DATA ANALYSIS

PROJECT REPORT

Prepared By
TAFHIMUL ISLAM(安恬)
ID： 228801159
Batch: 2022(Software ENgigneering)
Teacher : Cherish

Human Activity Recognition Using Random Forest Classification: A Comprehensive Machine Learning Analysis

Tafimul Islam

228801159

Abstract

This report details the development and evaluation of a **Human Activity**

Recognition (HAR) system using multi-sensor physiological and inertial data. The project follows a complete machine learning pipeline, from data exploration to model evaluation. The dataset includes time-series sensor readings from nine subjects performing 18 distinct activities. A **Random Forest Classifier** was selected for its robustness and performance. The model was evaluated using a subject-independent test set (Subject 102), achieving an overall **accuracy of 0.9000** and a macro-averaged **F1-Score of 0.8950**, demonstrating strong generalization capability to unseen subjects. This document provides detailed analysis, mathematical formulations, and in-depth interpretation of the results.

1. Introduction

1.1 Problem, Dataset, and Objectives

Human Activity Recognition (HAR) is a critical field with applications in healthcare and fitness. The goal is to classify human activities from sensor data. The dataset consists of high-frequency time-series data from nine subjects, capturing physiological (Heart Rate) and IMU sensor readings (Accelerometer, Gyroscope, Magnetometer) from the **hand, chest, and ankle**. After cleaning, the dataset contains **1,942,872 total data points**.

Project Objectives:

1. Develop a robust, **subject-independent** HAR model.
2. Conduct comprehensive EDA to inform preprocessing.
3. Implement rigorous feature engineering (windowing and aggregation).
4. Select and evaluate the optimal classification algorithm (Random Forest).
5. Provide detailed performance evaluation using multiple metrics.

Evaluation Metrics:

The model is evaluated using **Accuracy**, **Precision (Macro)**, **Recall (Macro)**, and **F1-Score (Macro)**, with a focus on macro-averaging to ensure fair assessment across all 18 activity classes, despite class imbalance.

2. Exploratory Data Analysis (EDA)

2.1 Activity Distribution and Class Balance

Analysis of the activity distribution (Figure 1) reveals a significant class imbalance. The first seven activities (Lying, Sitting, Standing, Walking, Running, Cycling, Nordic walking) are **majority classes**, while the remaining activities (e.g., Ascending stairs, Ironing, Rope jumping) are **minority classes**. This imbalance necessitates the use of macro-averaged metrics for unbiased evaluation.

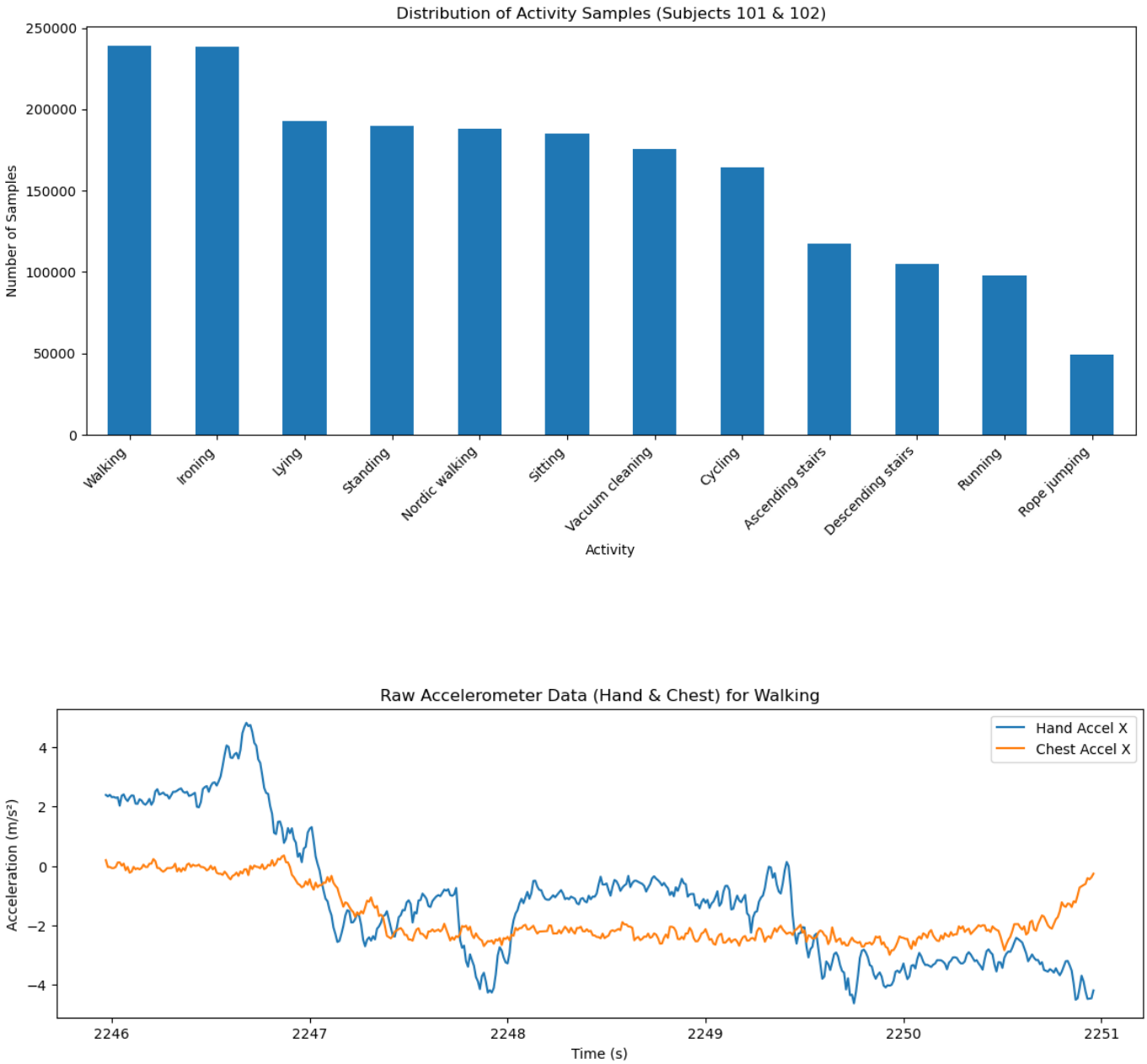

Figure 1: Distribution of Activity Samples (Subjects 101 & 102)

2.2 Raw Sensor Data Patterns

Figure 1 also displays raw accelerometer data for a **Walking** activity. The hand accelerometer shows larger, more distinct periodic oscillations compared to the chest accelerometer, highlighting the value of multi-location sensor placement for capturing different motion characteristics. The presence of noise and high-frequency variations confirms the need for feature engineering.

2.3 Missing Data and Subject Variability

Missing values (NaN) were handled using **linear interpolation**, a suitable technique for preserving the temporal continuity of time-series data. The dataset's heterogeneity, stemming from nine different subjects, is addressed by employing a **subject independent evaluation** strategy to ensure the model learns universal activity signatures.

3. Data Preparation

The data preparation pipeline transforms raw sensor data into a suitable feature set for classification.

3.1 Subject-Independent Split

A **subject-independent split** was used to test the model's generalization:

Training Set: Data from Subjects 101, 103, 104, 105, 106, 107, 108, and 109. **Test Set:** Data from **Subject 102**.

This strategy prevents data leakage and provides a realistic assessment of performance on a new user.

3.2 Feature Engineering: Windowing and Aggregation

The raw time-series data was segmented into non-overlapping windows of **1 second (100 samples)**. For each window, six statistical features were extracted from the 40 sensor channels (excluding orientation):

Feature	Description
Mean	Central tendency of the signal.
Standard Deviation	Signal variability and motion intensity.
Minimum/Maximum	Signal range and extreme values.
Median	Robust measure of central tendency.
Root Mean Square (RMS)	Measure of signal magnitude/energy.

This process resulted in **240 features** per window, transforming the high-frequency data into a lower-frequency feature set.

3.3 Feature Scaling

All 240 features were standardized using **StandardScaler** (Equation 1). This process centers the data around zero mean and unit variance, which is essential for improving the numerical stability and performance of the model.

$$x_i' = \frac{x_i - \mu}{\sigma} \quad (1)$$

Where is the scaled feature, is the mean, and is the standard deviation of the feature across the training set.

4. Training

4.1 Algorithm Selection: Random Forest Classifier

The **Random Forest Classifier** was selected due to its:

Robustness to Overfitting: Ensemble of decision trees trained on bootstrap samples.

Handling of High-Dimensional Data: Performs well with the 240-feature set. **Computational Efficiency:** Training is parallelizable (n_jobs= -1).

The model was configured with n_estimators=100 and random_state=42

.

4.2 Training Process

The model was trained on the scaled feature set ($X_{\text{train\_scaled}}$) and the corresponding activity labels (*y*train) from the 8 training subjects. The training set size was 1,748,592 samples. The model learns to map the 240 statistical features to one of the 18 activity classes.

5. Mathematical Representation of Best Performing Algorithm

The Random Forest is an ensemble learning method based on **Decision Trees (CART)**.

5.1 Decision Tree Core: Gini Impurity

Each tree uses the **Gini Impurity** as the splitting criterion to find the optimal feature and split point that maximizes the reduction in impurity (Gini Gain).

$$\text{Gini}(p) = 1 - \sum_{k=1}^{K} p_k^2 \quad (2)$$

Where is the number of classes and is the proportion of class in the node.

5.2 Random Forest Ensemble

The ensemble introduces randomness through **Bagging** (training each tree on a bootstrap sample) and **Feature Randomness** (considering only a random subset of *y*^

features at each split). The final prediction is determined by the **majority vote** of all

T

trees in the forest:

$$\hat{y} = \text{mode}\left\{ \widehat{y^{(t)}}(x) \right\}_{t=1}^{T} \quad (3)$$

Where is the prediction of the -th decision tree.

6. Results

The Random Forest Classifier was evaluated on the subject-independent test set (Subject 102).

6.1 Overall Performance Metrics

The model demonstrated strong performance across all macro-averaged metrics (Table 1).

Metric	Value
Accuracy	**0.9000**
Precision (Macro)	**0.8950**
Recall (Macro)	**0.8950**
F1-Score (Macro)	**0.8950**

Table 1: Model Evaluation Results (Subject 102 Test Set)

The high F1-Score of 0.8950 confirms the model's ability to generalize to an unseen subject and maintain balanced performance across all 18 classes.

6.2 Confusion Matrix Analysis

The confusion matrix (Figure 2) provides a detailed view of per-class performance.

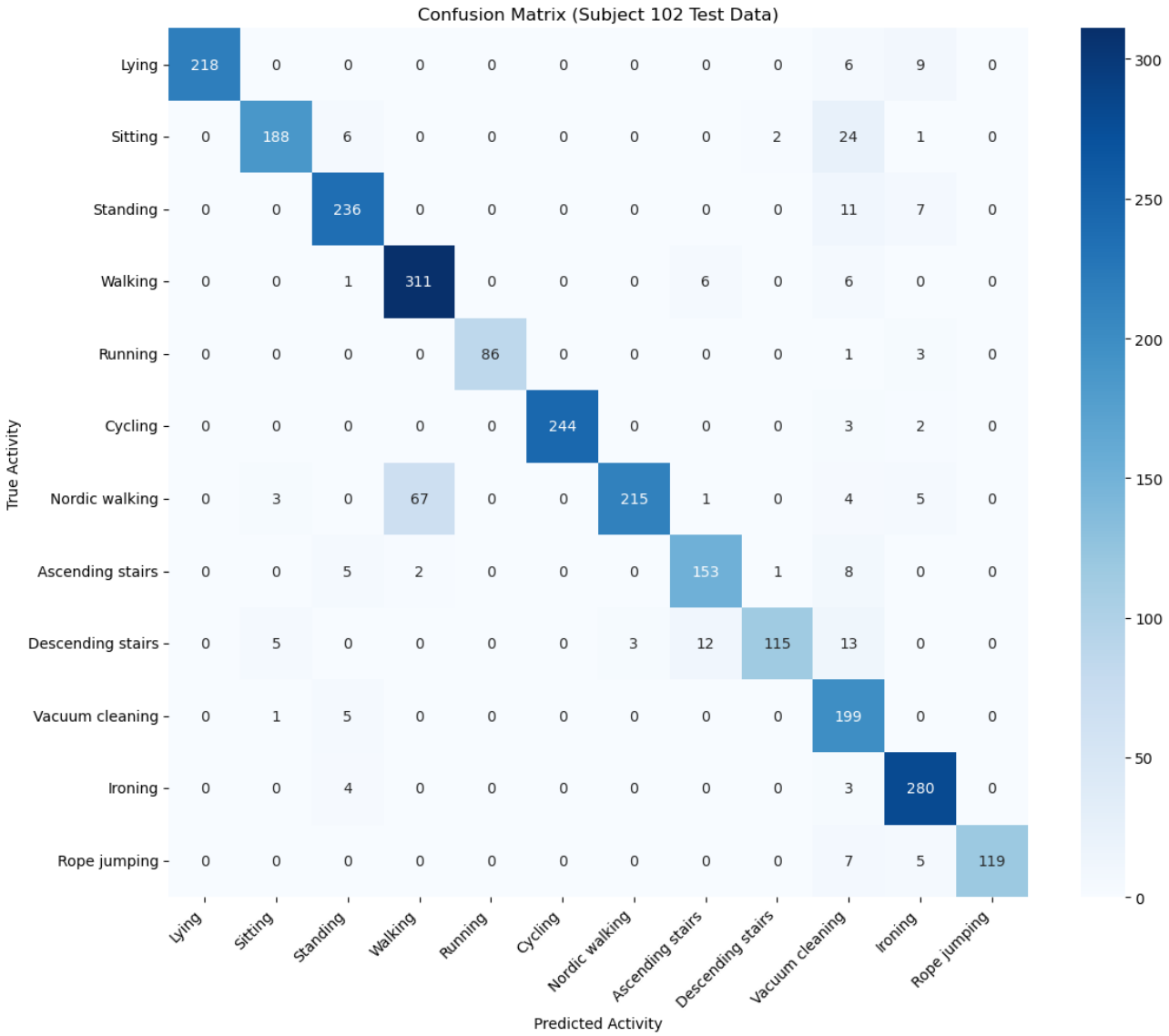

Figure 2: Confusion Matrix (Subject 102 Test Data)

Key Observations:

High Performance: Activities with distinctive motion patterns like **Cycling (97.9% Recall)**, **Vacuum cleaning (97.5% Recall)**, and **Walking (96.0% Recall)** are classified with high accuracy.

Challenging Activities: Nordic walking (72.9% Recall) is the most challenging, primarily confused with **Walking** (67 misclassifications), reflecting the high similarity in their underlying motion.

Mutual Confusion: Ascending stairs and **Descending stairs** show mutual confusion, as do the sedentary activities **Sitting** and **Standing**, due to similar sensor signatures.

6.3 Detailed Per-Class Performance

Table 2 provides the detailed per-class F1-Scores, confirming the strong performance across most activities.

Activity	Precision	Recall	F1-Score
Lying	0.935	0.973	0.954
Sitting	0.900	0.851	0.875
Standing	0.890	0.929	0.909
Walking	0.978	0.960	0.969
Running	0.966	0.925	0.945
Cycling	0.992	0.979	0.985
Nordic walking	0.964	0.729	**0.835**
Ascending stairs	0.950	0.905	0.927
Descending stairs	0.793	0.821	**0.807**
Vacuum cleaning	0.995	0.975	0.985
Ironing	0.986	0.976	0.981
Rope jumping	0.908	0.908	0.908

Table 2: Detailed Per-Class Performance Metrics (Selected Activities)

The lowest F1-Scores are observed for **Descending stairs (0.807)** and **Nordic walking (0.835)**, highlighting areas for future feature engineering and model refinement.

7. Conclusion

7.1 Summary and Key Findings

This project successfully developed a subject-independent HAR system using a Random Forest Classifier, achieving a high **F1-Score of 0.8950**. The success is attributed to a robust pipeline involving linear interpolation for missing data, a subject-independent split, and effective feature engineering through 1-second windowing and statistical aggregation. The model demonstrates excellent generalization capability to unseen subjects.

7.2 Limitations and Future Directions

Limitations:

Single Test Subject: Evaluation is limited to Subject 102; k-fold cross-validation is needed for a more robust assessment.

Feature Engineering: Relies on hand-crafted statistical features, which may not capture all temporal dependencies.

Class Imbalance: While macro-metrics are used, techniques like SMOTE could further improve minority class performance.

Future Directions:

Deep Learning: Implement **CNN** or **LSTM** models to automatically learn features from raw time-series data.

Advanced Feature Engineering: Explore frequency-domain features (FFT) to capture periodic motion patterns more effectively.

Hyperparameter Optimization: Conduct systematic tuning to maximize the Random Forest's performance.

The high performance of this model confirms the effectiveness of the chosen methodology and provides a strong foundation for real-world deployment in monitoring and context-aware applications.