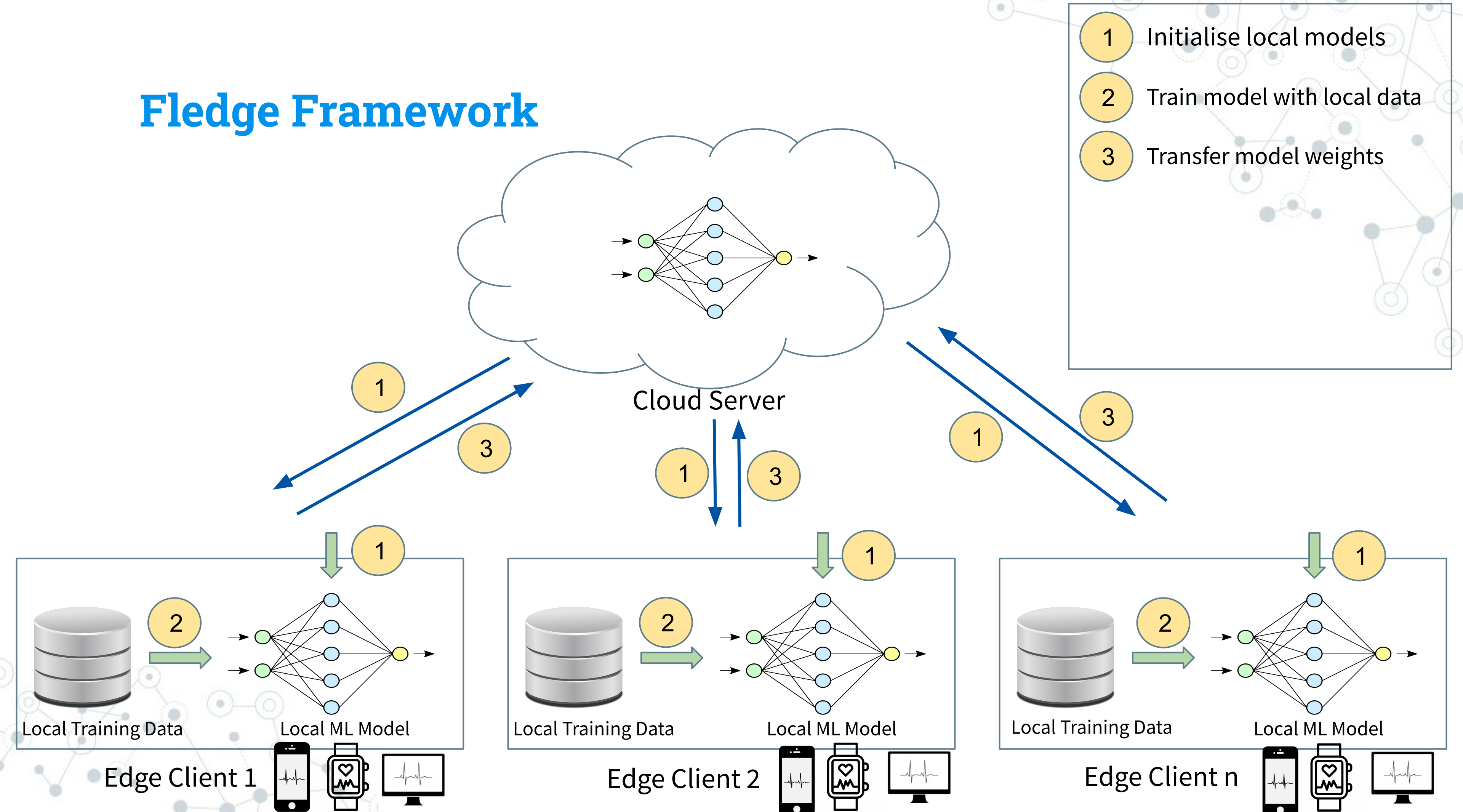
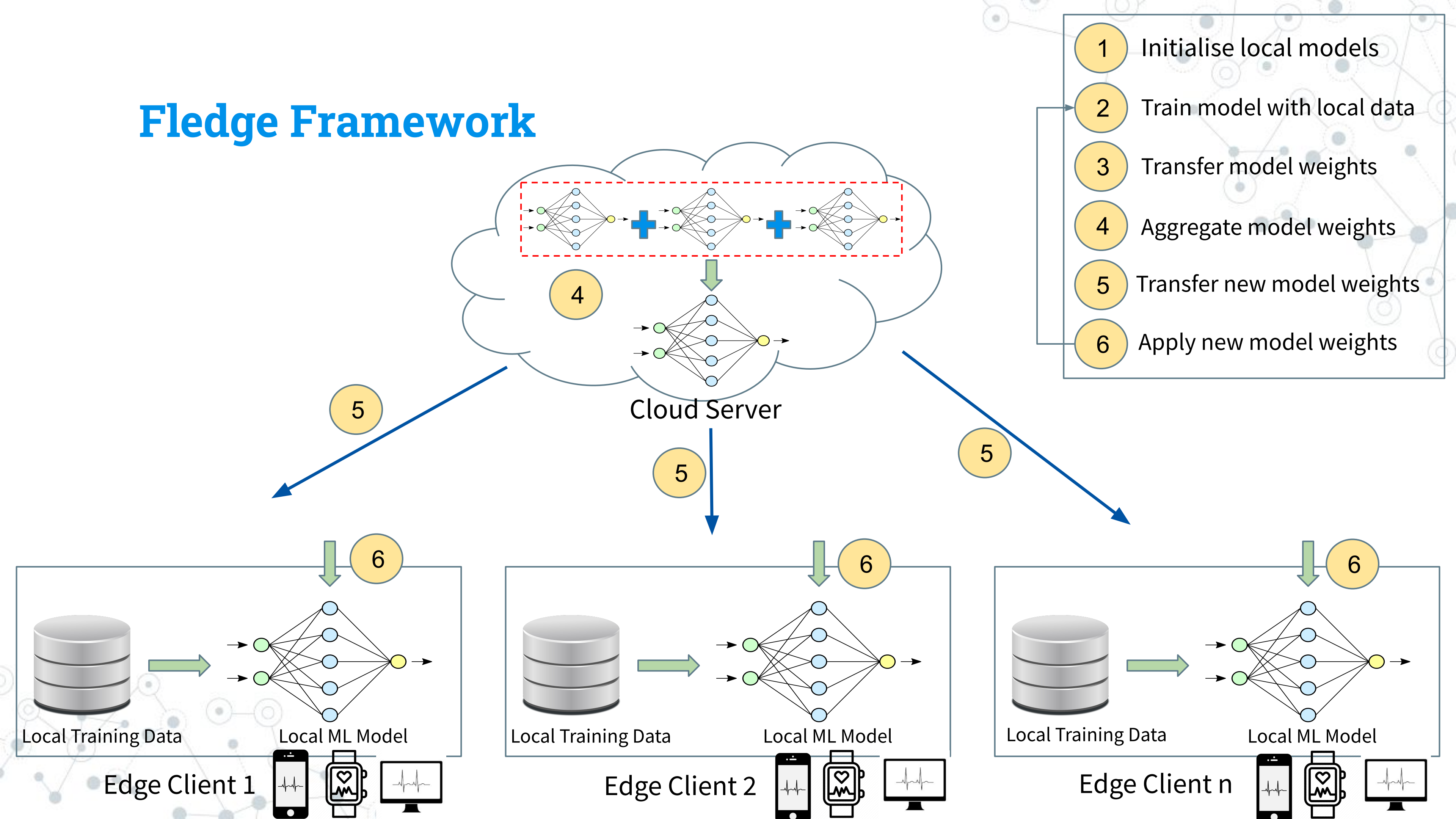
Fledge: Edge-based Federated Learning Framework for Mobile Healthcare

Presented By: Pritish Mishra

Fledge Framework




Fledge Framework
1 Initialise local models
2 Train model with local data
3 Transfer model weights
4 Aggregate model weights
5 Transfer new model weights
6 Apply new model weights
4
Cloud Server
5
5
5
6
6
6
Local Training Data
Local ML Model
Edge Client 1
Local Training Data
Local ML Model
Edge Client 2
Local Training Data
Local ML Model
Edge Client n

Fledge Framework : Key Features

- **Generic:** Provides support for any ML model uploaded by model/application developer.
- **Real-world Deployment:** Deployment environment supports actual edge devices on real-world infrastructure, rather than only simulated mode.
- **Resource Awareness:** Framework must split the model intelligently between available resources to ensure good resource utilisation and accuracy.

Existing Federated Learning Frameworks

- FATE
- PaddleFL (PFL)
- TensorFlow Federated (TFF)
- PySyft
- IBM Federated Learning

Existing Federated Learning Frameworks

	Generic	Real-world Deployment	Resource Awareness
FATE	❌	✅	❌
PaddleFL (PFL)	❌	✅	❌
TensorFlow Federated (TFF)	✅	❌	❌
PySyft	✅	❌	❌
IBM Federated Learning	✅	✅	❌

Prior Work in ML Healthcare

- “FEEL: A Federated Edge Learning System for Efficient and Privacy-Preserving Mobile Healthcare”
- Proposes splitting of model to manage resource consumption
- Just a proof-of-concept of the splitting idea, no algorithm proposed for a generic framework

Why Federated Learning for ML Healthcare?

- Proliferation of data collection points - smart wearables, electronic health records, personal health monitors
- Centralised processing of all information
- Privacy-sensitiveness of data
- Data ownership
- Complexity of ML models

Research Questions

- RQ1: Can the federated mode of the framework provide similar accuracy as centralised mode? (Accuracy vs Privacy)

- RQ2: Does the federated mode of the framework add any extra overheads on performance? (Training time, Communication latency, Processing overheads)

- RQ3: Can we design an algorithm that splits any given ML model to maintain good resource consumption and accuracy?

Perf Eval - Dataset

- MNIST Dataset: Images of handwritten numbers from 0-9.

  No. of datapoints: 70,000 images

- Breast Cancer Dataset: Multiple features of patient with cancer diagnosis - malignant or benign

  No. of datapoints: 648

  No. of features: 9

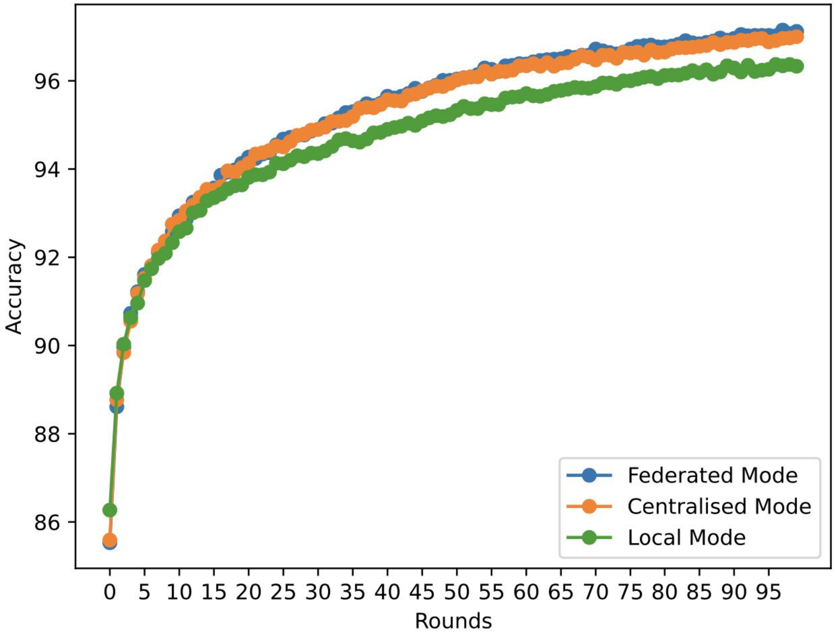
Perf Eval - Candidates

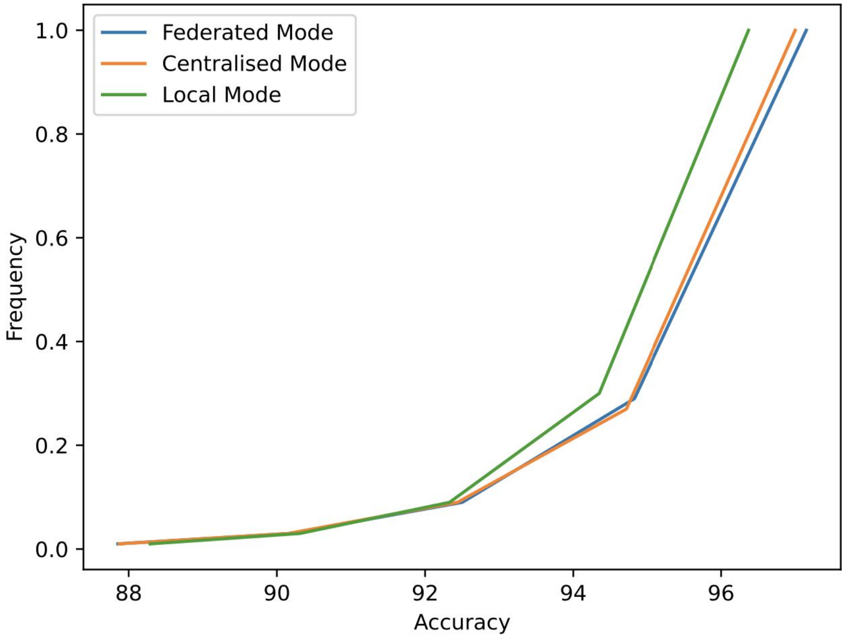
- Centralised Mode: Data collected from all clients available at the centralised server to train a global model.
- Local Mode: Each client has its own local data and local ML model to train.
- Federated Mode: Each client uses its own local data to train a globally distributed and aggregated model. Model params aggregated using FedAvg algorithm.

Perf Eval - Infra

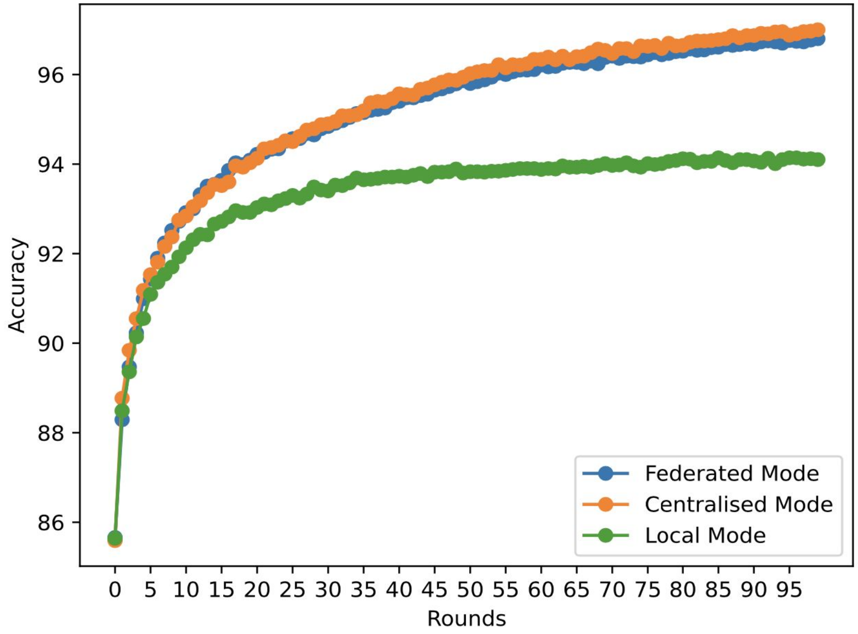
- Server Node (Cloud):
  - CPU: 3.1Gz @ Intel Xeon® Platinum 8175M
  - Memory: 16 GB
  - Cores: 4
- Client Node (Edge):
  - CPU: AWS Graviton Processor
  - Memory: 8 GB
  - Cores: 1

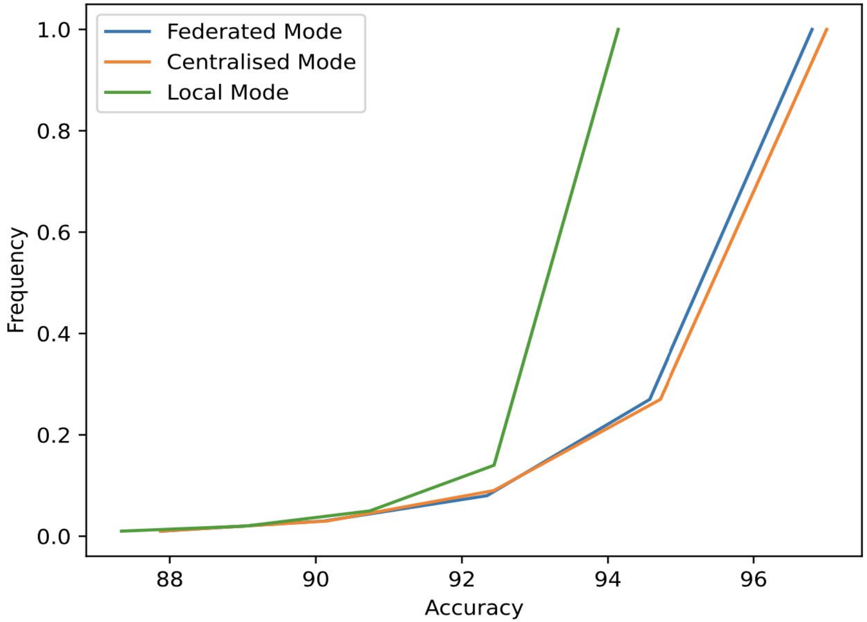
Performance Evaluation: Accuracy



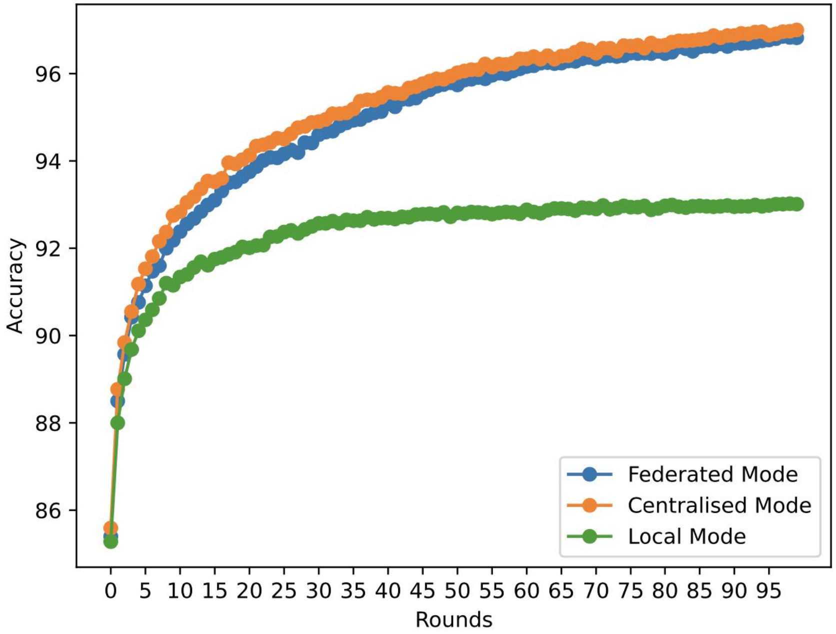


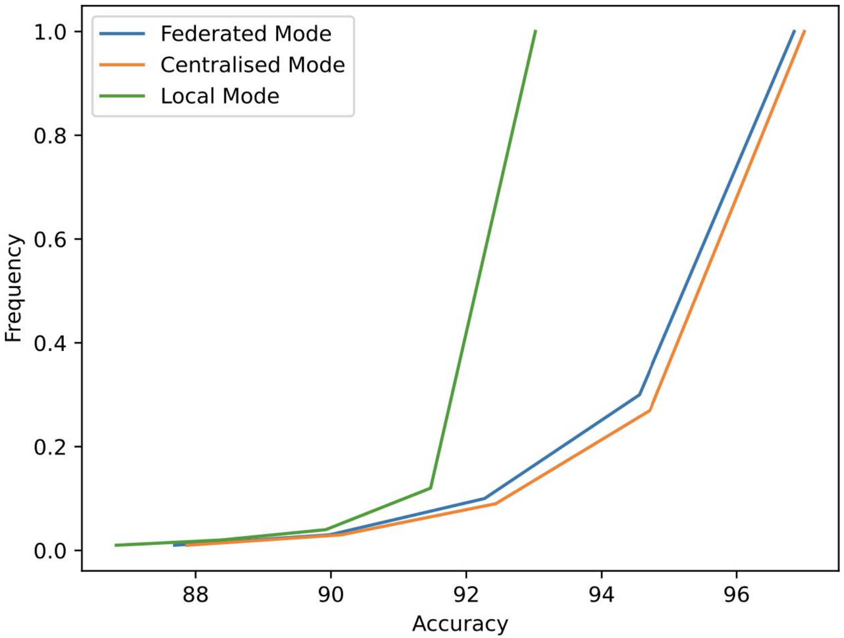
Performance Evaluation: Accuracy



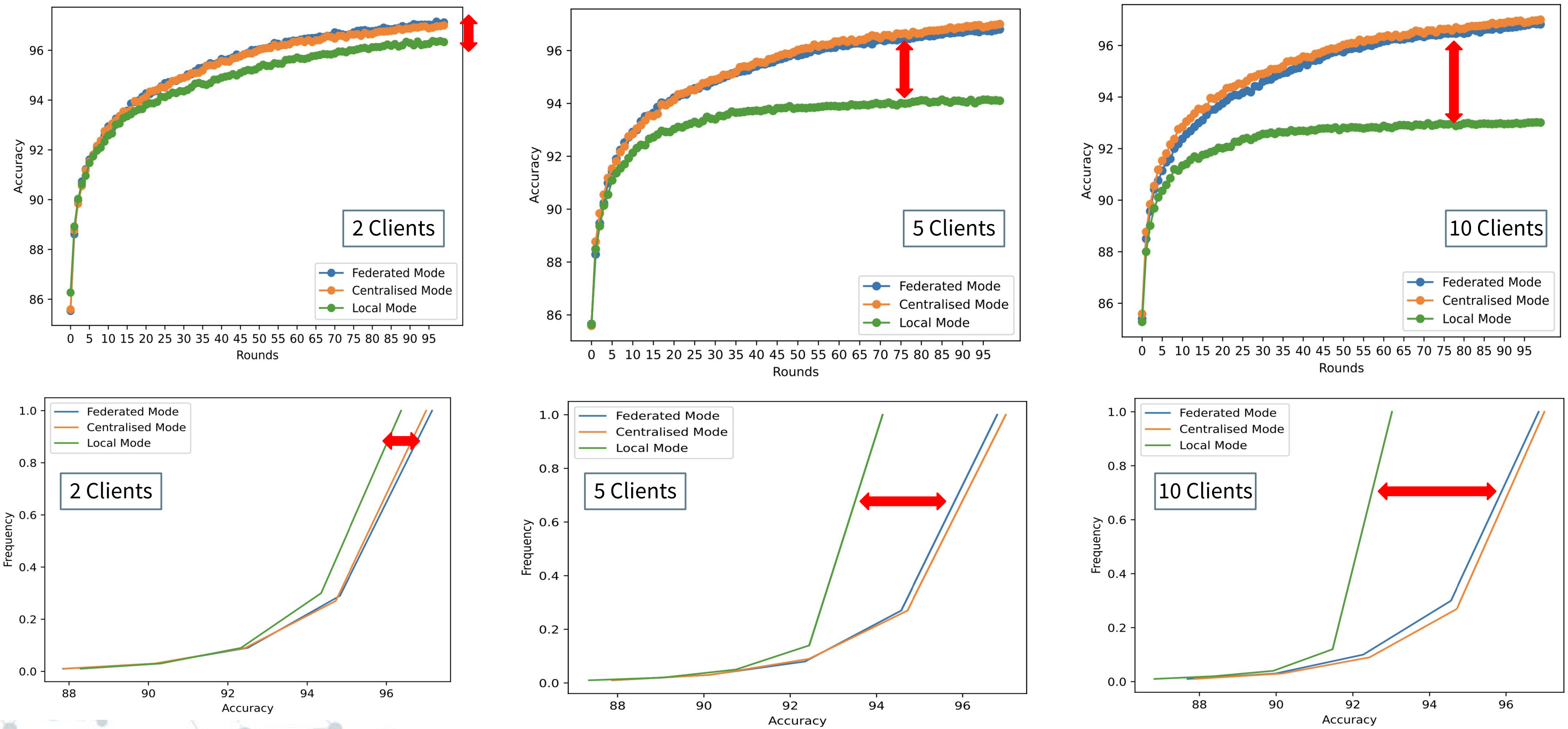


Performance Evaluation: Accuracy

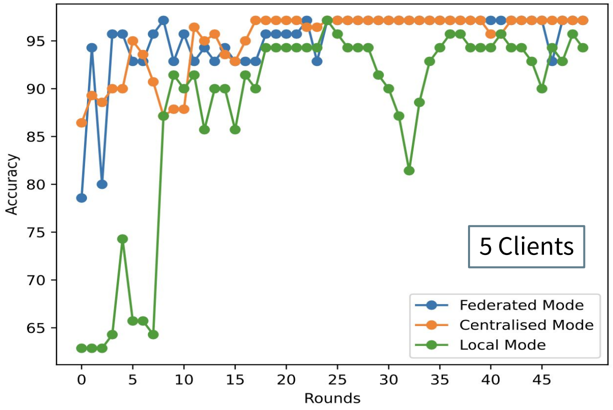


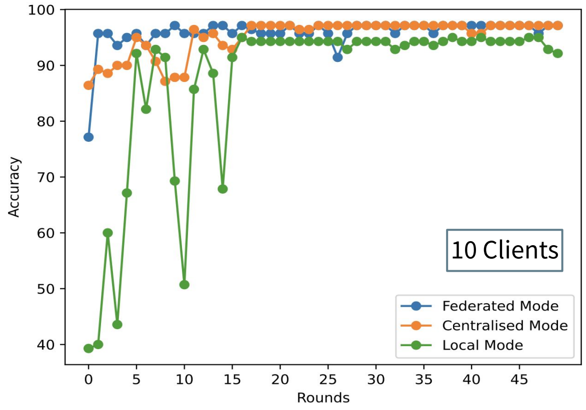


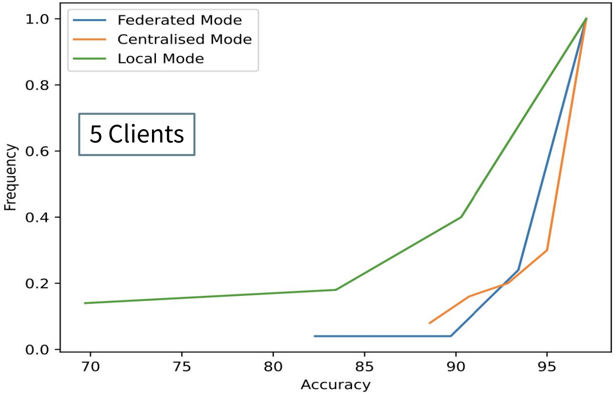
Perf Eval - Accuracy - MNIST Dataset

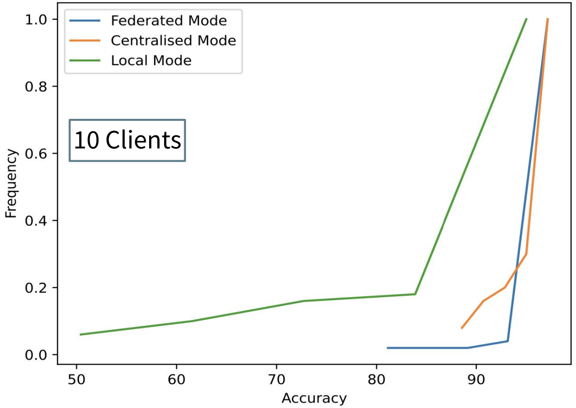


Perf Eval - Accuracy - Breast Cancer Dataset

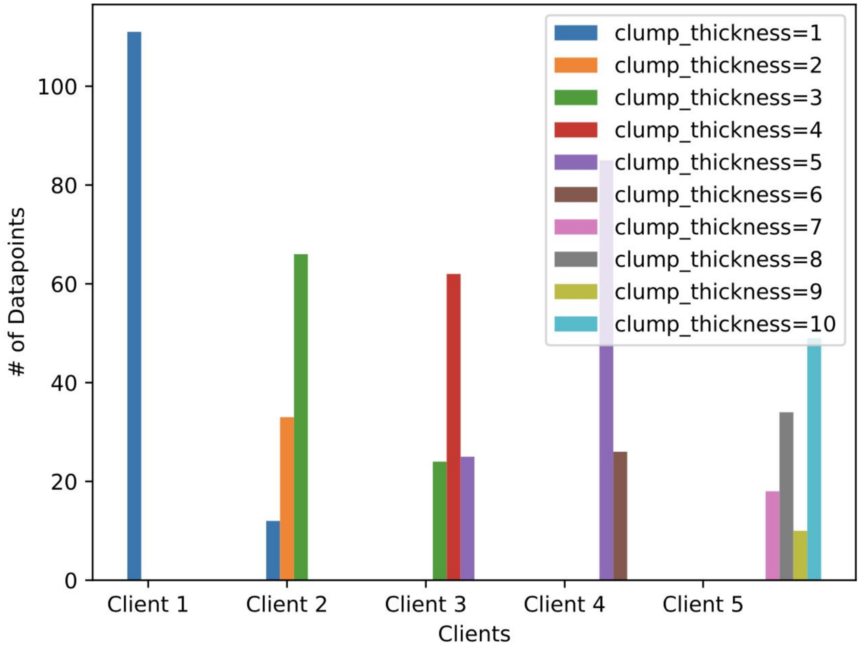




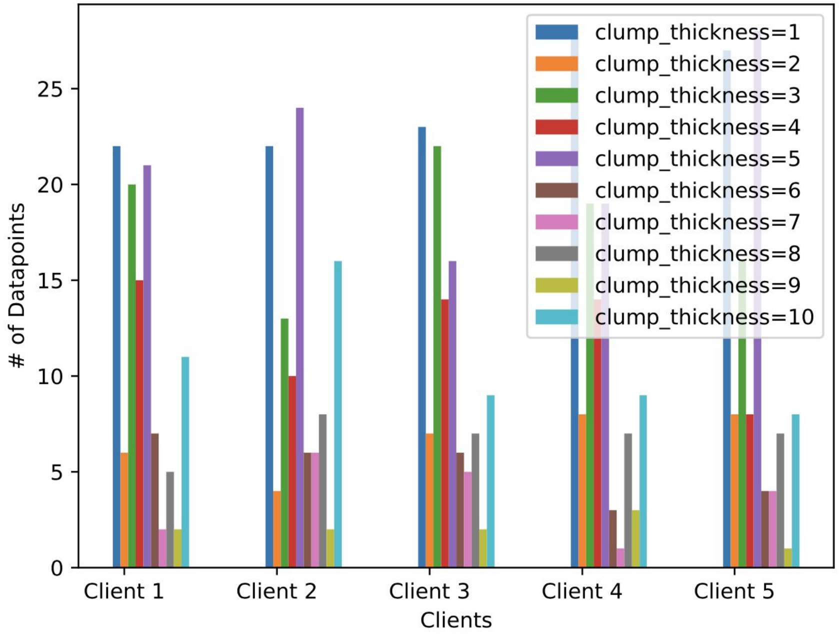




Perf Eval - Data Distribution

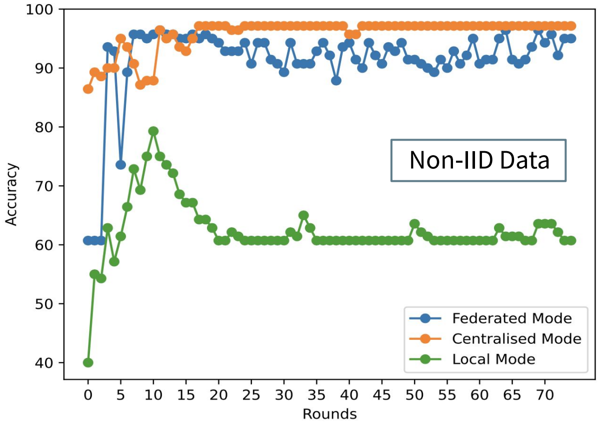


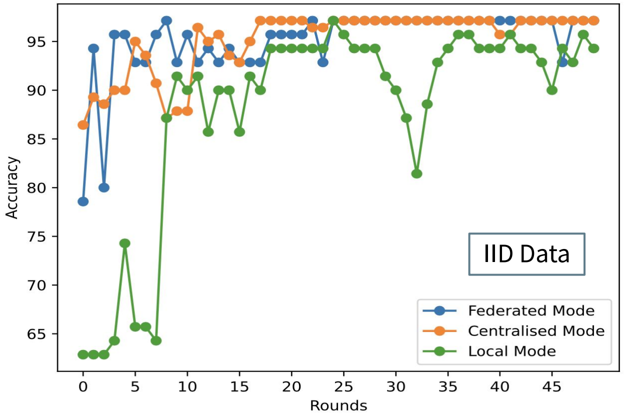
Non-IID Data Distribution

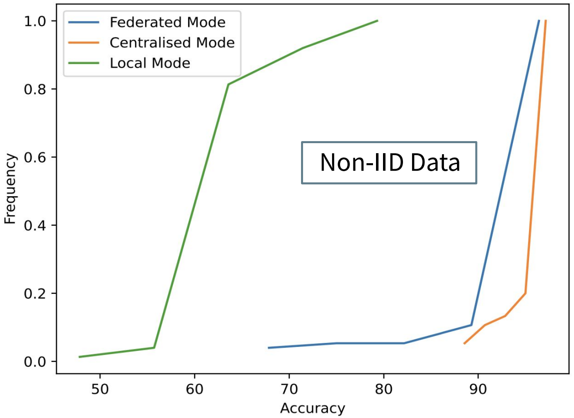


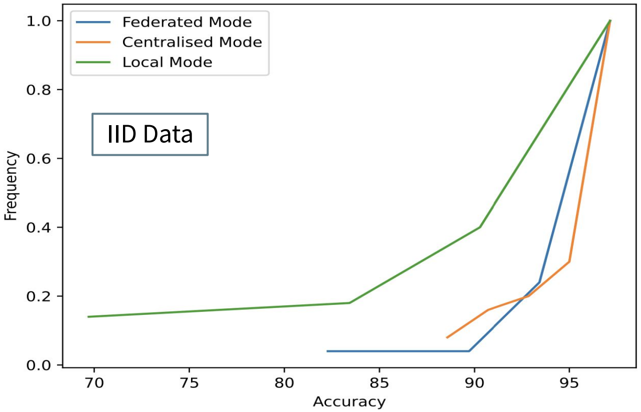
IID Data Distribution

Perf Eval - Accuracy - Breast Cancer Dataset

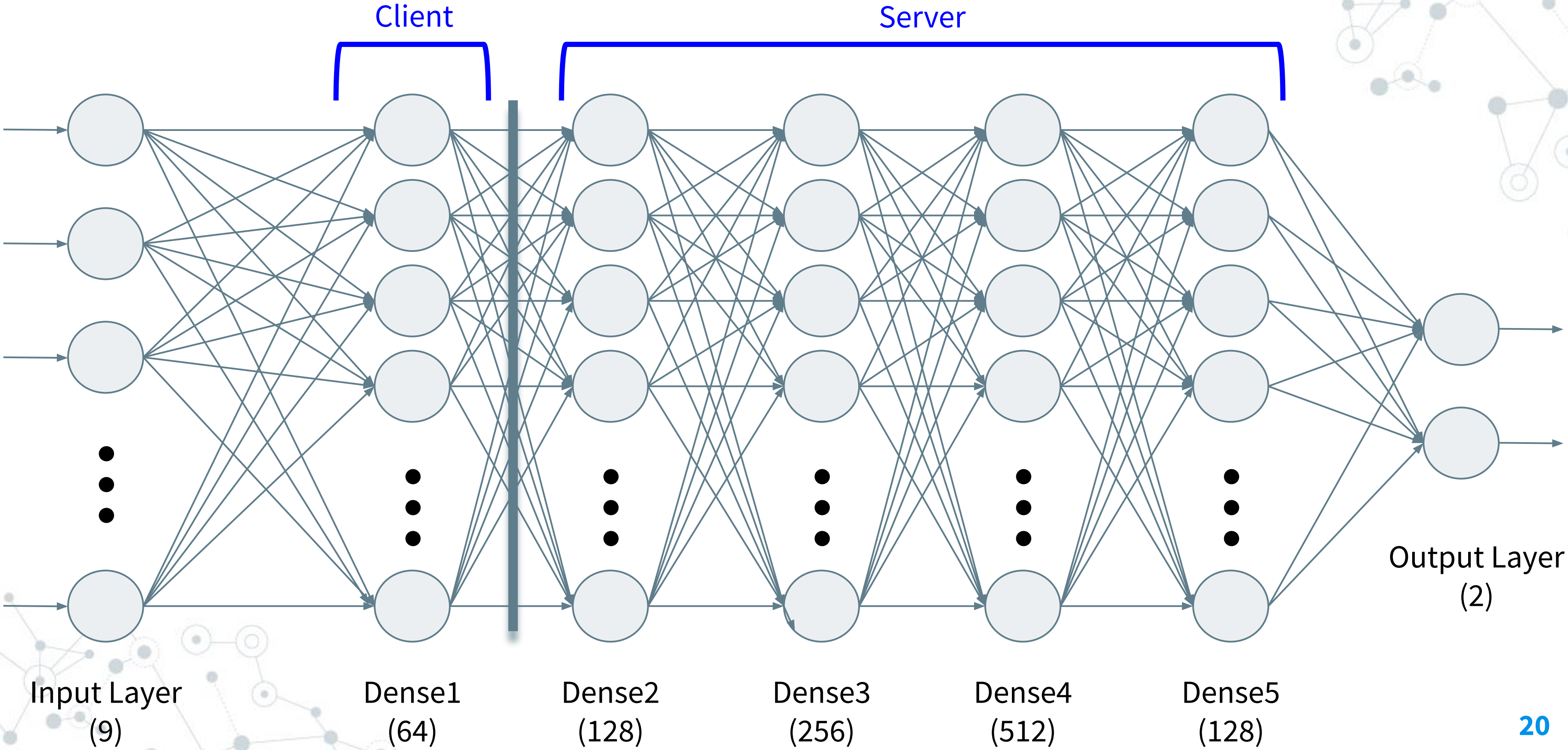








Splitting ML Model for Breast Cancer Dataset



Perf Eval - Pending Results

- Training times of each mode
- Resource consumption when entire model is deployed
- Resource consumption when the model is split
- Accuracy of model in split mode vs federated mode

Limitations / Future Work

- Algorithm to derive the split point of a model
- Minimising processing and communication latency
- Experiments to include more complex models, unsupervised models, larger infrastructure deployment, etc.
- Applicable only for fully connected layers in neural network

Thank You!