

# Federated Learning for Privacy-Preserving Heart Disease Prediction

**Alen Scaria**

Under the supervision of:

**Prof. Judhistir Mahapatro**

May 13, 2025



# Problem Statement

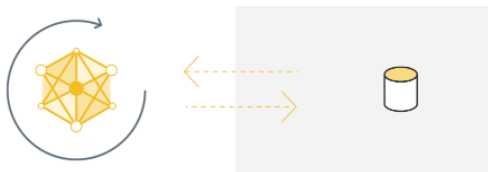
- Traditional heart disease prediction systems rely on centralized data collection, which raises serious concerns about data privacy and patient confidentiality.
- Monitoring in remote areas is difficult due to limited communication bandwidth and high latency in real-time queries.
- There is a need for a decentralized, privacy-preserving approach that allows collaborative model training without sharing raw patient data.

# Objective

- To develop a privacy-preserving heart disease prediction system using Federated Learning (FL) across distributed edge devices.
- To simulate a real-world healthcare environment by using edge devices for local training and a central server for model aggregation.
- To demonstrate that machine learning models can be effectively trained in a decentralized setting without compromising patient data privacy.

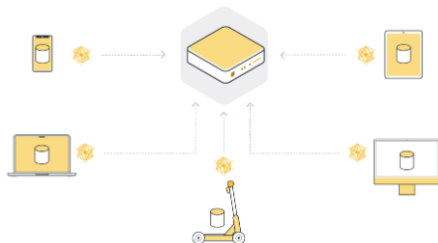
## Centralized ML and its Limitations

- Traditional ML aggregates data on a central server for training.
- Effective, but introduces risks: privacy breaches, security threats, and communication overhead.
- Especially problematic in sensitive domains like healthcare.



## Federated Learning (FL)

- FL enables model training without moving data from client devices.
- Clients train models locally and share only model parameters.
- Supports data privacy, reduced bandwidth use, and edge intelligence.



## Why Focus on Heart Disease & Decentralization

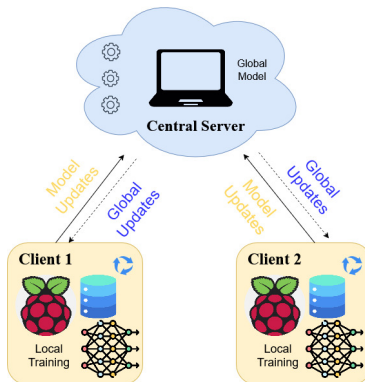
- Heart disease is a leading cause of death globally — often due to undiagnosed risk factors.
- Early detection is crucial: “Prevention is better than cure.”
- Healthcare data is highly sensitive; centralized AI is limited by privacy concerns.

## Literature Review

- A paper by Srinivasa Raju Rudraraju et al.[1] explored the application of FL to process heterogeneous sensor data in a fog computing-based smart home environment.
- Several studies have applied FL to heart disease prediction using datasets and various models (SVM, Logistic Regression, ANN, etc.).
- These lack hardware-level testbeds that mimic real-world edge deployments.
- Our work aims to bridge this gap by implementing a hardware-supported FL setup, simulating realistic edge nodes for heart disease prediction.

# System Architecture

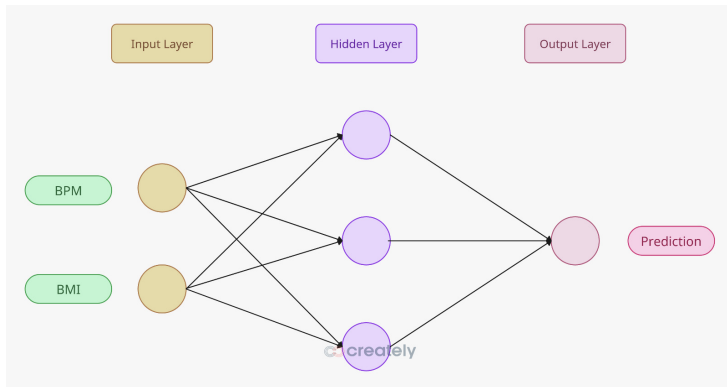
- The architecture follows a centralized server + multiple edge clients model using Federated Learning.
- The communication follows a cyclical update loop: local training → model update → global aggregation → updated model distribution.





# Model Architecture

- The model used is a simple feedforward Artificial Neural Network (ANN) with 2 input neurons, 1 hidden layer of 3 neurons, and 1 output neuron for binary classification.



## 1. Dataset Preparation & Split

- Used an existing heart disease dataset from Kaggle with relevant features such as BPM and BMI.
- The dataset was split into IID subsets and saved as separate files for each client.

## 2. Client Side Setup

- Two Raspberry Pi 3B devices (1GB RAM) were used to simulate real-world edge clients.
- Each client loads its respective data and trains a local model independently.
- Remote access was enabled via SSH using RealVNC due to resource limitations.

## 3. Server-Side Setup

- A central server (laptop) coordinates the federated learning process across all clients.
- Connected to the local WiFi to establish the connection with the clients over WiFi.
- Flower<sup>1</sup> — an open-source framework for Federated Learning — is used to build the server and client interfaces.
  - Provides a high-level API to implement FL without dealing with low-level networking.
  - Uses gRPC protocol to ensure seamless and efficient communication between server and clients.

---

<sup>1</sup>Source : <https://flower.ai/docs/>

## 3. Flower Workflow

- **Initialization:** Server starts a federated learning round.
- **Client Selection:** Clients (Raspberry Pis) are selected.
- **Local Training:** Each client trains the model on its local data for one or more epochs.
- **Model Update:** Clients send their trained model weights back to the server.
- **Aggregation:** Server performs model aggregation (FedAvg) to update the global model.
- **Repeat:** Process is repeated over multiple rounds to improve global model accuracy.

# Federated Learning Algorithm

---

**Algorithm 1** Federated Learning

---

1: **Server-Side:**

2: Initialize global model weight  $w_0$

3: **for** each round  $t = 0, 1, \dots$  **do**

4:   **for** each client  $i = 0, \dots, n - 1$  **in parallel do**

5:     Send  $w_t$  to client  $i$

6:     Receive updated weight  $w_{t+1}^{(i)}$  from client  $i$

7:   **end for**

8:

$$w_{t+1} \leftarrow \frac{1}{n} \sum_{i=1}^n w_{t+1}^{(i)}$$

9: **end for**

10:

11: **Client-Side:**

12: Split local dataset into batches of size  $B$

13: **for** each local epoch  $e = 0, \dots, E - 1$  **do**

14:   **for** each mini-batch  $b$  of size  $B$  **do**

15:     Update local model:

$$w_t = w_t - \eta \cdot \nabla L(w, b)$$

16:   **end for**

17: **end for**

18: Return updated weight  $w_{t+1}$  to the server

---

Algorithm [2]

## 4. Simulation

```
INFO : Starting Flower server, config: num_rounds=3, no round timeout
INFO : Flower ECE: gRPC server running (3 rounds), SSL is disabled
INFO : [INIT]
INFO : Requesting initial parameters from one random client
INFO : Received initial parameters from one random client
INFO : Starting evaluation of initial global parameters
INFO : Evaluation returned no results ('None')
INFO :
INFO : [ROUND 1]
INFO : configure_fit: strategy sampled 2 clients (out of 2)
INFO : aggregate_fit: received 2 results and 0 failures
```

```
[Round 1] Received weights from clients:
Client 0 weights:
  Layer 0 weights:
[[ -3.34147042 -3.25449577 -6.69596846]
 [ 6.41441036  5.85700266 -17.32869116]]
  mean=-3.058202, std=8.022990
Layer 1 weights:
```

Figure: Start of Simulation

```
weights_hidden_output:
[[ 1.74347161]
 [ 1.06463373]
 [-11.19361713]]
-----
[Epoch 1/10] Loss: 0.083674
[Epoch 2/10] Loss: 0.082465
[Epoch 3/10] Loss: 0.081537
[Epoch 4/10] Loss: 0.080954
[Epoch 5/10] Loss: 0.080676
[Epoch 6/10] Loss: 0.080972
[Epoch 7/10] Loss: 0.084355
[Epoch 8/10] Loss: 0.082252
[Epoch 9/10] Loss: 0.089465
[Epoch 10/10] Loss: 0.080680
INFO : Sent reply
INFO :
INFO : Received: evaluate message d301f894-f39d-4d9c-a464-86ed7837ad29
[EVALUATE] Loss: 0.072487, Accuracy: 0.9100, Samples: 3588
INFO : Sent reply
INFO :
INFO : Received: reconnect message 7c6b682d-6c16-4b9f-89f3-4f9ac9c37fb5
INFO : Disconnect and shut down
```

Figure: End of Simulation

## 5. Visualization Analysis

- Used the Matplotlib library to plot training metrics across federated rounds.
- Plotted client-wise accuracy and loss to monitor local training performance.
- Tracked how the global model accuracy improved after each round of aggregation.

## Global Model Performance

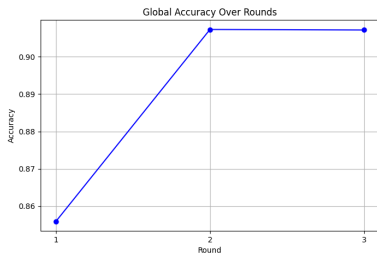


Figure: Global Accuracy

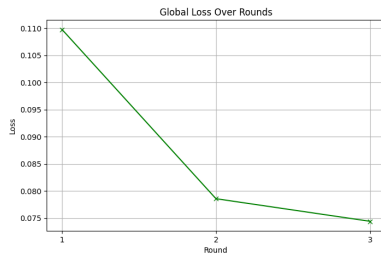


Figure: Global Loss



## Client Side Performance

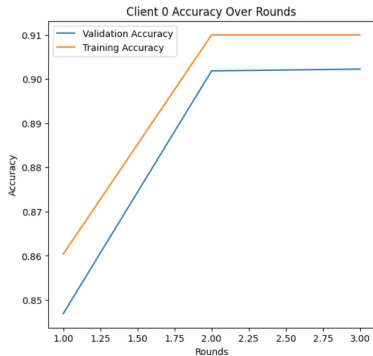


Figure: Client 0

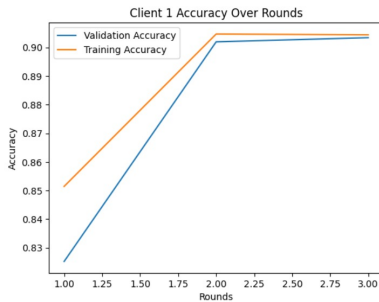


Figure: Client 1



# Challenges Faced

- Raspberry Pi's 32-bit ARM architecture and 1GB RAM limited support for standard ML libraries like TensorFlow and PyTorch.
- Manual implementation of the neural network using NumPy was required due to library incompatibilities.
- Limited memory on edge devices caused occasional instability during local training.
- Attempted real-time sensor integration failed due to unreliable heartbeat sensor performance.

- Integrate real-time physiological sensors for realistic health data simulation and better IoT-healthcare alignment.
- Upgrade to more powerful edge devices (e.g., Raspberry Pi 4 ) to support complex ML frameworks like TensorFlow/PyTorch.
- Scale up the number of clients to evaluate the performance and scalability of the Flower framework.
- Simulate non-IID data distributions across clients to test model robustness under real-world data heterogeneity.

# Conclusion

This project demonstrated the feasibility of deploying Federated Learning (FL) on low-powered edge devices such as Raspberry Pi for a privacy-sensitive task like heart disease prediction. Despite hardware and data limitations, the system effectively simulated a decentralized learning environment, highlighting FL's potential in healthcare scenarios. While the model was kept intentionally simple, its capacity to flag high-risk cases, even with modest accuracy, underscores its practical utility in early intervention. Overall, this work emphasizes how FL can address critical challenges in healthcare AI, such as data privacy and diverse data access, without centralizing sensitive patient information.

-  Rudraraju, S. R., Suryadevara, N. K., and Negi, A., 2023.  
“Heterogeneous sensor data acquisition and federated learning for resource constrained iot devices—a validation”.  
*IEEE Sensors Journal*, **23**(15), Jun., pp. 17602–17610.
-  McMahan, H. B., Moore, E., Ramage, D., and y Arcas, B. A., 2016.  
“Federated learning of deep networks using model averaging”.  
*CoRR*, **abs/1602.05629**.