# 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

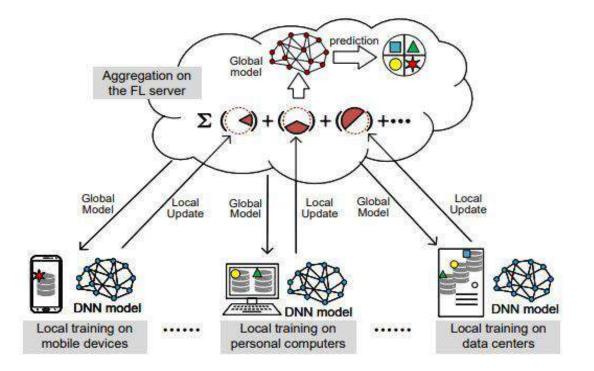
- What is Federated Learning
- Objective
- Applications
- Types Of Federated Learning (Data Partitioning)
- Types Of Federated Learning(Based On Client)
- Aggregation Algorithms
- Results & Observations
- References

#### Shift from centralized data to decentralized data

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

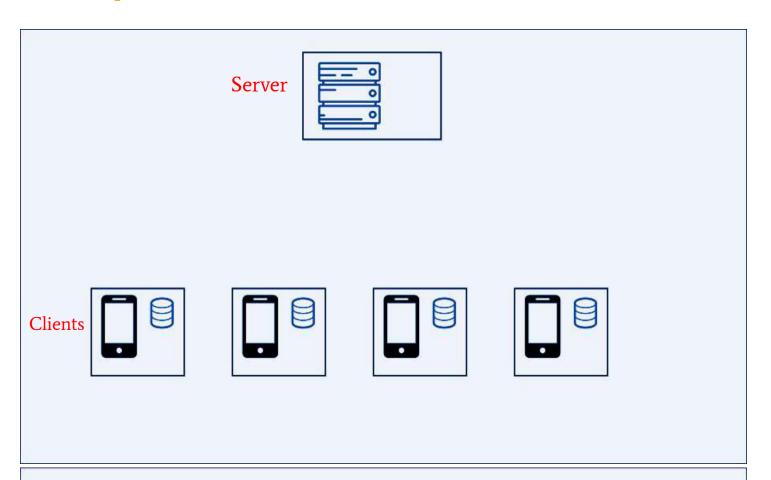
- Sending data to cloud for centralized ML is too costly
  - Self driving cars generate several TBs of data every day
  - Some wireless networks have limited bandwidth
- Dats is too sensitive(medical reports)
  - Data privacy
  - Keeping the control over data and give the competitive advantage in business and research

## What is Federated Learning?

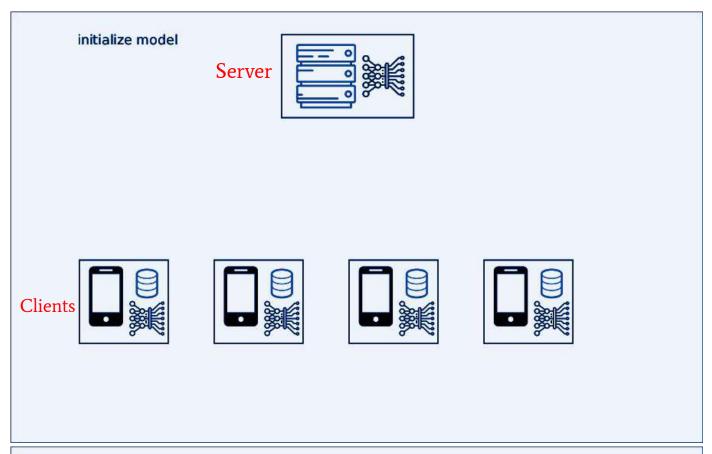


## Federated Learning Process

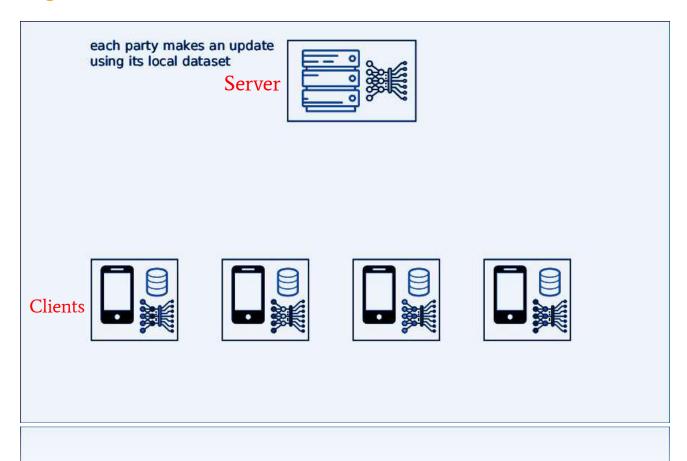
#### Client - Server Setup



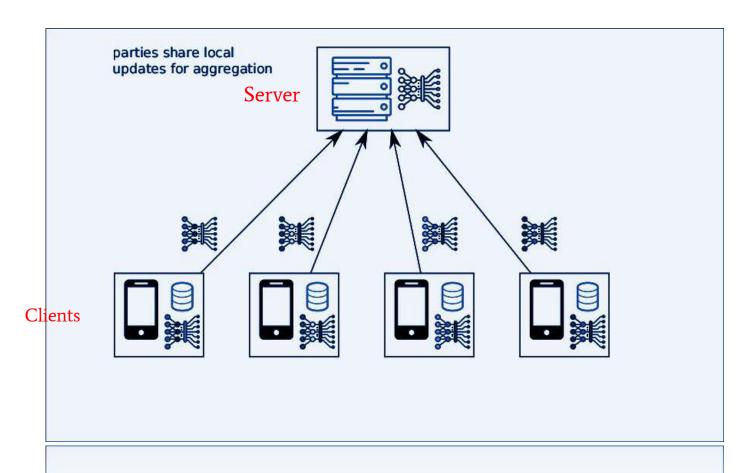
#### Step1: Initialization of Model



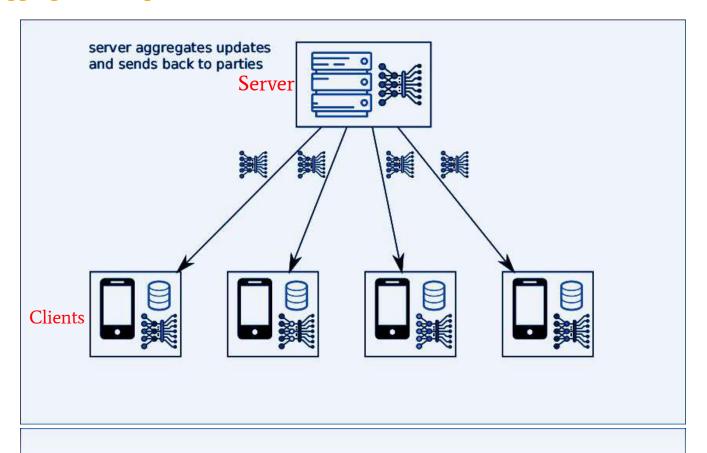
#### Step2: Training the model on client data



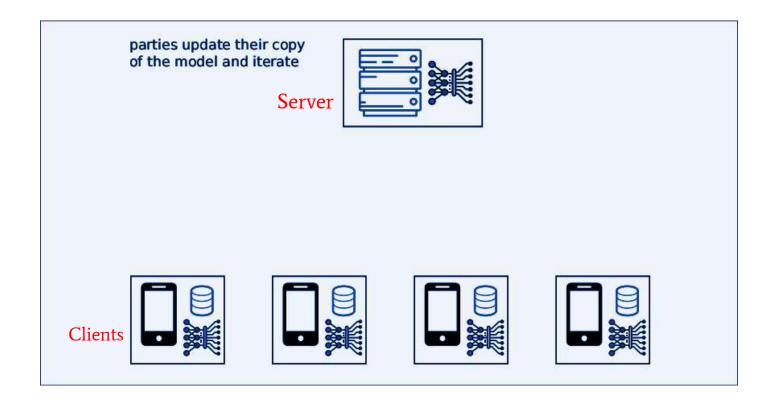
#### Step3: Sharing the parameters to Global model



#### Step4: Aggregation of parameters at server



#### Step5: Sharing the updated parameters to clients



## **Applications**

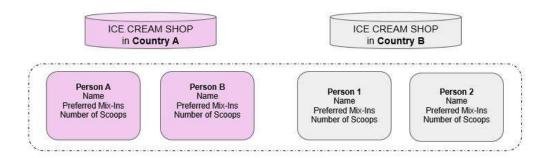
#### Google G-Board - Federated GRU(Gated Recurrent Unit)



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

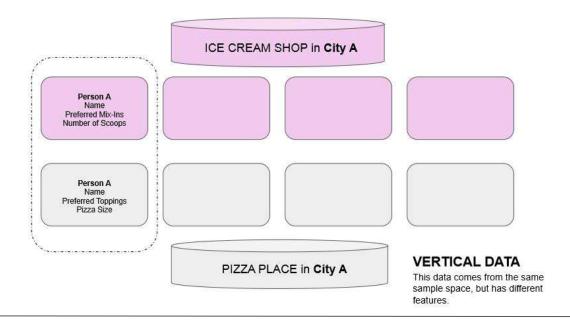


#### HORIZONTAL DATA

This data comes from different sample spaces, but has very similar features.

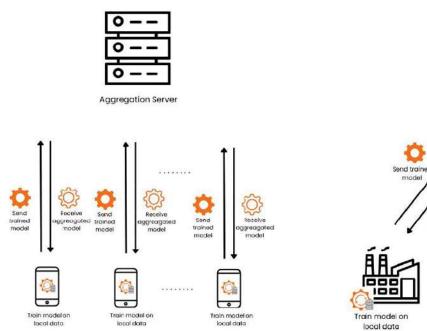
#### Vertical Federated Learning

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

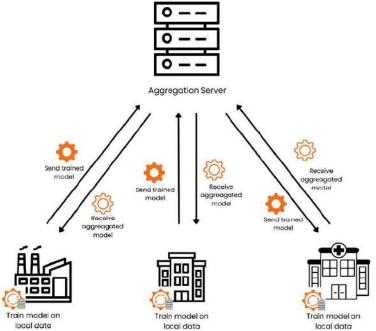


### Types of federated learning - Clients

#### 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

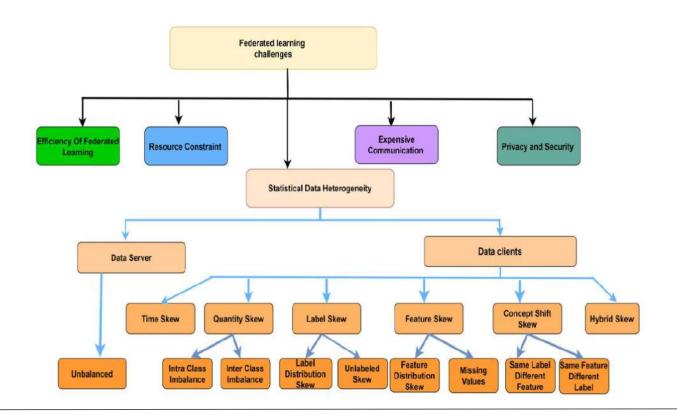
#### Cross-device FL

- 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?



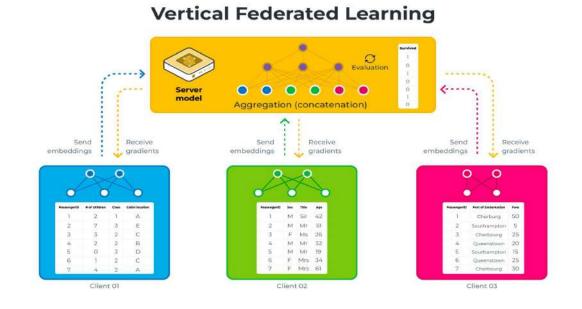
#### Problem Statement

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  $\eta$  is the learning rate.

```
Server executes:
   initialize wo
  for each round t = 1, 2, \dots do
      m \leftarrow \max(C \cdot K, 1)
      S_t \leftarrow \text{(random set of } m \text{ 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<sup>4</sup>
ClientUpdate(k, w): // Run on client k
   \mathcal{B} \leftarrow (\text{split } \mathcal{P}_k \text{ 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
```

#### FEDPROX- (Federated Optimization in Heterogeneous Networks)

#### Algorithm 2 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

#### **QFEDAVG**

#### Algorithm 2 q-FedAvg

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

$$\begin{split} & \overset{\mathbf{T}}{\Delta} w_k^t = L(w^t - \bar{w}_k^{t+1}) \\ & \Delta_k^t = F_k^q(w^t) \Delta w_k^t \\ & h_k^t = q F_k^{q-1}(w^t) \|\Delta w_k^t\|^2 + L F_k^q(w^t) \end{split}$$

- 7: Each selected device k sends  $\Delta_k^t$  and  $h_k^t$  back to the server
- 8: Server updates  $w^{t+1}$  as:

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

9: end for

## Open Source Federated Learning Frameworks











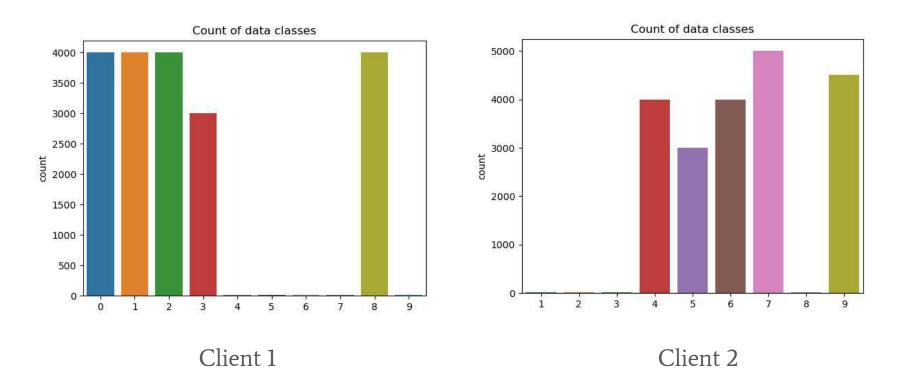






#### Baseline Models - Flower Framework

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



```
'53 | server.py:222 | fit_round 10: strategy sampled 2
                                                       'val_accuracy': [0.7257999777793884], 'v
                                                      .Eval accuracy: 0.9534000158309937
25 | server.py:236 | fit round 10 received 2 results
                                                       Fit history: {'accuracy': [0.98950129747]
                                                                                      3945], 'va'
its.
                     siva@siva-Swift-SF314-55G: ~/Desktop/Federated
                                                                                      237
129
                                                                                      9076116085
   Fit history: {'accuracy': [0.9826192855834961], 'loss': [0.05435093492269516],
                                                                                      2297], 'va
<sup>93</sup> 'val accuracy': [0.6518999934196472], 'val loss': [1.8747718334197998]}
                                                                                      976
   Eval accuracy: 0.930899977684021
                                                                                      9181103706
  Fit history: {'accuracy': [0.9859786033630371], 'loss': [0.04499607905745506],
                                                                                      44031], 'v
  | 'val accuracy': [0.6579999923706055], 'val loss': [1.7222992181777954]}
                                                                                      449
 (Eval accuracy: 0.9405999779701233
                                                                                      9254590272
.663Fit history: {'accuracy': [0.9871470332145691], 'loss': [0.036762069910764694],
                                                                                      5449], 'va
<sup>134</sup> 'val accuracy': [0.6499999761581421], 'val loss': [1.7135902643203735]}
                                                                                      398
  Eval accuracy: 0.9534000158309937
                                                                                      398 | conn
  Fit history: {'accuracy': [0.9898734092712402], 'loss': [0.029244087636470795],
                                                                                      98 | app.p
    'val_accuracy': [0.6620000004768372], 'val_loss': [2.1011500358581543]}
  Eval accuracy: 0.9473999738693237
   Fit history: {'accuracy': [0.9913339614868164], 'loss': [0.026174629107117653],
    'val accuracy': [0.6406999826431274], 'val loss': [1.861505389213562]}
   Eval accuracy: 0.9520999789237976
   Fit history: {'accuracy': [0.9929406046867371]. 'loss': [0.020072871819138527].
    'val_accuracy': [0.6643000245094299], 'val_loss': [1.7174336910247803]}
   Eval accuracy: 0.9546999931335449
   Fit history: {'accuracy': [0.992112934589386], 'loss': [0.021396998316049576],
    'val accuracy': [0.6766999959945679], 'val loss': [1.8145411014556885]}
   Eval accuracy: 0.9527999758720398
   DEBUG flwr 2024-03-15 01:57:52,398 | connection.pv:220 | gRPC channel closed
   INFO flwr 2024-03-15 01:57:52,398 | app.py:398 | Disconnect and shut down
   siva@siva-Swift-SF314-55G:~/Desktop/FederatedS
```

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)

```
Eval accuracy: 0.7864999771118164
                                            Fit history: {'accuracy': [0.9518635272979
erver.py:173 | evaluate round 10: strategy
                                             'val accuracy': [0.6815999746322632], 'val
                                            Eval accuracy: 0.7803999781608582
erver.py:187 | evaluate_round 10 received
                                            Fit history: {'accuracy': [0.9536482691764
                                             'val accuracy': [0.6883999705314636], 'val
erver.py:153 | FL finished in 36.2526219430
                                           Eval accuracy: 0.79830002784729
                                            Fit history: {'accuracy': [0.9561679959297
                  siva@siva-Swift-SF314-55G: ~/Des al_accuracy': [0.6604999899864197], 'val_lo
                                             Eval accuracy: 0.8104000091552734
6 Fit history: {'accuracy': [0.9194741845130 Fit history: {'accuracy': [0.9566929340362
val_accuracy': [0.5965999960899353], 'val_lo val_accuracy': [0.5964999794960022], 'val_
 Eval accuracy: 0.76419997215271
                                             Eval accuracy: 0.7509999871253967
p Fit history: {'accuracy': [0.9271665215492 Fit history: {'accuracy': [0.9591600894927
  val_accuracy': [0.6291000247001648], 'val_'val accuracy': [0.7106000185012817], 'val_
Eval accuracy: 0.7864999771118164
                                             Eval accuracy: 0.8256000280380249
p Fit history: {'accuracy': [0.9318403005599DEBUG flwr 2024-03-15 02:25:47,183 | connec
 'val_accuracy': [0.5871000289916992], 'val_INFO flwr 2024-03-15 02:25:47,184 | app.py:
 Eval accuracy: 0.7803999781608582
                                            siva@siva-Swift-SF314-55G:-/Desktop/Feder
 Fit history: {'accuracy': [0.9355890750885v1], toss : [v.109243/082229003], v.
 al accuracy': [0.5968000292778015], 'val loss': [3.11248779296875]}
 Eval accuracy: 0.79830002784729
 Fit history: {'accuracy': [0.9352483153343201], 'loss': [0.17073878645896912].
  'val_accuracy': [0.6484000086784363], 'val loss': [2.4795634746551514]}
 Eval accuracy: 0.8104000091552734
 Fit history: {'accuracy': [0.9387536644935608], 'loss': [0.1595885306596756],
 val accuracy': [0.5792999863624573], 'val loss': [3.3508050441741943]}
 Eval accuracy: 0.7509999871253967
 Fit history: {'accuracy': [0.9414800405502319], 'loss': [0.15539324283599854],
  'val accuracy': [0.5788000226020813], 'val loss': [3.385594367980957]}
 Eval accuracy: 0.8256000280380249
 DEBUG flwr 2024-03-15 02:25:47,183 | connection.py:220 | gRPC channel closed
 INFO flwr 2024-03-15 02:25:47,183 | app.py:398 | Disconnect and shut down
  siva@siva-Swift-SF314-55G:-/Desktop/Federated15
```

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
                                           Client 0 test accuracy: 73.034%
Iteration 3, loss = 0.49493030
                                           Client 1 test accuracy: 81.461%
Iteration 4. loss = 0.45459877
                                           Client 2 test accuracy: 71.348%
Iteration 5, loss = 0.43634872
                                           Combined test accuracy: 80.337%
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
```

#### **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

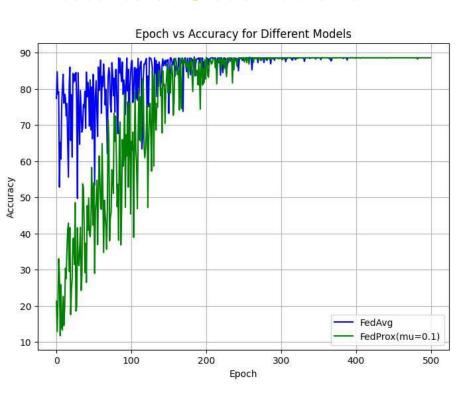
3.QFedavg

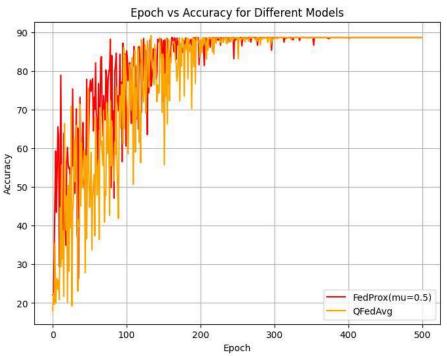
DATASET: TITANIC DATASET, WATER QUALITY DATASET

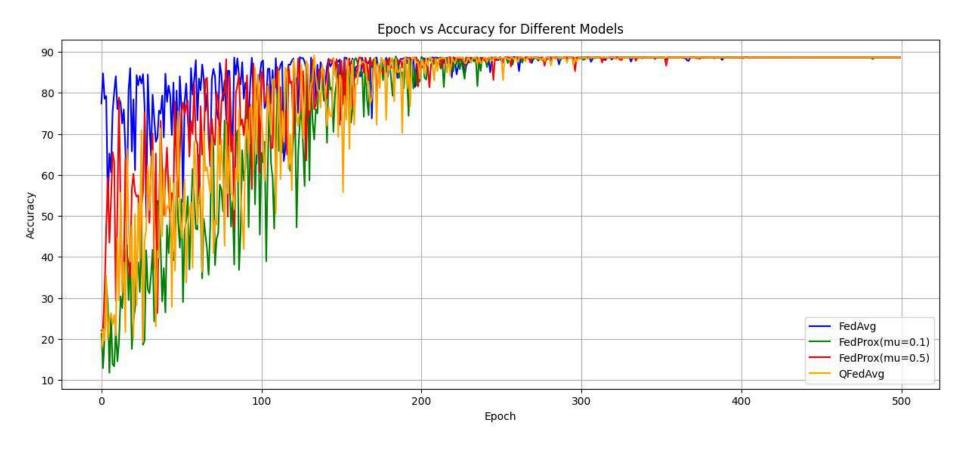
#### Water Quality Dataset

- 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







#### Titanic Dataset

- Dataset - Classification Problem(Two class)(12 features)(891 data points)

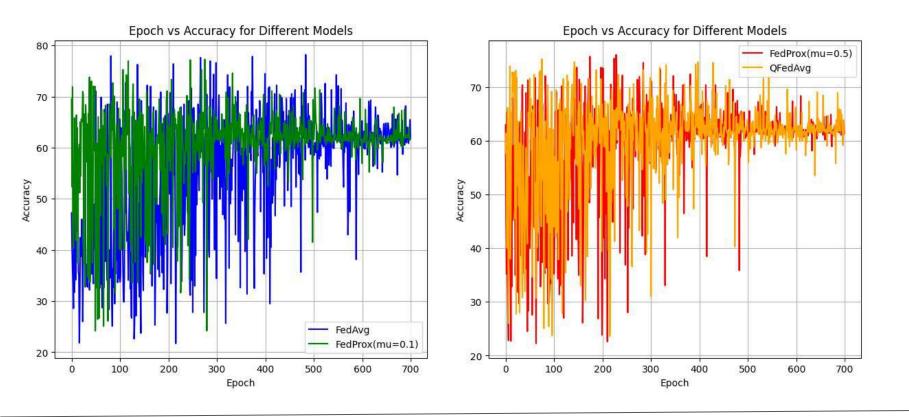
- Number of clients : 3

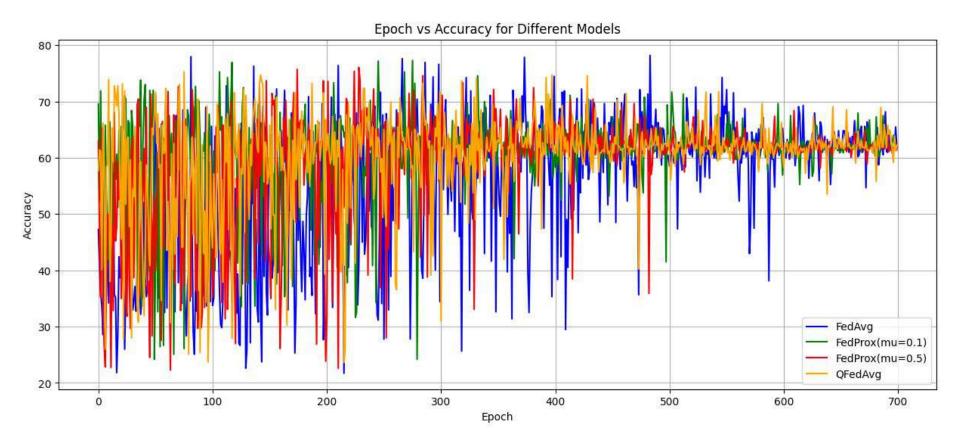
- Client 1 - Features {"Parch", "Cabin", "Pclass", "Survived"}

- Client 2 - Features {"sex", "Title", "Survived"}

- Client 3 - Features ('Age', 'SibSp', 'Embarked', Cabin)

#### Results & Observations





#### References

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

- 1. <a href="https://github.com/monk1337/Aweome-Heathcare-Federated-Learning?tab=readme-ov-file">https://github.com/monk1337/Aweome-Heathcare-Federated-Learning?tab=readme-ov-file</a>
- 2. <a href="https://github.com/albarqouni/Federated-Learning-In-Healthcare?tab=readme-ov-file">https://github.com/albarqouni/Federated-Learning-In-Healthcare?tab=readme-ov-file</a>
- 3. <a href="https://github.com/adap/flower">https://github.com/adap/flower</a> (flower framework)
- 4. <a href="https://github.com/FedML-AI/FedML-(fedml framework)-https://github.com/FedML-AI/FedML/tree/master/python/fedml">https://github.com/FedML-AI/FedML/tree/master/python/fedml</a>
- 5. <a href="https://github.com/OpenMined/PySyft">https://github.com/OpenMined/PySyft</a>(pysyft framework)

## Thank you