Federated Learning with Heterogeneous Models (ConvLSTM, GRU)

Group 5

1.Arkadeep Acharya (2101AI41) 2. Aritra Bhaduri (2101AI40)

3. Rakesh Kumar (2101AI26)

4.Yashveer (2101AI35) 5. Sunny Kumar Singh (2101AI33)

GitHub Link: aritroCoder/bigdata_proj

What is Federated learning?

- Federated Learning is a machine learning approach where multiple devices or systems collaboratively train a shared model without exchanging raw data.
- Instead of transferring data to a central server, each device trains the model on its own data and sends only the model updates, such as gradients, to the server.

Advantages

1. **Privacy Preservation**: Raw data remains on local devices, reducing the risk of data breaches or leaks.
2. **Reduced Bandwidth Usage**: Only model updates are shared, minimizing data transfer requirements.
3. **Access to Diverse Data**: Enables training on data from multiple sources without centralizing it, improving model generalization.

How federated learning helps in big data?

- **Data Privacy and Security:**
  - **Data Silos:** In many industries, data is fragmented across different organizations or devices due to privacy regulations or proprietary concerns. FL enables collaboration without compromising data security.
  - **Sensitive Data:** FL can be used to train models on sensitive data like medical records or financial transactions, ensuring that the data remains private
- **Scalability**
  - Distributed Training: FL can distribute the training process across multiple devices or servers, accelerating the training process and enabling the handling of massive datasets
  - Heterogeneous Devices: It can accommodate devices with varying computational capabilities, making it suitable for diverse environments
- **Efficiency**

- **Reduced Communication:** By minimizing the amount of data transferred between devices and the central server, FL reduces network traffic and bandwidth requirements.
- **Local Training:** Training models locally on devices reduces the computational burden on a central server, improving efficiency

Why Time Series is Considered Big Data?

High Volume:

- Time series data is generated continuously by devices, sensors, and applications, leading to vast datasets.
- Examples include IoT devices, financial transactions, and website logs.

High Velocity:

- Data points are recorded at high frequencies, such as milliseconds or seconds, leading to rapid data accumulation.

High Variety:

- Data comes from diverse sources, including sensors, stock markets, weather stations, and application logs.
- Can include numerical readings, categorical events, or metadata.

Why Can Videos be considered as Big Data

High Volume:

- Video datasets typically consist of over 13,000 video clips, resulting in massive data storage requirements due to high-resolution frames, numerous action categories, and detailed annotations for each clip.

High Velocity:

- Videos are captured and processed at high frame rates (e.g., 30 frames per second), generating a rapid influx of sequential data for continuous temporal analysis and real-time processing.

High Variety:

- Videos can encompass diverse action categories and scenarios, combining spatial (frame-based features) and temporal (sequence-based transitions) dimensions, which increase data complexity and representation challenges.

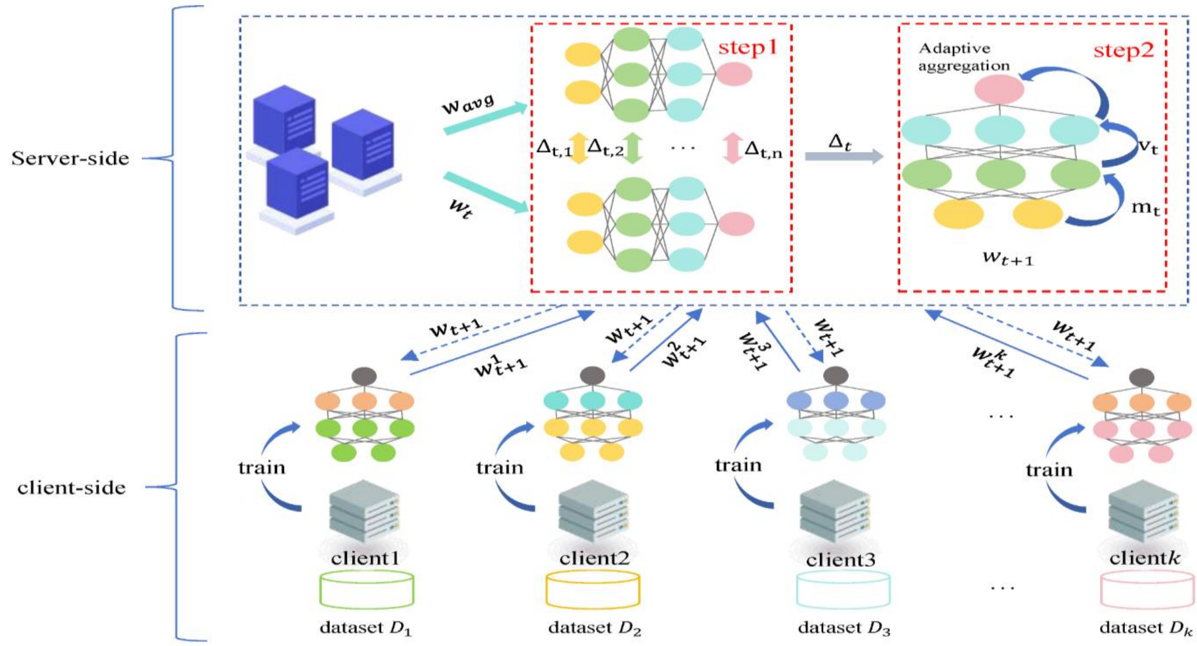
Working

- **Dataset Generation**:The datasets are divided into multiple non overlapping files to simulate the datasets from each client.

- **Trainer Algorithm**: The system supports different training approaches like FedAvg (a common federated learning algorithm) and a custom aggregation function. This allows flexibility to experiment with various training methods.

- **Model Configuration**: Users can specify models like GRU for sequence data or ConvLSTM for spatio-temporal (Video) data.

- **Training Rounds**: The process involves multiple iterations (rounds) where updates are gathered from all participating devices or datasets and combined to improve the global model.



Components of Federated Learning.

Clients:

1. Train models locally based on specific data types (e.g., GRU for time-series, ConvLSTM for Spatio-Temporal Data), ensuring privacy by keeping raw data on-device.
2. Share model parameters or gradients (not data) with the central server, enabling secure and distributed training.

Server:

1. Aggregates client model updates using algorithms like FedAvg or custom aggregation techniques to create a global model that handles heterogeneous architectures.

2. Evaluates the global model on a centralized test dataset, ensuring it generalizes across diverse data sources.

Aggregator Functions

1. FedAvg

$$w_{\text{global}} = \frac{1}{N} \sum_{i=1}^{N} n_i \cdot w_i$$

where:

- w_{global} is the updated global model.
- N is the total number of clients.
- n_i is the number of training samples on client i.
- w_i is the model weights from client i.

2. Custom

For each parameter k, the global model weight G_k is computed as:

$$G_k = \frac{\sum_{i=1}^{N} w_i' \cdot P_{i,k}}{\sum_{i=1}^{N} w_i'}$$

where:

- $w_i' = w_i \cdot b$ (10% chance) or $w_i \cdot (2 - b)$ (90% chance).
- w_i: Original dataset size of client i.
- $P_{i,k}$: Parameter k from client i.
- b: Bias factor (e.g., 0.65).

Results

Task 1: Times Series Analysis on NAB: Numenta Anomaly Benchmark

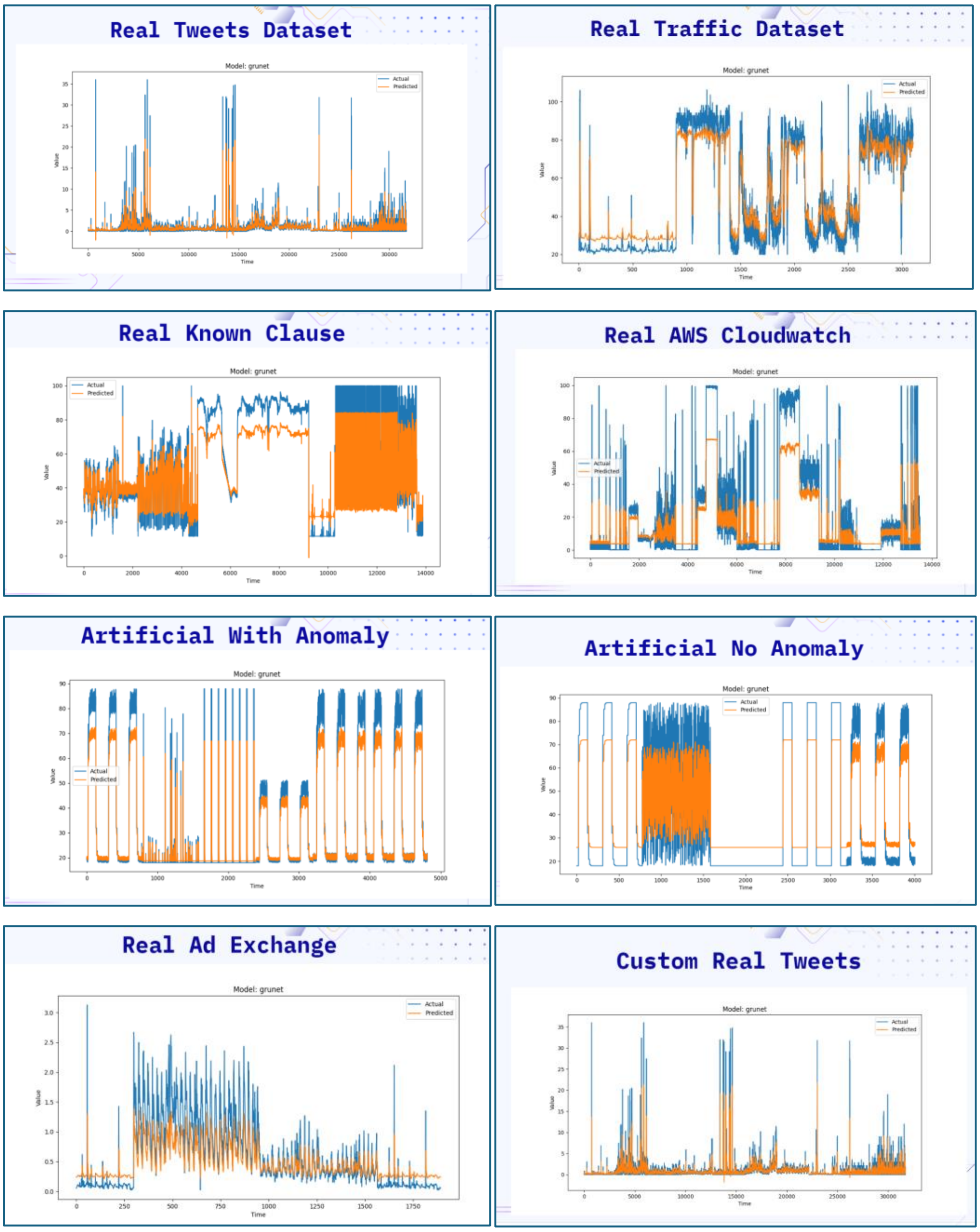
Training hyperparameters

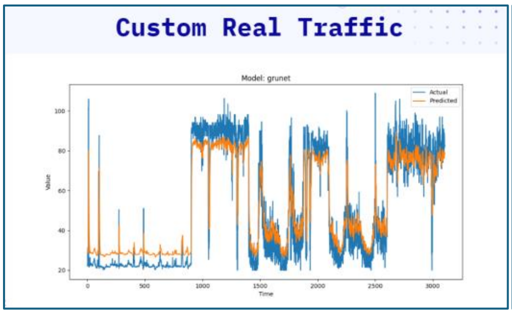
Model Type: GRU

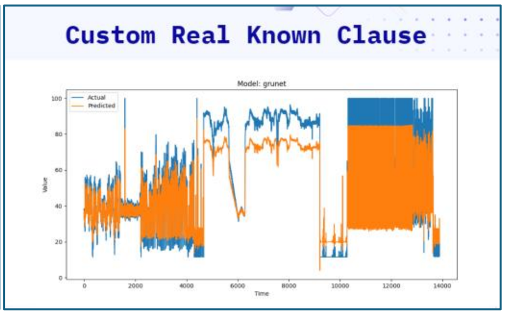
- Number of clients:
  - Artificial No Anomaly: 5
  - Artificial With Anomaly: 6
  - Read Ad Exchange: 6
  - Real AWS Cloudwatch: 17

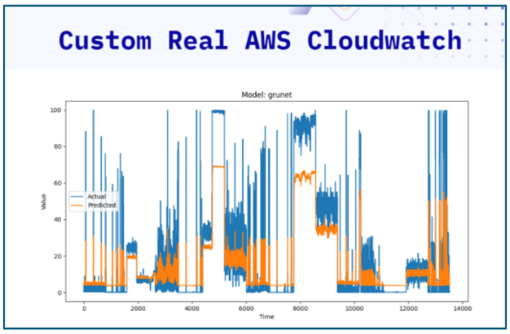
    - Real Known Cause: 7
    - Real Traffic: 7
    - Real Tweets: 10
- Sequence length: 30
- Train-test split: 80%-20%
- Rounds: 5
- Learning rate: 0.001
- Batch size: 32
- Epochs: 50

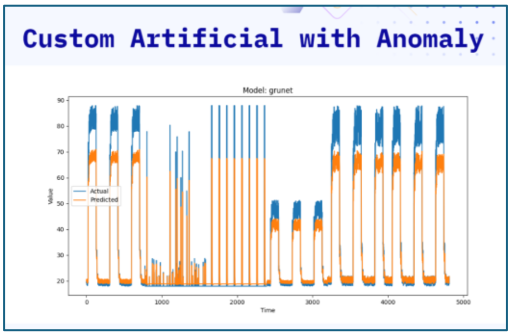
Result plots

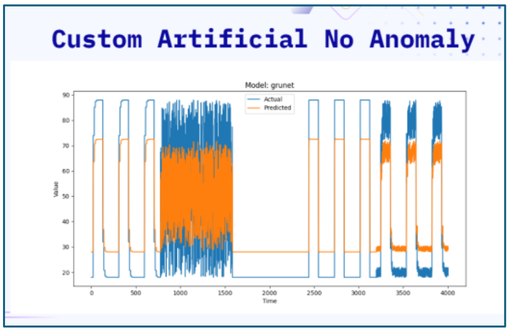


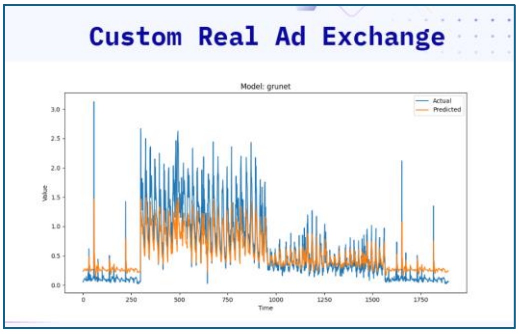












Metrics

Trainer: Federated Averaging (FedAvg)

Dataset: NAB

Dataset	MSE	MAE	R^2-score
Artificial No Anomaly	165.39	10.62	0.74
Artificial With Anomaly	80.50	4.45	0.87
Read Ad Exchange	0.36	0.41	0.60
Real AWS Cloudwatch	160.52	7.27	0.81
Real Known Cause	157.21	8.27	0.82
Real Traffic	91.38	6.75	0.89
Real Tweets	1.11	0.38	0.51

Trainer: Custom Averaging (adding small random biases to model parameters)

Dataset: NAB

Dataset	MSE	MAE	R^2-score
Artificial No Anomaly	228.48	13.05	0.73
Artificial With Anomaly	84.75	4.77	0.86
Read Ad Exchange	0.12	0.24	0.64
Real AWS Cloudwatch	162.62	7.97	0.81
Real Known Cause	159.09	9.13	0.79
Real Traffic	92.78	7.55	0.89
Real Tweets	1.12	0.39	0.50

Task 2: Video Classification on UCF-101 Action Recognition Dataset

Training Hyperparams

Model Type: ConvLSTM

We used the following hyper-params:

1. Batch size = 64
2. Rounds =2
3. Learning Rate = 2e-3
4. Epochs per client =5
5. Seq length =16
6. Frame dim = 224x224
7. LSTM dim = 512

Task 2 Evaluation

We evaluated the ConvLSTM model on UCF101 dataset in 3 setting:

- Normal training on Part -1 and testing on Part-3
- Federated Learning on Part-1 and Part-2 with FedAvg
- Federated Learning on Part-1 and Part-2 with Custom Aggregator Function

In the table shown below we report the accuracy under 3 different setting.

FedAvg	Custom Aggregator	Normal

77.8 %	67.2 %	58.2 %

---THANK YOU---