Human Activity Recognition and Context Prediction using the

Extrasensory Dataset

CS256 Project Report

By

Naga Rohan Kumar Bayya

016980275

## Contents

[**Ⅰ. Dataset Overview 3**](#_ty3er9ml7fui)

[1. Data Collection Process: 3](#_kgy1hq6kcmou)

[2. Users and Devices: 3](#_afk1efgtnkl9)

[3. Sensors 3](#_btd7yfmdvl9f)

[4. Activities/Context Labels: 4](#_q1dcbzrufi0r)

[5. Challenges: 4](#_wblwjjgjcr2h)

[**Ⅱ. Data Preprocessing 5**](#_f44pfqo267kj)

[**Ⅲ. Feature Engineering 6**](#_t9k50ncl282w)

[Dimensionality Reduction using Correlation Analysis: 7](#_4i3r66keplo2)

[**Ⅳ. Model Development 8**](#_cx0643ph45gu)

[1. Model Selection: 8](#_q437n3hjzxg8)

[2. Cross-Validation: 8](#_eq8bbh3qkh75)

[3. Model Evaluation Metrics: 8](#_8dv3w3xd8yc)

[**Ⅴ. Advanced Model Exploration 9**](#_te36p54h7s7j)

[**Ⅵ. Evaluation 10**](#_zdm2lhag6kmz)

[1. Model Evaluation Function: 10](#_pac3v29sgh03)

[2. Cross Validation: 10](#_j2qbmuzaibmj)

[3. Model Performance: 10](#_njtamrv42479)

[4. Model Comparison: 10](#_xk30plewp7hk)

[5. Discussion of Performance: 10](#_221yc117gwwc)

[**Ⅶ. Conclusion 12**](#_48k2wxr3rzvx)

[**Ⅷ. Appendix 13**](#_5ml0ac1lda1i)

# Ⅰ. Dataset Overview

The ExtraSensory dataset is a rich resource for behavioral context recognition in-the-wild, collected from mobile sensors including smartphones (iPhone/Android) and smartwatches. Here's a detailed exploration of its structure:

## Data Collection Process:

Data was collected through the ExtraSensory mobile application, which performed a 20-second recording session automatically every minute. The application collected measurements from various sensors including accelerometer, gyroscope, magnetometer, location services, audio, watch accelerometer, watch compass, phone state indicators, and additional sensors like light, air pressure, humidity, and temperature. The sensors provided measurements at different frequencies and durations, with some sampled once a minute and others sampled at higher frequencies (e.g., 40Hz for accelerometer, gyroscope, and magnetometer).

## Users and Devices:

The dataset includes data from 60 users, mainly students and research assistants from the UCSD campus, with diverse ethnic backgrounds. Users had a variety of phone devices, including different iPhone generations (4, 4S, 5, 5S, 5C, 6, and 6S) and Android devices from various manufacturers. There were differences in sensor availability among devices, with some sensors not present in certain phones (e.g., iPhones lacking an air pressure sensor).

## Sensors

The dataset comprises measurements from heterogeneous sensors, including motion-reactive sensors (accelerometer, gyroscope, magnetometer), location services, audio, watch accelerometer, watch compass, and additional sensors. Sensor measurements were recorded at different frequencies and provided diverse types of data, such as tri-axial direction and magnitude of acceleration, rotation rates, magnetic field information, audio features, and location coordinates.

## Activities/Context Labels:

The dataset includes both cleaned labels and original self-reported labels from users.

Cleaned labels consist of context labels derived from user self-reports, covering a wide range of activities and contexts (e.g., indoors, sitting, walking, sleeping, computer work).

Original labels include main activities (e.g., lying down, sitting, standing) and secondary activities (e.g., sports, transportation, basic needs, company, location).

Some challenges with labels include class imbalance, missing values, and the need for cleaning and processing to ensure reliability.

## Challenges:

* Class Imbalance: The dataset exhibits class imbalance, with some activities having significantly more examples than others, which can affect the performance of machine learning models.
* Missing Values: Due to various reasons such as sensor unavailability or user behavior, some examples may have missing sensor measurements or context labels, requiring strategies for handling missing data during analysis.
* Diverse Data Modalities: The dataset contains data from diverse sensor modalities, each with its own data format and characteristics, posing challenges for feature extraction, fusion, and model development.
* Handling data from different sensors, each providing different types of information (e.g., motion, location, audio), requires careful consideration of feature engineering and model architecture.

# Ⅱ. Data Preprocessing

In the data preprocessing phase, I addressed several key aspects to ensure the quality and usability of our dataset. Firstly, I tackled the issue of missing values. Identifying missing values was crucial, so I meticulously examined our dataset to locate any instances of missing data. Once identified, I employed various imputation techniques to fill in these missing values, ensuring that our dataset remained as complete as possible. Additionally, I implemented advanced techniques to handle missing values in cases where simple imputation was not feasible. Throughout this process, I also flagged missing values to keep track of the changes made to the dataset.

Next, I focused on reducing noise in our data. Noise reduction was essential to improve the accuracy of our analysis. I applied various smoothing techniques and frequency filtering methods to minimize noise interference in our dataset. Additionally, I developed outlier detection algorithms to identify and eliminate noisy data points that could skew our results.

Normalization was another critical step in our data preprocessing pipeline. I employed normalization techniques to scale our data and bring it within a standardized range. This ensured that features with different scales did not disproportionately influence our analysis. Techniques such as min-max scaling, standardization, and robust scaling were applied to achieve this normalization.

In terms of data segmentation strategies, I devised methods to segment our time-series data effectively. I implemented fixed-length segments to divide our data into consistent intervals, facilitating analysis across different time periods. Additionally, I incorporated overlap between segments to ensure continuity and capture nuanced changes in the data. Determining segmentation criteria and applying window functions were crucial steps in this process, allowing us to extract relevant features from each segment accurately.

# Ⅲ. Feature Engineering

Time-domain features are directly extracted from the time-series sensor data. These features capture the characteristics of the data over time. The time-domain columns present in the dataset include:

1. raw\_acc:magnitude\_stats:mean
2. raw\_acc:magnitude\_stats:std
3. raw\_acc:3d:mean\_x
4. raw\_acc:3d:mean\_y
5. raw\_acc:3d:mean\_z
6. raw\_acc:3d:std\_x
7. raw\_acc:3d:std\_y
8. raw\_acc:3d:std\_z
9. watch\_acceleration:magnitude\_stats:mean
10. watch\_acceleration:magnitude\_stats:std
11. watch\_acceleration:3d:mean\_x
12. watch\_acceleration:3d:mean\_y
13. watch\_acceleration:3d:mean\_z
14. watch\_acceleration:3d:std\_x
15. watch\_acceleration:3d:std\_y
16. watch\_acceleration:3d:std\_z
17. proc\_gyro:magnitude\_stats:mean
18. proc\_gyro:magnitude\_stats:std
19. proc\_gyro:3d:mean\_x
20. proc\_gyro:3d:mean\_y
21. proc\_gyro:3d:mean\_z
22. proc\_gyro:3d:std\_x
23. proc\_gyro:3d:std\_y
24. proc\_gyro:3d:std\_z
25. raw\_magnet:magnitude\_stats:mean
26. raw\_magnet:magnitude\_stats:std

Frequency-domain features capture the periodic patterns and frequency components of the data. The frequency-domain columns present in the dataset include:

raw\_acc:magnitude\_spectrum:log\_energy\_band0

1. raw\_acc:magnitude\_spectrum:log\_energy\_band4
2. raw\_acc:magnitude\_spectrum:spectral\_entropy
3. proc\_gyro:magnitude\_spectrum:log\_energy\_band0
4. proc\_gyro:magnitude\_spectrum:log\_energy\_band4
5. proc\_gyro:magnitude\_spectrum:spectral\_entropy
6. raw\_magnet:magnitude\_spectrum:log\_energy\_band0
7. raw\_magnet:magnitude\_spectrum:log\_energy\_band4
8. raw\_magnet:magnitude\_spectrum:spectral\_entropy
9. watch\_acceleration:magnitude\_spectrum:log\_energy\_band0
10. watch\_acceleration:magnitude\_spectrum:log\_energy\_band4
11. watch\_acceleration:magnitude\_spectrum:spectral\_entropy
12. watch\_acceleration:spectrum:x\_log\_energy\_band0

## Dimensionality Reduction using Correlation Analysis:

Features that are highly correlated are deduplicated to prevent redundancy and multicollinearity, which can negatively impact the performance of machine learning models.

# 

# Ⅳ. Model Development

## Model Selection:

Different machine learning models are explored and compared. I implemented several models, including Decision Tree Classifier, Random Forest, Support Vector Machines (SVM), as well as deep learning models such as RNN (Recurrent Neural Network) and LSTM (Long Short-Term Memory). Each of these models has its strengths and weaknesses, and they are suitable for different types of data and tasks.

## Cross-Validation:

To evaluate model performance robustly, I implement cross-validation using TimeSeriesSplit. Cross-validation is essential for assessing how well a model generalizes to new data and helps in detecting overfitting. By splitting the data into multiple folds and training the model on different subsets while testing on others, cross-validation provides a more accurate estimate of the model's performance.

## Model Evaluation Metrics:

I evaluated each model's performance using common metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into different aspects of model performance, such as the proportion of correctly classified instances (accuracy), the ability to correctly identify positive cases (recall), and the balance between precision and recall (F1-score).

# Ⅴ. Advanced Model Exploration

Advanced model exploration is conducted through the use of sequence models like LSTM and ensemble models like Random Forest classifiers to improve prediction accuracy.

1. Sequence Models (LSTM):

* The code defines a function run\_lstm\_model specifically for running an LSTM model.
* Inside this function, a Sequential model is created using Keras (from TensorFlow). An LSTM layer is added to the model to capture temporal dependencies in the activities.
* The model is compiled with appropriate loss function, optimizer, and metrics.
* During training, the model is fit to the data and trained for a certain number of epochs. Early stopping is used to prevent overfitting.
* The model's performance is evaluated using evaluation metrics such as accuracy, precision, recall, and F1-score.

1. Ensemble Models (Random Forest):

* The code defines a function run\_random\_forest\_model for running a random forest classifier.
* Setting the class\_weight to 'balanced' helps address class imbalance by adjusting weights inversely proportional to class frequencies.
* min\_samples\_split determines the minimum number of samples required to split an internal node, set to 10, which can prevent overfitting by limiting splits

# Ⅵ. Evaluation

Evaluation is performed using a series of metrics including accuracy, precision, recall, and F1-score. Let's break down how evaluation is done and discuss the performance of different models and feature sets:

## 1. Model Evaluation Function:

Evaluation is done using a function called `evaluate\_model`, which calculates the accuracy, precision, recall, and F1-score for a given model and test data. This function likely compares the model's predictions with the actual labels and computes these metrics based on the results.

## 2. Cross Validation:

Cross-validation is employed using TimeSeriesSplit. This method splits the dataset into multiple folds while preserving the temporal order of the data. Each fold is used as a test set once while the rest are used for training. This ensures that the model is evaluated on unseen data.

## 3. Model Performance:

Different models, including Random Forest, SVM, RNN, and LSTM, are trained and evaluated using the cross-validation approach. For each model and feature set, the average accuracy, precision, recall, and F1-score are calculated across the cross-validation folds.

## 4. Model Comparison:

After evaluating each model using cross-validation, the performance metrics (accuracy, precision, recall, and F1-score) are aggregated and plotted for comparison. This allows for a visual comparison of how different models perform on the given task and which feature sets lead to better performance.

## 5. Discussion of Performance:

The performance of different models and feature sets can be discussed based on the aggregated metrics. For example:

- A model with higher accuracy indicates better overall performance in classifying activities.

- Precision measures the ratio of correctly predicted positive observations to the total predicted positives, which is important if false positives are costly.

- Recall measures the ratio of correctly predicted positive observations to the all observations in actual class, which is important if false negatives are costly.

- F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics.

- By comparing these metrics across different models and feature sets, one can identify which combination yields the best performance for the given task.

# Ⅶ. Conclusion

In conclusion, the analysis of the provided dataset suggests that LSTM outperformed RNN in terms of classification performance. This observation can be attributed to LSTM's ability to retain and utilize information from previous time steps more effectively compared to traditional RNN architectures. The superior performance of LSTM highlights the importance of memory retention in capturing temporal dependencies within the data.

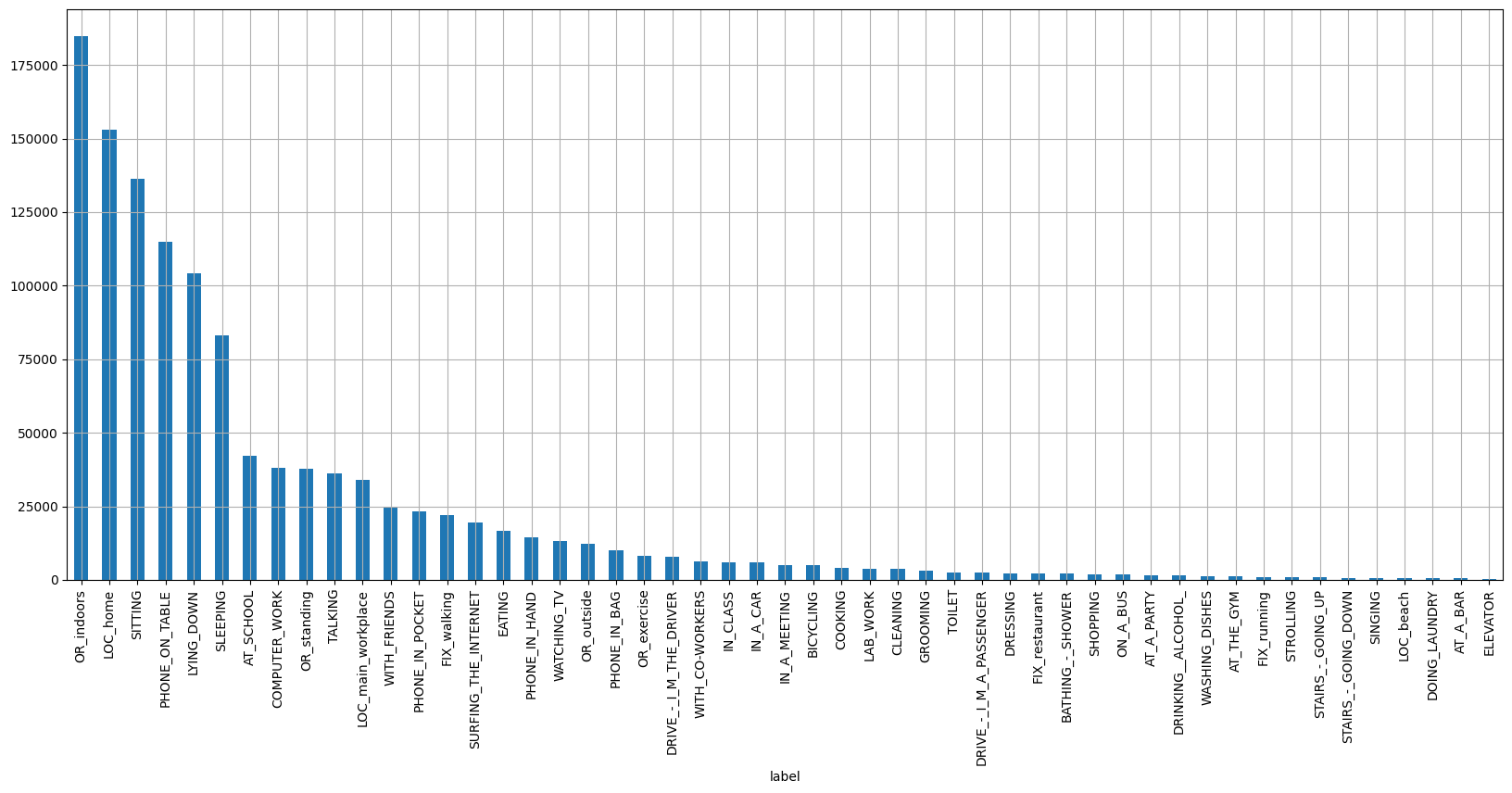
Secondly, few primary labels like Running had a very few data points and resulted in very skewed and inaccurate results.

Moreover, machine learning models such as SVM and ensemble models like random forest classifiers also exhibited good performance in the classification task. This suggests that while deep learning models like LSTM offer advantages in handling sequential data, traditional machine learning approaches remain competitive and effective for certain tasks.

Potential improvements and future research directions for the ExtraSensory dataset include exploring advanced sensor fusion techniques, multimodal learning approaches, temporal modeling with LSTM or Transformer-based architectures, transfer learning and domain adaptation strategies

# Ⅷ. Appendix

Number of examples for every context label



1. Performance metrics:  
   