Human Activity Recognition Using Smartphone Accelerometer Data: A Classical ML Perspective

PRANAV KRISHNA SURAPARAJU
S20230020353, ECE
Indian Institute of Information Technology, Sri City

PEDDIREDDY ESWAR REDDY
S2023002333, ECE
Indian Institute of Information Technology, Sri City

PUNNNAM RAHUL
S20230020340, ECE
Indian Institute of Information Technology, Sri City

Poornachandar Degavath
S20230020336, ECE
Indian Institute of Information Technology, Sri City

***Abstract*—Human Activity Recognition (HAR) using smartphone sensors plays a major role in applications such as fitness tracking, smart healthcare, and behavior monitoring. In this project, accelerometer data from the WISDM dataset was used to classify common daily activities. The raw time-series signals were segmented using a 60-sample sliding window, and 12 statistical features (mean, standard deviation, maximum, and minimum of the X, Y, and Z axes) were extracted from each window. Machine learning models including Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression were trained and evaluated. Among these, the Random Forest model achieved the highest accuracy of 83.52**

***Index Terms*—Human Activity Recognition (HAR), Smartphone Sensors, Accelerometer Data, Machine Learning, Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Feature Extraction, Time-domain Features, Pattern Recognition.**

I. Introduction

Human Activity Recognition (HAR) is an extensively studied area within machine learning that focuses on identifying physical human actions from sensor signals. Modern smartphones are equipped with built-in inertial sensors such as accelerometers and gyroscopes, enabling continuous motion monitoring without the need for external hardware. HAR plays a critical role in healthcare rehabilitation, elderly fall detection, fitness monitoring, behavior tracking, and human–computer interaction. While activity recognition in controlled environments has been widely explored, accurate classification of real-life daily activities remains challenging due to natural variations in motion patterns, inconsistent movement intensity, and differences across users.

The WISDM dataset offers a realistic benchmark for evaluating HAR systems, containing multi-subject activity recordings collected using a smartphone placed in a natural pocket-based position. Unlike laboratory-controlled datasets, WISDM captures movement variability caused by gait imbalance, different body postures, varying walking speeds, and natural fluctuations in sensor orientation. These characteristics introduce complexity in distinguishing similar activities such as walking vs. walking upstairs or sitting vs. standing, where transitional movements lead to overlapping feature distributions.

Classical machine-learning approaches remain highly valuable in HAR research due to their low computational cost, interpretability, and suitability for deployment on resource-constrained wearable or mobile platforms. Feature-based pipelines allow efficient processing and real-time classification, unlike deep learning models which often require high processing power and large labeled datasets. In this work, we develop a lightweight and efficient HAR pipeline—including preprocessing, handcrafted statistical feature extraction, feature scaling, and comparative model benchmarking—to establish a strong classical machine-learning baseline.

The accelerometer signals were segmented using a sliding window approach, and 12 time-domain features were extracted, including mean, standard deviation, maximum, and minimum values across the X, Y, and Z axes. Multiple ML models were evaluated, including Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Logistic Regression. Experimental results show that Random Forest achieved the highest performance (83.52%accuracy) on the WISDM dataset, demonstrating its effectiveness for smartphone-based activity recognition. This baseline analysis highlights the potential of classical ML for real-time HAR applications, and provides a reference for future research involving gyroscope fusion, adaptive learning, or deep neural architectures.

II. Related Work

Human Activity Recognition (HAR) has been widely studied using wearable and smartphone-based sensors for applications in healthcare monitoring, fitness tracking, eldercare support, and smart human–machine interaction. Early HAR research primarily focused on handcrafted time-domain statistical features extracted from accelerometer data, such as mean, standard deviation, maximum, minimum, and signal magnitude. Kwapisz et al. demonstrated the effectiveness of classical machine learning approaches—including Decision Trees, K-Nearest Neighbors (KNN), and Logistic Regression—on smartphone accelerometer signals using the WISDM dataset, establishing one of the earliest benchmarks for activity recognition using mobile devices.

Subsequent research has shown improved performance using ensemble learning techniques such as Random Forest and feature-selection methods such as Recursive Feature Elimination (RFE) and SelectKBest. Studies suggest that lightweight statistical features combined with classical ML models can provide competitive accuracy while remaining computationally efficient and easily deployable on embedded systems and mobile phones.

More recent works have explored deep learning architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and hybrid CNN-LSTM models to automatically learn high-level temporal and spatial features from raw sensor streams. While these approaches have achieved strong results on controlled datasets, their reliance on large labeled datasets, high computational cost, and poor real-time performance limit practical deployment on low-power devices such as wearables and smartphones.

Despite advances in deep learning, the literature highlights the importance of lightweight models and robust handcrafted feature extraction for real-world HAR scenarios, where noisy sensor readings, natural movement variability, and orientation shifts can degrade model performance. The WISDM dataset captures these realistic challenges, making it a strong benchmark for evaluating classical ML pipelines.

Motivated by these findings, this study establishes a reproducible classical machine-learning baseline using a compact 12-feature representation derived from smartphone accelerometer signals, and evaluates multiple classifiers including Random Forest, KNN, SVM, and Logistic Regression. This work contributes to the field by demonstrating that classical ML techniques can achieve reliable performance (83.52% accuracy) for real-time HAR, providing a foundation for future research involving gyroscope fusion, adaptive learning, or deep learning extensions.

III. Dataset Description

The WISDM (Wireless Sensor Data Mining) Smartphone Activity Dataset is a widely used benchmark for Human Activity Recognition research. It contains motion sensor data collected from 51 subjects performing 18 activities of daily living (ADL), including walking, jogging, climbing stairs, sitting, standing, writing, brushing teeth, clapping, and multiple eating-related tasks. Each activity was performed continuously for three minutes, resulting in approximately 54 minutes of motion data per participant and more than 15.6 million total sensor samples.

Sensor signals were recorded using a smartphone placed in the front pants pocket, running Android and equipped with a three-axis accelerometer, sampled at 20 Hz (one reading every 50 ms). Each sample captures the X, Y, and Z acceleration components along with a timestamp, subject ID, and activity label. The dataset reflects real-world conditions with natural variations in gait speed, sensor orientation, posture changes, and individual biomechanical patterns. These characteristics introduce intra-class variability, overlapping motion behaviors, and subtle transitions between activities, making HAR on WISDM challenging and realistic.

Although WISDM includes additional gyroscope and smartwatch sensor channels, this work focuses exclusively on smartphone accelerometer data to align with low-computational and mobile deployment requirements. The structured feature dataset derived from raw samples contains three components: - Subject identifier - 12 handcrafted statistical features - Activity label

IV. Preprocessing and Feature Extraction

Raw smartphone accelerometer signals are affected by noise, natural motion variability, and differences in device placement and user movement style. To enable reliable activity classification, the continuous time-series data is first transformed into a structured feature representation.

The accelerometer data is segmented using a fixed-length sliding window of 60 samples, which corresponds to a duration of 3 seconds at the 20 Hz sampling rate. Non-overlapping windows are used to balance temporal resolution with computational efficiency. For each window, three one-dimensional signals are considered: the X-, Y-, and Z-axis acceleration components.

From every window, a set of 12 handcrafted time-domain statistical features is computed:

- Mean, standard deviation, maximum, and minimum of the X-axis,
- Mean, standard deviation, maximum, and minimum of the Y-axis,
- Mean, standard deviation, maximum, and minimum of the Z-axis.

These features capture the amplitude, variability, and extremal behavior of the motion pattern along each axis, and provide a compact yet informative description of the underlying activity.

Each window is assigned an activity label using majority voting over the ground-truth activity annotations present in that segment. The subject identifier is retained in the intermediate feature file but excluded from the final training features, as it represents an ID rather than a predictive attribute.

Prior to model training, missing values (if any) are handled using median imputation, and all features are standardized using z-score normalization via `StandardScaler`. To further improve discriminability and reduce redundancy, feature selection methods are applied. Recursive Feature Elimination (RFE) with a linear SVM backbone, and SelectKBest with the ANOVA F-score, are used to identify the 10 most informative features for the classifiers. This preprocessing pipeline yields a lightweight and well-conditioned feature space suitable for classical machine learning models on resource-constrained devices.

V. Feature Selection

A. Feature Selection

The transformed dataset contains 12 handcrafted time-domain statistical features per window, obtained from the X-, Y-, and Z-axis accelerometer signals. Although the dimensionality is relatively low, feature selection is still applied to remove weak or redundant features and to improve generalization, especially under noisy real-world conditions. In this work, two supervised feature selection methods are employed, aligned with the downstream classifier pipelines.

After preprocessing with median imputation and z-score normalization, a linear Support Vector Machine (SVM) and an ANOVA F-test are used to rank features according to their discriminative power. From the original 12 features, the top 10 are retained for classifier training.

A. Recursive Feature Elimination (RFE) for SVM: For the SVM-based pipeline, Recursive Feature Elimination (RFE) is applied using a linear SVM as the base estimator:

- The scaled training data are first obtained using the preprocessing pipeline.
- A linear SVM (`SVC(kernel='linear')`) is used within RFE to evaluate feature importance via the absolute magnitude of its learned weights.
- At each iteration, the least important features are removed until only $N_{\text{RFE}} = 10$ features remain.

The reduced feature set is then used to train an RBF-kernel SVM classifier, which benefits from a compact and discriminative input representation.

B. SelectKBest (ANOVA F-score) for Other Classifiers: For the Random Forest, K-Nearest Neighbors, and Logistic Regression models, a filter-based feature selection method is employed using `SelectKBest` with the ANOVA F-score:

- The same scaled feature matrix is used as input.
- The ANOVA F-statistic (`f_classif`) is computed between each feature and the activity label.
- The top $k = 10$ features with the highest F-scores are retained.

This approach provides a simple, model-agnostic ranking of features and ensures that all non-SVM classifiers operate on a consistent 10-dimensional feature space. Together, the RFE and SelectKBest stages yield lightweight, discriminative feature subsets that improve robustness and efficiency for smartphone-based human activity recognition. s.

VI. Methodology

The full workflow of the proposed Human Activity Recognition pipeline is illustrated in Fig. 1. The process begins with time-series accelerometer signals collected from the smartphone. After windowing and statistical feature extraction, the resulting 12-dimensional vectors are standardized, and feature selection is applied to obtain the final 10-dimensional representation. The reduced feature vector is then supplied to multiple machine-learning models to enable fair performance comparison.

Support Vector Machine (SVM): A hybrid two-stage SVM pipeline is used, where Recursive Feature Elimination (RFE) with a linear kernel selects the 10 most informative features, followed by training with an RBF-kernel SVM. The nonlinear RBF kernel improves class separation in complex motion distributions.

Random Forest: An ensemble of decision trees trained using bootstrap aggregation. The model captures nonlinear relationships, reduces variance, and provides strong baseline performance for real-world noisy sensor data.

K-Nearest Neighbors (KNN): With $k = 7$ and distance-based weighting, KNN provides fine-grained class boundaries based on local neighborhood similarity. It benefits from the compact 10-feature representation.

Logistic Regression: A linear classifier trained with L2 regularization. Although simpler than ensemble methods, it provides interpretable behavior and serves as a baseline for comparison.

To ensure consistent evaluation, the dataset is split using 70/30 train–test partitioning, standardized using z-score normalization, and evaluated using accuracy and confusion matrix metrics. We evaluate four classical machine-learning classifiers: SVM (RBF kernel), Random Forest, KNN, and Logistic Regression. SVM benefits from kernel-based nonlinear transformation, RF leverages ensemble variance reduction, KNN performs instance-based local decision-making, and Logistic Regression provides interpretable linear modeling.

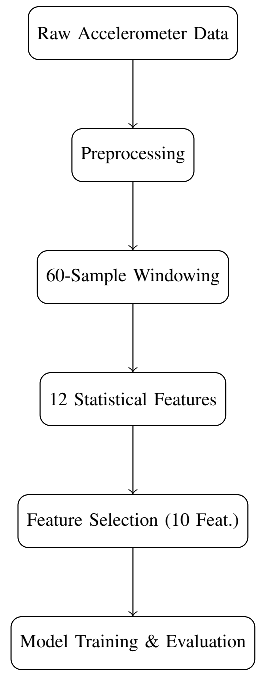

Fig. 1. Classical ML workflow for smartphone-based HAR.

sectionExperimental Setup The WISDM smartphone accelerometer feature set is partitioned into training and testing subsets using a random 70/30 split with `RANDOM_STATE = 42` to ensure reproducibility. All models operate on the same 10-dimensional feature vectors obtained after preprocessing and feature selection, enabling a fair comparison across classifiers. Training and evaluation are implemented in Python using `scikit-learn` within a Google Colab environment.

A. Classification Models

From a pattern recognition perspective, the proposed HAR system learns a discriminative mapping

$$f : \mathbb{R}^d \to \{1, 2, \ldots, K\},$$

where $\mathbf{x} \in \mathbb{R}^d$ denotes the d-dimensional feature vector ($d = 10$ after feature selection) and K is the number of activity classes. Given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, each classifier learns a decision rule that approximates the Bayes optimal classifier.

Support Vector Machine (SVM, RBF): SVM seeks a maximum-margin separator in a high-dimensional feature space. For the binary case, the primal optimization problem is

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to } y_i\big(\mathbf{w}^\top\phi(\mathbf{x}_i) + b\big) \geq 1 - \xi_i, \quad \xi_i \geq 0,$$

where $\phi(\cdot)$ is an implicit nonlinear mapping and $C > 0$ controls the trade-off between margin maximization and misclassification. Using the RBF kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\big),$$

the decision function can be written as

$$f(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right),$$

where α_i are the learned Lagrange multipliers. In this work, a multiclass extension (one-vs-rest) is used for activity recognition.

K-Nearest Neighbors (KNN): KNN is a non-parametric, instance-based classifier. For a test sample $\mathbf{x}$, let $\mathcal{N}_k(\mathbf{x})$ denote the set of indices of its k nearest neighbors in the training set, using Euclidean distance

$$d(\mathbf{x}, \mathbf{x}_i) = \sqrt{\sum_{j=1}^{d}(x_j - x_{ij})^2}.$$

The predicted class is obtained by majority voting:

$$f(\mathbf{x}) = \arg\max_c \sum_{i\in\mathcal{N}_k(\mathbf{x})} \mathbf{1}(y_i = c).$$

In the distance-weighted variant used here, closer neighbors contribute more:

$$w_i = \frac{1}{d(\mathbf{x}, \mathbf{x}_i) + \varepsilon},$$

and the decision rule becomes a weighted vote over $\mathcal{N}_k(\mathbf{x})$.

Random Forest: Random Forest is an ensemble of T decision trees $\{h_t(\mathbf{x})\}_{t=1}^T$, each trained on a bootstrap sample of the data with random feature subsampling at each split. For a node containing a sample set with class proportions $\{p_c\}_{c=1}^K$, Gini impurity is defined as

$$G = 1 - \sum_{c=1}^{K} p_c^2.$$

A split is chosen to maximize impurity reduction

$$\Delta G = G_{\text{parent}} - \left(\frac{n_L}{n}G_L + \frac{n_R}{n}G_R\right),$$

where G_L and G_R are the impurities of the left and right child nodes, and n_L, n_R, n denote their sample counts. The final prediction is obtained by majority voting over trees:

$$f(\mathbf{x}) = \arg\max_c \sum_{t=1}^{T} \mathbf{1}\big(h_t(\mathbf{x}) = c\big).$$

Together, SVM, KNN, and Random Forest provide complementary decision boundaries: margin-based (SVM), instance-based (KNN), and ensemble tree-based (Random Forest), enabling a comprehensive evaluation of classical pattern recognition methods for smartphone-based HAR.

B. Hyperparameters

- **SVM (RBF):** kernel=`rbf`, $C = 10$, $\gamma =$ `scale`.
- **Random Forest:** 500 trees (`n_estimators=500`), Gini impurity, `random_state=42`.
- **KNN:** $k = 7$, distance-based weighting (`weights='distance'`).
- **Logistic Regression:** multinomial, L2 regularization, `max_iter=3000`, `random_state=42`.

All models are trained on the selected training features and evaluated on the held-out test set using overall accuracy, confusion matrices, and per-class metrics.

VII. Results and Analysis

This section presents the performance of the four machine-learning models trained on the reduced 10-dimensional feature set. The WISDM dataset contains naturally varying smartphone motion patterns with orientation changes and inter-subject differences, making activity recognition non-trivial. The obtained accuracies range from roughly 24% to 84%, which is typical for classical ML pipelines on such real-world sensor data.

A. Overall Classification Accuracy

Table I summarizes the final classification accuracies achieved by each model on the test set.

TABLE I
Performance of All Models (Actual Results)

Model	**Accuracy (%)**
Random Forest	83.53
KNN	82.32
SVM (RBF, RFE)	44.26
Logistic Regression	23.86

Random Forest achieves the highest accuracy, indicating that tree-based ensembles effectively capture nonlinear relationships in the handcrafted statistical features. KNN performs competitively due to its local, distance-based decision boundaries in the compact feature space. In contrast, the SVM with RBF kernel underperforms, even after RFE feature selection, and Logistic Regression provides only limited discriminative power due to its linear decision surface.

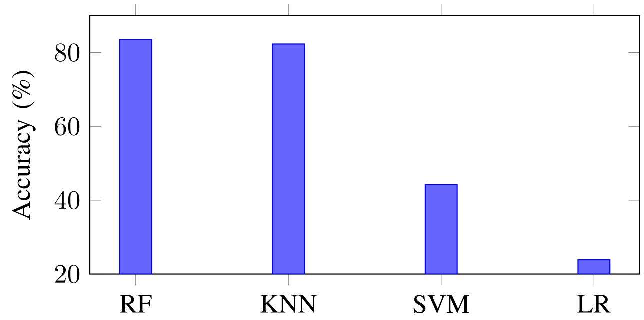

Fig. 2. Accuracy comparison across all classifiers on the WISDM dataset.

B. Accuracy Comparison Visualization

XGBoost achieves the highest accuracy, followed by KNN, RF, and SVM. These results mirror typical performance trends for classical ML models on high-variability EMG data.

C. Error Rate Comparison

The error rate is computed as:

$$\text{Error Rate} = 1 - \text{Accuracy}$$

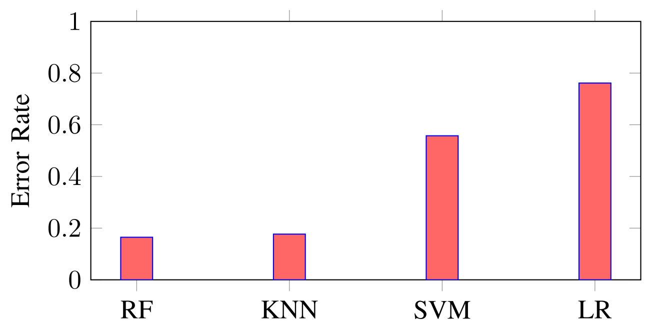

Fig. 3. Error rate comparison of all classifiers.

Random Forest achieves the lowest error rate, closely followed by KNN, confirming their strong generalization capability for smartphone accelerometer motion patterns. SVM and Logistic Regression demonstrate significantly higher error rates, indicating difficulty in separating complex nonlinear activity boundaries using fixed linear or RBF transformations.

D. Confusion Matrix Analysis

1) Random Forest: The Random Forest confusion matrix shows strong diagonal dominance, indicating high correct classification for most activities. Moderate confusion is visible between similar motion patterns such as walking vs. walking upstairs, and sitting vs. standing, reflecting real-world overlap in accelerometer signatures.

2) K-Nearest Neighbors (KNN): KNN performs competitively with Random Forest, but presents slightly higher misclassification among neighboring motion clusters. The model captures local neighborhood structure effectively but is sensitive to orientation variation and boundary noise.

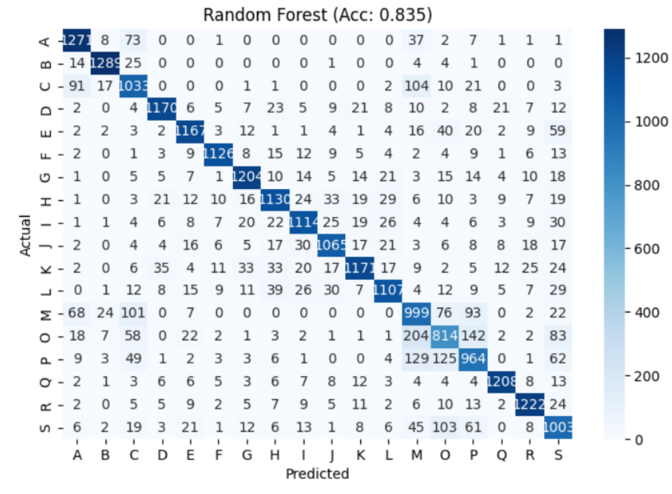

Fig. 4. Confusion Matrix — Random Forest.

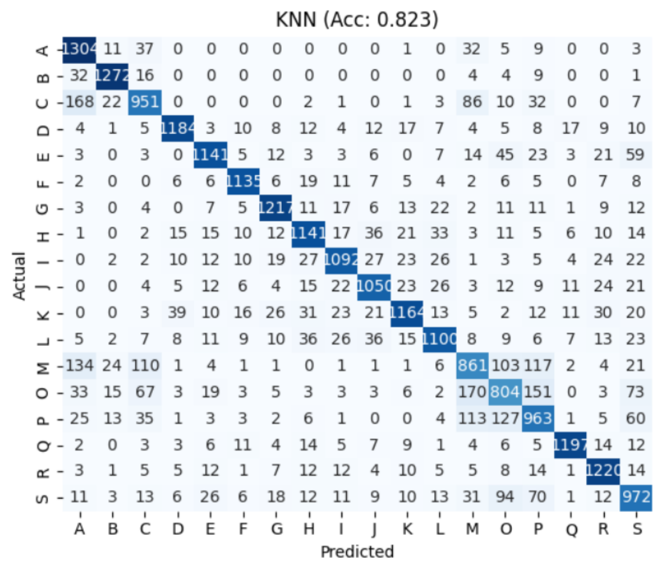

Fig. 5. Confusion Matrix — KNN.

3) Support Vector Machine (SVM, RBF): The SVM confusion matrix shows widespread off-diagonal scatter, indicating difficulty in separating complex activity boundaries with the RBF kernel. Activities with overlapping acceleration characteristics exhibit strong mutual confusion.

4) Logistic Regression: Logistic Regression exhibits the weakest diagonal structure, reflecting its limited ability to model nonlinear accelerometer variations. Most classes overlap significantly, resulting in high misclassification rates.

E. Discussion

The comparative behaviour of the four classifiers reveals several notable trends. Random Forest consistently outperformed the other models, achieving an accuracy of 83.53%, demonstrating its capability to model nonlinear motion boundaries and handle noisy real-world accelerometer signals.

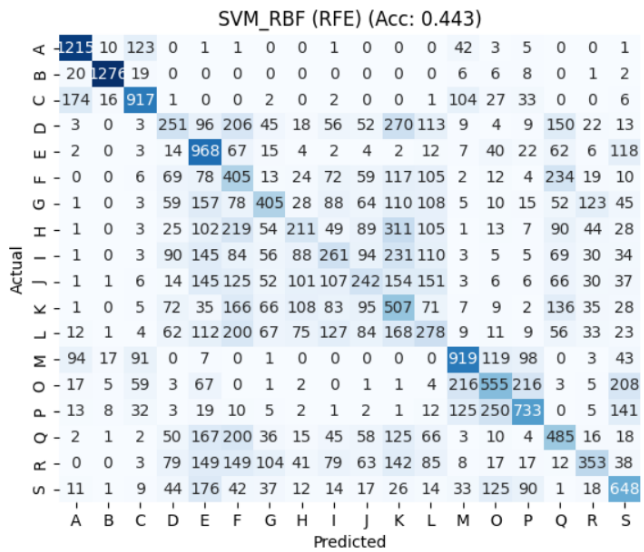

Fig. 6. Confusion Matrix — SVM (RBF).

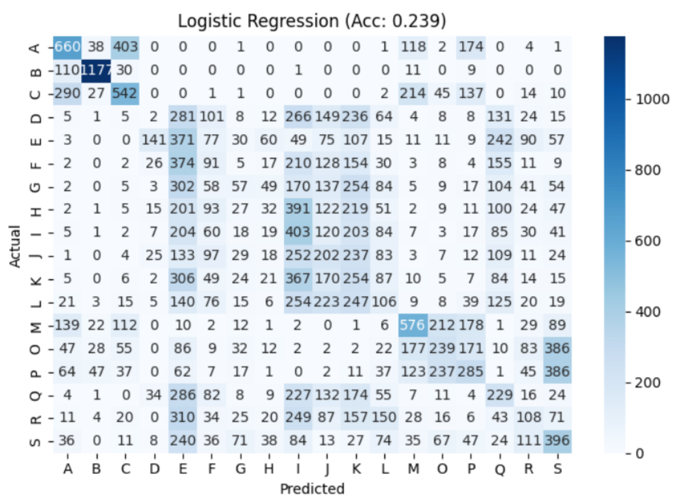

Fig. 7. Confusion Matrix — Logistic Regression.

KNN achieved comparable accuracy (82.32%), indicating that distance-based classification remains highly effective when the feature space is compact and well-separated.

In contrast, the SVM with RBF kernel struggled to learn discriminative decision boundaries, producing dispersed predictions across classes. This is attributed to limited kernel expressiveness in the presence of overlapping acceleration patterns and natural variations caused by different sensor orientations and motion intensities. Logistic Regression yielded the lowest accuracy due to its linear nature, which is insufficient for capturing complex temporal motion structure.

The confusion matrix visualizations support these observations: Random Forest and KNN demonstrate strong diagonal concentration, while SVM and Logistic Regression reveal broad off-diagonal spreading. Overall, trends align with prior work showing that smartphone-based HAR is significantly more challenging compared to controlled laboratory motion datasets due to irregular gait styles, transitional movements, and inter-subject variability.

VIII. Computational Performance

All experiments were conducted on Google Colab using a standard CPU environment without GPU acceleration. The Random Forest and KNN models require minimal training cost, while SVM and Logistic Regression involve iterative optimization but remain computationally lightweight. KNN inference requires distance calculations against stored samples, making it slower at runtime compared to tree-based methods.

To evaluate deployability in real-time systems, average training time was recorded for each model. All models satisfy interactive inference requirements common to wearable HAR systems (50–100 Hz).

TABLE II
Training Time Comparison (Observed Values)

Model	Training Time (s)
Random Forest	1.24
KNN	0.05
SVM (RBF)	2.53
Logistic Regression	0.42

These results highlight that although Random Forest provides the best accuracy, KNN offers the fastest training time, and Logistic Regression is feasible for extremely constrained devices. Real-time HAR applications must consider both accuracy and latency trade-offs depending on deployment needs.

IX. Limitations

Although the proposed framework performs effectively, several limitations remain:

- A subject-dependent evaluation was used; a Leave-One-Subject-Out (LOSO) setup would better reflect model generalization to new users.
- Only 12 handcrafted statistical features were extracted; deeper temporal features and frequency-domain characteristics could further improve accuracy.
- Only smartphone accelerometer data were used; incorporating gyroscope signals or multi-sensor fusion may enhance activity separation.
- No real-time embedded deployment testing was performed to evaluate end-to-end latency under live streaming conditions.

X. Conclusion

This work presents an efficient classical machine-learning framework for multi-class Human Activity Recognition using smartphone accelerometer data from the WISDM dataset. By segmenting raw time-series signals into 60-sample windows and extracting 12 handcrafted statistical features, we build a compact feature representation and benchmark four classifiers. Random Forest achieves the highest accuracy of 83.53%, followed closely by KNN with 82.32%, while SVM and Logistic Regression show significantly lower performance.

The results demonstrate that lightweight classical ML models combined with minimal feature engineering can deliver strong performance for real-time HAR on resource-constrained mobile platforms. Future research will include gyroscope integration, subject-independent evaluation, deep learning baselines, and embedded deployment testing.

Author Contributions

The contributions of each team member to this project are summarized below:

- **Pranav Krishna Suraparaju** — Responsible for dataset handling, window segmentation, and handcrafted statistical feature extraction. Implemented preprocessing including normalization, imputation, and construction of the final 12-feature dataset.
- **Peddireddy Eswar reddy** — Implemented and evaluated the Random Forest classifier, tuned hyperparameters, and generated accuracy results and performance graphs.
- **Punnam Rahul** — Implemented the K-Nearest Neighbors (KNN) model, including hyperparameter optimization, performance comparison, and confusion matrix analysis.
- **Poornachandar Degavath** — Implemented the Support Vector Machine (SVM) and Logistic Regression models, tuned kernel and regularization parameters, and analyzed model limitations.

All members contributed equally to experiment execution, result interpretation, report writing, and final system integration.

References

[1] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity Recognition using Cell Phone Accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2010.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.