1- 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)

```
sub10 folders: ['ADL', 'FALLS']
Data loaded. Shape: (355313, 7)
Labels: ['Standing' 'Sitting' 'Walking' 'Back_Sitting_Chair' 'Fall_Forward_Lying'
 'Sideward_Lying']
Final Data Shape: (355313, 7)

Labels Distribution:
 label
Walking               148408
Standing              145934
Sitting                17466
Sideward_Lying         14538
Back_Sitting_Chair     14521
Fall_Forward_Lying     14446
Name: count, dtype: int64
```

Dataset Quality

- No missing sensor values after preprocessing
- Mild class imbalance (ADLs > Falls)
- Noise present due to real-world sensor readings

Feature Engineering

Because this is time-series data, we cannot directly feed raw rows to classical ML.

Approach Used

- Segment signals into fixed-size windows
- Extract statistical features per window

Extracted Features (per axis)

For each window:

- Mean
- Standard Deviation

- Minimum
- Maximum

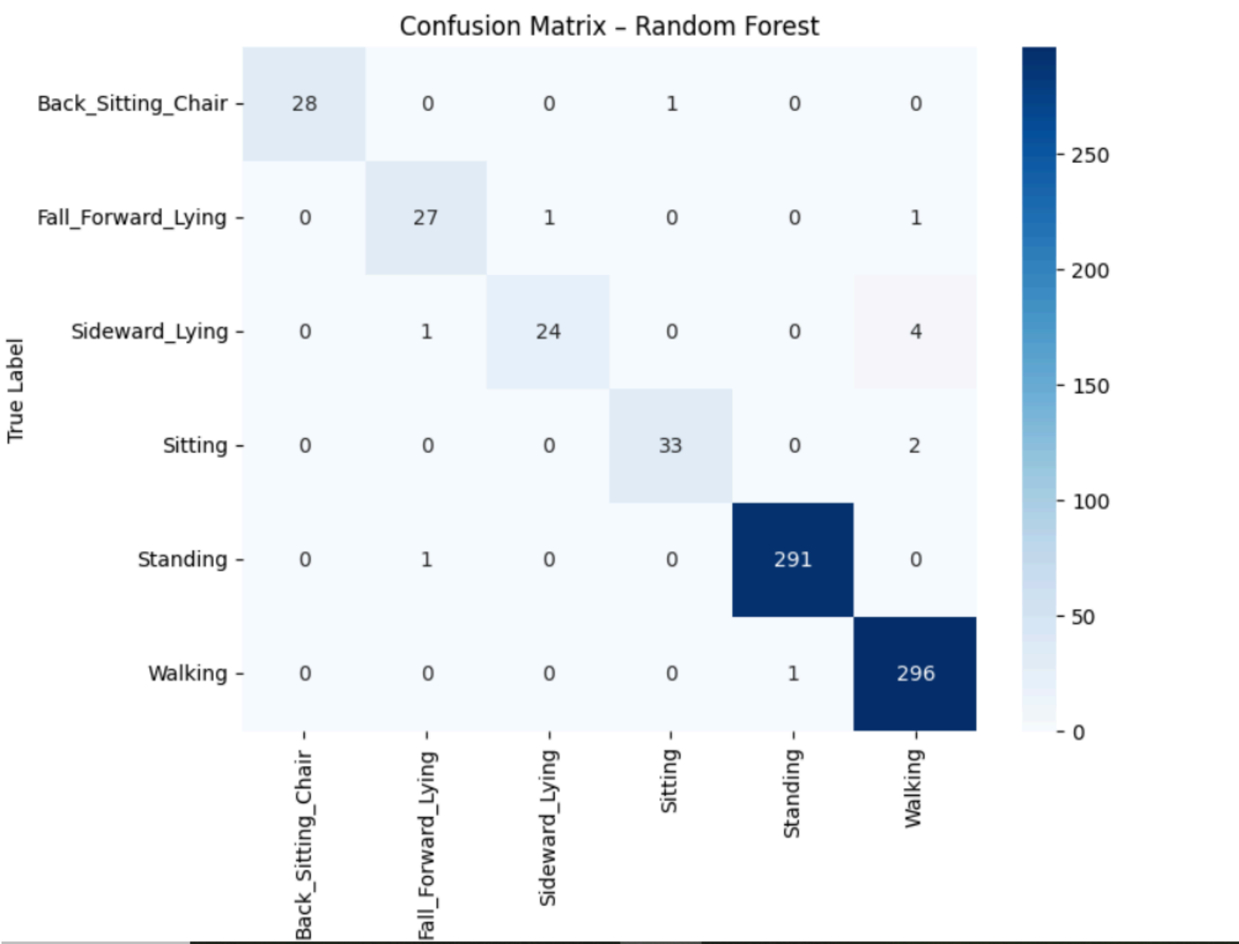
Total features = `3 axes × 4 stats = 12 features`

2- Preprocessing Steps

Steps	Why Important?
Parse Timestamps	• Timestamps allow us to understand the temporal sequence of events • Proper datetime format enables time-based operations and analysis • Essential for time series modeling and feature extraction
Chronological Sorting	• Time series analysis requires data in correct temporal order • Many algorithms assume sequential data (e.g., LSTM, sliding windows) • Prevents errors in calculating time-based features (velocity, acceleration)
Handle missing values	• Missing values can cause errors in machine learning models • Gaps in sensor data can lead to incorrect analysis • Proper imputation maintains data continuity for time series
Remove duplicates	• Duplicate rows can bias model training • Outliers/impossible values indicate sensor errors") • Clean data improves model accuracy and reliability

Rename columns	Better readability
Feature Scaling	For models sensitive to feature magnitude (such as SVM and Logistic Regression), feature scaling was applied to normalize sensor values.

Confusion Matrix (Random Forest)



Per-Class Metric

	precision	recall	f1-score	support
Back_Sitting_Chair	1.000000	0.965517	0.982456	29.000000
Fall_Forward_Lying	0.931034	0.931034	0.931034	29.000000
Sideward_Lying	0.960000	0.827586	0.888889	29.000000
Sitting	0.970588	0.942857	0.956522	35.000000
Standing	0.996575	0.996575	0.996575	292.000000
Walking	0.976898	0.996633	0.986667	297.000000
accuracy	0.983122	0.983122	0.983122	0.983122
macro avg	0.972516	0.943367	0.957024	711.000000
weighted avg	0.983051	0.983122	0.982823	711.000000

Results & Baseline Comparison Table

	Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-score (Macro)
0	Logistic Regression	0.939522	0.879962	0.820022	0.844654
1	SVM	0.956399	0.937330	0.842054	0.883921
2	Decision Tree	0.971871	0.948942	0.931462	0.938862
3	Random Forest	0.983122	0.982149	0.945338	0.962636
4	XGBoost	0.983122	0.972516	0.943367	0.957024

Conclusion:

Among all evaluated models, **Random Forest** achieved the best overall performance.

- **Highest F1-score (Macro) → 0.9626**
  Best balance between precision & recall across all 6 classes
- **Highest Precision (Macro) → 0.9821**
  Fewer false positives

- **Highest Recall (Macro) → 0.9453**
  Better detection of all activities (including falls)

- Accuracy is tied with XGBoost, **but RF is more balanced overall**

XGBoost vs Random Forest

- XGBoost has **same accuracy** (0.9831)

- But Random Forest:

  - Slightly better Recall
  - Slightly better F1-score
- For **healthcare / fall detection**, **Recall & F1-score matter more than accuracy**