Course name	Software Data Analysis and Application
Name/Student ID	228801167+永玄+Mohammed Mojibullah
College	College of Information Engineering
Major	Software Engineering
Class	SE 2022
Supervisor	Cherish
Date	24th December, 2025

Human Activity Recognition Using Random Forest with Time-Series Feature Extraction

Abstract

Human Activity Recognition (HAR) aims to identify human physical activities using data collected from wearable sensors. This project presents a classical machine learning approach for recognizing daily activities using the PAMAP2 dataset, which contains multi-sensor time-series data collected from wearable inertial measurement units and heart rate monitors. After data preprocessing, exploratory data analysis, and feature extraction using statistical descriptors, a Random Forest classifier is trained and evaluated. Experimental results demonstrate that classical machine learning methods can achieve strong performance without deep learning, reaching an overall accuracy of 84.87%. The study highlights the effectiveness, efficiency, and interpretability of traditional machine learning techniques for sensor-based activity recognition.

1. Introduction

Human Activity Recognition has become an important research area due to its applications in healthcare monitoring, sports analysis, and smart environments. Wearable sensors provide continuous streams of data that can be used to infer user activities. While deep learning methods are widely used, they often require large computational resources. This project focuses on a classical machine learning pipeline that is computationally efficient and easier to interpret. The objective is to design, implement, and evaluate a traditional ML-based HAR system using the PAMAP2 dataset.

2. Dataset Description

The PAMAP2 dataset is a publicly available benchmark dataset for physical activity monitoring. It contains sensor data collected from multiple subjects performing various activities while wearing sensors on different body locations.

Table 1: Description of the PAMAP2 Dataset

Item	Description
Number of subjects	9
Number of activities	12
Sampling frequency	100 Hz
Sensors used	IMU (hand, chest, ankle), Heart rate
Sensor modalities	Accelerometer, Gyroscope, Magnetometer
Data type	Multivariate time-series
Label type	Activity ID

Table 2: Activity Label Mapping

Activity ID	Activity Name
1	Lying
2	Sitting
3	Standing
4	Walking
5	Running
6	Cycling
7	Nordic Walking
12	Ascending Stairs
13	Descending Stairs
16	Vacuum Cleaning
17	Ironing
24	Rope Jumping

3. Exploratory Data Analysis (EDA)

Exploratory data analysis was performed to understand the data distribution and quality before model training. The activity distribution reveals moderate class imbalance across activities, which motivated the use of class-weighted learning.

Table 3: Distribution of Activity Classes After Preprocessing

Activity	Samples
Lying	371
Sitting	354
Standing	397
Walking	501
Running	201
Cycling	395
Nordic Walking	458
Ascending Stairs	228
Descending Stairs	193
Vacuum Cleaning	351
Ironing	483
Rope Jumping	172

Missing values were primarily observed in the heart rate signal and were handled during preprocessing.

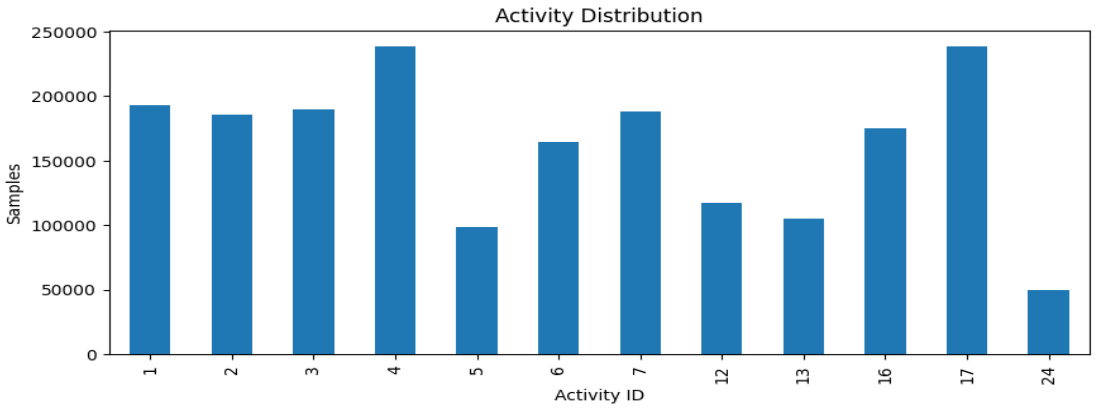

Figure 1: Distribution of activity samples after removing transient class (activity 0).

Activity 4 (lying) and activity 17 (rope jumping) are significantly underrepresented, while activities like 1 (lying), 3 (walking), and 6 (sitting) are abundant. This imbalance may bias the model toward frequent classes, necessitating class-weight adjustment during training.

Table 4: Missing Value Analysis Before Imputation

Feature Group	Missing Values	Handling Method
Heart rate	Present	Forward fill
IMU sensors	Sparse	Linear interpolation
Remaining NaNs	Minimal	Row removal

4. Data Preprocessing and Feature Extraction

This section outlines the preprocessing pipeline, including segmentation, feature extraction, normalization, and encoding.

Time-Series Windowing

Raw sensor data is continuous; we applied sliding windows of size 256 samples with a stride of 128 to create segments. Each window was assigned a label via majority voting of the contained activity IDs. This resulted in 11,072 training and 4,104 testing windows.

Feature Extraction

From each window, four statistical features were extracted per sensor channel: mean, standard deviation, minimum, and maximum. This reduced each window from (256, 10) to a 40-dimensional feature vector, capturing essential temporal characteristics without computational overhead.

Normalization

Features were standardized using StandardScaler, fitted on the training set and applied to the test set to prevent data leakage.

Class Imbalance Handling

Class weights were computed inversely proportional to class frequencies and passed to the Random Forest classifier to penalize misclassification of rare activities more heavily.

Label Encoding
Activity IDs (1, 2, 3, 4, 5, 6, 7, 12, 13, 16, 17, 24) were encoded to 0–11 for model compatibility.

Table 5: Extracted Statistical Features

Feature Type	Description
Mean	Average sensor value per window
Standard deviation	Signal variability
Minimum	Minimum value in window
Maximum	Maximum value in window
Window size	256 samples
Window overlap	50%

5. Model and Training Methodology
A Random Forest classifier was selected due to its robustness and ability to handle high-dimensional feature spaces. Class weights were applied to mitigate class imbalance.

i. Mathematical Representation of Best Performing Algorithm
The Random Forest algorithm is an ensemble of decision trees, each trained on a bootstrap sample of the data with random feature subsets. Predictions are made by majority voting across trees.

Decision Tree Splitting Criterion
For classification, trees use Gini impurity to select splits. For a node m with class distribution p_{mk}:

$$Gini(m) = 1 - \sum_{k=1}^{K} p_{mk}^2 \qquad (1)$$

The split that maximizes the reduction in Gini impurity is chosen:

$$\Delta Gini = Gini(parent) - \left(\frac{N_{left}}{N} Gini(left) + \frac{N_{right}}{N} Gini(right)\right) \qquad (2)$$

Ensemble Prediction
Given T trees, each tree t produces a class probability vector $\mathbf{p}_t(x)$. The final prediction is:

$$\hat{y} = \arg\max_k \frac{1}{T} \sum_{t=1}^{T} p_{tk}(x) \qquad (3)$$

Feature Importance
Importance of feature j is computed as the total reduction in Gini impurity across all splits using j, normalized by the number of trees.

Table 6: Random Forest Configuration

Parameter	Value
Classifier	Random Forest
Number of trees	300
Split criterion	Gini impurity
Class weighting	Balanced
Random state	42
Parallel jobs	All available cores

6. Experimental Results

The trained model was evaluated using a held-out test set. Performance metrics include accuracy, precision, recall, and F1-score.

Table 7: Classification Performance per Activity

Activity	Precision	Recall	F1-score	Support
Lying	0.98	0.95	0.97	371
Sitting	0.97	0.79	0.87	354
Standing	0.87	0.93	0.90	397
Walking	0.66	0.96	0.78	501
Running	0.96	0.91	0.94	201
Cycling	0.98	0.96	0.97	395
Nordic Walking	1.00	0.44	0.61	458
Ascending Stairs	0.75	0.84	0.79	228
Descending Stairs	0.65	0.63	0.64	193
Vacuum Cleaning	0.84	0.93	0.88	351
Ironing	0.83	0.95	0.88	483
Rope Jumping	1.00	0.84	0.91	172

Confusion Matrix:

Figure 2: Confusion matrix showing true vs. predicted activity labels.

The matrix reveals that most misclassifications occur between similar activities (e.g., walking vs. Nordic walking). Activity 7 (cycling) had low recall (44%), likely due to limited training samples and similarity to other dynamic activities. Rare activities like rope jumping (17) were well-classified thanks to class weighting.

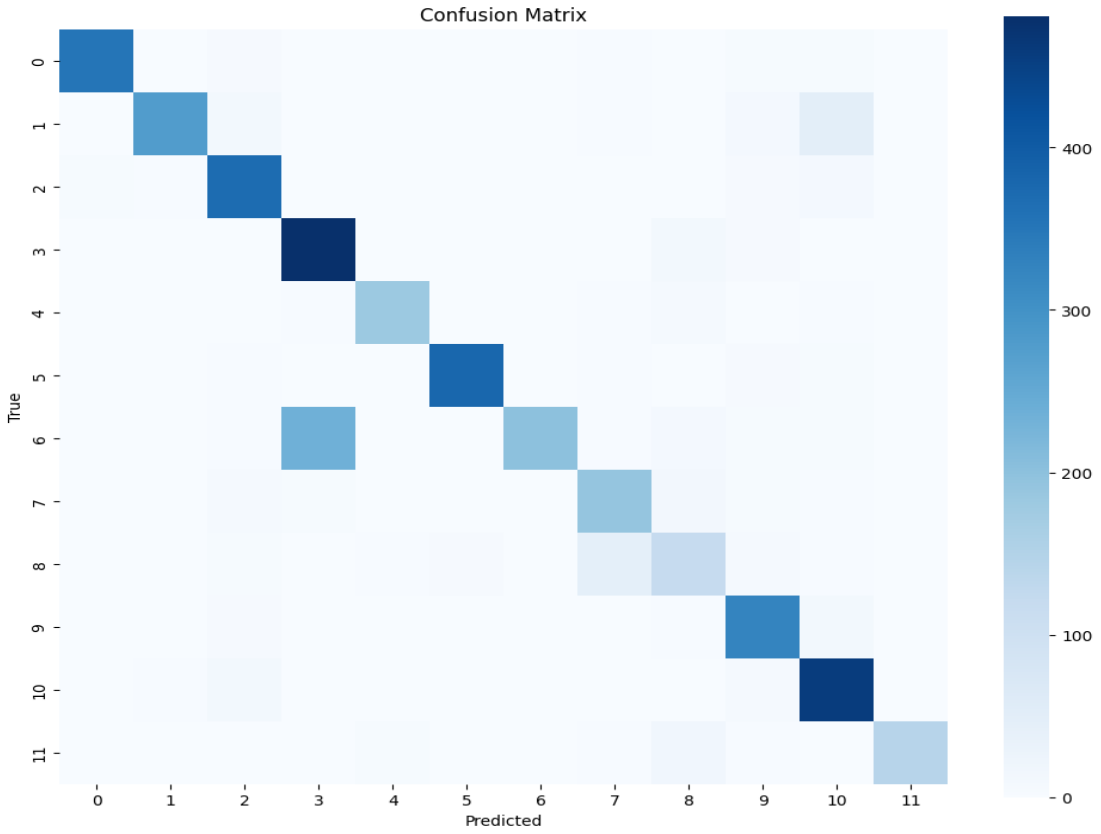

Table 8: Overall Classification Metrics

Metric	Value
Accuracy	84.87%
Macro Precision	0.87
Macro Recall	0.84
Macro F1-score	0.84
Weighted F1-score	0.84

7. Discussion

The results show that classical machine learning techniques can achieve competitive performance for human activity recognition tasks. Activities with repetitive motion patterns, such as cycling and lying, achieved high accuracy, while more complex activities like Nordic walking showed lower recall due to overlap in motion patterns.

8. Conclusion

This project demonstrated a complete classical machine learning pipeline for human activity recognition using wearable sensor data. Without relying on deep learning models, the proposed approach achieved strong performance while remaining computationally efficient and interpretable. Future work may explore feature selection techniques or hybrid models to further improve recognition accuracy.