Human Activity Recognition Dataset - A Comparative Study
STAT 8056: Statistical Learning and Data Mining **-** Final Report
Team Members: Shriya Rai (RAI00016) **|** Rachit Jas (JAS00001)

Introduction

Problem Statement

The goal of this project is to extract meaningful features and use them to predict human activity (such as walking, sitting, etc.) from garnered smartphone actigraphy time-series data using a slew of popular machine learning algorithms. Based on empirical results, this comparative study also attempts to discover the most suitable feature extractor and machine learning algorithm for the actigraphy dataset. The importance of human activity recognition from actigraphy data is motivated and extended by active medical research such as in the efficient detection of sleep-related disorders.

Data Overview

The Human Activity Recognition dataset available on Kaggle, was chosen for the predictive task. Multivariate activity-series sensor data of 30 volunteers, (each belonging to the age bracket of 19-48 years), was collected using a Samsung SII phone equipped with an accelerometer and gyroscope. The key characteristics of the dataset have been summarized in the table given below:

Data Attribute	Attribute Description/Value
Total Number of Samples	7352
Total Number of Features	561
Total Number of Classes	6 {Walking, Standing, Sitting, Laying, Walking Downstairs, Walking Upstairs}
Dataset Size	25 MB
Number of Volunteers Who Participated	30
Train-Test Split	70%-30% {21 volunteers' series data + 9 volunteers' series data}
Mean / Median Total Time Steps Recorded for Volunteer	350, 347
Mean / Median Activity Sequence Length	26, 27

Challenges to be Addressed

This actigraphy dataset is quite challenging to work with due to its following properties:

1. The total activity data collected for each volunteer is not constant, as evident from Figure 1.

2. The sequence length (i.e., consecutive number of time steps recorded for a particular activity for a specific volunteer) for each activity throughout the dataset is not constant, giving rise to the **Activity Sequence Length Problem**.

3. Each volunteers' worth of data comprises an array of multivariate time series data corresponding to an activity. Being a multiclass prediction problem, further complexity is introduced.

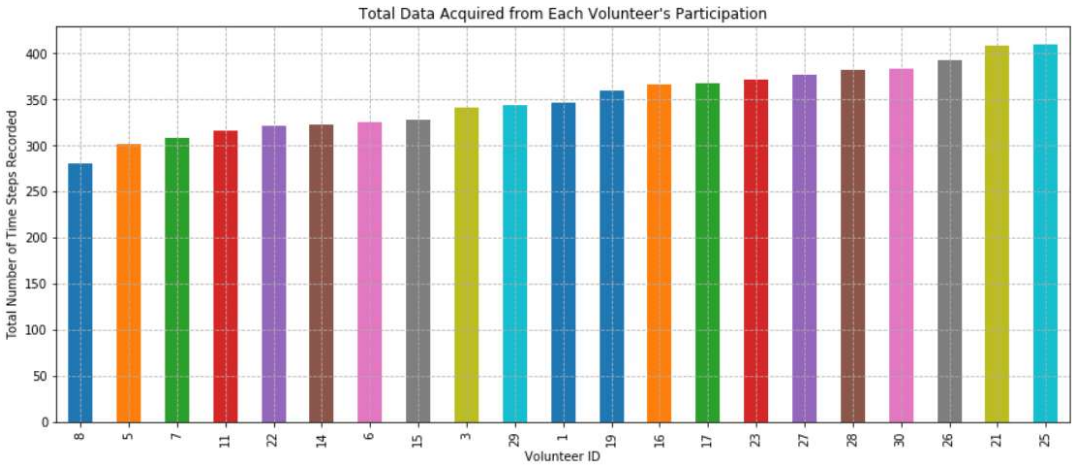
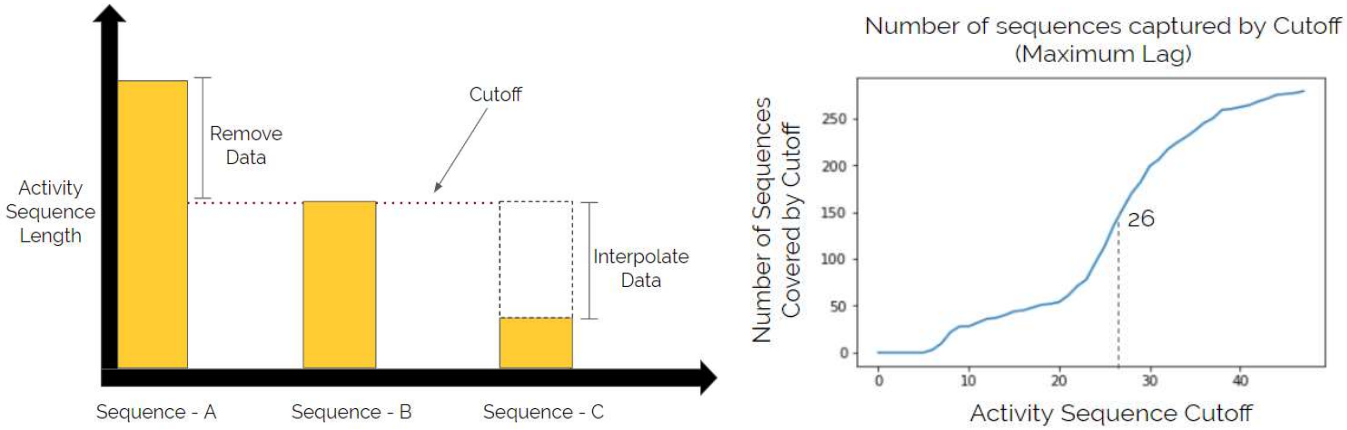


Figure 1: Total Activity Series Data for Each Volunteer

Methods Utilized

Data Cleaning and Transformations

About 80% of the project time was consumed in preparing the activity series data for predictive modeling, especially to solve the Activity Sequence Length Problem mentioned earlier. This problem can be visualized more clearly in Figure 2(a).



(a) (b)

Figure 2: (a) Activity Sequence Length Problem, (b) Search for Suitable Activity Sequence Cutoff

Four data transformations were applied on the original dataset to achieve a total of 280 activity sequences. each sequence being an input vector of 14586 dimensions (cutoff length x 561). These transformations can be summarized as follows:

1. **Data Standardization** is applied to ensure that the numerical sensor data is scaled appropriately.
2. Additional series **Data Removal** is done to conform the activity sequences to the cutoff length.
3. Missing series **Data Interpolation** is done using order-3 splines to conform data to cutoff length. Splines of order 2, 3 and 4 were applied to the data and analyzed separately, wherein it was found that order-3 spline gave the most consistent and representative results for data interpolation.
4. **Data Flattening** is performed to incorporate the time lags of each feature for the prediction task.

Data Loss - Interpolation Error Tradeoff

Unfortunately, there is always a tradeoff between data loss (by removing the observed series data) and interpolation error (introduced by order-3 spline interpolation). Concretely, if the activity sequence cutoff is very high, then the data loss is minimized as more activity sequences are incorporated for prediction, as visible in Figure 2(b), but the interpolation error is expected increase sharply since more time steps' worth of data would need to be interpolated. On the other hand, if the activity sequence cutoff is very low, then the interpolation error is minimized since most of the activity sequences would touch/surpass the cutoff length thereby decreasing the need for interpolation, however the high data loss is expected since the probability of sequences surpassing the small cutoff length is high, resulting in drastic removal of useful data for prediction purposes.

To solve the activity sequence problem, a cutoff sequence length of 26 was found to be adequate, which was inferred from the central measures of tendency (i.e., mean / median activity sequence length), in order to achieve favourable activity sequences of constant length.

Predictive Models with Feature Extractor/Selector

As the data is high dimensional, 4 different methods to extract or select important features were employed - PCA based extractor, Tree based selector, Variance Threshold based selector and Univariate selector. Recursive Feature Elimination was not utilised as one of the methods as it would have had high computation overhead due to high dimensional data.

PCA based extractor returned the principal components which had the largest 100 eigenvalues. Tree based selector utilised random forest to compute importance of the features and returned top 100 features which had the highest information gain. Univariate Selector selected the top 100 features which have the strongest relationship with the output variable based on the ANOVA F-value. Variance Threshold based selector is the simplest approach which returned features which had variance greater than or equal to .85.

These methods were coupled with 6 different machine learning algorithms to perform a comparative study - Penalised Logistic Regression, Logistic Regression, SVM, Xgboost, Random Forest, kNN. These models were chosen to have a holistic set of algorithms in order to perform the comparative study. Penalty function used in Penalised Logistic Regression models was L2 norm. The number of tree estimators in Random Forest models were decided by utilising OOTB error elbow plot. KNN models predicted the class label by making use of the 20 nearest neighbors. SVM models were trained by utilised radial kernel. Logistic Regression and Xgboost models were trained with their default parameters.

Standalone Models with Automated Feature Extraction

Due to their inherent nature of automating feature extraction, the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are chosen for this experiment. The Cross Entropy loss function and Adam optimizer were utilized for training each of these models. The proposed architecture of CNN and RNN for the predictive task is summarized in the table given below:

Model	Layers	Architecture	Details
CNN	6 + 1	Conv/Max → Dropout → Conv/Max → Dropout → Fully Connected → Fully Connected → Softmax	Layers = 7; Kernel Size = 5 Stride = 1; Padding = 0 Pool Size = 2

RNN	4 * 1	RNN Unit → RNN Unit → RNN Unit → RNN Unit → Softmax	Hidden Dimensions = 100 Output Dimensions = 6

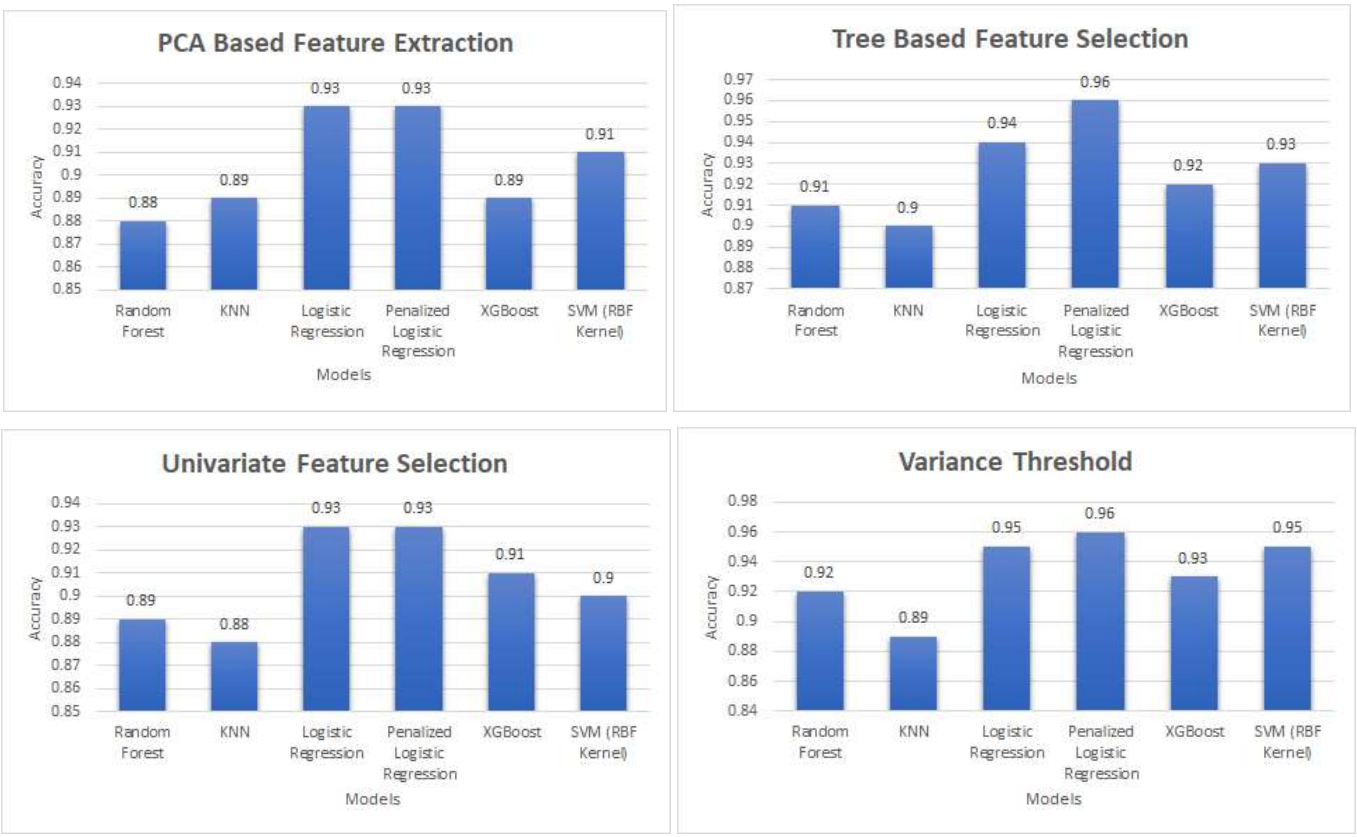
Results

Being a balanced-class classification problem, accuracy metric was used for model evaluation.

Predictive Models with Feature Extractor/Selector

For of all the feature extractor/selectors methods, Penalised Logistic Regression model performed the best and achieved **96%** test data accuracy with two of the methods, while kNN model performed consistently worse for all feature selectors and achieved on average **89%** test data accuracy. However, random forest model gave the worst performance for the PCA based feature extractor.

The order of relative model performance (best to worst) is nearly the same in all the feature selectors methods that is - Penalised Logistic Regression, Logistic Regression, SVM, Xgboost, Random Forest, kNN. Tree-based selector and Variance Threshold based selector performed equally well when coupled with Penalised Logistic Regression to achieve **96%** accuracy. Univariate feature selection and PCA based extractor performed equally well when coupled with Penalised Logistic Regression to achieve **93%** accuracy. When coupled with all machine learning algorithms, the Variance Threshold based selector performed better overall than the Tree-based selector. Due to its simplicity in implementation, the Variance Threshold based selector is preferred over the Tree-based selector method when coupled with Penalised Logistic Regression model. The test accuracies for this experiment can be viewed below in Fig 3:



.Figure 3: Test Accuracies of Machine Learning Algorithms Along With Different Feature Extractor/Selectors

Standalone Models with Automated Feature Extraction

The reported test accuracies for each of the models have been summarized in the table given below:

Data Models	CNN (with Dropout)	CNN (without Dropout)	RNN (with Dropout)	RNN (without Dropout)
Observed Test Accuracy	95%	94%	98%	99.6%

Overall, it was found that both, the standard as well as the regularized (dropout) versions of CNN and RNN performed at par respectively. However, it was noted that the dropout versions of the CNN and RNN took more training time when compared to their non-dropout versions respectively. From the table, it is clear that the RNN outperforms the CNN models. Surprisingly, the RNN model gave **99.6%** test accuracy, thereby beating the highest claimed test accuracy (**99.3%**) for this dataset on Kaggle.

Discussions and Future Works

From the above empirical results, a number of observations were documented as follows:

1. **Are complex models worth the computational time wait?:** In the present scenario, it is often thought that complex neural network models, such as the RNN and CNN should always be the 'go-to' options for any predictive modeling task at hand due to their strong predictive power. For the human activity recognition dataset, the **CNN took 2 hours** for training to achieve a **95%** accuracy. Likewise, the **RNN took 30 minutes** for training to achieve a **99.6%** accuracy. However, the **Penalized Logistic Regression took** only **2 minutes** for training to produce an accuracy of about **96%**. Since the data is scarce for this prediction problem, it can be concluded that the Penalized Logistic Regression, being a much simpler method, is a much more suitable option for this dataset. Thus, complex models, especially neural networks, need not always triumph over simple machine learning methods for predictive modelling.

2. **Room for Improvement:** Although the order-3 spline gives decent interpolation results, the problem of missing data interpolation should be studied more in depth, since the spline order might change over time. Moreover, since bigger sequence datasets generally guarantee better prediction estimate results, it would have been more ideal if additional training data were available (since data transformations decrease the effective sample size but increase the number of dimensions). It is also possible that the true infinite class data distribution could be imbalanced. In this case, as more unseen actigraphy data is gathered in the future (for example, in a streaming fashion), the preferences for the 'best model' may need to be adapted in real-time using metrics such as the F1-score.

Appendix :

GitHub Link : https://github.com/shriya2909/human-activity-recognition-with-smartphones