Sequence-Based Deep Learning Classification (RNN / LSTM / GRU)

1. Objective

The objective of this deliverable is to implement and evaluate sequence-based deep learning models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), or Gated Recurrent Units (GRU) for classification tasks.

This approach directly operates on raw sequential data and aims to capture temporal dependencies that cannot be effectively modeled using traditional machine learning or fully connected neural networks.

2- Dataset Description

Dataset Name

MobiFall Dataset v2.0

Dataset Source

- Downloaded from Kaggle
  Link: **https://www.kaggle.com/datasets/kmknation/mobifall-dataset-v20**

Domain

Human Activity Recognition using Motion Sensor Data

Data Characteristics

- **Type:** Time-series sensor data
- **Sampling:** Continuous sensor readings during activities
- **Selected Subject:** sub10

Type of Sensors

This dataset uses **smartphone sensors**:

- **Accelerometer**
- **Gyroscope**

Both record motion in **x, y, z axes**.

Collected Variables (Columns)

Typical file format contains:

Column	Description
timestamp	Time of recorded event
x	Acceleration along x-axis
y	Acceleration along y-axis
z	Acceleration along z-axis

Each `.txt` file contains thousands of rows → one time-series sample.

Activities (Classes Used)

6 classes are selected:

Normal Activities (ADL):

1. **Walking**
2. **Sitting**
3. **Standing**

Fall Activities:

4. **Fall Forward**

5. **Fall Backward**
6. **Fall Sideways**

This makes a **multiclass classification problem**.

Dataset Size

- **Total Samples:** 355,313 rows
- **Features (raw):** accX, accY, accZ + timestamp
- **Target Variable:** Activity label (6 classes)

3- Preprocessing Steps

Removal of Metadata:

Header information and comments were ignored, and only numerical sensor readings were retained.

Timestamp Handling:

Sensor timestamps were converted into datetime format and used to maintain chronological order.

Sorting:

Data samples were sorted based on timestamps to preserve temporal consistency.

Missing Value Handling:

Missing values were handled using forward filling to maintain continuity in time-series data.

Duplicate Removal:

Duplicate rows were removed to avoid redundancy and bias in training.

Label Encoding:

Activity labels were converted from categorical text format to numeric form using label encoding for compatibility with neural networks.

Feature Scaling:
Standardization was applied to accelerometer features to improve neural network convergence and stability.

4. Data Preparation

Raw accelerometer signals were segmented into fixed-length sequences and reshaped into 3D tensors. No handcrafted features were used.

5. Model

An LSTM network with dropout regularization and softmax output layer was trained using categorical cross-entropy loss.

6. Results

Overall Performance

- **Accuracy = 96.48%**
- **Macro F1 = 91.65%**

This means:

- The model performs very well overall
- It also performs well across all classes, not just dominant ones (Standing, Walking)

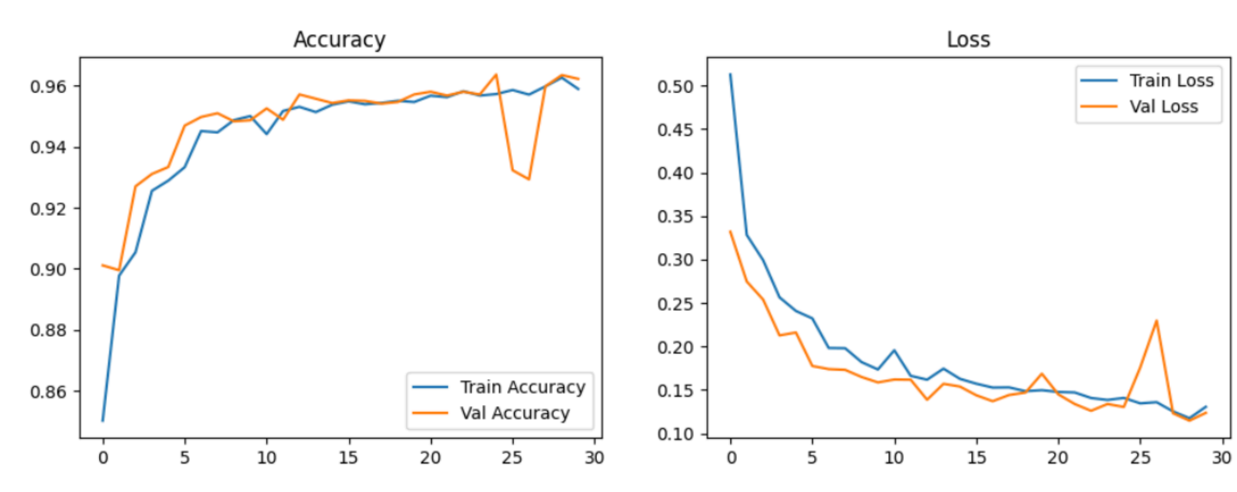
```
Accuracy: 0.9648
Macro F1: 0.9165

--- Classification Report ---
                    precision    recall  f1-score   support

Back_Sitting_Chair       1.00      0.85      0.92       291
Fall_Forward_Lying       0.93      0.81      0.87       288
    Sideward_Lying       0.97      0.84      0.90       291
           Sitting       0.96      0.76      0.85       349
          Standing       0.99      1.00      1.00      2919
           Walking       0.94      0.99      0.96      2968

          accuracy                           0.96      7106
         macro avg       0.97      0.88      0.92      7106
      weighted avg       0.97      0.96      0.96      7106
```

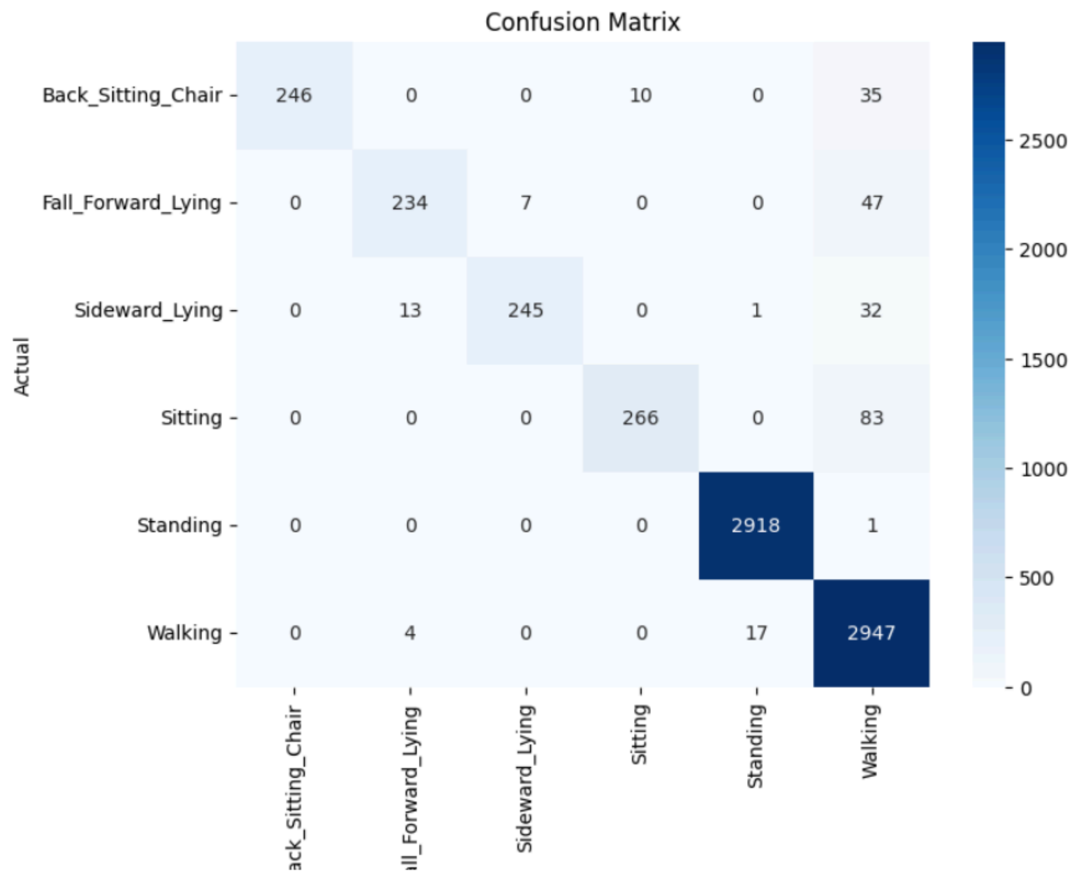
Graphical representation



Per-Class Interpretation

Activity	Interpretation
Standing	Perfect recall (1.00) → model almost never misses standing
Walking	Very high recall (0.99) → excellent movement detection
Back Sitting Chair	Recall 0.85 → some confusion with sitting
Fall Forward Lying	Recall 0.81 → hardest fall to detect
Sideward Lying	Recall 0.84 → good but some confusion

Sitting	Recall 0.76 → sometimes confused with standing or BSC



7. Conclusion

Sequence-based LSTM models perform better because they learn temporal dependencies in sensor data. Falls are dynamic events that occur over time, and LSTM captures this information better than traditional classifiers.