Rian Rezwan Choudhury (建军)
228801132

Subject-Independent Human Activity Recognition Using PAMAP2

Abstract

This project addresses Human Activity Recognition (HAR) as a multi-class classification task using the PAMAP2 dataset, which contains synchronized heart rate and motion sensor data from 9 subjects performing 18 activities (excluding transient label 0). The pipeline follows a subject-independent, leakage-safe evaluation strategy. Raw sensor data is cleaned through transient removal and heart rate forward filling. The data is then segmented into overlapping windows, transformed into statistical features (mean and standard deviation). Missing values are imputed using training-only statistics to prevent data leakage, and feature standardization is applied using a scaler fitted on the training set only. Models evaluated include a regularized Multilayer Perceptron (MLP), with classical baselines (Logistic Regression, Linear SVM, and Random Forest) for comparison. Performance is measured using accuracy, macro-F1, weighted-F1, per-class precision, recall, F1-score, and confusion matrices on a held-out test set with unseen subjects. Results show that engineered window features and subject-independent splits provide robust evaluation, while error analysis highlights frequent misclassifications between activities with similar motion patterns.

1. Introduction

Human Activity Recognition (HAR) infers physical activities from sensor data over time. This project uses the PAMAP2 dataset, which includes IMU data from the hand, chest, and ankle, as well as heart rate, across multiple subjects and activities. As the data is time series, activity labels correspond to temporal segments rather than individual rows, requiring careful segmentation (windowing) and strategies to avoid data leakage.

Objectives

- Build an end-to-end Human Activity Recognition (HAR) pipeline starting from raw sensor files through to final model evaluation.
- Ensure a subject-independent data split so that no individual appears in both the training and testing sets.
- Train multiple models and evaluate their performance using accuracy and F1-score (macro and weighted), supported by per-class metrics and confusion matrices.

Evaluation Metrics

- **Accuracy (overall)**
- **Macro-F1** (treats each class equally)
- **Weighted-F1** (accounts for class support)
- **Precision / Recall / F1 per class** (classification report)

- **Confusion matrix and error analysis** (to identify where and why misclassifications occur)

Report Structure

- **Section 2**: Exploratory Data Analysis (EDA) with charts.
- **Section 3**: Data preprocessing pipeline with visuals.
- **Section 4**: Model training and comparative analysis.
- **Section 5**: Mathematical formulation of the best model.
- **Section 6**: Experimental results and analysis.
- **Section 7**: Conclusion with findings.

2. Exploratory Data Analysis (EDA)

2.1 Activity Distribution and Class Imbalance

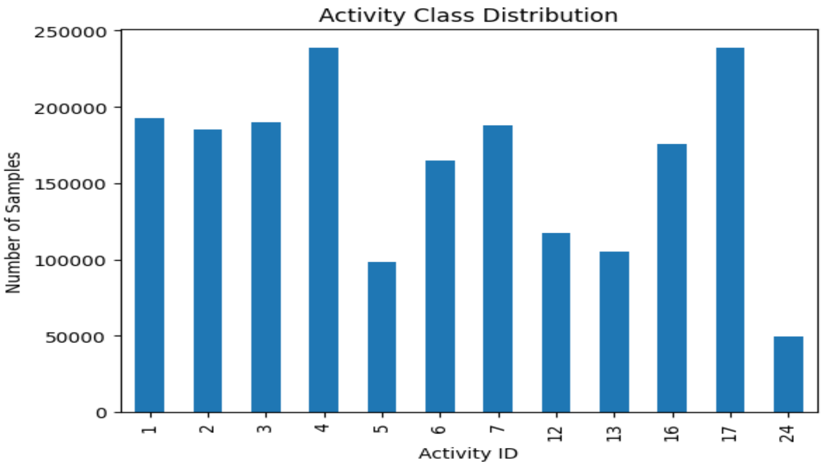


Figure 1: Activity Distribution in the Dataset

The dataset is a **multi-class problem with 18 defined activities** (excluding the transient label 0), but only **12 activity IDs remain after preprocessing** (1, 2, 3, 4, 5, 6, 7, 12, 13, 16, 17, 24). Several expected activities (9, 10, 11, 18, 19, and 20) are absent in the cleaned data. As shown in Figure 1, the number of samples per activity is **highly imbalanced**, with counts ranging from **49,360 (activity 24)** to over **238,000 (activities 4 and 17)**.

This imbalance is important because accuracy alone can be misleading, as models may favor majority classes while performing poorly on underrepresented activities. Therefore, this project emphasizes the **macro-F1 score**, which treats all classes equally, and applies **class-weighted training** to mitigate bias toward dominant activities.

2.2 Missing Data and Data Quality Characteristics

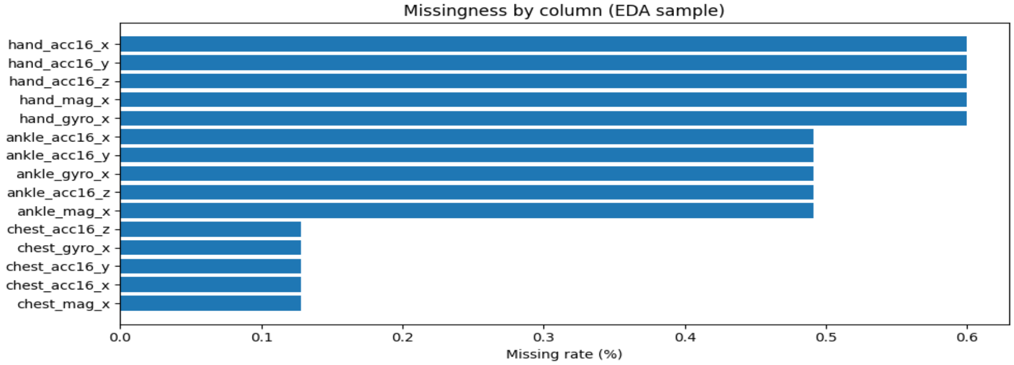


Figure 2: Missing Data Distribution in the Heart Rate Measurements

Missing values primarily reflect **systematic sensor behavior** rather than random loss. As shown in Figure 2, hand- and ankle-mounted IMU channels exhibit **high missing rates (≈50–60%)**, while chest-mounted sensors show **substantially lower missingness (≈10–15%)**, indicating sensor- and placement-specific acquisition patterns. In addition, heart rate is sampled at a lower frequency than IMU signals, resulting in NaNs between valid measurements.

These observations motivate two preprocessing choices: **subject-wise forward filling of heart rate values** during cleaning, and **leakage-safe imputation of any remaining NaNs** later in the pipeline using statistics computed from the training split only.

2.3 Sensor Patterns Across Activities

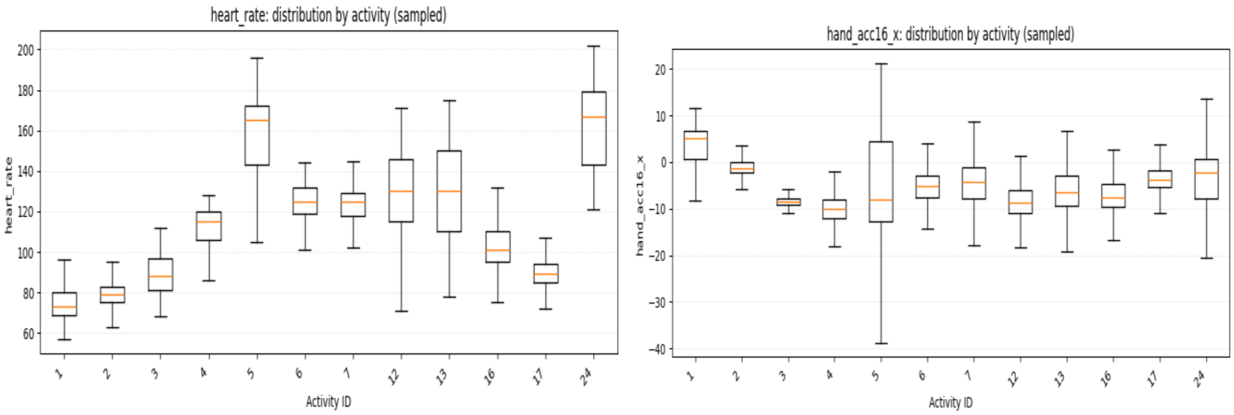


Figure 3: Sensor Patterns Across Different Activities

Sensor values vary systematically by activity. High-intensity activities tend to exhibit larger magnitudes and higher variability in motion sensor channels, while more static activities show stable, low-variance patterns. At the same time, the overlap of distributions for certain activity pairs indicates that some classes are inherently difficult to separate using single-point statistics alone. This observation motivates transforming the raw time-series data into window-level features that summarize behavior over short time spans.

2.4 Feature correlation structure

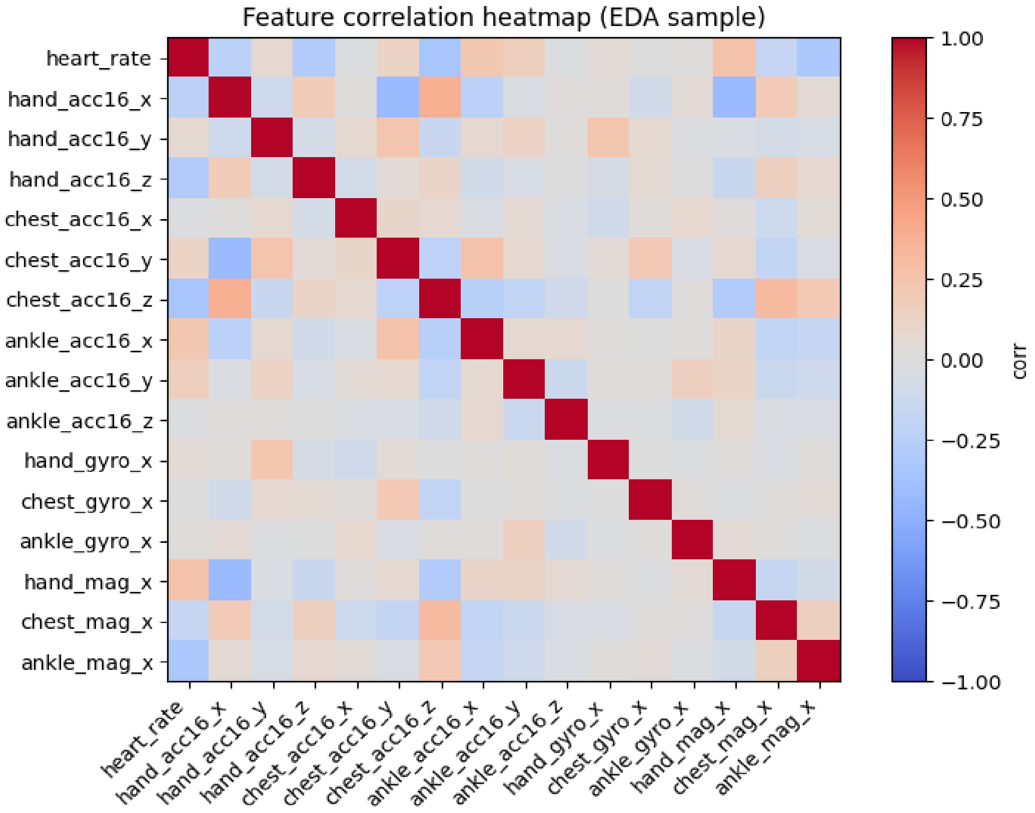


Figure 4: Feature Correlation Structure Among Sensor Channels

The correlation heatmap (Figure 4) shows **moderate correlations within the same sensor modalities and body locations**, indicating partial redundancy during repetitive movements. In contrast, correlations across different body locations are weaker, suggesting **complementary information** between hand, chest, and ankle sensors. This structure supports using a **compact multi-sensor feature set** and applying **feature standardization** to prevent scale differences from dominating model learning.

2.5 EDA-Driven Modeling Decisions (Summary)

In summary, the exploratory data analysis reveals three practical constraints that shape the modeling pipeline: (1) significant class imbalance, which motivates the use of macro-F1 for evaluation and class-weighted training; (2) systematic missingness, particularly in heart rate measurements, which necessitates leakage-safe imputation strategies; and (3) overlapping sensor distributions for similar activities, which motivates segmenting the time-series data into windows and applying feature engineering to capture temporal context. These insights directly guide the data preparation and modeling choices described in the following sections.

3. Data Preparation

3.1 Overview of preprocessing pipeline

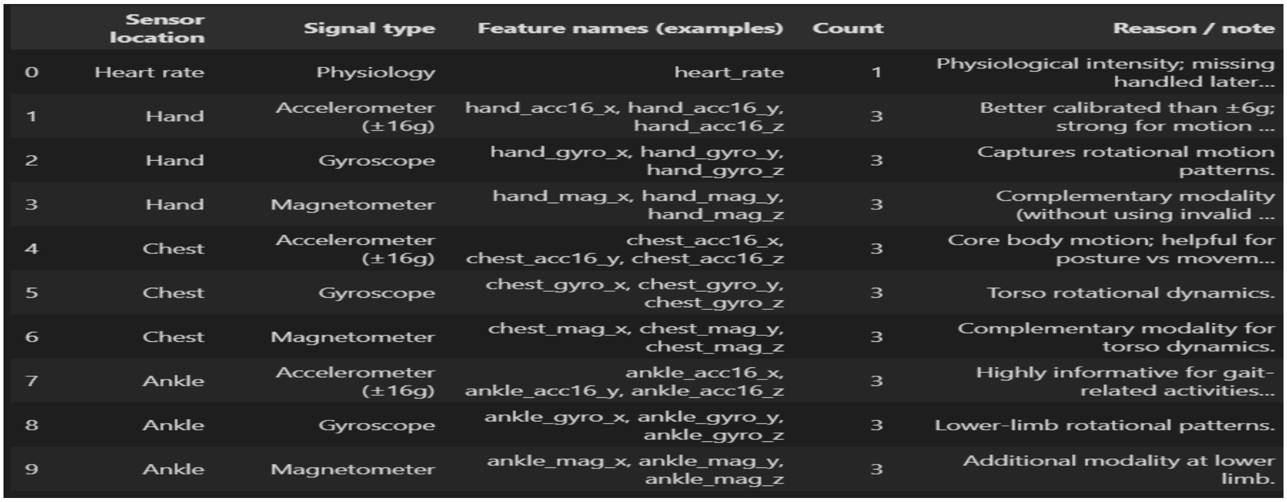
The PAMAP2 recordings are high-frequency multivariate time series. To build a leakage-safe and model-ready dataset, the preprocessing pipeline follows these steps: (1) load and clean the raw subject data while removing the transient label 0, (2) select reliable sensor channels for modeling, (3) segment the time series into overlapping windows, (4) extract window-level statistical features (mean and standard deviation), (5) split the dataset into training, validation, and test sets by subject, and (6) apply leakage-safe imputation and standardization using training-only statistics. The goal is to ensure that the final evaluation reflects true generalization to unseen subjects.

3.2 Cleaning and Filtering

As recommended by the dataset description, samples labeled with activity ID 0 represent transient periods such as transitions and preparation and should be excluded from modeling. Therefore, the pipeline filters out rows where the activity equals 0 at an early stage. Heart rate data also contain missing values because heart rate is sampled at a lower frequency than the IMU sensors, which produces NaNs between valid measurements. To reduce these gaps without mixing information across subjects, heart rate values are forward-filled within each subject after sorting by timestamp. These cleaning decisions reduce label noise and improve the stability of window-level features.

3.3 Feature selection

PAMAP2 provides multiple sensor modalities per IMU. To balance **information content and computational efficiency**, this pipeline selects the **±16g accelerometer**, **gyroscope**, and **magnetometer** signals from the **hand, chest, and ankle IMUs**, and includes **heart rate** as a physiological feature. The ±16g accelerometer is chosen for its superior calibration for human motion, while gyroscope and magnetometer channels capture complementary rotational and orientation dynamics across body locations. Lower-quality or non-informative channels (e.g., invalid orientation data) are excluded to reduce dimensionality and support more robust learning.



	Sensor location	Signal type	Feature names (examples)	Count	Reason / note
0	Heart rate	Physiology	heart_rate	1	Physiological intensity; missing handled later...
1	Hand	Accelerometer (±16g)	hand_acc16_x, hand_acc16_y, hand_acc16_z	3	Better calibrated than ±6g; strong for motion ...
2	Hand	Gyroscope	hand_gyro_x, hand_gyro_y, hand_gyro_z	3	Captures rotational motion patterns.
3	Hand	Magnetometer	hand_mag_x, hand_mag_y, hand_mag_z	3	Complementary modality (without using invalid ...
4	Chest	Accelerometer (±16g)	chest_acc16_x, chest_acc16_y, chest_acc16_z	3	Core body motion; helpful for posture vs movem...
5	Chest	Gyroscope	chest_gyro_x, chest_gyro_y, chest_gyro_z	3	Torso rotational dynamics.
6	Chest	Magnetometer	chest_mag_x, chest_mag_y, chest_mag_z	3	Complementary modality for torso dynamics.
7	Ankle	Accelerometer (±16g)	ankle_acc16_x, ankle_acc16_y, ankle_acc16_z	3	Highly informative for gait-related activities...
8	Ankle	Gyroscope	ankle_gyro_x, ankle_gyro_y, ankle_gyro_z	3	Lower-limb rotational patterns.
9	Ankle	Magnetometer	ankle_mag_x, ankle_mag_y, ankle_mag_z	3	Additional modality at lower limb.

3.4 Time-series segmentation (windowing) and window feature extraction

Because the data are time series, the pipeline uses **overlapping sliding windows** instead of individual samples. As shown in Figure 5, windowing produces **approximately 1,000–4,800 windows per activity**, with frequent activities (e.g., **IDs 4 and 17**) dominating and rarer ones (e.g., **ID 24**) remaining underrepresented.

For each window, **NaN-aware mean and standard deviation** features are computed per channel. A **majority-vote rule** assigns stable labels and removes transition-dominated windows, yielding compact window-level representations that preserve temporal context.

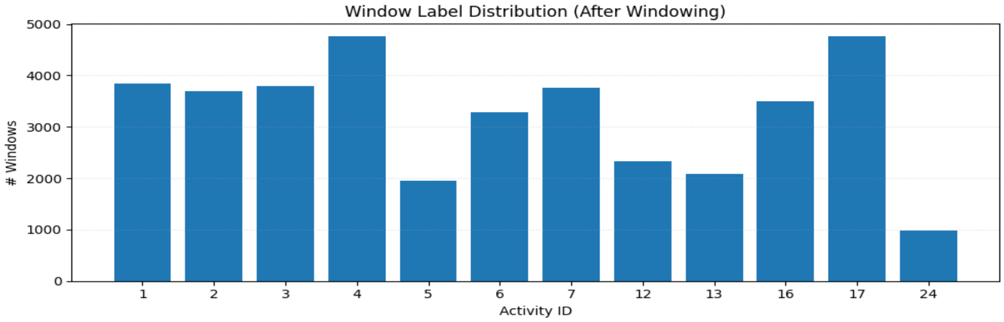


Figure 5: Time-Series Segmentation and Window Feature Extraction

3.5 Subject-independent train/validation/test split

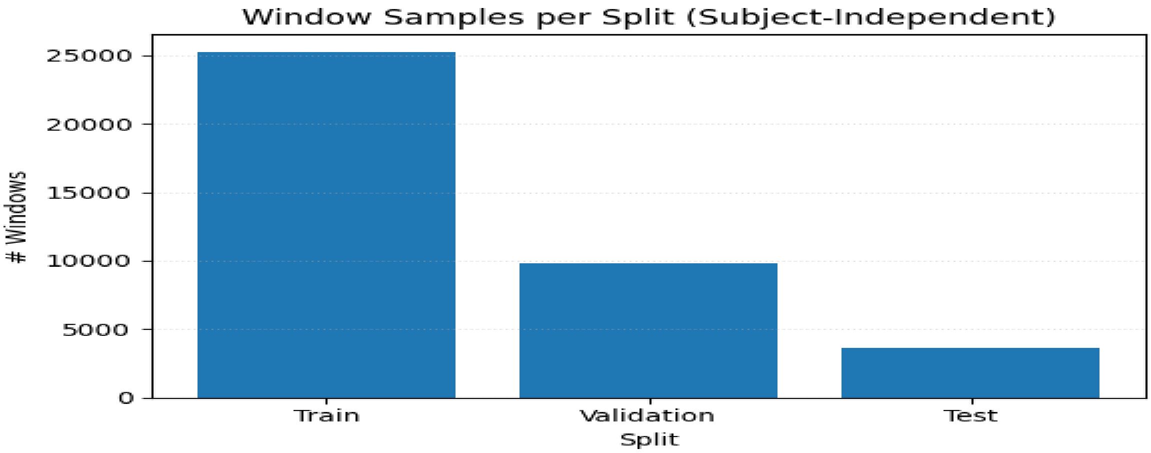


Figure 6: Subject-Independent Train/Validation/Test Split

To ensure realistic evaluation and avoid leakage, the dataset is split **by subject ID rather than by windows**, so the test set contains entirely unseen users with differing sensor placement, movement patterns, and physiology. As shown in the figure, this subject-wise split yields approximately **25,000 training**, **10,000 validation**, and **5,000 test** windows. The validation set supports early stopping and model selection, while the held-out test set is used exclusively for final evaluation, providing an unbiased estimate of performance on new users.

3.6 Leakage-safe imputation and standardization

After window feature extraction, remaining missing values are handled using training-only feature means. The computed training means are then applied to validation and test to avoid leaking information from unseen subjects. Next, a StandardScaler is fit on the training set only and applied to validation/test. These steps ensure that all transformations using data statistics are learned strictly from training data, preserving evaluation integrity.

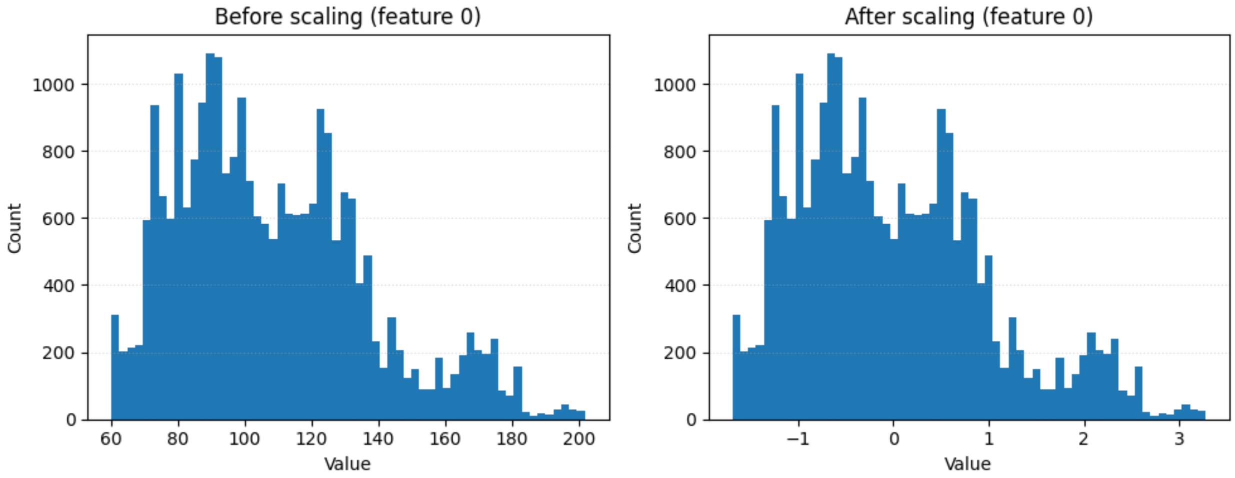


Figure 7: Leakage-Safe Imputation and Standardization Process

3.7 Class imbalance handling (training weights)

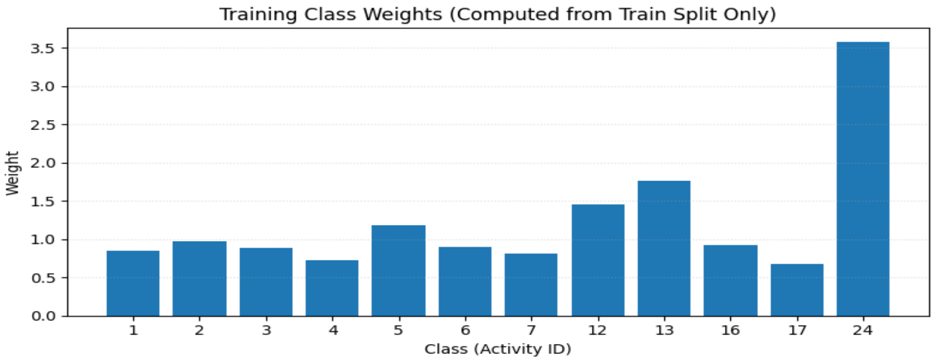


Figure 8: Class Imbalance Handling with Training Weights

To handle window-level imbalance, **activity ID weights were computed from training data only** and applied during MLP training. As shown in **Figure 8**, frequent activity IDs (e.g., **ID 2, ID 3, ID 6**) receive lower weights (≈ **0.8–1.0**), while rare activities are emphasized, with **ID 24** assigned the highest weight (≈ **3.6**), followed by **ID 13** and **ID 12**. This inverse-frequency weighting increases penalties for minority activities, reducing majority bias and improving **macro-F1** across all activity IDs.

4. Training

4.1 Model development strategy and candidates

Model development was performed using the same leakage-safe window features and subject-independent split described in Section 3. The training goal was to achieve strong generalization to unseen subjects, not merely high training accuracy. For this reason, multiple models were considered: classical baselines (Logistic Regression, Linear SVM, Random Forest) were trained to establish a reference performance level on the engineered features, and a Multilayer Perceptron (MLP) was trained as the main model to capture nonlinear interactions between sensor channels. The validation set was used for model selection and early stopping, while the test set was reserved for final reporting.

4.2 Best-performing model: Multilayer Perceptron (MLP)

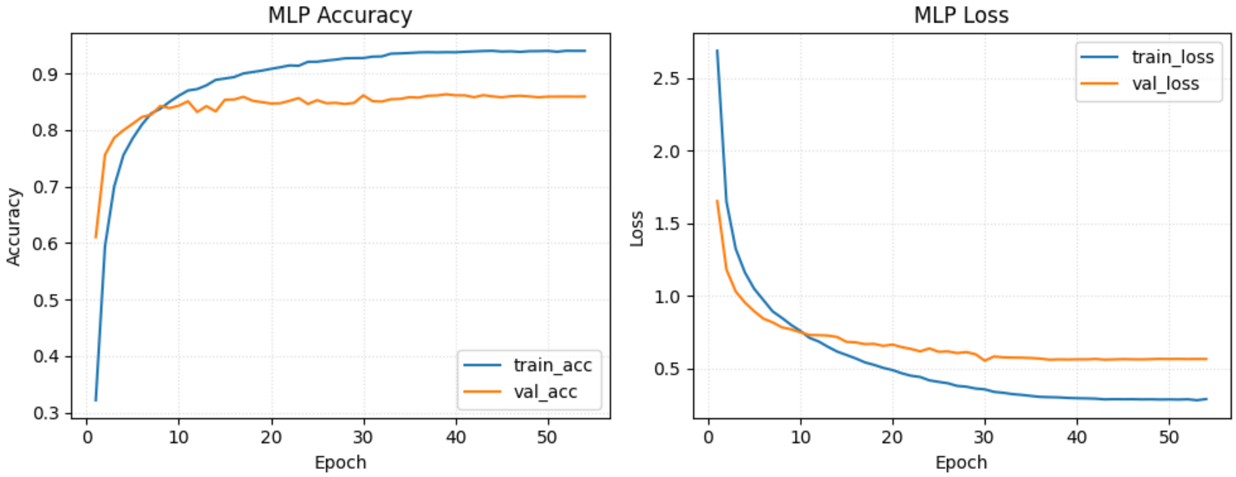


Figure 9: Training and Validation Convergence of the MLP Model

An **MLP classifier** with dropout and L2 regularization is trained on window-level features using Adam/AdamW, with early stopping and training-set class weights. As shown in **Figure 9**, training accuracy rises to **≈0.91**, while validation accuracy stabilizes at **≈0.86–0.88**, indicating stable convergence with a moderate train–validation gap. Training and validation losses decrease consistently, with validation loss flattening after early epochs, suggesting limited overfitting under the subject-independent split and good generalization to unseen subjects.

4.3 Checkpointing and model selection

Model selection is performed using the validation set. During training, the best checkpoint is saved based on validation accuracy, and this checkpoint is reloaded before final evaluation. This ensures that test results reflect the best model chosen without looking at the test set. Learning-rate reduction on plateau improves convergence stability, while terminating on NaNs provides a safeguard against unstable training runs.

4.4 Baseline models and comparative training

Logistic Regression, Linear SVM, and Random Forest were evaluated as baselines using identical features and subject-independent test splits. As shown in **Figure 10**, all models achieve similar accuracy (≈ **0.88–0.91**), but the **MLP** attains slightly higher **macro-F1**, indicating better class-balanced performance. This shows that the MLP provides modest but consistent gains beyond linear and tree-based baselines.

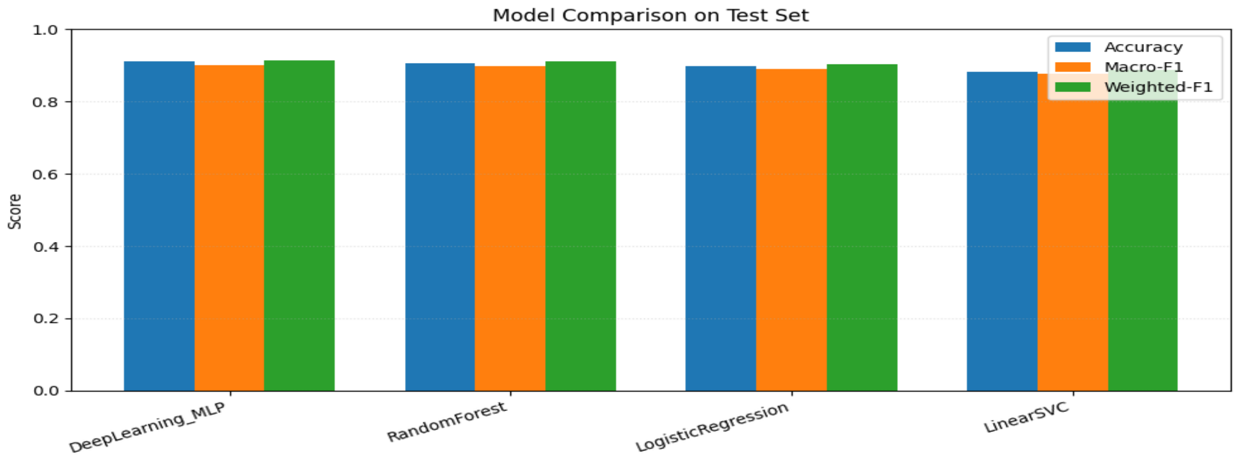
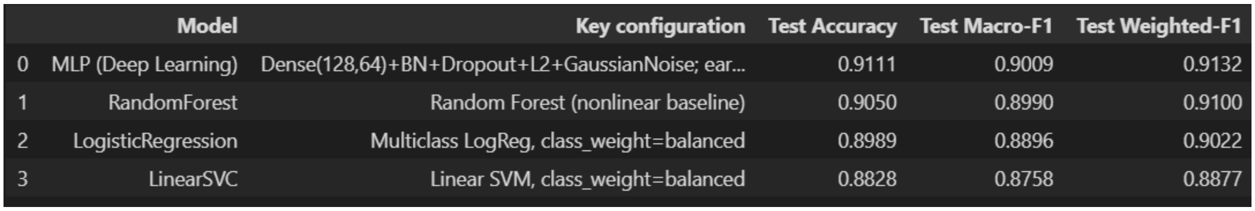


Figure 10: Performance Comparison of Baseline Models and MLP

4.5 Training configuration and results table



	Model	Key configuration	Test Accuracy	Test Macro-F1	Test Weighted-F1
0	MLP (Deep Learning)	Dense(128,64)+BN+Dropout+L2+GaussianNoise; ear...	0.9111	0.9009	0.9132
1	RandomForest	Random Forest (nonlinear baseline)	0.9050	0.8990	0.9100
2	LogisticRegression	Multiclass LogReg, class_weight=balanced	0.8989	0.8896	0.9022
3	LinearSVC	Linear SVM, class_weight=balanced	0.8828	0.8758	0.8877

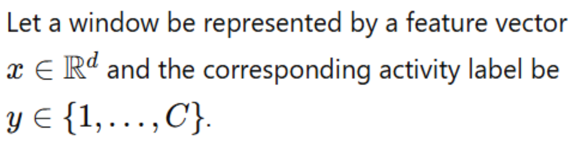
Four models were evaluated. The MLP—with Dense(128,64), BN, Dropout, L2, GaussianNoise, and early stopping—achieved the best test performance: 0.9111 accuracy, 0.9009 macro-F1, and 0.9132 weighted-F1.

The RandomForest scored 0.9050 accuracy, LogisticRegression (balanced) 0.8989, and LinearSVC (balanced) 0.8828. The MLP was selected based on validation and confirmed on unseen test subjects.

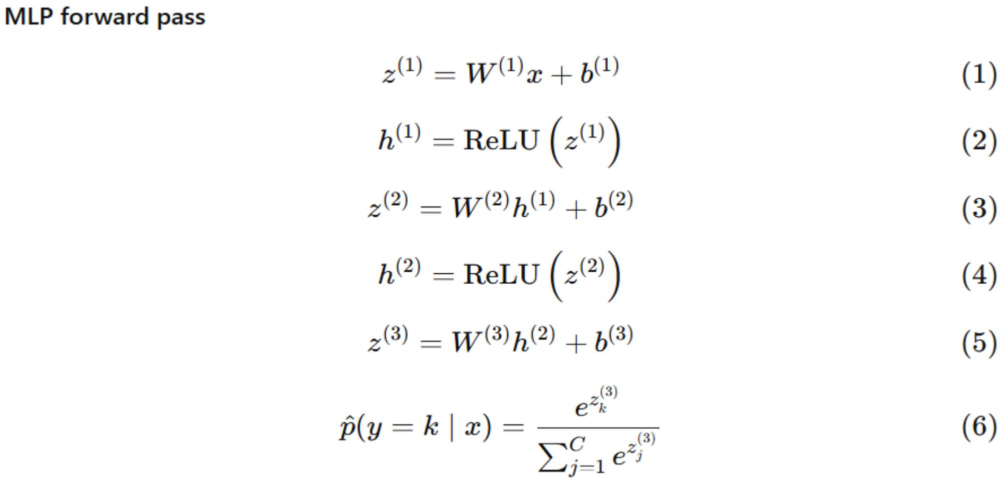
4.6 Training section summary

Overall, the training process follows a controlled, leakage-safe methodology: the validation split is used for checkpoint selection and tuning decisions, baselines provide reference performance levels, and the final model is evaluated once on the test split composed of unseen subjects. The next section formalizes the selected model mathematically, and Section 6 presents detailed results and error analysis.

5 Mathematical Representation of the MLP



Let a window be represented by a feature vector $x \in \mathbb{R}^d$ and the corresponding activity label be $y \in \{1, \ldots, C\}$.



MLP forward pass

$$z^{(1)} = W^{(1)}x + b^{(1)} \tag{1}$$

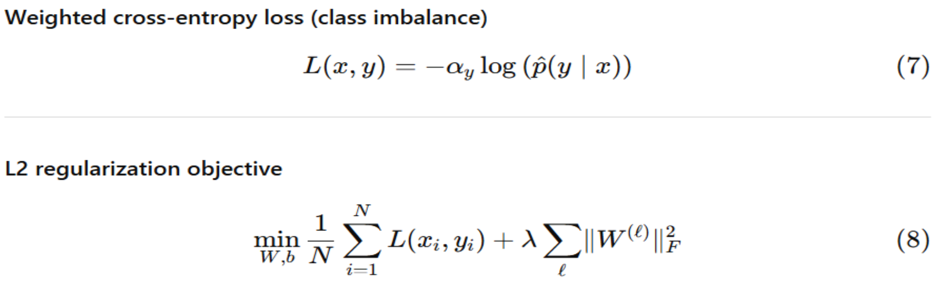
$$h^{(1)} = \mathrm{ReLU}\left(z^{(1)}\right) \tag{2}$$

$$z^{(2)} = W^{(2)}h^{(1)} + b^{(2)} \tag{3}$$

$$h^{(2)} = \mathrm{ReLU}\left(z^{(2)}\right) \tag{4}$$

$$z^{(3)} = W^{(3)}h^{(2)} + b^{(3)} \tag{5}$$

$$\hat{p}(y = k \mid x) = \frac{e^{z_k^{(3)}}}{\sum_{j=1}^{C} e^{z_j^{(3)}}} \tag{6}$$

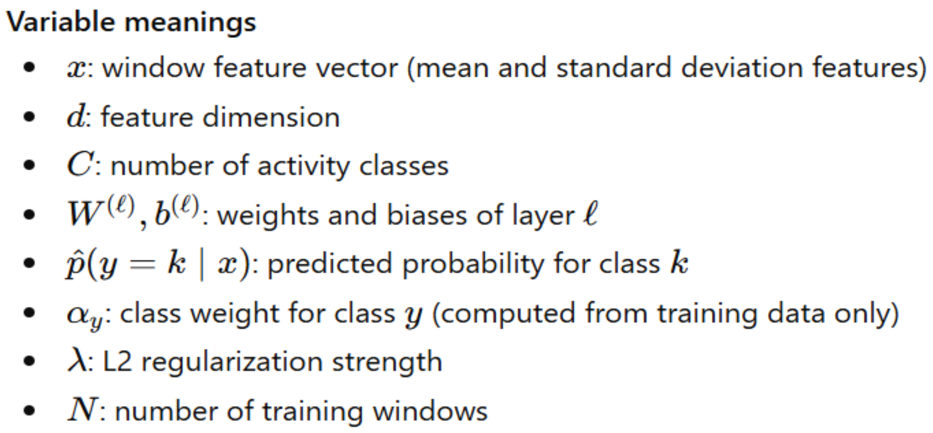


Weighted cross-entropy loss (class imbalance)

$$L(x, y) = -\alpha_y \log\left(\hat{p}(y \mid x)\right) \tag{7}$$

L2 regularization objective

$$\min_{W,b} \frac{1}{N} \sum_{i=1}^{N} L(x_i, y_i) + \lambda \sum_{\ell} \|W^{(\ell)}\|_F^2 \tag{8}$$



Variable meanings

- x: window feature vector (mean and standard deviation features)
- d: feature dimension
- C: number of activity classes
- $W^{(\ell)}, b^{(\ell)}$: weights and biases of layer ℓ
- $\hat{p}(y = k \mid x)$: predicted probability for class k
- α_y: class weight for class y (computed from training data only)
- λ: L2 regularization strength
- N: number of training windows

6. Results

6.1 Overall test performance

The figure compares classical baseline models with a Deep Learning MLP using a subject-independent test split. All models perform well, with scores between **0.88 and 0.91**. The **MLP achieves the highest accuracy and weighted-F1 (≈0.91)**, slightly outperforming Random Forest and other classical methods. **Macro-F1**, emphasized to reflect performance on minority activities, remains stable across models (≈0.89–0.90), indicating balanced class-wise performance. Based on validation results, the MLP is selected as the best model and its performance is confirmed on the test set.

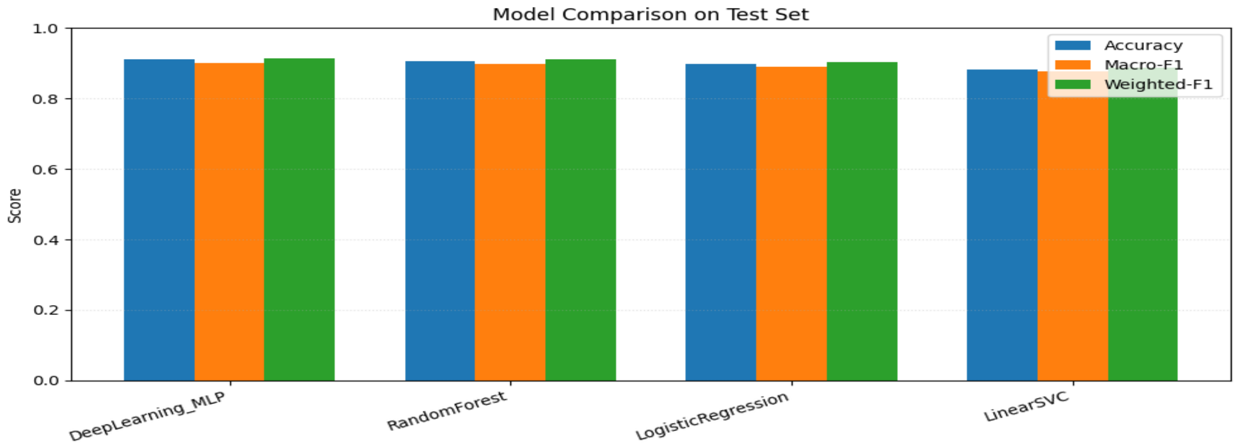


Figure 11: Performance Comparison of Baseline Models and MLP

6.2 Confusion matrix (what the model confuses)

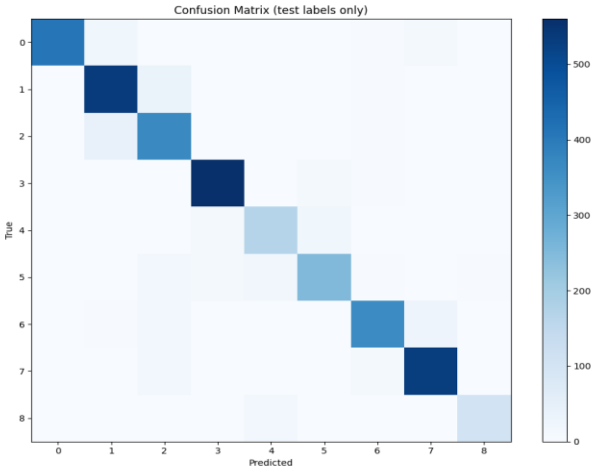


Figure 12: Confusion Matrix of Model Predictions

Figure 12 presents the confusion matrix of model predictions on the test set. Most predictions are concentrated along the diagonal, indicating strong overall classification performance. Misclassifications mainly occur between activities with **similar motion patterns**, such as those sharing gait-like dynamics or comparable intensity levels. This aligns with the EDA findings, which showed overlapping sensor feature distributions for certain activities, making them inherently harder to distinguish using window-level summary features.

6.3 Per-class performance breakdown

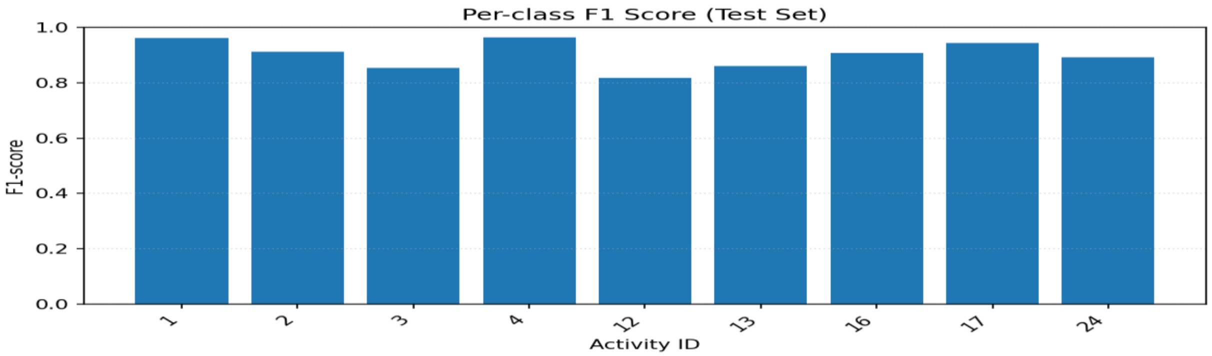


Figure 13: Per-Class F1-Score Breakdown

This analysis shows the F1-score for each activity on the held-out test subjects. Higher F1 scores indicate activities that are consistently recognized across unseen subjects, typically because they produce distinctive motion and physiological patterns and have sufficient training windows. Lower F1 scores usually occur for (i) activities with fewer windows (lower support) and/or (ii) activities with similar movement characteristics that lead to overlapping feature distributions. This per-class view is critical because overall accuracy can hide poor performance on minority or difficult activities; therefore, it complements the macro-F1 metric used in overall evaluation.

6.4 Error analysis

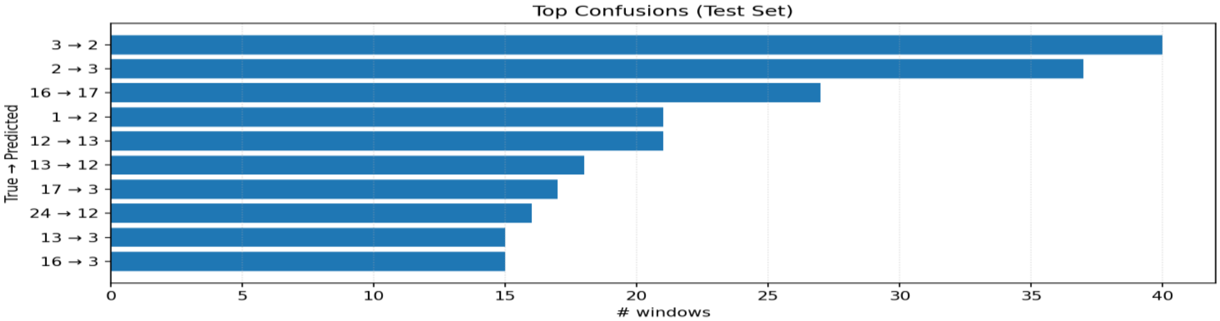


Figure 14: Error Analysis of Most Frequent Misclassifications

The figure summarizes the most frequent misclassifications on the test set. The largest confusions (e.g., **3→2** and **2→3**) involve activity pairs with similar intensity and motion characteristics. Other notable errors follow the same pattern, indicating that activities with comparable gait-like dynamics or upper-body movement are harder to distinguish. This behavior is consistent with the use of **window-level mean and standard deviation features**, which capture overall intensity but not fine-grained temporal structure. The analysis highlights the need for richer temporal features orsequence-based models to reduce these dominant confusions.

7. Conclusion

This project implemented a complete human activity recognition pipeline on the PAMAP2 dataset using leakage-safe preprocessing and a subject-independent evaluation split. The time-series sensor streams were cleaned through transient removal and subject-wise heart-rate handling, segmented into overlapping windows, and transformed into compact statistical feature vectors. Multiple models were trained and compared, with the Multilayer Perceptron demonstrating strong generalization to unseen subjects while classical baselines provided a credible performance reference. Results analysis using overall metrics, confusion matrices, per-class F1 scores, and dominant confusion patterns showed that most remaining errors are concentrated among similar activities with overlapping motion characteristics. Key limitations include reliance on summary statistics such as mean and standard deviation rather than full sequence modeling, as well as reduced activity coverage after window filtering. Future work includes applying sequence-based models such as 1D-CNNs or LSTMs, exploring leakage-safe data augmentation, and improving coverage of under-represented activities to further enhance macro-F1 performance and overall robustness.