CMPT 353 Project

2024 Summer

Greg Baker

An Analysis of Diverse Models Predicting Actions on 5-Dimensional Data

Hanjie Liu(SID:301404949)
Yuan Gao (SID: 301365417)

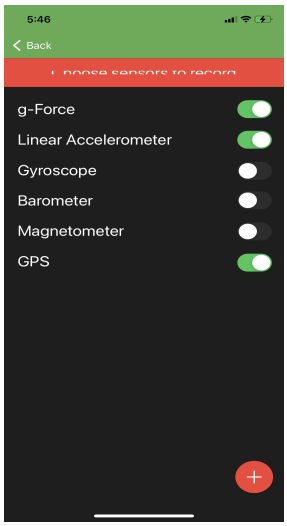
Table of Contents

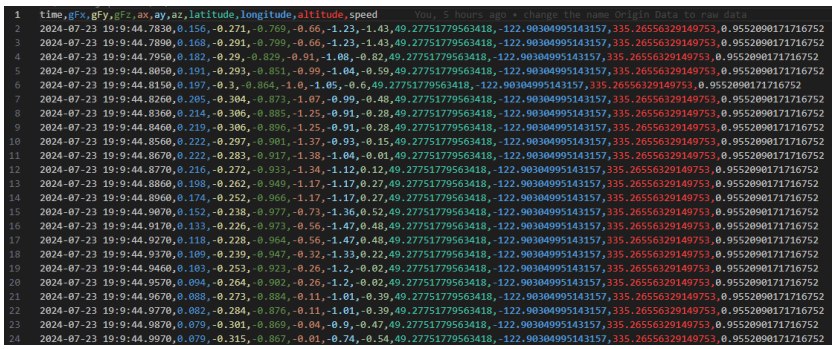
Introduction

In this project, we aim to predict user motion using a data model. Data is collected from the Physics Toolbox Sensor Suite app on mobile devices, and four states of motions are collected: walking, running, jumping, and remaining still. An additional set of dataset was collected under different motion states and was labeled ‘multi’. Data cleaning methods such as LOWESS regression are then used to smooth the data. The model then uses various machine learning algorithms to try to predict motion status based on the given data. By using multiple different algorithms on processed data, the model will be robust and accurate, which could be used to understand user activity at any given time.

Data Collection

Data collection is done by the Physics Toolbox Sensor Suite app. More precisely, the multi-record function of the toolbox. A combination of g-force, linear accelerometer, and gps measures were used in collecting data. The linear accelerometer and g-force measures acceleration and g-force in three different directions; ax and gFx, measures the left to right direction, ay and gFy, measure the front to back direction, and finally, az and gFz, measure the up and down direction. The GPS simply measures four things, latitude, longitude, and altitude, same as the linear accelerometer and g-force, but on a larger scale, and speed. Since the app is on mobile devices, during the data collection, the mobile device is held in hand to simulate devices capable of motion detection like fitbits and apple watches. Each data collection lasted around 10 secs and saved in *.csv* file format. The data for participating in the training are: *‘time’, ‘ax’,’ay’,az’*,and ’*speed*’.

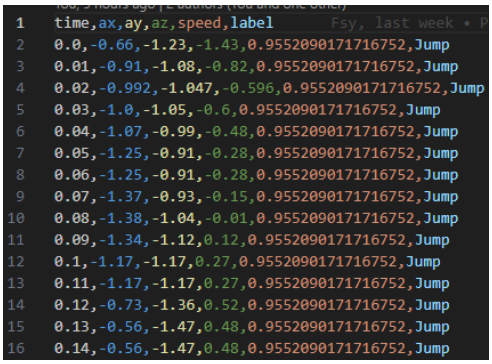




time,gFx,gFy,gFz,ax,ay,az,latitude,longitude,altitude,speed

2024-07-23 19:9:44.7830,0.156,-0.271,-0.769,-0.66,-1.23,-1.43,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.7890,0.168,-0.291,-0.799,-0.66,-1.23,-1.43,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.7950,0.182,-0.29,-0.829,-0.91,-1.08,-0.82,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8050,0.191,-0.293,-0.851,-0.99,-1.04,-0.59,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8150,0.197,-0.3,-0.864,-1.0,-1.05,-0.6,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8260,0.205,-0.304,-0.873,-1.07,-0.99,-0.48,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8360,0.214,-0.306,-0.885,-1.25,-0.91,-0.28,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8460,0.219,-0.306,-0.896,-1.25,-0.91,-0.28,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8560,0.222,-0.297,-0.901,-1.37,-0.93,-0.15,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8670,0.222,-0.283,-0.917,-1.38,-1.04,-0.01,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8770,0.216,-0.272,-0.933,-1.34,-1.12,0.12,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8860,0.198,-0.262,-0.949,-1.17,-1.17,0.27,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.8960,0.174,-0.252,-0.966,-1.17,-1.17,0.27,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9070,0.152,-0.238,-0.977,-0.73,-1.36,0.52,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9170,0.133,-0.226,-0.973,-0.56,-1.47,0.48,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9270,0.118,-0.228,-0.964,-0.56,-1.47,0.48,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9370,0.109,-0.239,-0.947,-0.32,-1.33,0.22,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9460,0.103,-0.253,-0.923,-0.26,-1.2,-0.02,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9570,0.094,-0.264,-0.902,-0.26,-1.2,-0.02,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9670,0.088,-0.273,-0.884,-0.11,-1.01,-0.39,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9770,0.082,-0.284,-0.876,-0.11,-1.01,-0.39,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9870,0.079,-0.301,-0.869,-0.04,-0.9,-0.47,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752
2024-07-23 19:9:44.9970,0.079,-0.315,-0.867,-0.01,-0.74,-0.54,49.27751779563418,-122.90304995143157,335.26556329149753,0.9552090171716752

Dataset before process(above)



time,ax,ay,az,speed,label

0.0,-0.66,-1.23,-1.43,0.9552090171716752,Jump
0.01,-0.91,-1.08,-0.82,0.9552090171716752,Jump
0.02,-0.992,-1.047,-0.596,0.9552090171716752,Jump
0.03,-1.0,-1.05,-0.6,0.9552090171716752,Jump
0.04,-1.07,-0.99,-0.48,0.9552090171716752,Jump
0.05,-1.25,-0.91,-0.28,0.9552090171716752,Jump
0.06,-1.25,-0.91,-0.28,0.9552090171716752,Jump
0.07,-1.37,-0.93,-0.15,0.9552090171716752,Jump
0.08,-1.38,-1.04,-0.01,0.9552090171716752,Jump
0.09,-1.34,-1.12,0.12,0.9552090171716752,Jump
0.1,-1.17,-1.17,0.27,0.9552090171716752,Jump
0.11,-1.17,-1.17,0.27,0.9552090171716752,Jump
0.12,-0.73,-1.36,0.52,0.9552090171716752,Jump
0.13,-0.56,-1.47,0.48,0.9552090171716752,Jump
0.14,-0.56,-1.47,0.48,0.9552090171716752,Jump

Dataset after process(left)

Fig 1. Physic Toolbox Suite

Data Plotting

To better understand and visualize the collected data, 2 distinct plots were constructed. The first method is a 3D scatter plot (fig 2) that visualizes the relationship between three variables: Ax, Ay, and Az. Additionally, a color map is used to represent a fourth variable, Time, where the color of the points indicates their position in the timeline.

This plot allows for the visualization of three variables simultaneously, giving a more comprehensive understanding of the data's spatial relationships. Better for visualizing the relationship between three spatial variables and incorporating time as a fourth dimension through color coding.

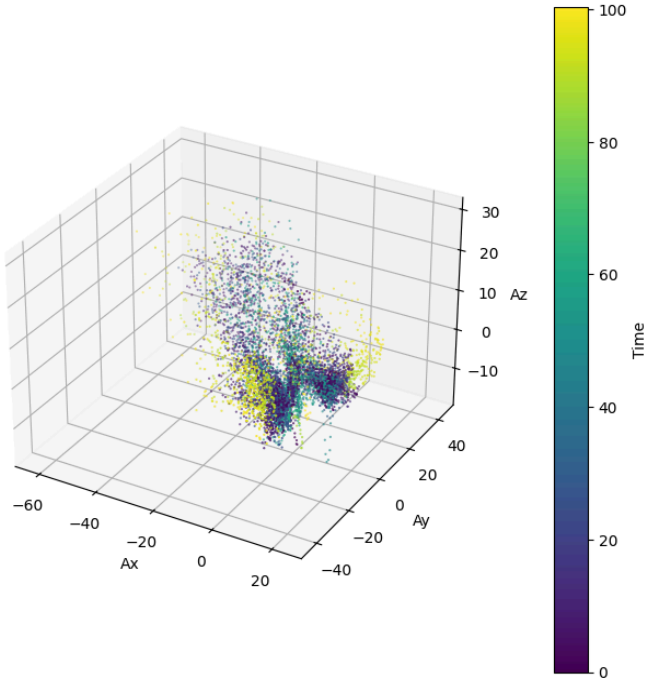


Fig 2
Example of Running Data from 5 time periods recorded.
Note the axis unit change depending on the data

The second method (fig 3) contains a series of line plots displaying the data over a certain time frame for four variables: ax, ay, az, and speed. Each subplot represents a different variable, plotted against the same x-axis, which denote time by unit 0.01s

Each variable is plotted separately, making it easy to identify patterns and trends specific to each one. Having all plots on the same x-axis allows for easy comparison of the different variables over the same time frame. Later, when we compare different data clean methods, the performance of this method is also more intuitive.

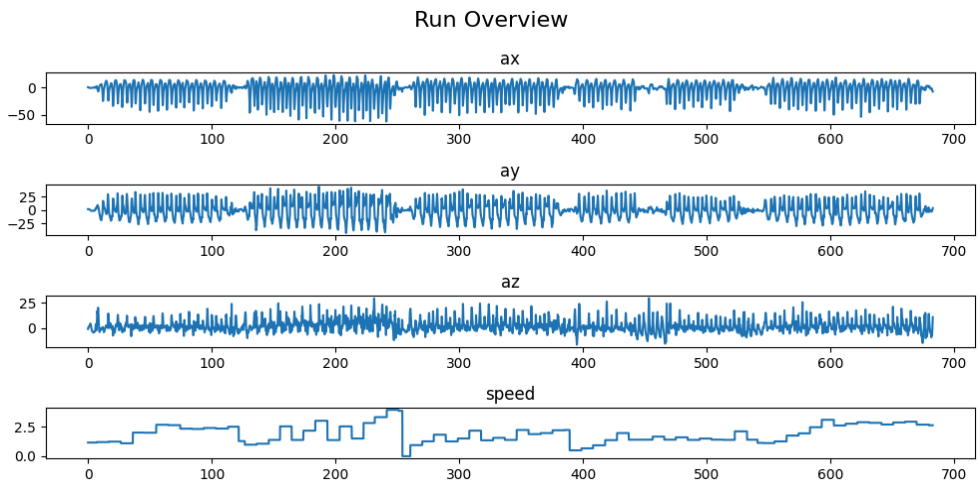


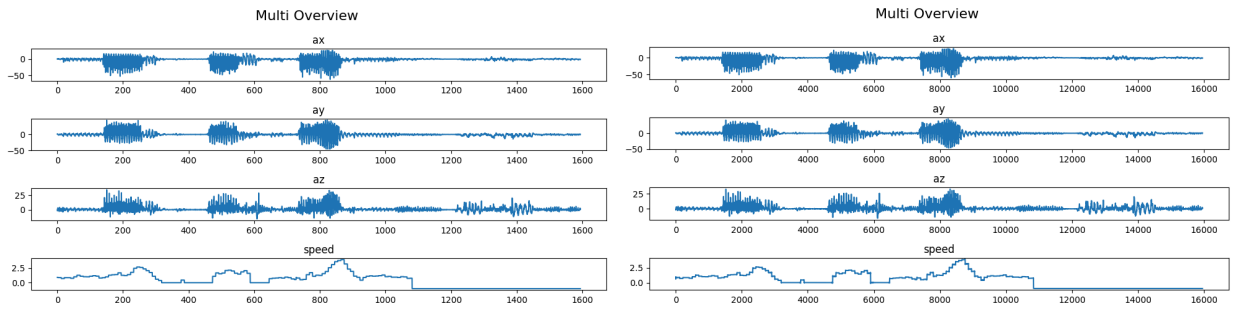
Fig 3
Example of Running Data from 5 time periods recorded.

Note the axis unit change depending on the data

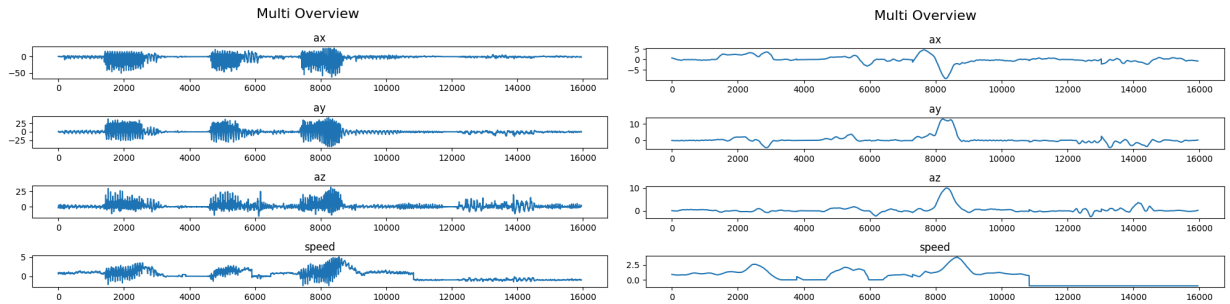
Data Cleaning

For data cleaning, only the linear accelerometer measures and speed was preserved. This is because the g-force is very similar to linear accelerometer and only one was needed, and GPS coordinates were not useful because the changes are not perceivable. A new column was then created to label the data motion status in file for easier training.

Data smoothing algorithms were used to combat any potential outliers and remove any potential sensor noise. Three different algorithms were employed, LOWESS smoothing, Kalman Filtering, and FFT denoise. Having different data cleaning processes allows for comparison and ensures the best approach is chosen.



Example of Multi-action data's plot, original data(left), FFT_denoised data(right)



Example of Multi-action data's plot, Kalman smoothed(left), Lowess smoothed(right)

Among them, the best performer is LOWESS smoothed. It performs best on all training methods. This may be due to the complexity of our data, which is mostly non-linear and has a lot of noise. LOWESS smooth is good at handling this situation.

	Classification Model	**Accuracy (0 - 1)**
FFT	MLP	0.9345
	GaussianNB	0.7691
	K-nearest Neighbour	0.9257
	RandomForest	0.9842
Kalman	MLP	0.9255
	GaussianNB	0.6861
	K-nearest Neighbour	0.9131
	RandomForest	0.9644
LOWESS (highest)	MLP	0.9950
	GaussianNB	0.8143
	K-nearest Neighbour	0.9956
	RandomForest	0.9996

Table 1: Table showcasing accuracies of classification models under different filters based on standard scaler.

Scaler Determination

For data reprocessing, we tried two different Scaler methods, namely StandardScaler and MaxMinScaler, which perform differently on different training models, but in general StandardScaler will perform better. This analytical approach emphasizes the importance of data preprocessing in shaping the efficacy of classification models, allowing for a comprehensive evaluation of different algorithms and scaling techniques.

	Classification Model	**Accuracy (0 - 1)**
No Scaler	MLP	0.9337
	GaussianNB	0.7677
	K-nearest Neighbour	0.9008
	RandomForest	0.9925
Standard Scaler (highest)	MLP	0.9445
	GaussianNB	0.7677
	K-nearest Neighbour	0.9255
	RandomForest	0.9936
Min-Max Scaler	MLP	0.9131
	GaussianNB	0.7677
	K-nearest Neighbour	0.8971
	RandomForest	0.9912

Table2: Table showcasing accuracies of classification models under different scalers base on original data training

After evaluating the model using both scaling methods, a significant improvement was observed using the standard scaler as shown in the model above. The reason for this difference is that the min-max scaler scales the values linearly to a fixed range of 0 to 1. Since the collected data is a wave from -x to x, where x is the peak of the wave, and all three datasets produce very similar ranges of values. For these reasons, it was a confident decision to adopt Standard Scaler as the preferred preprocessing scaler.

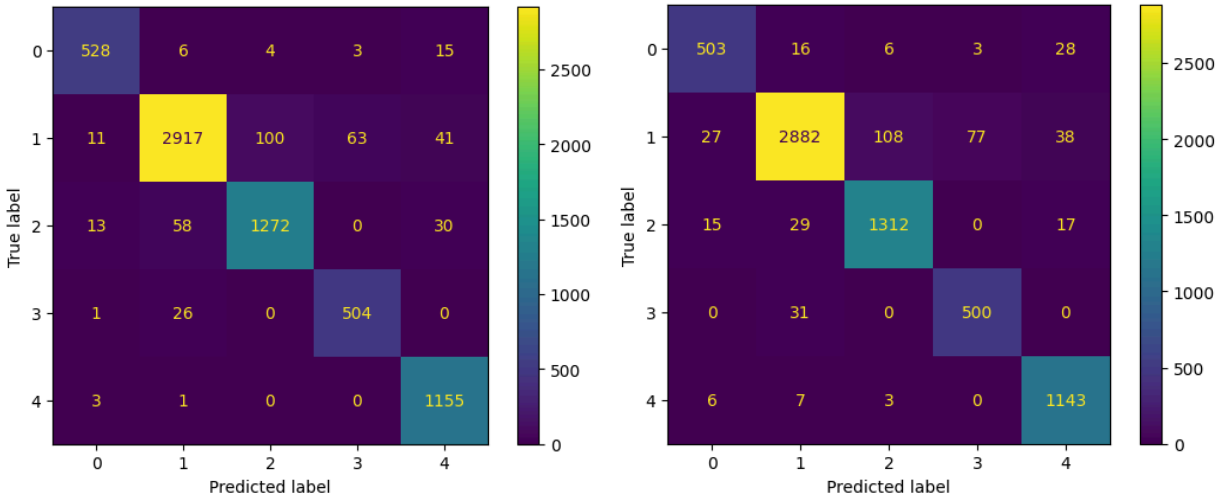
Model Selection

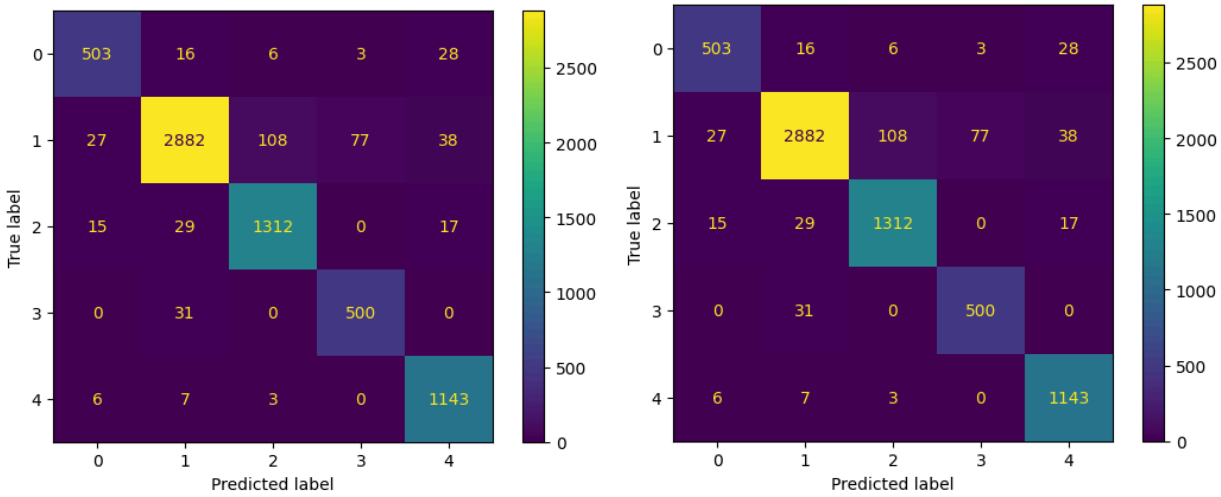
To obtain the most optimal classification model, the training process was initiated by pre-processing datasets using the Standard Scaler. The pre-processed data was then applied to four distinct models: Gaussian Naive Bayes (GaussianNB), K-Nearest Neighbours (KNN), Random Forest, and the Multi-Layer Perceptron (MLP) model. The scaler we choosed is Standard Scaler. The MLP layers are [16,32,64,128].

Model Comparison

To express the predictive power of each model, we used the Confusion Matrix and accuracy as our metrics, we tested the capabilities of all models for all datasets, and visualized our confusion Matrix. The demonstrated results are tested on the original dataset.

Table 1 and Table 2 clearly show the performance of different models in different situations. The results consistently demonstrated that Random Forest exhibited the highest accuracy across various conditions. Despite MLP's sophisticated architecture, it does not surpass the accuracy achieved by the Random Forest model. The confusion matrices are shown below (original dataset):





MLP(top-left), GaussianNB(top-right), KNN(lower-left), RandomForest(lower-right)

Conclusion

After a comprehensive series of tests and training iterations, a clear consensus has emerged on which classification model best determines activity types. It is evident that meticulous data cleaning significantly enhances prediction accuracy. Furthermore, the application of the Standard Scaler consistently outperforms the Min-Max Scaler. In the pursuit of an optimal classification model, the performance of four selected models was meticulously evaluated: Gaussian Naive Bayes (GaussianNB), K-Nearest Neighbors (KNN), Random Forest, and Multi-Layer Perceptron (MLP). The outcomes consistently highlight the superiority of the Random Forest model across various conditions in our experiments. Together, these factors determine the predictive power of our final model

Project Experience Summary

Hanjie Liu (301404949):

In this project, I was responsible for implementing the code for visualizing data, dataloader, and model training workflow. I also produced, and organized all the results. In this paper, I was also responsible for the corresponding parts of the above content with tables and conclusions.

Yuan Gao (301365417):

For the project, I collected the entire data sets for the models using the Physics Toolbox Suite, and created the code for smoothing data with different algorithms. I also created the smoothed data using different smoothing algorithms. For the report, I have written the introduction, data collection, and data cleaning.