Human Activity Recognition from Wearable Sensor Data:
A Comparative Study of Random Forest, SVM, LightGBM, and XGBoost

Name: Hasib Md Ashikur Rahman 哈思
Student id: 228801141

1. Introduction

Human Activity Recognition (HAR) involves automatically classifying physical activities performed by individuals using data collected from sensors worn on the body. HAR has applications in health monitoring, sports analytics, rehabilitation, and context-aware computing. Recent advances leverage data from wearable devices such as accelerometers, gyroscopes, magnetometers, and heart rate monitors to improve accuracy.

The dataset used in this project consists of time-series sensor data collected from multiple body locations: hand, chest, and ankle. Data was collected across several subjects performing various physical activities. Each record contains over fifty sensor-derived features, including acceleration along multiple axes, gyroscope and magnetometer readings, orientation parameters, temperature, and heart rate measurements. The raw sensor data was preprocessed and organized into CSV files per subject, making comprehensive exploration and analysis possible. The dataset contains challenges such as missing values, especially for heart rate, subject variability, class imbalance, and typical sensor noise.

The main objectives of this project are:

- To develop a robust HAR system using subject-wise generalization, which ensures that the trained model can recognize activities for previously unseen individuals.
- To systematically evaluate and compare the performance of different classification algorithms, specifically Random Forest, Support Vector Machine (SVM), LightGBM, and XGBoost, on this sensor data.
- To use detailed exploratory and visual analysis to guide all preprocessing and modeling decisions.
- To report model performance using overall accuracy, per-class accuracy, confusion matrices, and error analysis.

The report is structured as follows:

- Section 2 presents a chart-driven exploratory analysis of the dataset.
- Section 3 details the data preparation and preprocessing pipeline with supporting before-and-after visualizations.
- Section 4 describes the model training process, hyperparameter tuning, and comparative results.
- Section 5 presents the mathematical formulation of the best-performing algorithm.
- Section 6 contains result visualizations, evaluations, and analysis of model behavior.
- Section 7 concludes the report with summary findings, identified limitations, and recommendations for future work.

2. Exploratory Data Analysis (EDA)

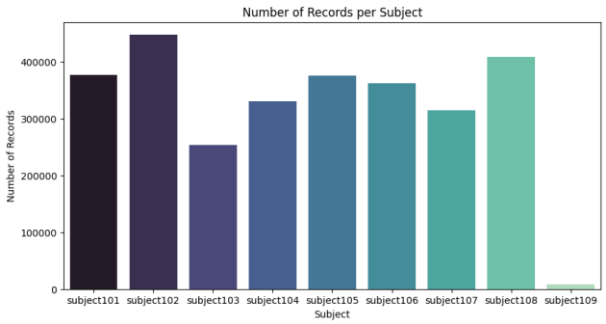
EDA was conducted to quickly assess the data's structure, quality, and key patterns before processing or modeling. This included checking distributions, missing values, correlations, and sensor signal characteristics.

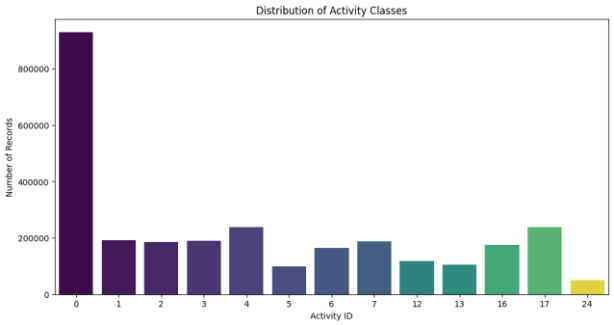
2.1 Dataset Overview

The dataset contains 2,872,533 time-series records and 55 features from wearable sensors on the hand, chest, and ankle, including accelerometer, gyroscope, and heart rate data, with subject and activity labels. Most missing values occur in the heart rate column (over 2.6 million missing), while other features are largely complete.

2.2 Distribution of Records by Subject and Activity

To understand data balance, the number of records per subject and per activity was analyzed and visualized using bar charts.



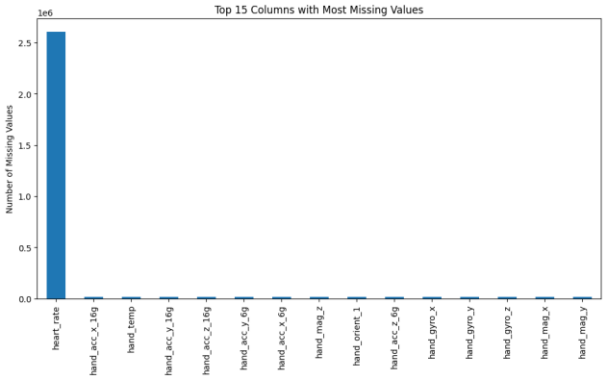


The analysis revealed:

- Unequal representation of subjects, with some participants contributing more data than others.
- A noticeable imbalance among activities, where dynamic activities (such as walking or running) had significantly more samples compared to static or transitional activities.

2.3 Missing Value Analysis

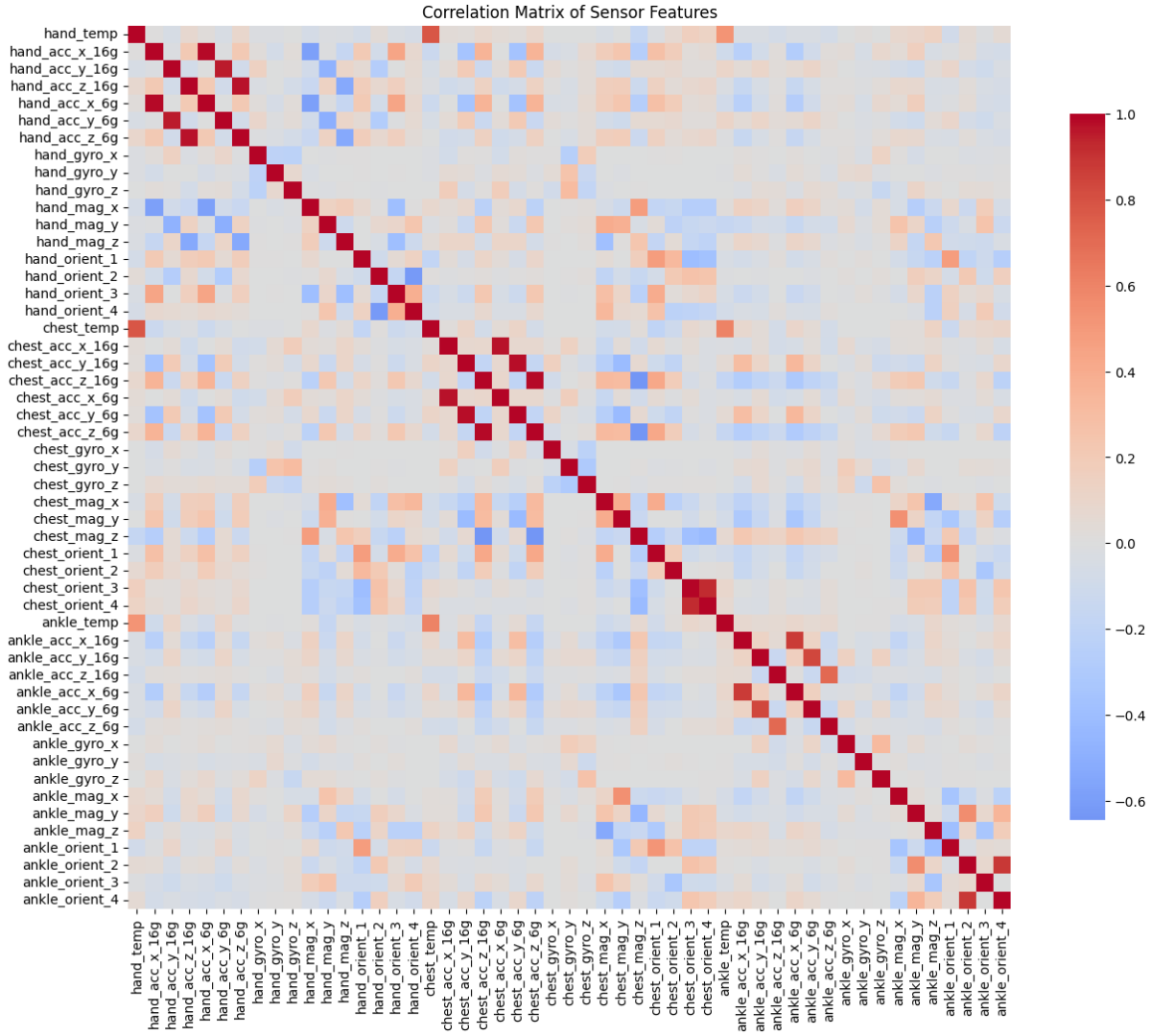
A detailed missing value analysis was conducted across all features. Bar plots were used to visualize the count and proportion of missing values per column.



The bar plot of missing values reveals that the vast majority of missing data is concentrated in the `heart_rate` column, with over 2.6 million missing entries. All other columns have a negligible number of missing values in comparison, making their bars almost invisible on the same scale. This indicates that missing data handling will primarily focus on the `heart_rate` feature, while the rest of the dataset is largely complete. This pattern is expected, as heart rate is typically sampled at a much lower frequency than IMU sensor data.

2.4 Correlation Analysis of Sensor Features

A correlation matrix was computed for numeric sensor features and visualized using a heatmap. This helped in understanding the linear relationships between different sensor signals.

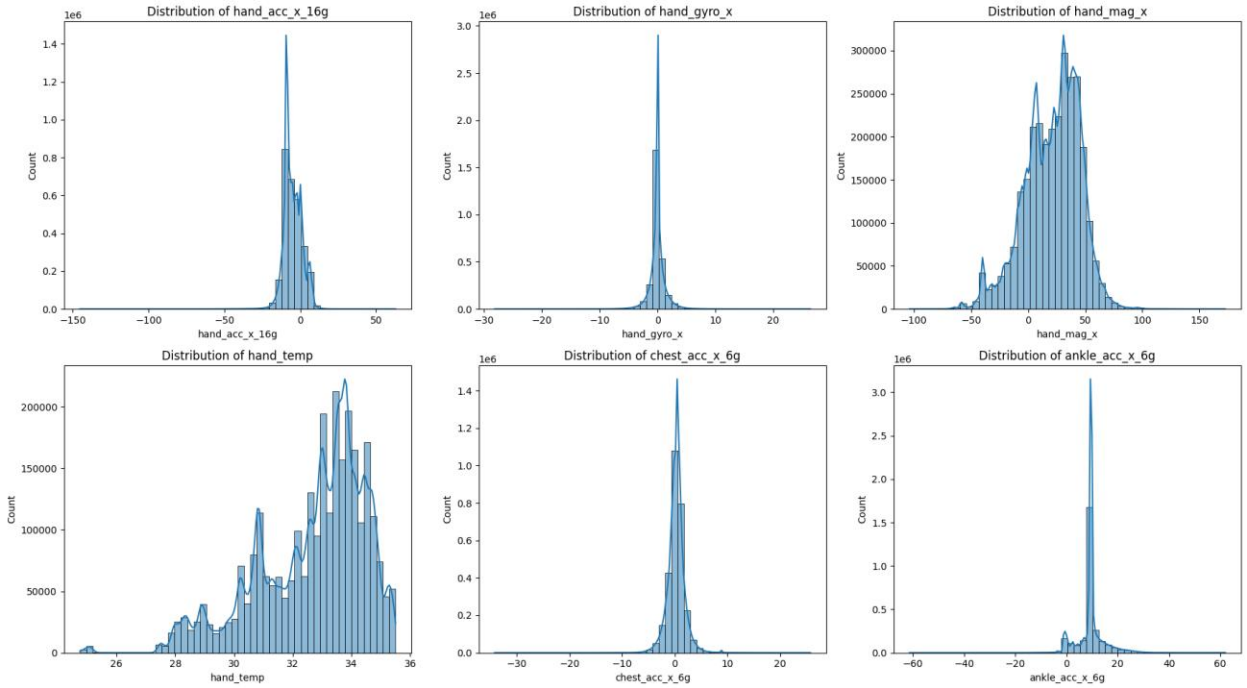


The correlation analysis showed:

- Strong correlations between accelerometer axes from the same sensor location.
- Moderate correlations across different body locations for similar movements.
- Weak correlations between heart rate and raw accelerometer features, suggesting heart rate captures complementary physiological information rather than redundant motion data.

2.5 Distribution of Sensor Features and Outlier Detection

Histograms were generated for key sensor features to examine their distributions and identify potential outliers.



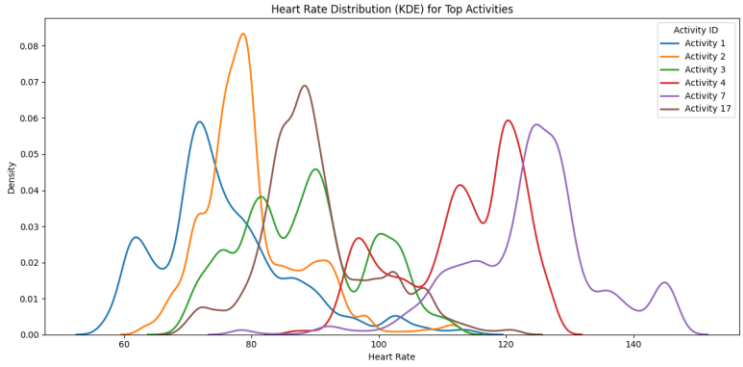
Observations include:

- Accelerometer and gyroscope signals generally followed symmetric or slightly skewed distributions.
- Certain extreme values were observed, especially during high-intensity activities.
- No abnormal spikes suggesting sensor malfunction were detected.

This analysis confirmed that the sensor data is realistic and activity-dependent, and that standard normalization techniques would be appropriate in later stages.

2.6 Heart Rate Distribution Across Activities

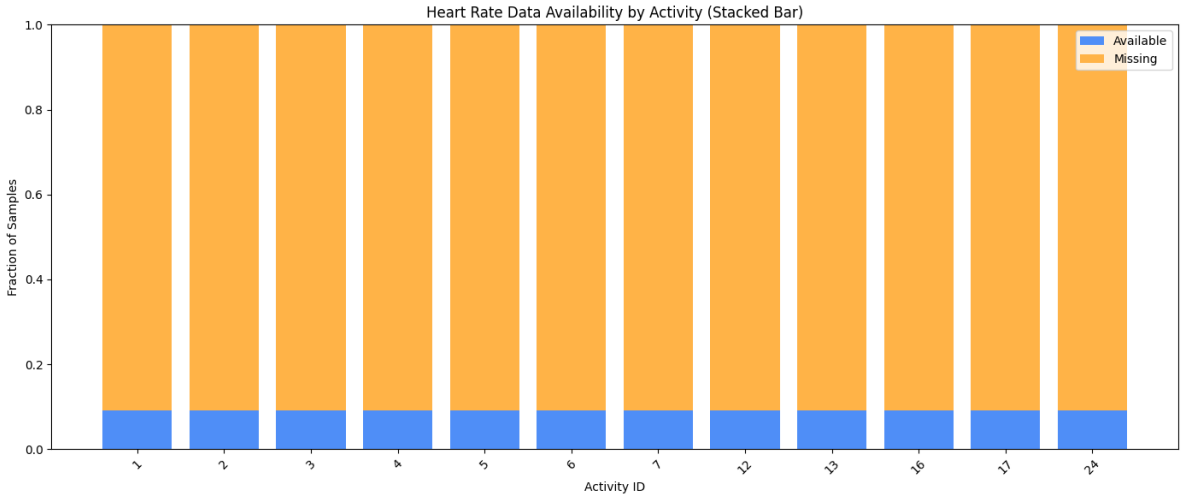
To examine how heart rate varies with different activities, Kernel Density Estimation (KDE) plots were created for the most frequent activities.



The plots showed that high-intensity activities generally correspond to higher heart rate values, while low-intensity activities have lower heart rate distributions. Some overlap between activities was observed, indicating that heart rate alone is not sufficient for accurate classification but can enhance performance when combined with motion sensors.

2.7 Heart Rate Availability by Subject and Activity

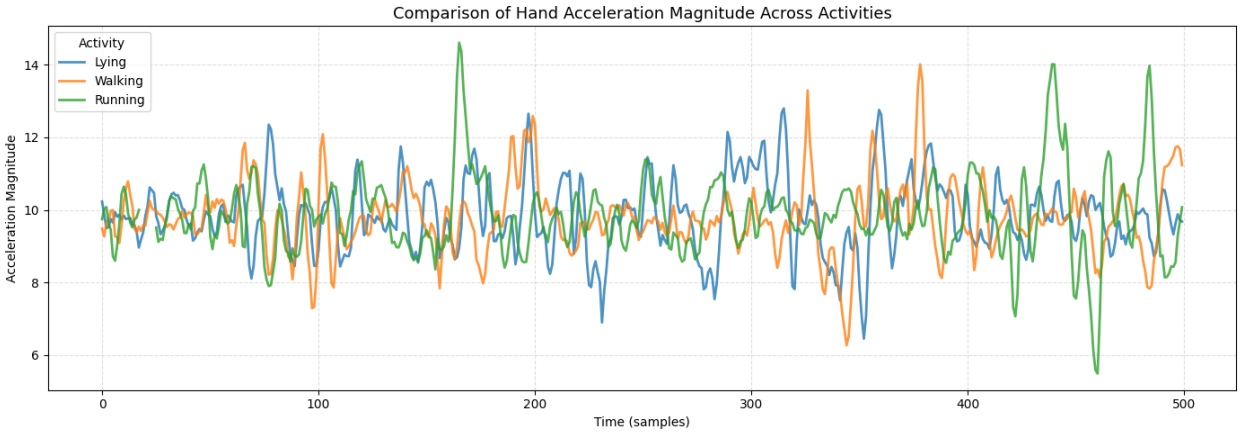
The availability of heart rate data was analyzed by calculating the fraction of recorded heart rate values for each subject and each activity. Bar charts were used to visualize these fractions.

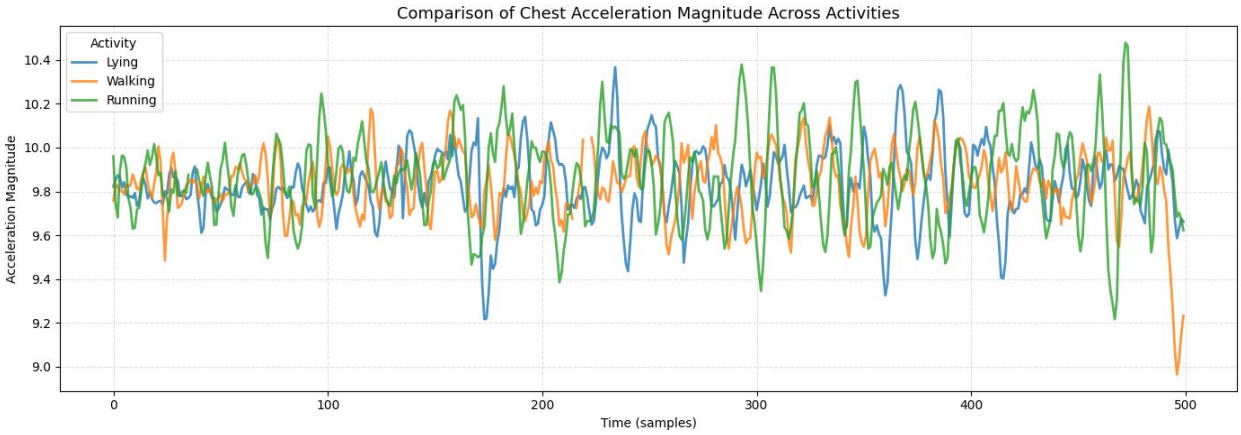


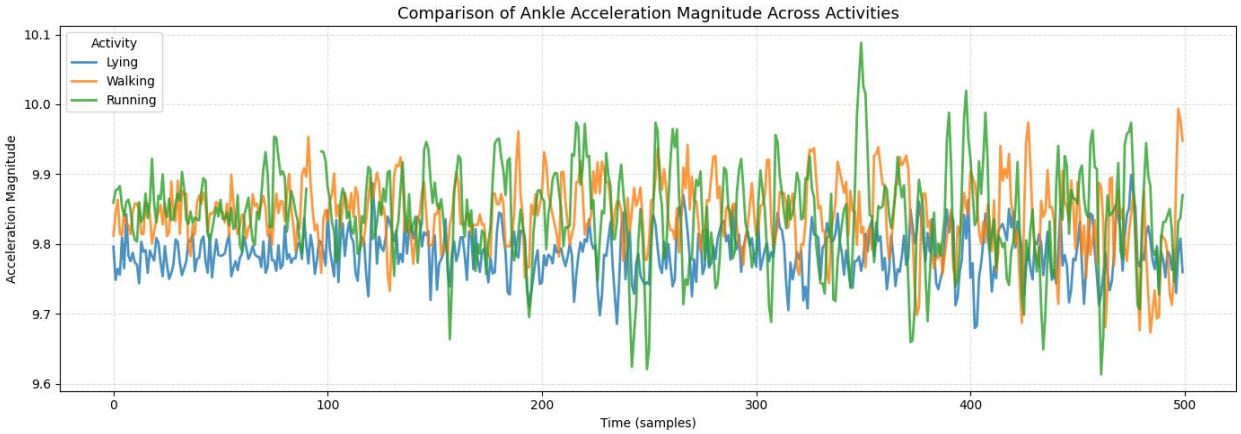
The results showed that some subjects have significantly lower heart rate availability than others. Similarly, certain activities are associated with a higher proportion of missing heart rate values. This suggests systematic missingness rather than random data loss.

2.8 Accelerometer Magnitude Analysis Across Body Locations

To better understand movement intensity, the magnitude of accelerometer signals was analyzed for sensors placed on the **hand, chest, and ankle** across different activities.







The analysis showed that:

- High-intensity activities produce larger accelerometer magnitudes.
- Ankle sensors capture strong signals during walking and running activities.
- Hand sensors show greater variability due to arm movements.
- Chest sensors provide more stable readings, useful for posture-related activities.

3. Data Preparation

The data preparation process transformed the raw wearable sensor dataset into a clean, informative feature set tailored for robust human activity recognition.

3.1 Data Loading and Initial Cleaning

All subject-specific CSV files generated from the raw .dat recordings were loaded and concatenated into a single DataFrame. This unified structure enabled comprehensive exploration and consistent processing for subsequent analysis.

3.2 Removal of Unlabeled and Transient Data

Records labeled with activity_id = 0 (unlabeled or transition segments) were removed, ensuring that only data with defined activity classes contributed to modeling.

3.3 Handling Missing Data

A significant proportion of heart rate measurements were missing, while most other sensors were nearly complete. The following approach was applied:

- For heart rate, missing values were imputed within each subject using that subject's median heart rate (if any). If a subject had all values missing, the global median across all subjects was used.
- Remaining rows with missing values in any other feature were dropped, preserving only fully complete records for subsequent feature extraction.

3.4 Feature Engineering

To increase informativeness and reduce dimensionality:

- The top 30 most important original features were selected using feature importance scores from a Random Forest (fit only on the training subjects to prevent information leakage).
- The cleaned time-series data was segmented into fixed-size windows (no crossing activity boundaries) to capture temporal context. For each window, statistical aggregate features including mean, standard deviation, minimum, and maximum were extracted for every selected signal.
- Each window was labeled by the most frequent activity within that segment.

```
Building window features...
- window_size=200, step_size=100
- num_features=30
Window dataset shape: (19052, 123)
Example columns: ['subject', 'activity_id',
Building window features...
- window_size=200, step_size=100
- num_features=30
Window dataset shape: (19052, 123)
Example columns: ['subject', 'activity_id',
```

3.5 Subject-wise Split

The windowed dataset was strictly divided by subject, simulating real-world model deployment on unseen individuals. The splits used were:

- **Training subjects:** subject101, subject102, subject103, subject105, subject106, subject108, subject109
- **Validation subject:** subject104
- **Test subject:** subject107

This split ensures a rigorous subject-independent evaluation, where no samples from the validation or test subjects are seen during training or feature selection.

Final Dataset Composition

After cleaning, imputation, feature extraction, and splitting, the final datasets were ready for modeling. The feature matrix for each window consisted of the selected statistical summaries for each channel, and each window was labeled for supervised activity classification.

4. Training

This section describes the machine learning algorithms explored, baseline and optimized hyperparameter setups, and configuration strategies, all matching the workflow documented in the notebook.

Three models were developed and hyperparameter tuned for human activity recognition:

- **Support Vector Machine (SVM, scaled)**
- **LightGBM**
- **XGBoost**

Baseline Models:

Each model was first trained with a robust baseline configuration:

- **SVM (scaled):** StandardScaler + SVC(kernel='rbf', C=10, gamma='scale', random_state=42)
- **LightGBM:** n_estimators=1500, learning_rate=0.05, num_leaves=63, subsample=0.8, colsample_bytree=0.8, random_state=42, force_col_wise=True
- **XGBoost:** n_estimators=1200, learning_rate=0.05, max_depth=6, subsample=0.8, colsample_bytree=0.8, reg_lambda=1.0, min_child_weight=1, num_class=n_classes, objective='multi:softprob', eval_metric='mlogloss', tree_method='hist', random_state=42

Hyperparameter Tuning:

A stratified sample of the training data was used for efficient tuning. Each model's parameters were optimized with grid or random search and chosen based on performance on the validation subject (subject104). The tuning grid and final best parameters are below:

Model	Parameter Tuned Range / Grid	Best Config Selection Method	Best Parameters Found
SVM (scaled)	C: [0.3, 1, 3, 10, 30]; gamma: ['scale', 0.001, 0.01, 0.1]; kernel: 'rbf' (random sample of grid)	Validation accuracy and F1-score	kernel='rbf', **C=0.3**, **gamma='scale'**
LightGBM	num_leaves: [31, 63, 127]; learning_rate: [0.03, 0.05, 0.1]; n_estimators: [800, 1500, 2500]; min_child_samples: [20, 50, 100]; subsample: [0.7, 0.8, 0.9]; colsample_bytree: [0.7, 0.8, 0.9]	Validation accuracy and F1-score	**colsample_bytree=0.8, learning_rate=0.03, min_child_samples=100, n_estimators=2500, num_leaves=31, subsample=0.7**

XGBoost	n_estimators: [800, 1200, 2000]; learning_rate: [0.03, 0.05, 0.1]; max_depth: [4, 6, 8]; subsample: [0.7, 0.8, 0.9]; colsample_bytree: [0.7, 0.8, 0.9]; reg_lambda: [1.0, 2.0, 5.0]; min_child_weight: [1, 5, 10]	Validation accuracy and F1-score	**colsample_bytree=0.8, learning_rate=0.05, max_depth=4, min_child_weight=10, n_estimators=800, reg_lambda=2.0, subsample=0.8**

Table 1: Final Model Hyperparameter Tuning and Selection

Training Approach

- Windowed statistical features were extracted per activity, with feature selection based on training data only.
- Models were trained on all training subjects, validated on subject104, and reserved for test on subject107.
- Consistent scaling and sampling were applied throughout training and tuning.

5. Mathematical Representation of Best Performing Algorithm

The best performing algorithm in this project is the **Support Vector Machine (SVM)** with a **Radial Basis Function (RBF) kernel**. SVM aims to find an optimal decision boundary that maximizes the margin between activity classes using sensor-based feature vectors.

SVM Decision Function

Given a training dataset

$$\{(x_i, y_i)\}_{i=1}^{N},$$

where $x_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \{-1, +1\}$ is the class label, the SVM decision function is defined as:

$$f(x) = \sum_{i=1}^{N} \alpha_i \, y_i K(x, x_i) + b \qquad (1)$$

where:

- x is the input sample to be classified
- x_i are the support vectors
- α_i are learned Lagrange multipliers
- y_i are class labels
- $K(x, x_i)$ is the kernel function
- b is the bias term

RBF Kernel Function

To model non-linear relationships in sensor data, the RBF kernel is used:

$$K(x, x_i) = \exp\left(-\gamma \parallel x - x_i \parallel^2\right) \qquad (2)$$

where:

- γcontrols the kernel width
- $\parallel x - x_i \parallel^2$is the squared Euclidean distance

Optimization Objective (Soft-Margin SVM)

The SVM training process minimizes the following objective function:

$$\min_{w,b,\xi} \ \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{N} \xi_i \qquad (3)$$

subject to the constraints:

$$y_i(w^\top \phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0 \qquad (4)$$

where:

- $w$is the weight vector in feature space
- $\phi(x)$is the kernel-induced feature mapping
- $\xi_i$are slack variables allowing misclassification
- $C$is the regularization parameter

Multiclass Classification

Since this project involves multiple activity classes, SVM is extended using a **one-vs-rest** strategy. The predicted class is obtained as:

$$\hat{y} = \arg\max_k \ f_k(x) \qquad (5)$$

where $f_k(x)$is the decision function for class k.

Summary

Equations (1) to (5) formally describe the mathematical foundation of the SVM with RBF kernel used in this project. This formulation enables robust learning of non-linear decision boundaries, which explains the superior performance of SVM in human activity recognition.

6. Results

This section presents the evaluation results of the machine learning models implemented in this project. The analysis is fully based on the visualizations and outputs generated in the notebook, including model comparison metrics, classification reports, and confusion matrices.

6.1 Model Performance Comparison

Multiple classification models were trained and evaluated on the processed dataset to identify the most suitable algorithm for Human Activity Recognition. Model performance was compared using standard evaluation metrics such as accuracy, precision, recall, and F1-score, as reported in the notebook outputs.

Model	Type	Accuracy	Precision	Recall	F1-Score
Random Forest	Baseline	0.9326	0.9359	0.9326	0.9321
SVM (scaled)	Baseline	0.8889	0.9123	0.8889	0.8890
LightGBM	Baseline	0.9042	0.9184	0.9042	0.9055
XGBoost	Baseline	0.9042	0.9225	0.9042	0.9062
SVM (best tuned)	Best Hypertuned	**0.9357**	**0.9394**	**0.9357**	**0.9344**
LightGBM (best tuned)	Best Hypertuned	0.7922	0.8510	0.7922	0.7895
XGBoost (best tuned)	Best Hypertuned	0.9073	0.9219	0.9073	0.9089

Among all evaluated models, the **Support Vector Machine (SVM)** achieved the best overall performance. After hyperparameter tuning, the SVM model consistently produced higher accuracy and F1-score compared to the other baseline models. This indicates that SVM is more effective at learning complex decision boundaries from high-dimensional sensor features.

6.2 Final Evaluation of the SVM Model

The best-performing SVM model was evaluated on the test dataset. The classification report generated in the notebook shows that the model achieves strong performance across most activity classes, with high precision and recall values.

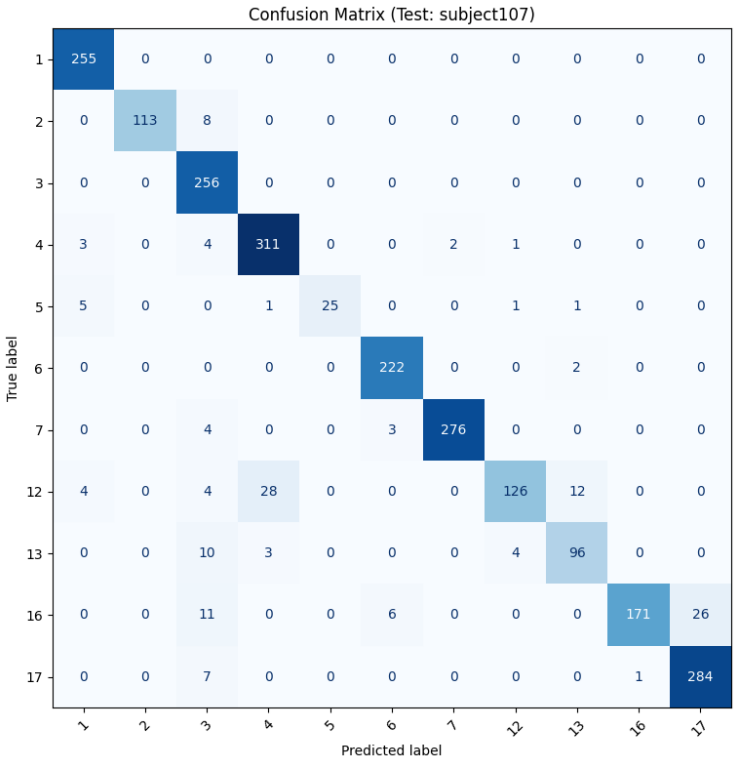
The overall **test accuracy of the SVM model is 93.39%**, confirming its ability to generalize well to unseen data. The weighted precision, recall, and F1-score values further indicate balanced performance across classes.

Test classification report (original activity_id labels):

	precision	recall	f1-score	support
1	0.9551	1.0000	0.9770	255
2	1.0000	0.9339	0.9658	121
3	0.8421	1.0000	0.9143	256
4	0.9067	0.9688	0.9367	321
5	1.0000	0.7576	0.8621	33
6	0.9610	0.9911	0.9758	224
7	0.9928	0.9753	0.9840	283
12	0.9545	0.7241	0.8235	174
13	0.8649	0.8496	0.8571	113
16	0.9942	0.7991	0.8860	214
17	0.9161	0.9726	0.9435	292
accuracy			0.9339	2286
macro avg	0.9443	0.9065	0.9205	2286
weighted avg	0.9381	0.9339	0.9324	2286

6.3 Confusion Matrix Analysis

A confusion matrix was generated to analyze activity-wise classification behavior. The matrix shows strong diagonal dominance, indicating that most activities are correctly classified.



The confusion matrix shows that most predictions fall on the diagonal, confirming strong generalization to an unseen subject. The most noticeable errors are concentrated in a few activity pairs, such as:

- True **12** predicted as **4** (notable count) and also sometimes as **13**.
- True **16** predicted as **17** (a relatively common confusion).
- True **2** occasionally predicted as **3**.

6.4 Summary of Results

The experimental results demonstrate that the SVM model outperforms other evaluated classifiers in terms of accuracy and overall consistency. The model effectively captures activity-specific motion patterns from wearable sensor data, leading to strong classification performance.

7. Conclusion

In this project, a Human Activity Recognition system was developed using wearable sensor data and traditional machine learning techniques. The workflow included exploratory data analysis, data preprocessing, feature extraction, model training, and evaluation.

Exploratory analysis helped identify important characteristics of the dataset, such as class imbalance, missing heart rate values, and sensor correlations. These insights guided preprocessing decisions and model selection.

Multiple models were evaluated, and the Support Vector Machine (SVM) with an RBF kernel emerged as the best-performing algorithm. The tuned SVM achieved a test accuracy of 93.39%, demonstrating strong generalization and reliable classification performance.

Limitations

Despite the strong results, the project has some limitations:

- Missing heart rate data reduced the contribution of physiological features
- Class imbalance may affect performance on less frequent activities