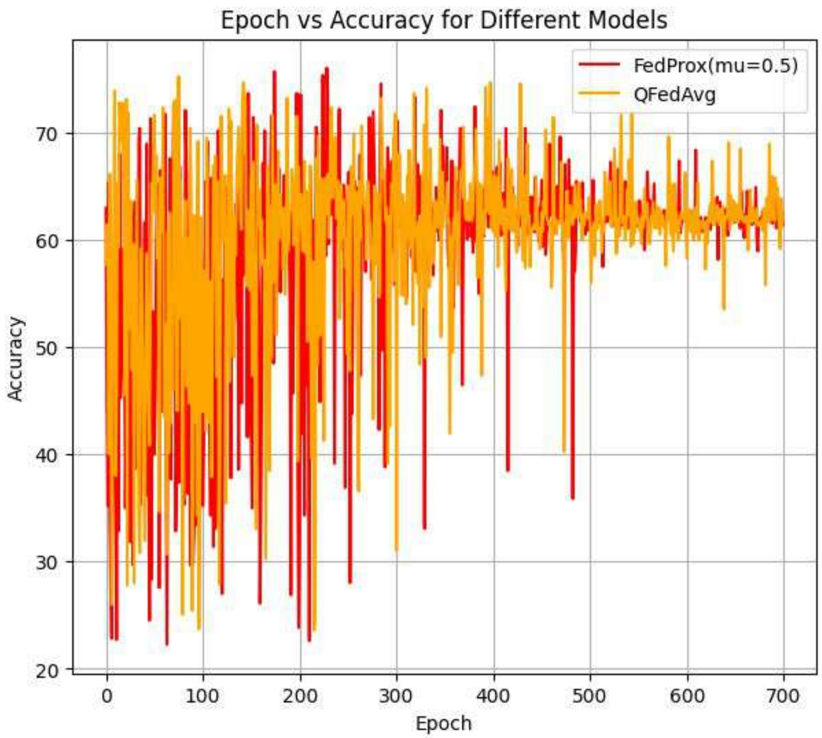
Titanic Dataset

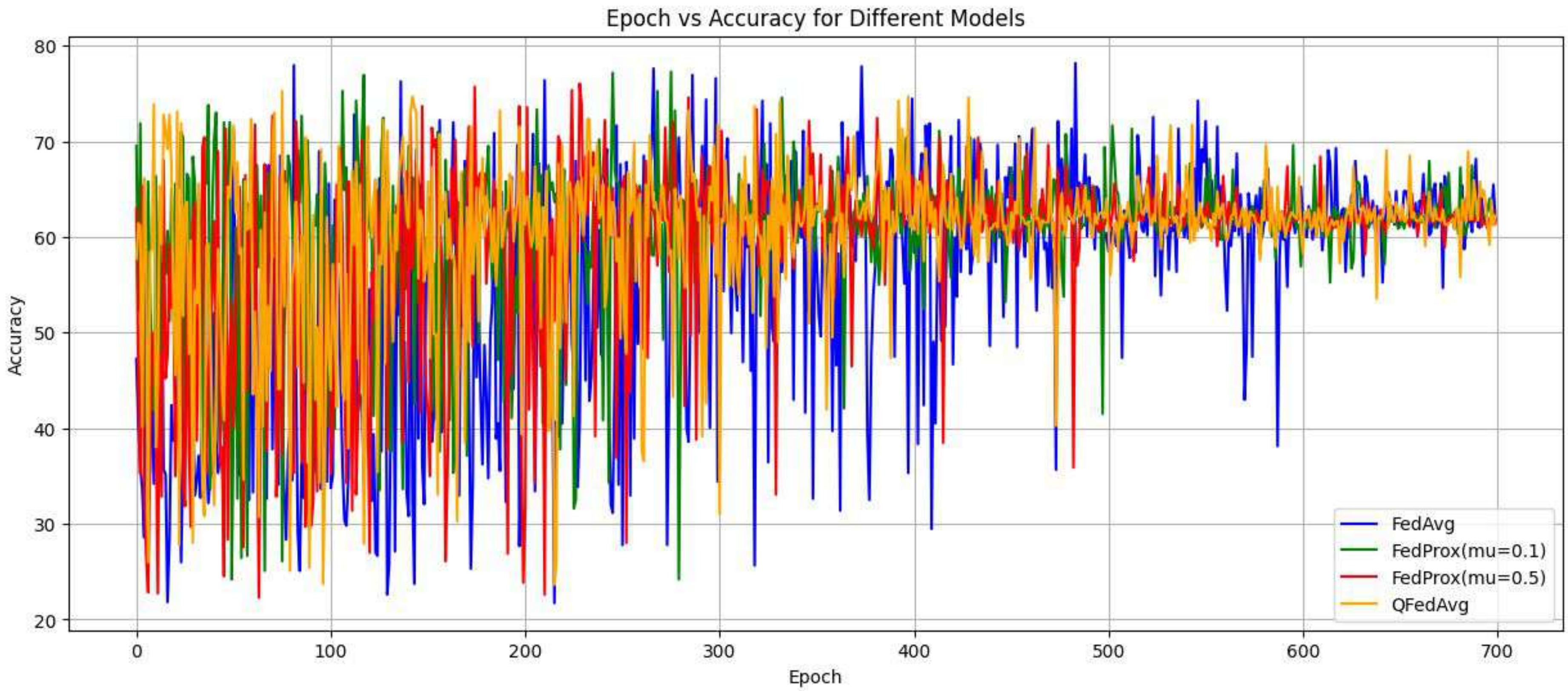
- Dataset - Classification Problem(Two class)(12 features)(891 data points)
- Number of clients : 3
- Client 1 - Features {"Parch", "Cabin", "Pclass",”Survived”}
- Client 2 - Features {“Sex", "Title”,”Survived”}
- Client 3 -Features{‘Age', 'SibSp',’ Embarked’,Cabin}

https://github.com/datasciencedojo/datasets/blob/master/titanic.csv

Results & Observations






Epoch vs Accuracy for Different Models
Accuracy
Epoch
80
70
60
50
40
30
20
0
100
200
300
400
500
600
700
FedAvg
FedProx(mu=0.1)
FedProx(mu=0.5)
QFedAvg

References

Github link for resources(Research papers and codes for frameworks):

1. https://github.com/monk1337/Aweome-Heathcare-Federated-Learning?tab=readme-ov-file

2. https://github.com/albarqouni/Federated-Learning-In-Healthcare?tab=readme-ov-file

3. https://github.com/adap/flower(flower framework)

4. https://github.com/FedML-AI/FedML-(fedml framework)-https://github.com/FedML-AI/FedML/tree/master/python/fedml

5. https://github.com/OpenMined/PySyft(pysyft framework)

Thank you