Longboarding techniques classification using Machine Learning

Tuan (Kevin) Le[1,2,3], Evans Sajtar[1], McKenzie Lamb P.hD[2]
[1]Department of Computer Science, DePauw University, Greencastle, IN, [2]Department of Mathematical Science, DePauw University, Greencastle, IN, [3]Department of Physics, DePauw University, Greencastle, IN

Background

- An application aiming to classify a rider's longboard ride's techniques
- Experimentations with different methods for reducing loss and improving accuracy in predicting Longboard activities.

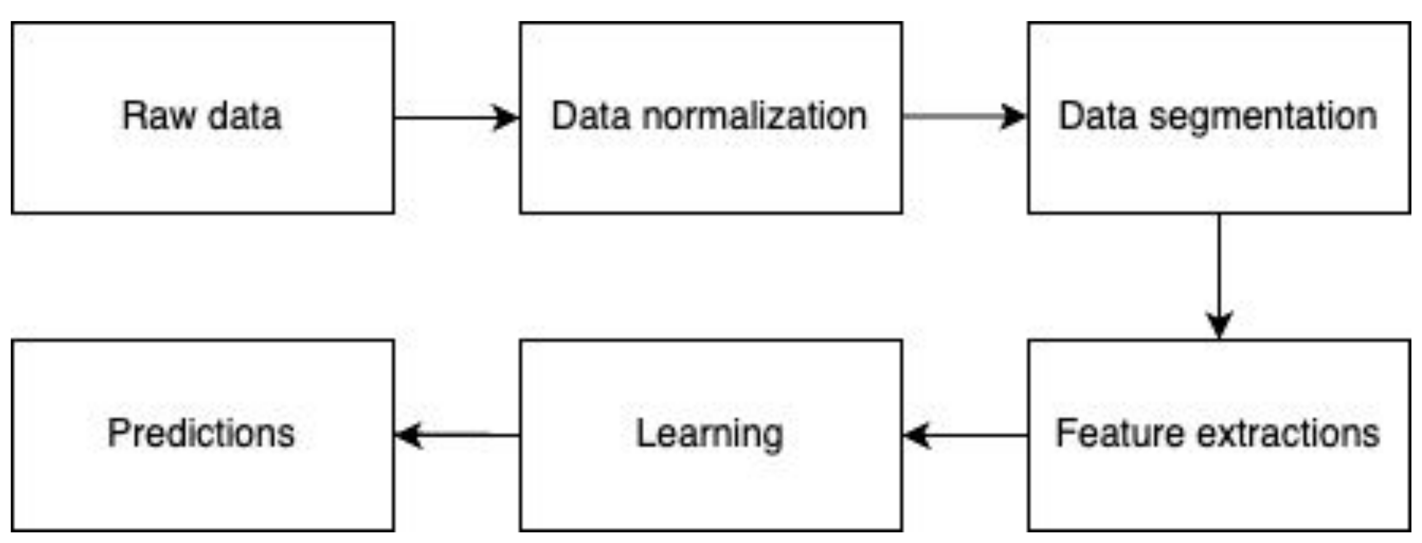
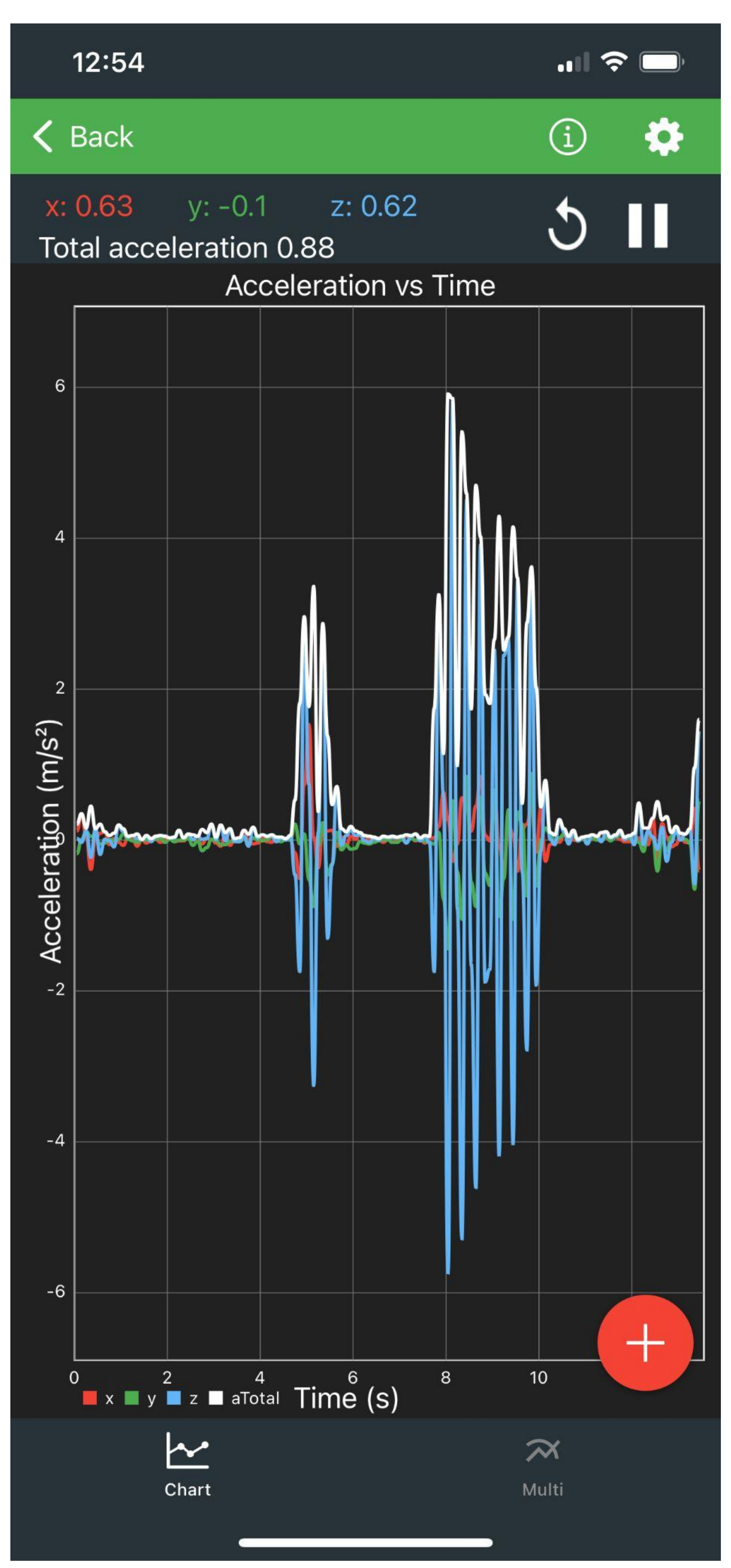


Figure 2: Predictor's workflow, i.e. all the steps happening from raw data to predictions within the model.

Data collection



- Android mobile application called "Physics Toolbox Sensor Suite"
- which records data in non-harmonic time intervals, frequently 0.01s-0.03s.
- Linear acceleration changes whenever the mobile device speeds up, slows down, or changes direction.
- When the device is at rest with respect to the surface of the earth, it reads acceleration values of 0, 0, 0.

time	ax	ay	az	Azimuth	Pitch	Roll
0.023363	1.5628	-2.0029	-4.6351	227.0835	-20.5221	-4.7116
0.030611	1.5628	-2.0029	-4.6351	227.0835	-20.5221	-4.7116
0.031195	1.5628	-2.0029	-4.6351	226.8786	-19.7172	-4.1708
0.047697	1.5628	-2.0029	-4.6351	226.8786	-19.7172	-4.1708
0.056661	1.5628	-2.0029	-4.6351	226.8786	-19.7172	-4.1708
0.066371	1.7575	-1.5602	-4.2442	226.8786	-19.7172	-4.1708
0.067007	1.7575	-1.5602	-4.2442	227.2102	-18.3254	-3.2866
0.084393	1.7575	-1.5602	-4.2442	227.2102	-18.3254	-3.2866
0.093233	1.7575	-1.5602	-4.2442	227.2102	-18.3254	-3.2866
0.102311	1.7575	-1.5602	-4.2442	227.2102	-18.3254	-3.2866
0.102716	2.307	-1.0359	-3.7756	227.2102	-18.3254	-3.2866
0.120837	2.307	-1.0359	-3.7756	228.1911	-16.1515	-2.3014
0.13033	2.307	-1.0359	-3.7756	228.1911	-16.1515	-2.3014
0.138833	2.307	-1.0359	-3.7756	228.1911	-16.1515	-2.3014
0.139614	2.307	-1.0359	-3.7756	228.1911	-16.1515	-2.3014
0.157083	3.4595	0.3896	-2.626	228.1911	-16.1515	-2.3014
0.165859	3.4595	0.3896	-2.626	229.27	-14.4537	-1.9822

Data Normalization

- Vector normalization using 3-dimensional rotation matrices that rotate the 3 dimensions around a vector.
- These inclination data are reversed and used for rotating all acceleration data back to a fixed coordinate system where Azimuth, Pitch, and Roll are collectively 0's.

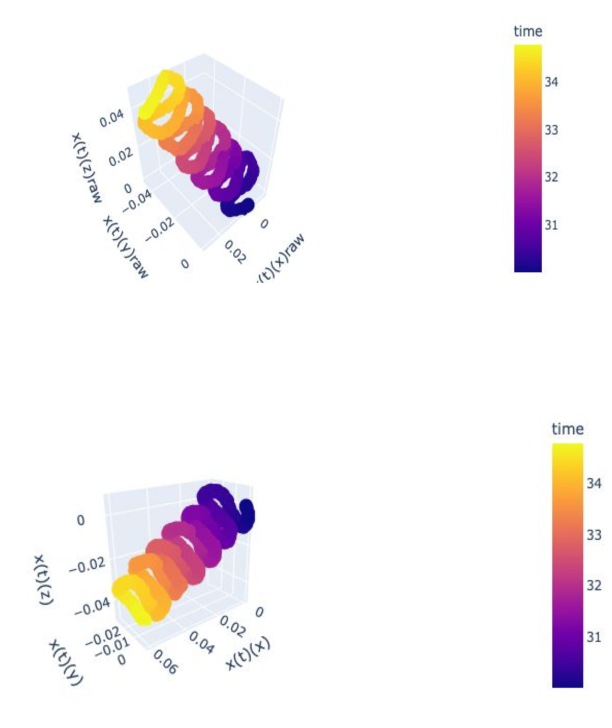


Figure: Rotation matrix applied on a small sample of the position displacement data. Above is the raw data, and below is the rotated data

Data segmentation

- data will be split into windows of similar time intervals. From each time window, we consider the last and first points to be vector data.
- After experiments with all possible time intervals, we found that 3s is the most optimal and high-performing time interval.
- This is largely based on the data so for different configurations of data, different periods of time may work better. As for all the configurations we have tested, 3s durations work best.

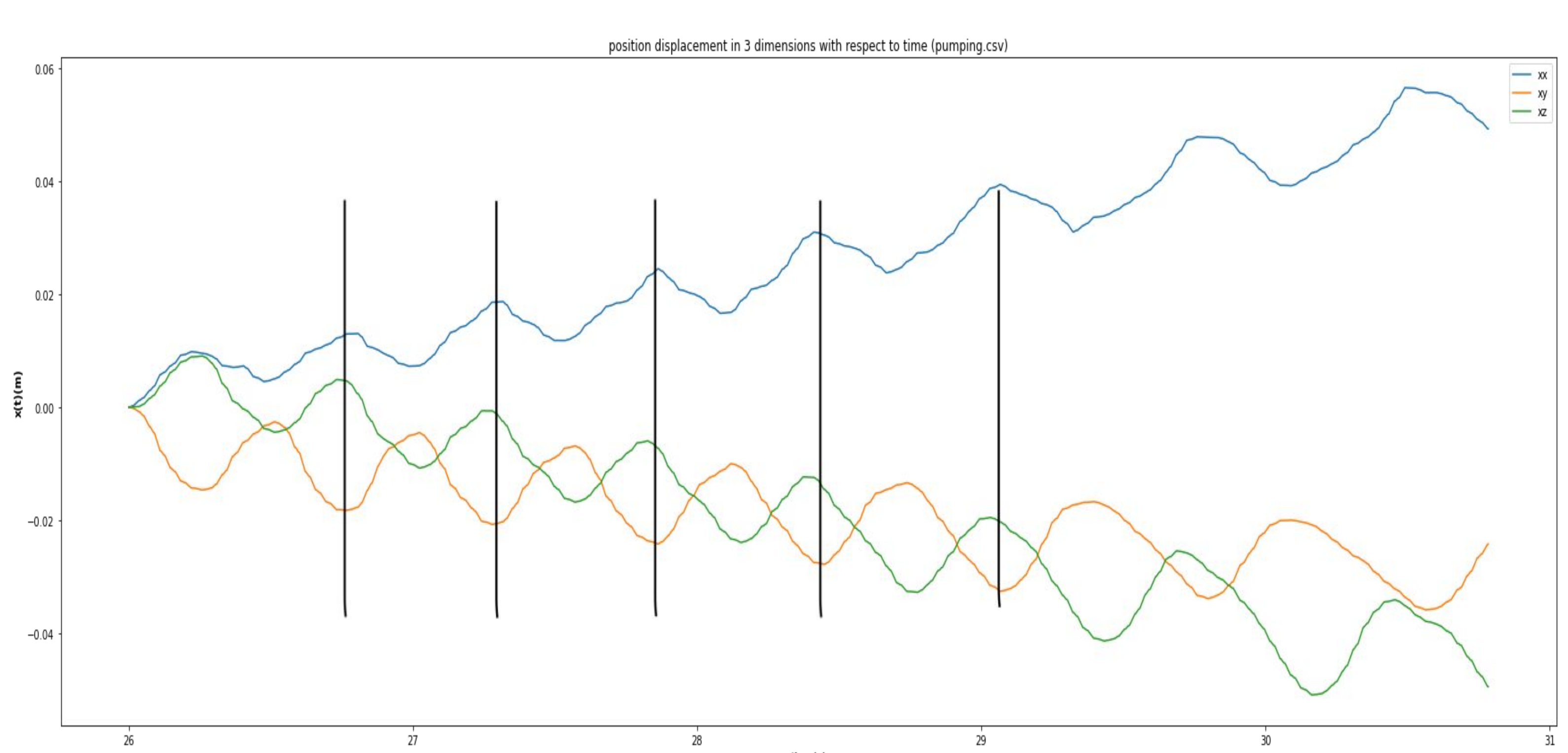


Figure: position displacement in 3 dimensions with respect to time (pumping.csv)

Feature extractions

- Sum of frequencies of position displacement vector in x and z dimensions
- Maximum and Minimum values of position displacement in x and z dimensions
- difference between the maximum and minimum accelerations in the y dimension
- Root mean squared of position displacement in x and z and acceleration in y (while ignoring NaN).

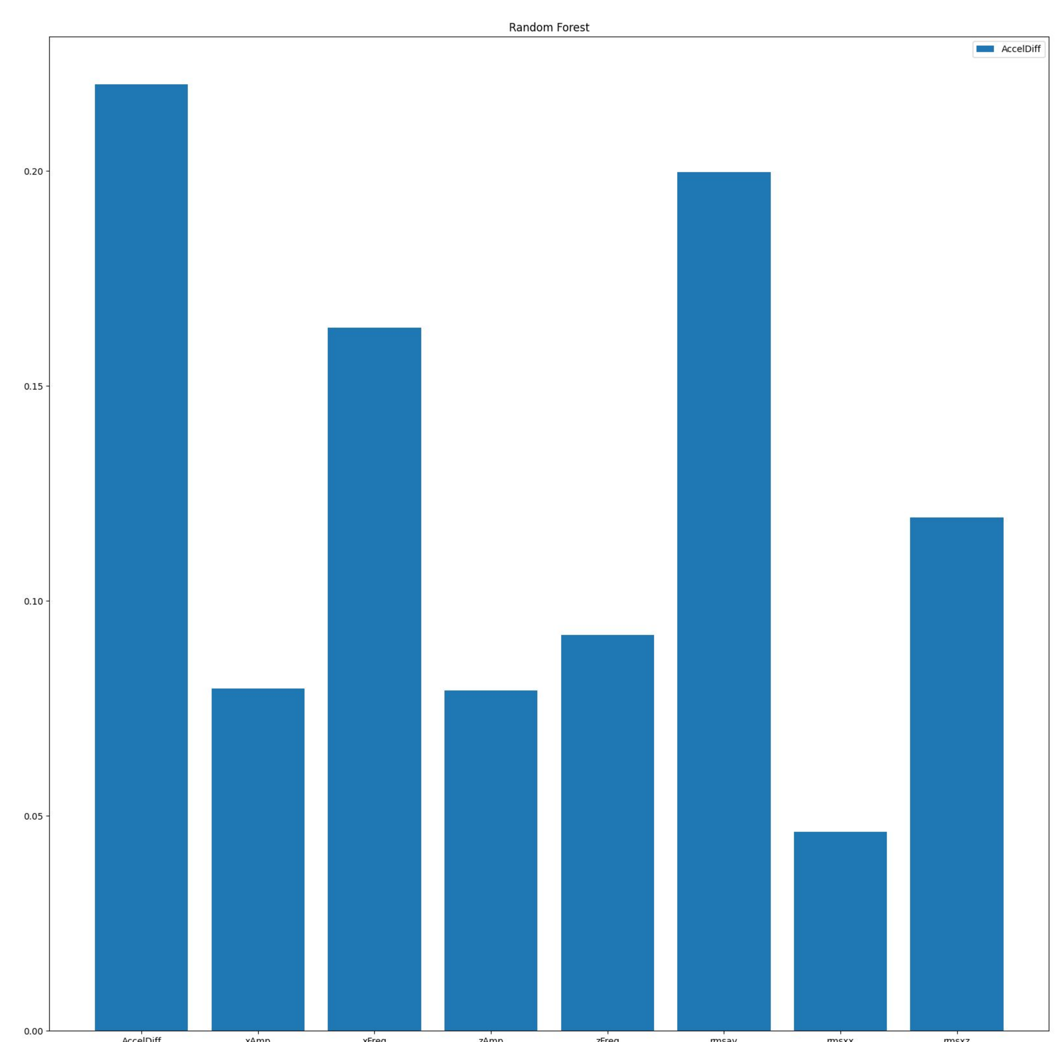


Figure 6: Feature importance when using Random Forest Classifier to classify Longboard Activity. From left to right, Acceleration di fference, Amplitude in x, Displacement frequency in x, Amplitude in z, Displacement frequency in z, Root-mean squared of acceleration in y, displacement in x and z

Results and future directions

id	Model	F1_score	Train Score	Test Score	Training Time
0	Gradient Boosting Classifier	0.953	1.00	0.97	0.599282
1	Random Forest	0.969	1.00	0.95	0.203462
2	Logistic Regression	0.905	0.88	0.86	0.011103
3	Nearest Neighbors	0.939	0.97	0.94	0.003243
4	Decision Tree	0.935	1.00	0.90	0.005140
5	Linear SVM	0.934	0.91	0.91	0.005235
6	Neural Net	0.931	0.90	0.88	0.248223
7	AdaBoost	0.913	0.90	0.86	0.122451
8	Gaussian Process	0.892	0.86	0.85	0.094242

Table: F1 score, train score (the fraction of correctly classified samples when learning the training set), test score (the fraction of correctly classified samples when learning the testing set) and training time of different classifier

Next steps:
- Higher accuracy with neural network models and modified versions of classifiers.
- Model deployment for an Android and IOS application.
- Do more test runs with different riders and configurations.

Acknowledgement

Appreciation for Dr. McKenzie Lamb for advising, Evans Sajtar for assisting on building the application, and all the individuals who helped refine and shape the direction and methods of the research.