‘

MACHINE LEARNING ASSIGNMENT

PROJECT REPORT

TEAM ID : 24

“Federated Learning for Privacy-Preserving Medical Diagnosis”

Name	SRN
Chinmayi Shravani Yellanki	PES2UG23CS152
Disha R	PES2UG23CS179

Abstract

This project explores federated learning (FL) as a privacy-protection method for medical diagnosis through the Kaggle publicly available sepsis prediction dataset. With structured clinical data available and pertinent to early sepsis diagnosis, the dataset is distributed among various clients to replicate groupwise collaborative learning without the sharing of central data. Globally via the Flower FL framework, locally trained federated multi-layer perceptron (MLP) models are aggregated. Non-IID data, communication latency, and client heterogeneity challenges are overcome. Differential privacy and homomorphic encryption privacy mechanisms are implemented to secure patient data. Performance is demonstrated to be within proximity to that of the centralized models with added privacy through the federated MLPs.

Introduction

Diagnostic healthcare models need extensive and heterogeneous datasets but decentralized sharing of anonymous patient data is faced with stringent privacy and governance barriers. Decentralized learning also enables numerous clients (like healthcare institutions) to collaboratively build models within local environments and share only the model's updates. This research employs the Kaggle sepsis dataset, stratified across artificial clients, to compare federated multilayer perceptron models to diagnose sepsis. It endeavors to overcome challenges common within federated settings, namely heterogeneously scattered data and minimal exchange, with firm privacy assurances.

Problem Statement

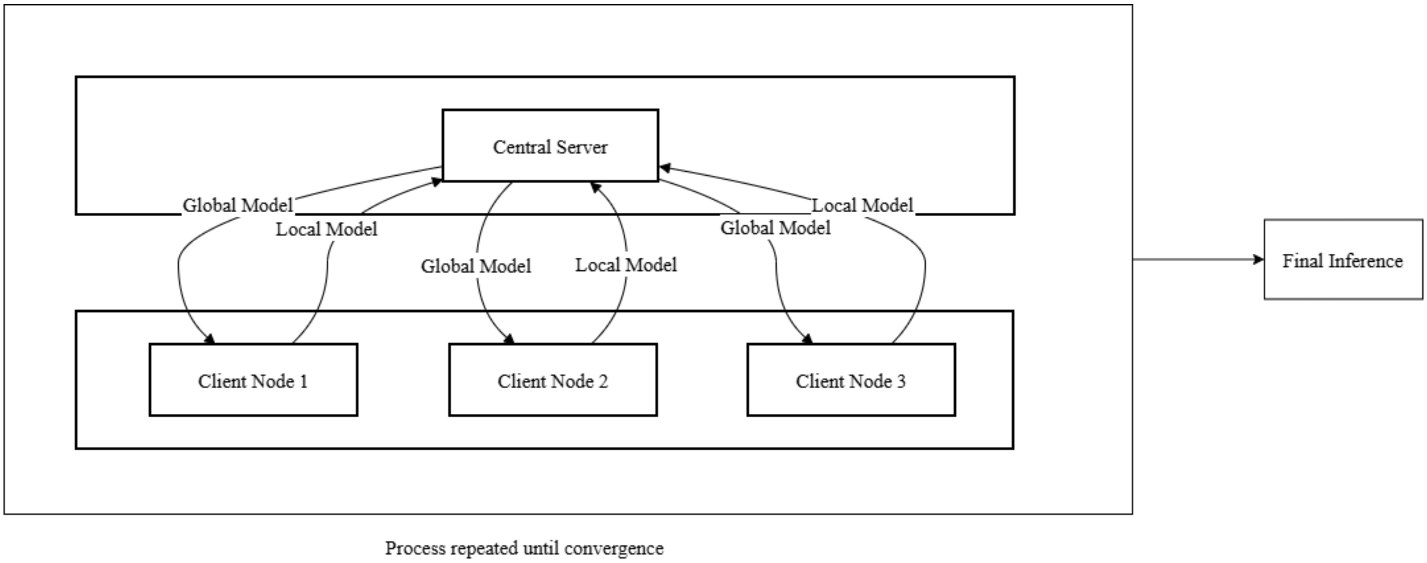
The main problem being addressed is training machine learning models for medical diagnosis. The problem is that patient data is personal and private. Different hospitals keep their data separate. Privacy laws prevent sharing data directly between hospitals.

Objective

Our goal is to create a system called federated learning. This system allows hospitals to work together to train a model. The model helps with diagnosing patients. It keeps patient information private. Hospitals do not share their actual data, only updates to the model.

Dataset Details:
Source: Kaggle
Size : 1048575 rows
Key features : Heart rate (HR), Temperature, (temp) Respiratory Rate, Blood pressure, Lactate level, WBC count, Age
Target variable : SepsisLabel

Architecture diagram



Methodology

Data Partitioning and Simulation

- Data division between four clients.
- IID partitions where the clients receive statistically comparable data.
- Non-IID partitions to allow for skewed distributions and dominant classes within each client.

Federated Learning Architecture

- Flower infrastructure coordinates federated learning and communication.
- Configuration consists of 4 clients, 3 comm rounds, 1 local epoch per comm round, and batch size 32.

- FedAvg algorithm averages model weights proportional to local dataset sizes.
- FedProx algorithm is used for non-iid data.

Architecture Overview

- A multi-layer perceptron MLP with 3 linear layers with size 128, 64, and output units respectively.
- Activation: ReLU; Regularization: dropout layers.
- Loss: cross-entropy, with focal loss option to combat class imbalance.
- Optimizer: Adam.

Privacy Mechanisms

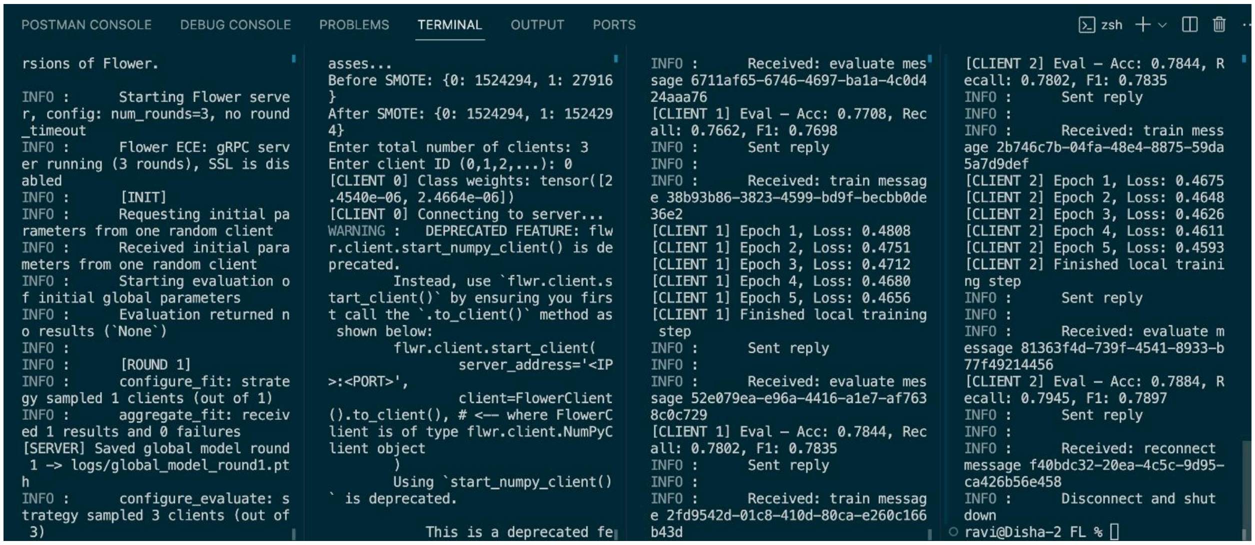
- Differential Privacy with Opacus incorporates calibrated noise in gradients that are measured by ε (epsilon).
- Homomorphic Encryption (HE) encrypts gradients sent from the clients to the server.
- These measures serve to alleviate risks, including gradient inversion and data reconstruction assaults.

Overcoming Problems

- Non-IID data processed with client-oriented learning and proper aggregation.
- Communication optimized by limiting rounds and parameter size.
- Client heterogeneity is dealt with through dynamic local training.

Results and Outcomes

Local FL training output:



```
rsions of Flower.

INFO :      Starting Flower serve
r, config: num_rounds=3, no round
_timeout
INFO :      Flower ECE: gRPC serv
er running (3 rounds), SSL is dis
abled
INFO :      [INIT]
INFO :      Requesting initial pa
rameters from one random client
INFO :      Received initial para
meters from one random client
INFO :      Starting evaluation o
f initial global parameters
INFO :      Evaluation returned n
o results (`None`)
INFO :
INFO :      [ROUND 1]
INFO :      configure_fit: strate
gy sampled 1 clients (out of 1)
INFO :      aggregate_fit: receiv
ed 1 results and 0 failures
[SERVER] Saved global model round
 1 -> logs/global_model_round1.pt
h
INFO :      configure_evaluate: s
trategy sampled 3 clients (out of
 3)
```

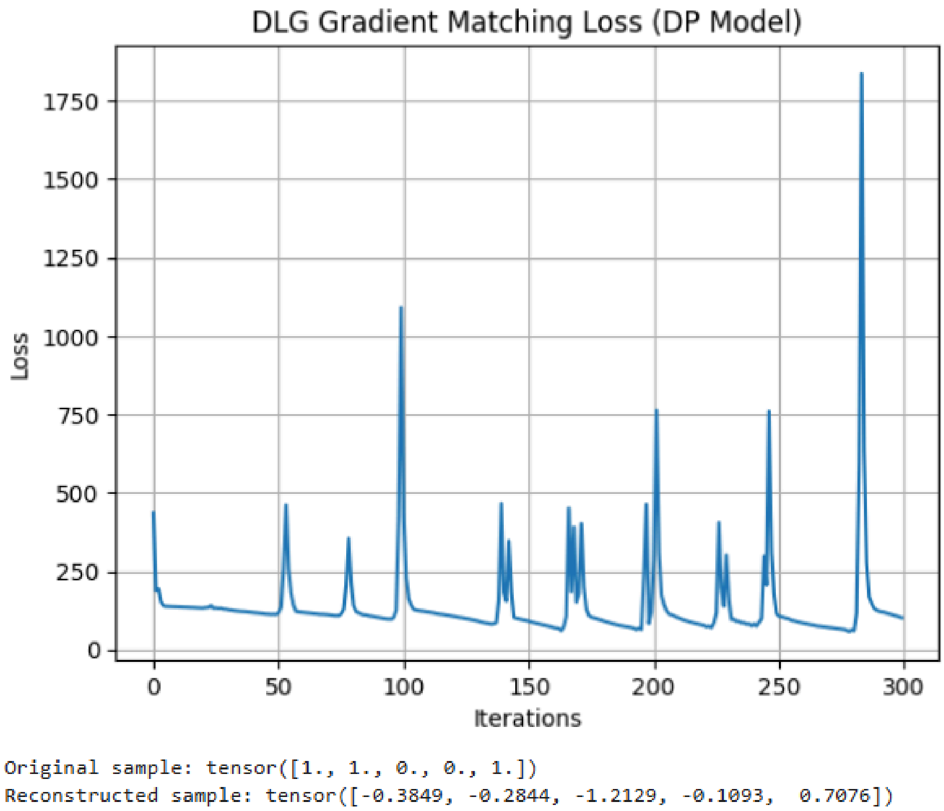
```
asses...
Before SMOTE: {0: 1524294, 1: 27916
}
After SMOTE: {0: 1524294, 1: 152429
4}
Enter total number of clients: 3
Enter client ID (0,1,2,...): 0
[CLIENT 0] Class weights: tensor([2
.4540e-06, 2.4664e-06])
[CLIENT 0] Connecting to server...
WARNING :   DEPRECATED FEATURE: flw
r.client.start_numpy_client() is de
precated.
        Instead, use `flwr.client.s
tart_client()` by ensuring you firs
t call the `.to_client()` method as
 shown below:
        flwr.client.start_client(
                server_address='<IP
>:<PORT>',
                client=FlowerClient
().to_client(), # <-- where FlowerC
lient is of type flwr.client.NumPyC
lient object
        )
        Using `start_numpy_client()
` is deprecated.

                This is a deprecated fe
```

```
INFO :      Received: evaluate mes
sage 6711af65-6746-4697-ba1a-4c0d4
24aaa76
[CLIENT 1] Eval - Acc: 0.7708, Rec
all: 0.7662, F1: 0.7698
INFO :      Sent reply
INFO :
INFO :      Received: train messag
e 38b93b86-3823-4599-bd9f-becbb0de
36e2
[CLIENT 1] Epoch 1, Loss: 0.4808
[CLIENT 1] Epoch 2, Loss: 0.4751
[CLIENT 1] Epoch 3, Loss: 0.4712
[CLIENT 1] Epoch 4, Loss: 0.4680
[CLIENT 1] Epoch 5, Loss: 0.4656
[CLIENT 1] Finished local training
 step
INFO :      Sent reply
INFO :
INFO :      Received: evaluate mes
sage 52e079ea-e96a-4416-a1e7-af763
8c0c729
[CLIENT 1] Eval - Acc: 0.7844, Rec
all: 0.7802, F1: 0.7835
INFO :      Sent reply
INFO :
INFO :      Received: train messag
e 2fd9542d-01c8-410d-80ca-e260c166
b43d
```

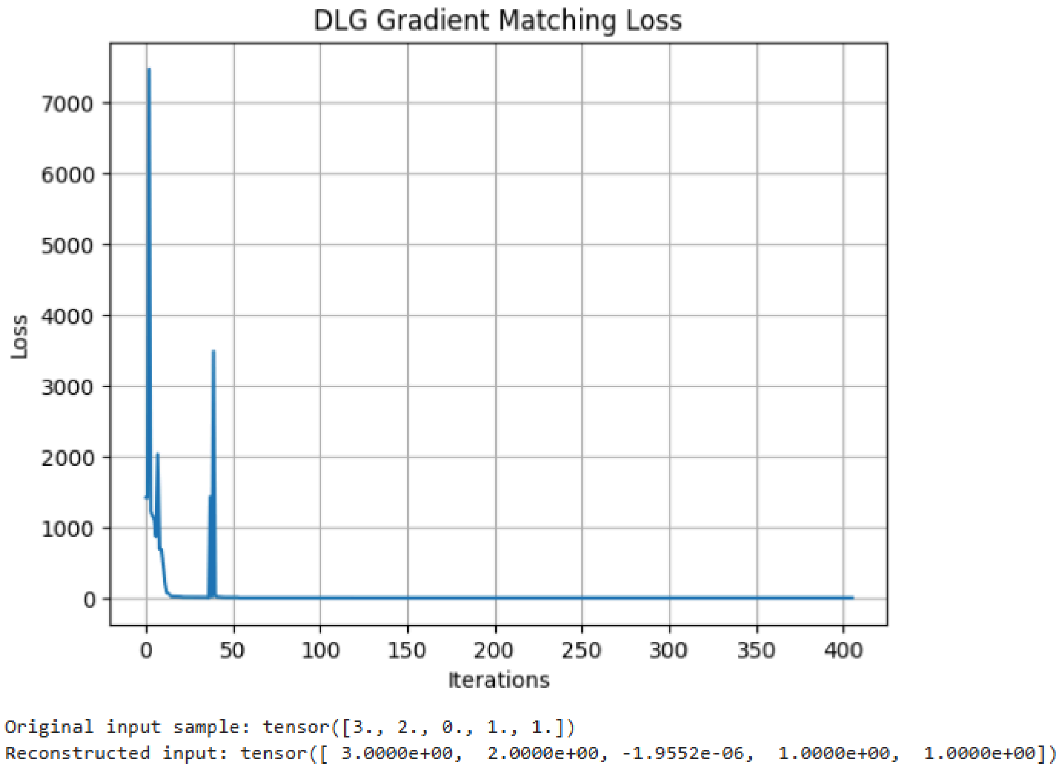
```
[CLIENT 2] Eval - Acc: 0.7844, R
ecall: 0.7802, F1: 0.7835
INFO :      Sent reply
INFO :
INFO :      Received: train mess
age 2b746c7b-04fa-48e4-8875-59da
5a7d9def
[CLIENT 2] Epoch 1, Loss: 0.4675
[CLIENT 2] Epoch 2, Loss: 0.4648
[CLIENT 2] Epoch 3, Loss: 0.4626
[CLIENT 2] Epoch 4, Loss: 0.4611
[CLIENT 2] Epoch 5, Loss: 0.4593
[CLIENT 2] Finished local traini
ng step
INFO :      Sent reply
INFO :
INFO :      Received: evaluate m
essage 81363f4d-739f-4541-8933-b
77f49214456
[CLIENT 2] Eval - Acc: 0.7884, R
ecall: 0.7945, F1: 0.7897
INFO :      Sent reply
INFO :
INFO :      Received: reconnect
message f40bdc32-20ea-4c5c-9d95-
ca426b56e458
INFO :      Disconnect and shut
down
○ ravi@Disha-2 FL % ▯
```

Gradient leakage with Differential Privacy (with reconstructed tensor)



```
Original sample: tensor([1., 1., 0., 0., 1.])
Reconstructed sample: tensor([-0.3849, -0.2844, -1.2129, -0.1093,  0.7076])
```

Gradient Leakage without Differential Privacy (with reconstructed tensor)



```
Original input sample: tensor([3., 2., 0., 1., 1.])
Reconstructed input: tensor([ 3.0000e+00,  2.0000e+00, -1.9552e-06,  1.0000e+00,  1.0000e+00])
```

Central Learning:

- ➔ Accuracy: 0.78
- ➔ Precision: 0.74
- ➔ Recall: 0.54
- ➔ F1 Score: 0.62
- ➔ AUC-ROC: 0.72
- ➔ Mean Absolute Error: 0.22
- ➔ Root Mean Squared Error: 0.47

IID Data:

Classification table:

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.98	0.94	0.96	342236
1	0.11	0.33	0.17	7417

- ➔ Accuracy: 0.93 (on 349,653 instances)
- ➔ Macro average: Precision = 0.55, Recall = 0.64, F1-score = 0.57
- ➔ Weighted average: Precision = 0.97, Recall = 0.93, F1-score = 0.95
- ➔ AUC: 0.7572
- ➔ Evaluation loss on test set: 0.48
- ➔ Evaluation accuracy on test set: 0.93

Non - IID data:

Classification table:

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.9869	0.9889	0.9879	751,215
1	0.0999	0.0861	0.0925	10,780

- Accuracy: 0.9761 (on 761,995 instances)
- Macro average: Precision = 0.5434, Recall = 0.5375, F1-score = 0.5402
- Weighted average: Precision = 0.9744, Recall = 0.9761, F1-score = 0.9752
- AUC: 0.7132

Performance Indicators

- Federated MLP models achieve test accuracy close to centralized baselines on IID and non-IID data.
- ROC-AUC and F1 scores confirm strong predictive power in the sepsis diagnosis.
- SMOTE oversampling enhances the minority imbalanced class's detection.

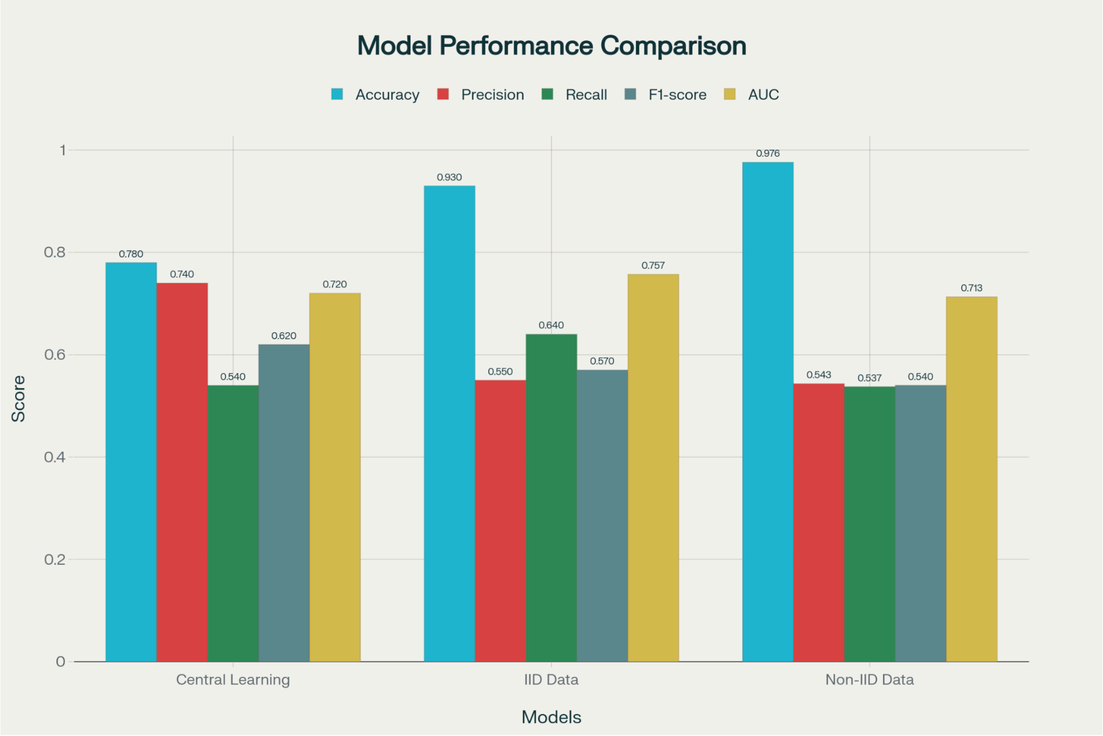
Privacy Analysis

- Differential privacy parameters provide tight privacy guarantees.
- HE prevents gradient-based privacy attacks.
- Conducted gradient leakage experiments to evaluate the privacy protection offered by the methods applied in this study

Efficiency

- Epochs and rounds of communication trade off between accuracy and time.
- Communication overhead is still an issue for scalability.

Visualisation of results:



Conclusion

This work certifies federated learning with MLPs on a real-world sepsis data set as an effective technique for privacy-protection medical diagnosis. This is achievable with performance similar to centralized ones but with stringent privacy requirements. Overcoming federated challenges like non-IID data, communication burden, and client heterogeneity is imperative to the success of its deployment to healthcare.