Human Activity Recognition Using Machine Learning: A Comprehensive Analysis of UCI Smartphones Dataset

Manchen Wang
Department of Statistics
University of Michigan
Ann Arbor, MI 48104
noriawmc@umich.edu

Rofia Chen
Department of Statistics
University of Michigan
Ann Arbor, MI 48104
rofiaa@umich.edu

Yingxi Chen
Department of Statistics
University of Michigan
Ann Arbor, MI 48104
yingxich@umich.edu

Yiran Fang
Department of Statistics
University of Michigan
Ann Arbor, MI 48104
yrfang@umich.edu

1 Introduction

Human Activity Recognition (HAR) is the process of using data from sensors to identify and classify human activities. These activities can range from basic actions (like walking, sitting, or running) to complex behaviors (like cooking or exercising). Human Activity Recognition (HAR) is essential for applications such as healthcare monitoring, fitness tracking, and smart environments, enabling accurate classification of human activities from sensor data. This project leverages the Human Activity Recognition with Smartphones dataset from the UCI Repository, containing sensor readings from 30 participants performing six activities using a waist-mounted smartphone, resulting in 561 extracted features. We will evaluate various machine learning models, including K-Nearest Neighbors, Logistic Regression, Support Vector Machines, Random Forests, etc. Our project takes aggregate signal data from smartphones, including accelerometer and gyroscope readings, as input and predicts the specific human activity being performed, such as walking, sitting, or standing, as the output. Key challenges include class imbalances, potential noise in signals, dependency on device placement, and high-dimensional feature space, which require careful preprocessing and feature selection to ensure generalizability and efficiency.

2 Related Work

Several studies have focused on enhancing Human Activity Recognition (HAR) using different machine learning and deep learning techniques. Anguita et al. (2013) used a multiclass SVM with a One-Vs-All approach and Gaussian kernels to achieve 96% accuracy on smartphone sensor data [1], while Kusuma et al. (2022) improved this by incorporating hyperparameter tuning, reaching 96.26% accuracy on the UCI HAR dataset [2]. Qian et al. (2010) proposed a hierarchical multi-class SVM for video-based HAR, achieving over 90% accuracy [3]. Papaleonidas et al. (2021) applied variable sampling techniques to raw wearable device signals, achieving 99.9% accuracy across nine activities [4], and Kaushal et al. (2023) showed that random forests could reach 99.7% accuracy in a real-time HAR setup [5]. Privacy-preserving methods were explored by Iravantchi et al. (2021) with PrivacyMic, a system that uses inaudible frequencies for activity detection, maintaining 95% accuracy [6]. Shashi (2022) developed a deep learning model using LSTM on smartphone data, achieving 91% accuracy [7], while Das (2018) combined feature-engineered data with LSTMs for improved performance [8]. Reviews by Zhang et al. (2022) and Labellerr (2023) discussed the advancements and limitations in open HAR datasets, calling for better standardization and feature alignment [9, 10]. Open dataset challenges were further analyzed by a recent review, which highlighted the need for improved data quality and annotation [11]. These studies collectively demonstrate the significance of tuning, hybrid models, and addressing data challenges for robust HAR systems.

2.1 Strength and Weakness

Papaleonidas et al. (2021) achieved 99.9% accuracy using variable sampling techniques, excelling in capturing detailed patterns but relying heavily on feature engineering. In contrast, Shashi (2022) used LSTM models with 91% accuracy, leveraging temporal dependencies but requiring larger datasets and more computational power. Thinking about their strength and disadvantage, in order to achieve a higher accuracy, our approach balances performance and practicality by combining efficient machine learning with dimensionality reduction.

3 Data

3.1 Data Preprocessing

The dataset was created from experiments with 30 volunteers performing six activities while wearing a smartphone on their waist, which recorded triaxial accelerometer and gyroscope signals at 50 Hz. Signals were preprocessed with noise filtering and segmented into 2.56-second sliding windows (50% overlap), yielding 128 readings per window. Acceleration signals were decomposed into body and gravity components using a Butterworth low-pass filter (0.3 Hz cutoff), and 561 features were extracted from time and frequency domains. Data was normalized to [-1,1], labeled manually via video recordings, and split into training (70%) and test (30%) datasets. Using `pandas.read_csv()`, the feature datasets (X_train, X_test) and activity labels (y_train, y_test) were loaded, mapped to descriptive activity names, and combined into structured data frames for machine learning analysis and modeling.

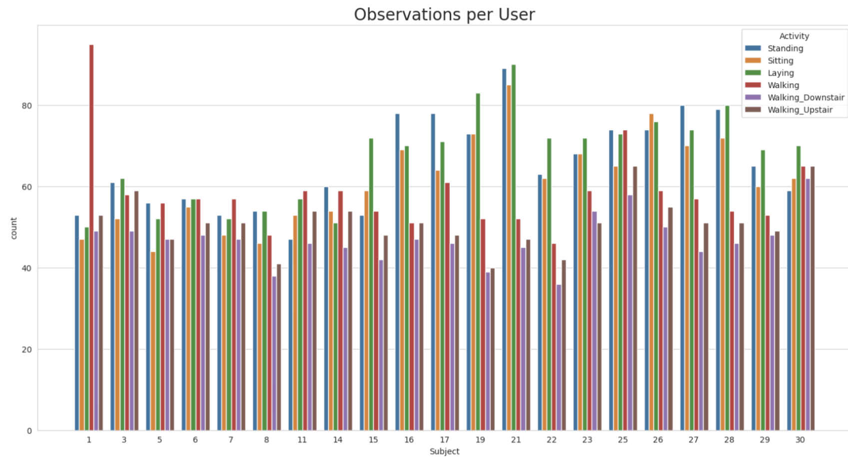
3.2 Exploratory Data Analysis (EDA)

3.2.1 Data Exploration

The dataset contains smartphone sensor data collected from 30 subjects performing six physical activities: WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, and LAYING. The dataset is divided into a training set with 7352 samples and a test set with 2947 samples, each represented by a 561-dimensional feature vector. These features are derived from raw accelerometer and gyroscope signals in both time and frequency domains, with normalization applied to the range [-1,1]. We observed that some columns have the same feature names but contain different values. This may be due to feature extraction processes generating features with similar descriptors across different axes or signal domains (e.g., time vs. frequency domain). These features are derived from the same raw signals but represent distinct transformations or statistical metrics, leading to differences in their values.

3.2.2 Data Cleaning

The dataset shows balanced activity label distribution across six classes, with varying sample contributions from subjects. Most features exhibit significant variance, capturing meaningful activity distinctions, though correlation analysis highlights redundancy in some time and frequency domain features. Such as "LAYING" having the highest count (1407 observations) and "WALKING DOWNSTAIRS" the lowest (986). A Gini coefficient of 0.0696 confirms minimal imbalance, ensuring the dataset's suitability for robust and unbiased machine learning models. This balance reduces subject-specific biases and supports accurate model generalization across diverse activities and subjects. However, the high feature dimensionality poses challenges, suggesting the need for dimensionality reduction techniques like PCA or feature selection. Overall, EDA confirms the dataset's richness and identifies redundancies and challenges, guiding model development.



PART 1: Feature-Based Analysis of Static and Dynamic Activities

Static activities (Standing, Sitting, Laying) exhibit consistently low energy values and minimal variability across all axes, reflecting limited motion and predictable gravitational acceleration with low entropy. In contrast, dynamic activities (Walking, Walking_Upstairs, Walking_Downstairs) show significantly higher energy levels and greater variability due to continuous and abrupt movements. As shown in Figure 1, Features like GravityAccMagentropy capture gravitational signal irregularity, effectively distinguishing static activities with low entropy from dynamic activities with higher entropy. Additionally, as shown in Figure 2, tBodyAccJerkenergyX, tBodyAccJerkenergyY, and tBodyAccJerkenergyZ highlight the intensity and variability of body motion along the axes, with stair-related activities showing the highest energy values, especially on the Z-axis, due to the vertical motion involved. These features, with significant discriminant weights, are crucial for separating static and dynamic motion patterns.

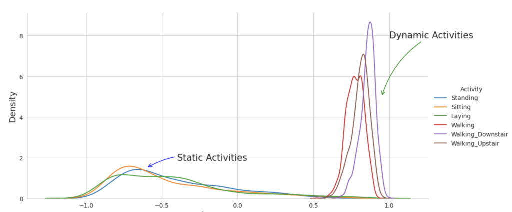


Figure 1: Density Plot of tGravityAccMagentropy Across Activities

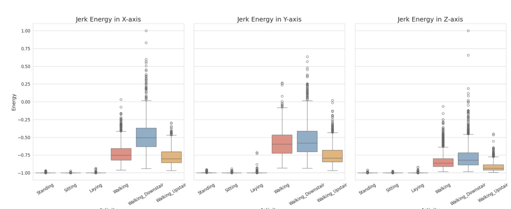
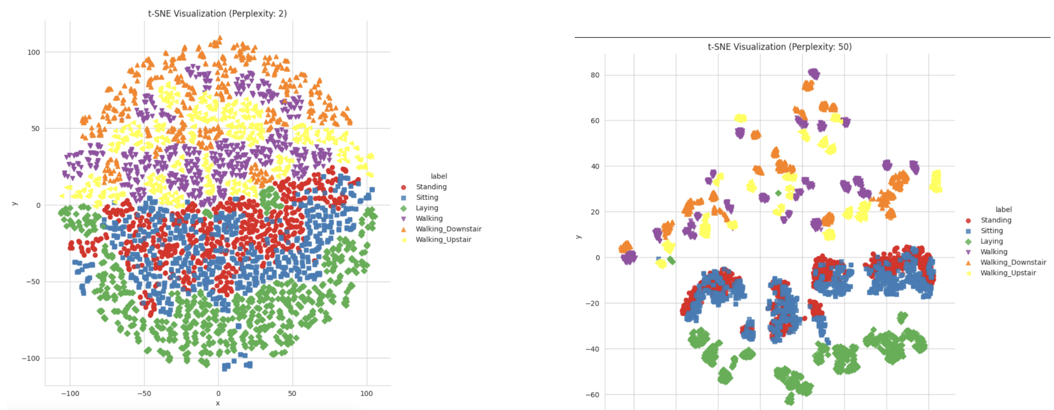


Figure 2: Comparison of Body Acceleration Jerk Energy Across Activities in X, Y, and Z Axes

PART 2: t-SNE Visualization

The t-SNE (t-Distributed Stochastic Neighbor Embedding) visualizations illustrate how high-dimensional data is reduced to two dimensions, making it easier to interpret patterns and clusters in the dataset. This method preserves local relationships between data points, allowing for an exploration of how the different activity labels (e.g., Standing, Sitting, Walking) cluster in the reduced space. By varying the perplexity parameter, which balances the focus between local and global data structure, we can observe how the clustering patterns shift. As shown in Figure 3(a), at very low perplexity values (e.g., 2), the visualization heavily emphasizes local relationships, resulting in tightly packed and distinct clusters. As the perplexity increases to 5 and 10, the clusters spread more naturally, with a better balance between local and global structures. With higher perplexity values, such as 20 and 50, the visualization shifts its focus towards capturing global relationships in the dataset, as shown in Figure 3(b).



(a) t-SNE Visualization (Perplexity: 2)

(b) t-SNE Visualization (Perplexity: 50)

Figure 3: Side-by-side comparison of t-SNE visualizations with different perplexity values.

4 Methods

4.1 Learning Pipeline

Our learning pipeline consists of data preprocessing, feature extraction, model training, and evaluation. The pipeline is designed to handle the given dataset's class imbalance while ensuring robust model generalization. We employed techniques such as oversampling the minority class and normalization for feature scaling.

4.2 Algorithm and Hyperparameter Selection

We implemented several widely-used algorithms for multi-class classification tasks, incorporating Linear Discriminant Analysis (LDA) for feature selection. Due to space constraints, we highlight only the algorithms that demonstrated exceptional performance here.

4.2.1 Logistic Regression

We used a **Logistic Regression model** as our primary classifier. Logistic regression is a statistical method that models the probability of a binary outcome based on one or more predictor variables. The model computes the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$, where $z = \mathbf{w}^T\mathbf{x} + b$. The weights $\mathbf{w}$ and bias b are learned during training by minimizing the binary cross-entropy loss:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right],$$

where N is the number of samples, y_i are the true labels, and $\hat{y}_i$ are the predicted probabilities.

To address the class imbalance, we used class-weighted loss functions, assigning higher weights to the minority class. The algorithm predicts the type of activity (e.g., sitting, standing, or walking) performed by a subject based on sensor data collected from smartphones or wearable devices. Logistic Regression employs a sigmoid function to estimate the probability of each activity class, selecting the class with the highest probability as the final prediction. Its interpretable nature allows researchers to understand the weight and impact of each feature, such as accelerometer or gyroscope readings, in determining the activity.

The Logistic Regression model was trained and evaluated using metrics such as accuracy, confusion matrix, and classification report to assess performance and computational efficiency. The confusion matrix highlighted the model's ability to distinguish similar activities, while the classification report detailed precision, recall, and F1-scores. Hyperparameters, including regularization strength (λ) and penalty type (L1 or L2), were optimized using grid search with 5-fold cross-validation. L1 promotes sparsity by reducing insignificant coefficients to zero, while L2 prevents overfitting by penalizing large coefficients. Regularization strength values ($\lambda \in \{0.01, 0.1, 1, 10\}$) were explored, with $\lambda = 0.1$

yielding the best performance. This systematic tuning balanced accuracy and generalizability, improving the model's capability for Human Activity Recognition tasks.

4.2.2 SVM

We employed a **Support Vector Machine (SVM)** classifier with a radial basis function (RBF) kernel. SVM works by finding the hyperplane that maximizes the margin between classes in a high-dimensional space. For the RBF kernel, the decision boundary is non-linear, calculated as:

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

where γ controls the influence of a single training example. The SVM optimization problem involves minimizing:

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

subject to:

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0,$$

where ξ_i are slack variables for handling misclassifications, C is the regularization parameter, and $\phi(\mathbf{x}_i)$ is the transformation induced by the RBF kernel.

Hyperparameters γ and C were selected using grid search with 5-fold cross-validation. We tested $\gamma \in \{0.01, 0.1, 1\}$ and $C \in \{1, 10, 100\}$, choosing $\gamma = 0.1$ and $C = 10$ as they provided the best validation accuracy. The kernel choice was motivated by its ability to model non-linear decision boundaries effectively.

4.2.3 k-Nearest Neighbors (kNN)

We used a **k-Nearest Neighbors (kNN)** classifier, a non-parametric algorithm that predicts the class label of a sample based on the majority class of its k-nearest neighbors. The neighbors are determined using the Euclidean distance:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{m}(x_{ik} - x_{jk})^2},$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are feature vectors in m-dimensional space. Features were normalized to have zero mean and unit variance using the StandardScaler to ensure fair distance computation.

Hyperparameters for the kNN classifier were tuned using grid search with 10-fold cross-validation. The primary hyperparameter, the number of neighbors (k), was explored over the range $k \in \{3, 5, 7, 9, 11\}$. The optimal value, $k = 5$, provided the best validation accuracy. Euclidean distance was used as the metric to measure similarity between data points. No additional weights were applied to neighbors in this implementation.

4.2.4 Quadratic Discriminant Analysis (QDA)

We used a **Quadratic Discriminant Analysis (QDA)** classifier for activity classification. QDA is a generative model that assumes each class follows a multivariate Gaussian distribution, characterized by its mean vector and class-specific covariance matrix. The quadratic decision boundary is determined by maximizing the posterior probability:

$$P(y = c|\mathbf{x}) \propto P(\mathbf{x}|y = c)P(y = c),$$

where $P(\mathbf{x}|y = c)$ is the class-conditional likelihood and $P(y = c)$ is the prior probability of class c. The likelihood is modeled as:

$$P(\mathbf{x}|y = c) = \frac{1}{(2\pi)^{d/2}|\Sigma_c|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_c)^T\Sigma_c^{-1}(\mathbf{x} - \mu_c)\right),$$

where μ_c and Σ_c are the mean vector and covariance matrix for class c, and d is the feature dimension.

The model computes separate covariance matrices for each class, allowing it to capture class-specific feature dependencies and resulting in quadratic decision boundaries. Features were normalized to zero mean and unit variance before training to ensure numerical stability.

QDA does not require traditional hyperparameter tuning as it is parameterized by the class-specific mean vectors and covariance matrices, estimated directly from the data. The priors for each class were automatically inferred from the training set distribution.

4.2.5 Linear Support Vector Classifier (LinearSVC)

We employed a **Linear Support Vector Classifier (LinearSVC)** for activity classification. LinearSVC is a discriminative model that finds the optimal hyperplane to separate classes in a high-dimensional feature space. The decision boundary is defined as:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b,$$

where $\mathbf{w}$ is the weight vector, $\mathbf{x}$ is the input feature vector, and b is the bias term. The classifier minimizes the hinge loss function, defined as:

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}\max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)) + \frac{\lambda}{2}\|\mathbf{w}\|^2,$$

where N is the number of training samples, y_i are the true labels (± 1), and λ is the regularization parameter.

Hyperparameters for the LinearSVC were optimized using grid search with 10-fold cross-validation. The primary hyperparameter, the regularization parameter C, was tested over the range $C \in \{0.125, 0.5, 1, 2, 8, 16\}$. The optimal value, $C = 1$, provided the best validation accuracy. To ensure numerical stability and consistent scaling during optimization, features were normalized to zero mean and unit variance. The convergence tolerance was set to 10^{-5}, and the maximum number of iterations was increased to 10,000 to handle potential convergence issues.

4.2.6 Stacking Classifier

The stacking classifier implemented in this study combines multiple machine learning models as base learners to enhance classification performance. The base models include Logistic Regression, which is configured with $C = 20$, l_2 regularization, and a maximum of 1000 iterations. A Support Vector Classifier (SVC) is used with a linear kernel, $C = 5$, and probability estimation enabled. The Linear Support Vector Classifier (LinearSVC) is configured with $C = 8$ and a maximum of 1000 iterations. The K-Nearest Neighbors (KNN) model is configured with 10 neighbors. Quadratic Discriminant Analysis (QDA) is used in its default configuration. Finally, the Random Forest Classifier is configured with 200 estimators, a maximum depth of 10.

The meta-model used in the stacking classifier is a Support Vector Classifier (SVC) with a linear kernel, $C = 5$, and probability estimation enabled. Cross-validation with 5 folds is applied during training to ensure robust model evaluation and prevent overfitting. The stacking classifier leverages the diverse strengths of its base models and combines their predictions to optimize classification accuracy.

5 Experiments and Results

5.1 Metrics and Evaluation

Since the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) and Precision-Recall Curve (AUPRC) are primary used in binary classification problems, we adopt the correlation matrix and accuracy score to evaluate the model performance result. The models were evaluated using overall accuracy, a normalized confusion matrix, and a standard confusion matrix. The evaluation aimed to assess classification performance across multiple human activity labels, with a focus on precision, recall, and class-wise prediction accuracy.

5.2 Results

KNN: The k-Nearest Neighbors (kNN) model, optimized with neighbors=5, achieved an impressive cross-validation score of 98.57% percents and an overall test accuracy of 96% . It performed exceptionally well across all activities, with F1-scores above 0.90, demonstrating strong precision and recall, though minor confusion persists between "Sitting" (recall: 85 percents) and "Standing" (precision: 88%) due to overlapping features. The normalized confusion matrix is shown in Figure 5.

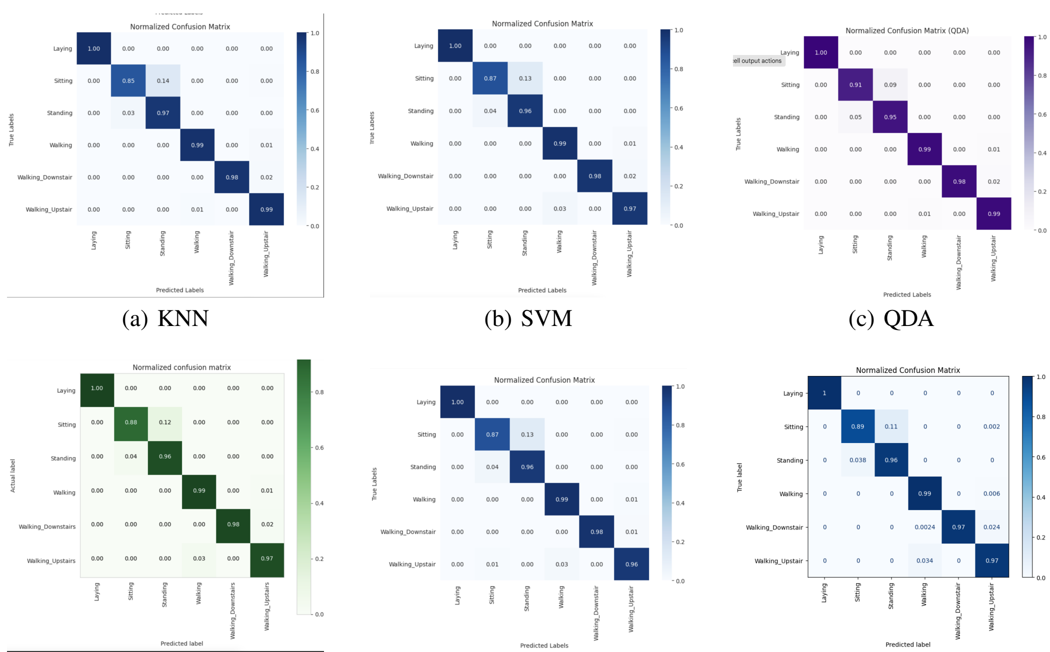
SVM: SVM model achieved an overall accuracy of 96.13%. The normalized confusion matrix is shown in Figure 5.

QDA: The QDA model achieved an overall accuracy of 96.91%, with high precision, recall, and F1-scores across all classes (macro and weighted averages of 0.97). The normalized confusion matrix is shown in Figure 5.

Logistic Regression: The logistic regression model, optimized with tune equals to 20 and L2 regularization, achieved an overall accuracy of 96% and a best cross-validation score of 98.38%, with macro and weighted average F1-scores of 0.96, demonstrating robust and reliable performance across all classes.The normalized confusion matrix is shown in Figure 5.

Linear SVC: The Linear SVC model, with the optimal tuning parameter `C=8`, achieved a cross-validation score of 98.56% and an overall test accuracy of 96%. It demonstrated strong performance across all activities, with F1-scores exceeding 0.90, though slight confusion remains between "Sitting" (recall: 87%) and "Standing" (precision: 89%).The normalized confusion matrix is shown in Figure 5.

Stack Classifier: The stacking classifier achieved a cross-validation mean accuracy of 98.14% and a test accuracy of 96.47%. Notable performance includes perfect precision, recall, and F1-score for "Laying," and high precision (97%) and recall (99%) for "Walking." Slight confusion was observed between "Sitting" (recall: 89%) and "Standing" (precision: 91%). The overall macro-averaged F1-score was 0.97, and the overall accuracy was 96.47%. The normalized confusion matrix is shown in Figure 5.



(a) KNN (b) SVM (c) QDA

(d) Logistic Regression (e) Linear SVC (f) Stacking Classifier

Figure 4: Normalized Confusion Matrix

Table 1: Accuracy of Different Models for Human Activity Recognition

Model	**Accuracy**
Logistic Regression	96%
k-Nearest Neighbors (kNN)	96%
Support Vector Machine (SVM)	96.13%
Quadratic Discriminant Analysis (QDA)	96.91%
Linear Support Vector Classifier (SVC)	96%
Stacking Classifier	96.47%

6 Conclusion and Discussion

6.1 Advantages and Limitations

In this study, we evaluated several machine learning algorithms for the task of Human Activity Recognition (HAR) using sensor data from smartphones and wearable devices. Our learning pipeline was carefully designed to address challenges such as class imbalance and feature scaling, employing techniques like oversampling and normalization to enhance model generalization.

Among the algorithms tested, Quadratic Discriminant Analysis (QDA) achieved the highest overall accuracy of **96.91%**, demonstrating robust performance with macro and weighted average F1-scores of **0.97**. QDA achieves the highest accuracy in human activity recognition due to its ability to model non-linear decision boundaries and capture class-specific feature variance effectively, which aligns well with the dataset's underlying patterns. The stacking classifier, which combined multiple base learners including Logistic Regression, Support Vector Machines, k-Nearest Neighbors, and Random Forests, also performed well with an accuracy of **96.47%** and a macro-averaged F1-score of **0.97**. Other models such as Logistic Regression, Linear Support Vector Classifier (LinearSVC), SVM with RBF kernel, and kNN showed comparable accuracies around **96%** with a weighted F-1 score or **0.96**, indicating that traditional classifiers can be highly effective for HAR tasks when appropriately tuned.

A consistent challenge across all models was the misclassification between the "Sitting" and "Standing" activities. This issue is evident in the confusion matrices (Figure 5) and reflected in the lower recall and precision scores for these classes. The overlapping feature space of these stationary activities makes them difficult to distinguish based solely on accelerometer and gyroscope readings. This suggests a need for more discriminative features or advanced modeling techniques that can capture subtle differences in sensor data.

The stacking classifier leveraged complementary model strengths, achieving perfect precision and recall for certain classes like "Laying," but with marginal overall improvement. This suggests the base models captured most relevant patterns, limiting the ensemble's added value.

The study highlights the effectiveness of traditional machine learning models in HAR, with high accuracy reflecting suitable features and preprocessing. However, confusion between certain activities points to sensor limitations or a need for advanced feature extraction. The focus on interpretable, efficient algorithms ensures suitability for real-time use on resource-constrained devices.

A key limitation is the risk of misuse if findings are not properly contextualized. Misapplication could lead to ethical violations, particularly in sensitive settings like healthcare or workplace monitoring.

6.2 Future Work

We will adopt a multivariant time series analysis approach (LSTM/RNN) using merely the raw data (Inertia signal) in the provided dataset and compare its results with the original dataset. By utilizing the raw inertial signals, such as accelerometer and gyroscope readings, these models can analyze the continuous flow of motion data and identify activity patterns that may not be fully captured by feature-engineered datasets. This approach also allows us to explore the extent to which raw data alone can provide sufficient discriminatory power, compared to datasets with pre-extracted features.

7 Contributions

Manchen Wang:

In this project, I made substantial and impactful contributions across multiple stages, showcasing technical expertise and a strong commitment to collaboration. My primary focus was on data processing, where I took the lead in writing the majority of the code for exploratory data analysis, data cleaning, and balancing checks to ensure the dataset's robustness for model development. I also implemented violin plots for data visualization, effectively analyzing sensor reading distributions by activity type, which added depth to the understanding of the dataset.

Additionally, I played a key role in documenting the results and data sections, where I presented critical insights into the data preparation process and conducted comparisons across different models to highlight performance variations. My contributions extended to interpreting key elements of the analysis, including data balancing, data cleaning, and t-SNE visualization, to uncover patterns across various activities. These insights were essential in guiding subsequent modeling decisions.

Furthermore, I built and fine-tuned logistic regression models, a cornerstone of the analysis, and authored the corresponding section to provide a comprehensive explanation of the methodology and findings. My detailed work ensured that the models were both accurate and interpretable, contributing significantly to the project's overall success.

Rofia Chen: In this Human Activity Recognition (HAR) project, I made significant contributions across multiple stages, including literature review, data pre-processing, feature analysis, and model development. I reviewed relevant studies to identify key methodologies and features commonly used in HAR research, establishing a strong foundation for the project. During the data pre-processing phase, I normalized the features and identified critical issues, such as multiple columns with different values but identical feature names, ensuring the dataset was clean and ready for analysis.

I conducted in-depth feature analysis to distinguish between static and dynamic activities, focusing on key features such as gravitational entropy and body acceleration jerk energy. Using visualizations like Kernel Density Estimation (KDE) plots and annotated boxplots, I highlighted distinct motion patterns across activities, facilitating a deeper understanding of the data. Significant features, such as `tGravityAccMagEntropy` and `tBodyAccJerkEnergyX`, `tBodyAccJerkEnergyY`, and `tBodyAccJerkEnergyZ`, were identified and prioritized based on their importance through Linear Discriminant Analysis (LDA) coefficients.

Dimensionality reduction techniques, including LDA and Principal Component Analysis (PCA), were applied to transform the feature space, improving both interpretability and model performance. I also developed and fine-tuned machine learning models, such as k-Nearest Neighbors (kNN), Quadratic Discriminant Analysis (QDA), and LinearSVC, optimizing their performance using GridSearchCV and cross-validation.

Additionally, I contributed to the research, design, and tuning of a Stacking Classifier, ensuring its effective implementation. My work throughout the project provided critical insights into the relationship between sensor data and physical activities, enhancing both the accuracy and interpretability of the models.

Yiran Fang In this project, I contributed significantly across various stages, beginning with data handling and exploratory analysis. I facilitated the loading and organization of the dataset, ensuring its readiness for preprocessing and modeling. By attending office hours and asking clarification questions, I helped address uncertainties and enhanced the team's understanding of the project requirements and dataset intricacies.

I conducted some literature review on Human Activity Recognition and machine learning techniques, providing a solid theoretical foundation that informed our methodological choices. Building on these insights, I developed and optimized the Random Forest model, which was unfortunately abandoned by us due to its low performance, utilizing `GridSearchCV` and cross-validation to fine-tune hyperparameters and maximize performance.

I played a key role in constructing the Stacking Classifier from scratch, carefully selecting and tuning base models for effective integration within the ensemble. This involved extensive experimentation and iterative adjustments to improve the classifier's efficacy. Additionally, I collaborated with the team on feature selection and contributing to discussions that identified the most impactful features for our models.

Beyond technical contributions, I synthesized our findings into clear, concise deliverables. I authored the data preprocessing section, co-wrote the conclusion and summary sections, and meticulously proofread the entire report to ensure coherence, clarity, and academic rigor.

Yingxi Chen In this project, I contributed by combining and synthesizing works for the related work and data sections. I reviewed multiple research papers to compare different approaches and model performances, summarizing key findings and their relevance to our study. For the data section, I analyzed the preprocessing methods used by previous researchers, fetched the dataset, and conducted exploratory data analysis to understand its structure and characteristics. I also developed an approach for dimensionality reduction by utilizing t-SNE to visualize and differentiate between various events in the dataset, providing insights into class separability and feature patterns.

For the modeling phase, I focused on fine-tuning the SVM model, optimizing hyperparameters for improved performance. Additionally, I explored the use of PCA for dimensionality reduction as a data transformation and feature selection technique, evaluating its impact on model performance.

Finally, I completed the write-up for the conclusion and discussion sections, summarizing the findings, highlighting the strengths and limitations of our methods, and proposing directions for future work.

References

[1] Jorge-L. Reyes-Ortiz, Luca Oneto, Alessandra Samà, Xavier Parra, and Davide Anguita. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing*, 171: 754–767, 2016. doi: https://doi.org/10.1016/j.neucom.2015.07.085.

[2] Wahyu Andhyka Kusuma, Agus Eko Minarno, and N. Dwi. Human Activity Recognition Utilizing SVM Algorithm with GridSearch. *Nucleation and Atmospheric Aerosols*, 2022. doi: https://doi.org/10.1063/5.0096708.

[3] H. Qian, Y. Mao, W. Xiang, and Z. Wang. Recognition of Human Activities Using SVM Multi-Class Classifier. *Pattern Recognition Letters*, 31(2):100–111, 2010. doi: https://doi.org/10.1016/j.patrec.2009.09.019.

[4] A. Papaleonidas, A. P. Psathas, and L. Iliadis. High Accuracy Human Activity Recognition Using Machine Learning and Wearable Devices' Raw Signals. *Journal of Information and Telecommunication*, pages 1–17, 2021. doi: https://doi.org/10.1080/24751839.2021.1987706.

[5] P. Kaushal, V. Gandhi, and Jasmeen Kaur Chahal. Human Activity Recognition with Machine Learning. *AIP Conference Proceedings*, 2916:030002–030002, 2023. doi: https://doi.org/10.1063/5.0178130.

[6] Yasha Iravantchi, K. Ahuja, Mayank Goel, Chris Harrison, and Alanson Sample. PrivacyMic: Utilizing Inaudible Frequencies for Privacy Preserving Daily Activity Recognition. *ACM Digital Library*, 2021. doi: https://doi.org/10.1145/3411764.3445169.

[7] Madhav Shashi. Human Activity Recognition Using Smartphones Sensor DataSet, 2021. URL `https://github.com/MadhavShashi/Human-Activity-Recognition-Using-Smartphones-Sensor-DataSet`.

[8] srvds (Sourav Das). GitHub - srvds/Human-Activity-Recognition: Predicts the human activities based on accelerometer and gyroscope data of Smart phones, 2018. URL `https://github.com/srvds/Human-Activity-Recognition`.

[9] S. Zhang, X. Zhang, M. Almahayan, S. Aloufi, M. Atallah, and K. El-Khatib. Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances. *Sensors*, 22(4): 1476, 2022. doi: https://doi.org/10.3390/s22041476.

[10] Labellerr. Human Activity Recognition (HAR): Fundamentals, Models, Datasets, 2023. URL `https://www.labellerr.com/blog/human-activity-recognition/`.

[11] Open Data Sets in Human Activity Recognition Research - Issues and Challenges: A Review, 2023. URL `https://pure.ulster.ac.uk/ws/portalfiles/portal/128003184/Open_Data_Sets_in_Human_Activity_Recognition_Research_-_Issues_and_Challenges_A_Review.pdf`. Available online (accessed: 10.11.2024).

A Appendix / supplemental material

Github Repository: https://github.com/YingxiC/stats415-final-project