

NUS ISS EBAC Capstone Phase 2 Report: LTA Mobility Sensing Project



Transport Mode Detection | Train Position Detection | Road Anomaly Detection

Internship Organisation

Land Transport Authority (Singapore)

Project Members

Ang Khee Hwa Randi (A0198526N) | Choong Pei Wen (A0199043Y) | Li Tiancheng (A0198530Y)

NUS ISS Supervisor

Brandon Ng Swee Tuan

Internship Host Organisation Supervisor

Rachel Lim

Special Thanks

Jonah Ong (Director of ITDC LTA)

Mo Guangquan (Dev-Ops)

Evan Chang (Dev-Ops Intern)

Abstract

The Mobility Sensing Project (MSP) by the Singapore Land Transport Authority (LTA) aims to identify travel patterns from GPS data and other multi-sensory data collected in smartphones to provide personalised travel information during service disruption and support the goals of Land Transport Master Plan (LTMP) 2040 of a smarter transportation network and the “20-Minute Towns, 45-Minute City”. It has four broad objectives, namely:

1. Improving the travel experience of commuters
2. Optimise planning of first and last-mile transportation journeys
3. Reducing journey times for commuters
4. Richer data to support qualitative survey data

In this report, we will be exploring 3 key projects under the MSP umbrella, namely:

1. Transport Mode Detection (TMD)
2. Train Position Detection (TPD)
3. Road Anomaly Detection (RAD)

The TMD project aims to detect the transport mode of commuters in real-time based on their smartphone sensor data. Insights could then be drawn to support other research into commuting patterns in Singapore and optimise transport planning at the macro level. To achieve this, accelerometer and gyroscope data was collected, followed using signal processing and assorted machine learning techniques to create a multi-classification model. Based on the experiments, Neural Network and Random Forest methods proved to be the most viable methods, with good overall accuracy, with the latter being the most robust.

In the TPD project, the use of Derivative Dynamic Time Warping (DDTW) of accelerometer data as mentioned in the research paper published in September 2019 (Khuong An Nguyen, 2019) was explored on its application to identify user location on underground trains in the absence of GPS availability. This method is premised on 2 key assumptions: the trains are fully autonomous (driverless) and have consistent acceleration and deceleration patterns which can be measured and collected from passengers' smartphones. The ability to detect would user location underground which typically lacks GPS signals, would allow LTA to better provide personalized analytics solutions and notifications to users based on their location. Based on the test results, it was found that 10Hz accelerometer data collection frequency achieved differentiable DDTW scores about 7 out of 10 times.

For the RAD project, it is primarily a proof of concept (POC) run on the feasibility of using smartphone sensors to detect potholes and crowdsource suspected potholes. Doing so will reduce the amount of manual effort by road surveyors, improve the timeliness of pothole detection and repairs, and ultimately increase commuter safety and comfort on the roads. A combination of wavelet transformation on the horizontal linear acceleration, Z-axis linear acceleration and X-axis gyroscope rotation was used to process the data; weighted thresholding and K-means clustering algorithms were used to create an ensemble voting system for pothole detection. Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) was used for the geospatial clustering of the crowdsourced data to filter out spurious results. The results were moderately successful with up to an 86% accuracy achieved in pothole detection across a multitude of different road surfaces, thus successfully proving the feasibility of the method.

Contents

Overview / Business Objectives	1
Transport Mode Detection	2
Train Position Detection	2
Road Anomaly Detection	3
Literature Review	4
Transport Mode Detection	4
Train Position Detection	5
Road Anomaly Detection	5
Overview of Key Smartphone Sensors	6
Data Preparation and Processing Overview	13
Data Collection, Processing, Analysis and Findings	14
Transport Mode Detection	14
Data Collection	14
Data Processing	15
Feature Engineering	16
Motorized Transportation Mode Detection (1Hz)	20
Multi-Class Transportation Mode Detection (1Hz)	21
Multi-Class Transportation Mode Detection (50Hz)	23
Multi-Class Transportation Mode Detection (50Hz + Differing Sliding Window Settings)	25
Limitations	28
Future Work	28
Train Position Detection	31
Data Collection	31
Key Technique Used	31
Key Assumption - One-stop or station-to-station Journeys Duration Stability	33
Overview of Data Processing	34
Higher-level features for Derivative Dynamic Time Warping	34
Test 1	35
Test 1 Observations	36
Test 2	37
Test 2 Observations	38
Test 3	39

Test 3 Observations	40
Test 4.....	41
Test 4 Observations	41
Test 5.....	43
Test 5 Observations	43
Summary of Observations.....	44
Limitations.....	45
Future Work	45
Road Anomaly Detection	46
Data Collection.....	46
Data Processing.....	49
Feature Engineering.....	53
Modelling	58
Findings & Conclusion	64
Limitations.....	65
Future Work	65
Appendix A – Brand Share by Sales Volume of Smartphone Market	I
Appendix B – Road Anomaly Detection	II
B01 – Fourier Transform Results.....	II
Linear Acceleration (20Hz)	II
Gyroscope Rotation (20Hz)	IV
B02 – Filtered Signal Data	VI
Linear Acceleration (20Hz)	VI
Gyroscope Rotation (20Hz)	VII
B03 – Principal Components	VIII
B04 – K-Means Clustering	XXI
B05 – Results from Other Clustering Methods	XXII
Local Outlier Factor (LOF) Method.....	XXII
Density-Based Spatial Clustering of Applications with Noise (DBSCAN).....	XXIII
Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)	XXV
Table of Figures	XXVI
References	XXX

Overview / Business Objectives

The Mobility Sensing Project by the Singapore Land Transport Authority (LTA) aims to identify travel patterns from GPS data and other multi-sensory data collected in smartphones to provide personalised travel information during service disruption and support the goals of Land Transport Master Plan (LTMP) 2040 of a smarter transportation network and the “20-Minute Towns, 45-Minute City”. It has four broad objectives, namely:

1. Improving the travel experience of commuters
2. Optimise planning of first and last-mile transportation journeys
3. Reducing journey times for commuters
4. Richer data to support qualitative survey data

Currently, most of the transport planning data come primarily from the EZ-Link transportation card transaction data which works well for the planning of main transportation lines, but it lacks the granularity for any form of personalisation and it does not capture the last mile walking journey of commuters very well. The advent of more powerful smartphones with a varied sensor suite being ubiquitous around us presents an opportunity to tap on it to gain richer insights to fill in information gaps for better transportation planning and personalization. Furthermore, it could also open up applications beyond transportation planning route planning, to assist in the detection of road anomalies, which we shall explore in this project. The other reasons for the use of smartphones include its powerful onboard computers that allow for complex applications to be developed, beyond just a dumb sensor logger; web connectivity to support crowdsensing and cloud technologies; its relatively low cost compared to dedicated precision sensor equipment which can cost in the tens of thousands of dollars.

For this project, we will be working on three major components, namely:

1. Transport Mode Detection (TMD)
2. Train Position Detection (TPD)
3. Road Anomaly Detection (RAD)

The workload of the three components would be split as follows: TMD would be developed primarily by Tian Cheng; Pei Wen focusing on TPD; lastly, Randi would be working on RAD. The team is further supported by Jonah, Rachel, Guangquan and Evan for guidance and Dev-Ops support.

These components may then be ported over the LTA’s existing *MyTransport.SG*¹ or a separate dedicated app for final deployment to support future analytics work.

Supporting this project would be the following sensor logger apps:

1. Mobility Sensing (MS)² – in-house developed app for Android and iOS
2. LTA Mobility Sensing – in-house developed prototype app for Android

¹ *MyTransport.SG* is a one-stop transport app developed by LTA to allow commuters to plan their journeys, check traffic conditions, check for parking availability, report errant behaviour by Personal Mobility Device (PMD) users and road defects.

² Mobility Sensing is the main app which we are using data collection and alpha testing the models created. It may also be customized to collect data from our desired variables and frequencies, but such customization is hardcoded which takes time.

3. Sensor Record³ – free app for Android devices by Martin Golpashin
4. AndroSensor – free app for Android devices by Fiv Asim

Transport Mode Detection

In this task, we aim to detect various transport modes using multi-sensory data collected by a smartphone.

The following chart shows the classification of different travel modes:

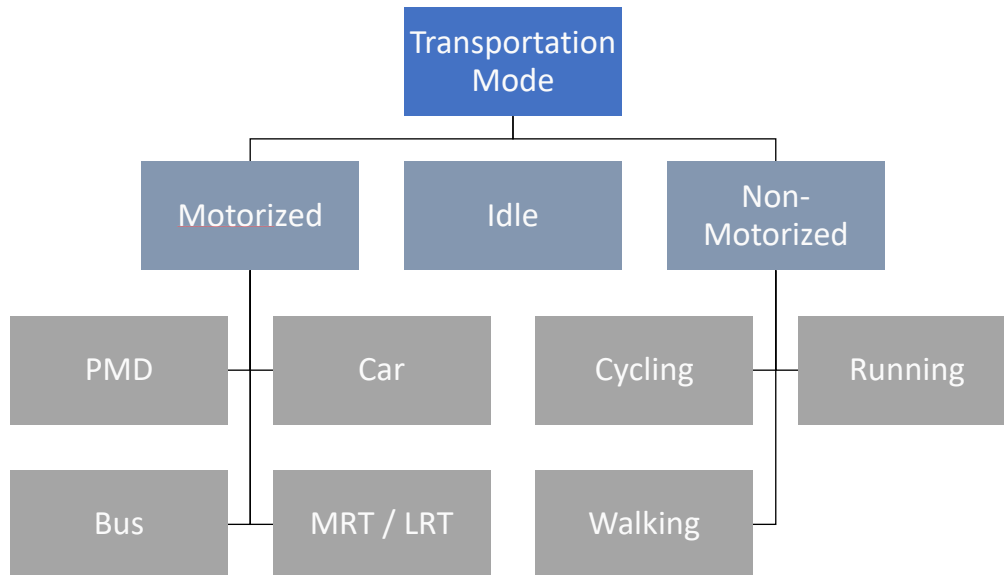


Figure 1: Transport Mode Classification

There are several benefits if we can detect transport modes correctly:

1. By identifying roads commuters cycle on, additional cycling paths may be planned to facilitate such modes of travel
2. Targeted improvement of travel options to reduce the travel time of citizens for key routes as part of the “45-minute City with 20-Minute Towns” as part of the Land Transport Master Plan 2040
3. Richer data on last-mile journeys of individual commuters for better transport planning

Train Position Detection

Currently, the GPS has limited signal connectivity underground and indoors, making geolocation via GPS alone almost impossible. Furthermore, the GPS sensor consumes a significant amount of battery power on smartphones, limiting sustained high-frequency usage over long periods. Therefore, LTA seeks to explore the use of smartphone accelerometer data from station-to-station train journeys to find a pattern signature which could be used as a reference for a pattern-matching algorithm to determine the location of a person on a fixed train route. Previous work by LTA on geolocation techniques without GPS used pressure signatures of different locations along the underground track recorded via the smartphone barometer. However, the barometer sensor is commonly absent on lower-tier smartphones, which would limit the widespread adoption of such methods. Accelerometer sensors are, however, more commonly

³ Sensor Record is the backup app which was used to collect data using Android devices during the initial testing. Ultimately, this sensor logger app was not used as more reliable alternatives such as AndroSensor and the upgraded MSP app.

found on almost all smartphones, making it possible for widespread adoption if the method is proven to be successful. The recording of accelerometer values can also be done on smartphones through an application at the background and this would be independent of the user's location sharing settings.

LTA's current methods deployed to collect and analyse commuter travel patterns are mainly through the tap-in and tap-out data at the train gantries, as well as, the use of Wifi-APs to determine the presence of commuters in a train station. These methods are successful in planning train frequencies based on the past number of commuters present. For instance, through the data, a regular spike in the number of passengers present at stations at certain timings indicates that that timing is a peak period and more trains should be deployed to increase train frequency and help ease the congestion. However, these methods cannot be applied in helping to develop the mobile application, MyTransport.SG (a mobile application that has been developed by LTA) into a mobile application that can provide useful and smart notifications for commuters to be notified of delays in their regular train journeys.

The following benefits can be achieved through tracing commuters' underground location and insights on commuter travel patterns:

1. Acceleration signals can be collected via MyTransport.SG and through pattern matching, users' travel patterns on the MRT can be recorded. According to the regular travel patterns recorded, push notifications can be sent to users if there are any disruptions along their regular travel route, instead of blasting indiscriminate notifications of train disruptions on all train lines that are mostly unimportant to the user.
2. With real-time matching, LTA may also inform the users of any disruption along the line, so that the user can plan alternative routes to their destinations.
3. Furthermore, with real-time commuter tracking underground, in times of disruptions where users are stuck in between two stations, LTA will be able to help users know how much longer they would need to reach their intended destinations. This can be done according to their location in the underground track and is useful should users be stuck in the middle of two stations.
4. In the event of a security breach, LTA may also use this information to track down where the passengers are along the train tunnels.
5. With travel routes obtained from the data in *MyTransport.SG*, LTA can better analyse the routes thus, optimise transport system planning.
6. The use of accelerometer values minimises the battery consumption of smartphones by reducing the reliance on GPS sensors that drains more battery.

Road Anomaly Detection

For this module, we are attempting to use the onboard smartphone sensors, namely the accelerometer, gyroscope, GPS, and magnetometer, to assist in the detection of road surface defects such as bumps and potholes. The hypothesis goes that when a vehicle travels over a surface defect, it will register a change in the accelerometer and gyroscope values from the norm, which we could then use to identify as a possible surface defect. This is then replicated for multiple users through crowdsensing. By combining multiple of such readings on a geospatial map, it would allow one to better pin-point zones where such defects exist, prompting the road surface maintenance team to dispatch an inspector to verify the event and also to send in a road repair crew where needed.

Doing so will result in the following benefits:

1. Reduce the cost of detecting road anomalies
2. Better coverage of roads being scanned for defects⁴
3. More timely detection and reporting of road defects island-wide
4. Lower road repair cost through timely early intervention
5. Improved road safety through timely removal of road anomalies
6. More comfortable road experience

For this phase of the project, it will be a Proof of Concept (POC) classification exercise between the following models:

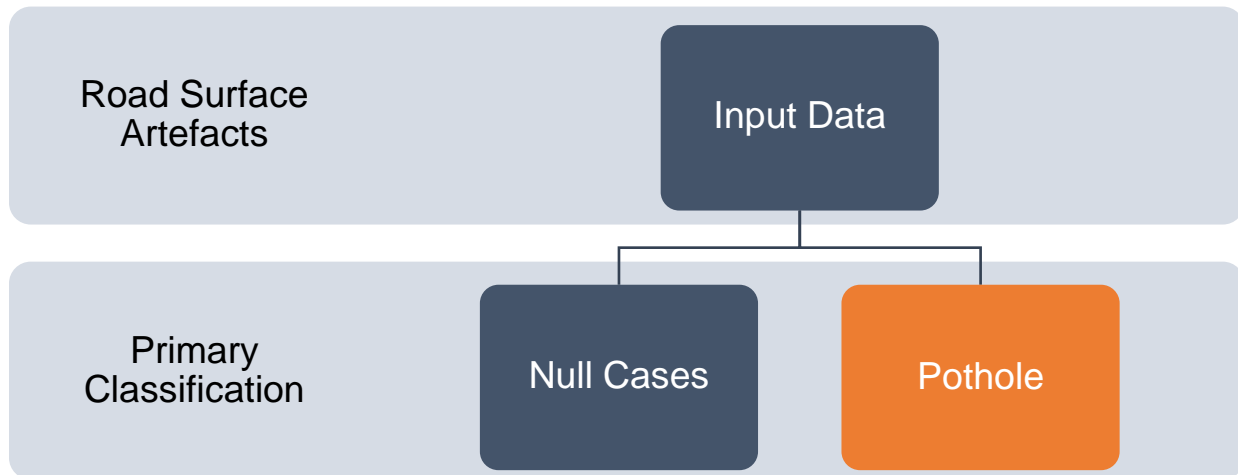


Figure 2: Classification of Different Road Surface Defects

Literature Review

Transport Mode Detection

In a previous study by Claudia Carpineti, et al., they used assorted smartphone sensor readings such as gyroscope (rate of rotation), orientation, linear acceleration, rotation vector and speed at a sampling rate of 20Hz. They applied a sliding window method with a 5s window size for feature extraction, namely maximum, the minimum, average, standard deviation in the time domain. Four modelling methods were then attempted (Decision Tree, Random Forest, SVM and Neural Network), and the Random Forest model was found to have the best performance. They found that acceleration, rate of rotation and speed were the most important variables (Claudia Carpineti, 2018) in transport mode detection.

In another study by Meng Chieh Yu, et al., three sensors (accelerometer, gyroscope and magnetometer) were used to collect the data. They also used the sliding window method for feature extraction in the time domain and frequency domain, of which seven key features were found, namely, acc_std, acc_mean, acc_FFT (peak), acc_FFT (ratio), mag_std, gyro_std and gyro mean (Meng-Chieh Yu, 2014). Other studies

⁴ It should be noted that using LTA's *MyTransport.SG*, members of the public are able to take photos and report such road anomalies on their phones. However, the need for additional effort to active tag such events, and the need to positively identify defects by untrained laymen generally results in an under reporting and misclassification of such road anomalies. This phenomenon is common in other crowdsourcing applications by other agencies globally.

also found that using a larger window size would have better predictive power than just using overlapping windows (Luca Bedogni, 2016).

Based on these findings, we decided to use the accelerometer, magnetometer and gyroscope sensor data to extract features in the time, frequency and wavelet domains, to attempt transport mode detection by mainstream machine learning approaches such as decision tree, random forest, SVM etc..

Train Position Detection

The methodology that has been used here was adapted from the research paper, *“Realtime Tracking of Passengers on the London Underground Transport by Matching Smartphone Accelerometer Footprints”* by Khuong An Nguyen, You Wang, Guang Li, Zhiyuan Luo and Chris Watkins. (Khuong An Nguyen, 2019) Though the setting of this paper was in London, a similar methodology seemed likely to be replicated on Singapore Mass Rapid Transit (MRT).

As the research paper presented a novel approach to track passengers in realtime with the smartphone accelerometer and a training database of the entire London underground network, we thought that the approach could be applied on Singapore’s driverless train lines. The rationale is that self-driving transport like Downtown Line or North-East Line trains in Singapore most likely possess predictable accelerations, decelerations and travelling time patterns, as they travel on the same fixed rail lines, which is similar to that of the London Trains experimented in this paper.

Road Anomaly Detection

This project took inspiration from an original pioneering study by MIT which used accelerometer and GPS data of Android phones to determine the existence of potential potholes (Eriksson, et al., 2008). Other teams have since built upon this proof of concept, evolving to other projects such as *Street Bump 1.0* developed by the City of Boston to automate pothole detection and reporting. However, that app had a lot of false-positive classifications, resulting in major tweaks that need to be made in its second iteration (Simon, 2014). However, even in its second iteration, *Street Bump 2.0* could not passively record and upload the data recorded, resulting in a lower app usage due to the additional cognitive effort required by users to report potential potholes (CEA, 2015). This highlights the importance of user experience in deploying the model as an application for users – especially for general masses who may not be as motivated or compelled to use an application, unlike a paid employee.

In terms of data processing methods, several studies have either deployed the use of Euler’s Angles matrix rotation to reorient sensor readings or simply used the overall magnitude of the 3-axis acceleration. The above methods are inferior to using quaternion rotation to reorient the values, especially in the event of a bump/pothole, it is expected that there will be a spike in the Z-axis acceleration’s magnitude (facing the sky and ground) which would best correspond with the bump/pothole, which we can only isolate if the values are orientated properly. The use of Euler’s Angles matrix rotation also has the problems of gimbal lock when orthogonal rotations are performed; and the order of the rotations performed will ultimately affect the final position of the rotated vector, resulting in an inconsistent realignment of sensor values. The use of magnitude of the 3-axis acceleration will prevent one from isolating the acceleration on the z-axis. To make things worse, if one is using raw acceleration values instead of linear acceleration, one will

also register a somewhat constant gravitation pull⁵ on the z-axis (assuming if the phone is in a neutral position aligned to world coordinates), and if the phone is off its neutral position, the force is then applied on all 3-axis, making it hard to denoise. Hence, we will propose using the quaternion rotation method using the quaternion units provided by the phone⁶ to reorient all values to a neutral orientation.

The subsequent data processing steps involve the use of a filter to denoise the signal from the other non-pothole events. Bump detection is then performed by either a thresholding method (Astarita, et al., 2012) or wavelet transform method (Li, et al., 2019). However, the former method tends to have lower accuracy and more false positives compared to wavelet transform methods. The use of continuous wavelet transformation combined with a Hierarchical Density-Based Clustering with Noise (HDBSCAN) by Li et al, was found to be superior in detecting potholes and eliminating false positives. Hence, for this study, we will be largely replicating their methods with some improvements made in using an expanded suite of sensors and the use of quaternion rotation for realigning the sensor values.

Overview of Key Smartphone Sensors

Current smartphone models as for 2020, have generally a wide array of sensors, with the flagship models tending to have a wider suite and possibly higher-end sensors with better precision. According to Euromonitor, iOS phones make up at least 30% of all smartphone sales in Singapore since 2014 (Euromonitor International, 2020)⁷, all of which are flagship models; based on web device detection, which estimates usage rates, the top 12 phone models in Singapore in 2019 making estimated 50% of the market share (Device Atlas, 2019), are also flagship models. Hence, one can safely use most sensors supported by the phone's OS regardless of the phone model. It is quite likely that the users of the apps which our models will be deployed will likely be younger, more tech-savvy individuals, who are also more likely to use higher-end or flagship phones with support for a larger array of higher-precision sensors. Common sensors include the following, with the accelerometer and GPS being one of the most common sensors available on all phones (Google, 2019):

1. Accelerometer
2. GPS
3. Magnetometer
4. Gyroscope
5. Barometer
6. Light sensor

The following are more details about the various sensors:

⁵ The Earth's gravity is somewhat constant at an average of -9.807m/s^2 however it varies depending on the location based on the Earth's density and centrifugal force from the Earth's rotation exerted at that location. The centrifugal force is strongest near the equatorial regions, making one slightly lighter there.

⁶ Android: rotation vector; iOS: CMAAttitude.quaternion.

⁷ Refer to "Appendix A – Brand Share by Sales Volume of Smartphone Market" for more info.

Sensor	Measures/captures	Advantages/disadvantages
Accelerometer	Linear acceleration	Gravity component is always present in the measurements. 3D position estimation requires double integration therefore it is susceptible to a large position drift
Gyroscope	Angular velocity	Angle orientation is estimated through integration of the angular velocity and prone to drift due to signal noise
Camera	Images and video	Data extraction from images is computationally expensive and battery inefficient. It brings concerns regarding user's privacy
Compass	Magnetic field	Measures orientation respect to the magnetic north but it is susceptible to electromagnetic noise
GPS	Geolocation	Directly measures global 3D positioning but it does not work indoors due to low signal strength from satellites
Barometer	Atmospheric pressure	Can deliver altitude coordinates and helps to rapidly acquire GPS location. Its precision is low and can be affected by atmospheric factors (e.g. air currents and changing weather)

Figure 3: Sensors details (from Smartphone-based Human Activity Recognition by Jorge Luis Reyes Ortiz)

For the most part in this project, sensor readings from the accelerometer and gyroscope would be the primary data used for analysis, though info from other sensors would also be used to augment the analysis. It should also be noted that the sensors will not necessarily fire at the same time and that there may be slight gaps in the sensor readings due to phone memory and processing issues. Hence, even values taken from the same timestamp are at best loosely synched together.

Table 1: List of variables used. We originally took readings sampled at 1Hz but it was not effective in capturing the wavelet signals. Hence, it was increased to 20Hz for RAD and 50Hz for TMD and TPD with subsequent upgrades to the sensor logger apps used.

S/N	Variable	UOM	Freq (Hz)	Est. Precision	Sensor(s) Used	RAD	TMD	TPD
1	3-Axis Acceleration	m/s ² or g	20 / 50	± 0.04 m/s ²	Accelerometer + Gyroscope	1	1	1
2	3-Axis Linear Acceleration	m/s ² or g	20 / 50	± 0.04 m/s ²	Accelerometer + Gyroscope	1	1	1
3	Rotation Vector (Android) or CMAAttitude.quaternion (iOS)	Quaternion Units	20 / 50	NA	Accelerometer + Gyroscope + Magnetometer	1	1	0
4	3-Axis Rotation Rate	rad/s	20 / 50	± 0.001 rad/s	Gyroscope	1	1	0
5	3-Axis Orientation	Degrees or rad	20 / 50	± 0.0001°	Gyroscope	0	1	0
6	Magnetic Field	μT	20 / 50	± 0.3 μT	Magnetometer	0	1	0

7	Air Pressure	hPa or mbar	20 / 50	± 0.05 mbar	Barometer	0	1	0
8	GPS Coordinates	Degrees or (Longitude & Latitude)	20 / 50	± 10 m	GPS + Wi-Fi + Bluetooth + 3G/4G/5G Comms Antennae	1	0	0
9	GPS Accuracy	m	20	NA	GPS + Wi-Fi + Bluetooth + 3G/4G/5G Comms Antennae	1	0	0
10	Speed	m/s	20	± 20 m	GPS + Wi-Fi + Bluetooth + 3G/4G/5G Comms Antennae	1	0	0

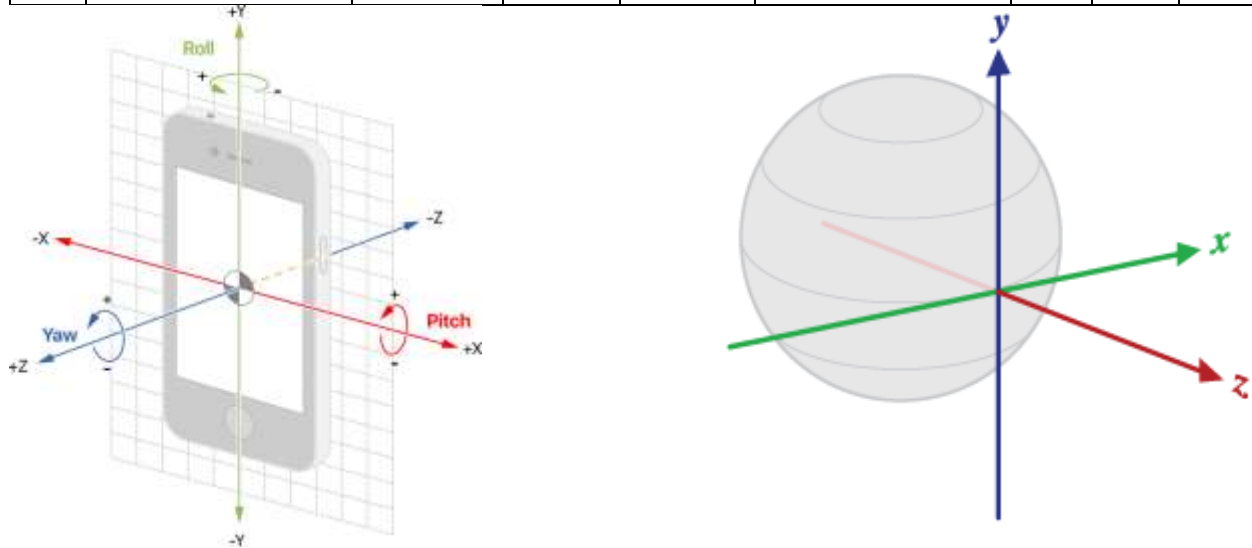


Figure 4: Neutral orientation of a smartphone aligned to world coordinates. Positive and negative values indicate the direction of the acceleration (g-force) applied. For android acceleration and rotation is measured in m/s^2 and degrees respectively, but in iOS g-force and radians are used, respectively.

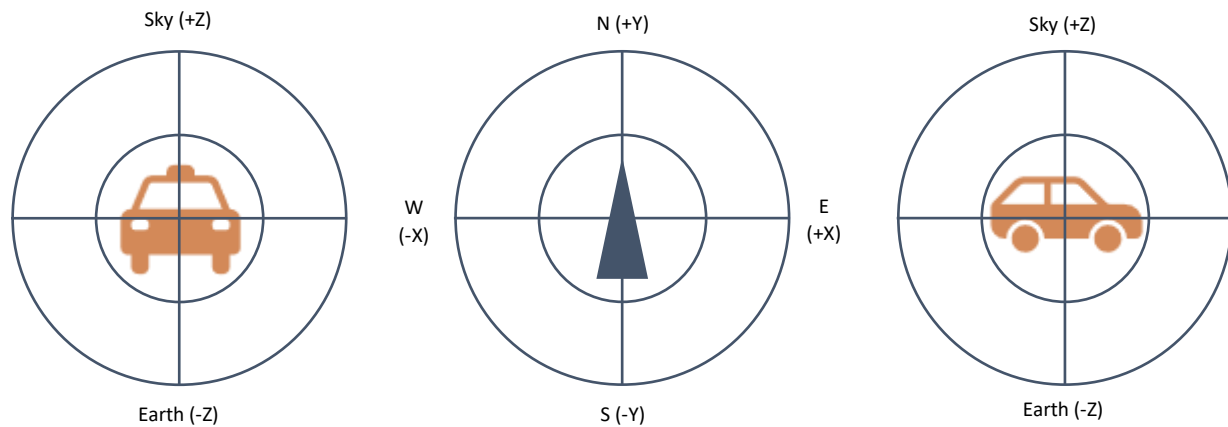


Figure 5: Neutral orientation of the smartphone superimposed over an object it is placed on, aligned to world coordinates.

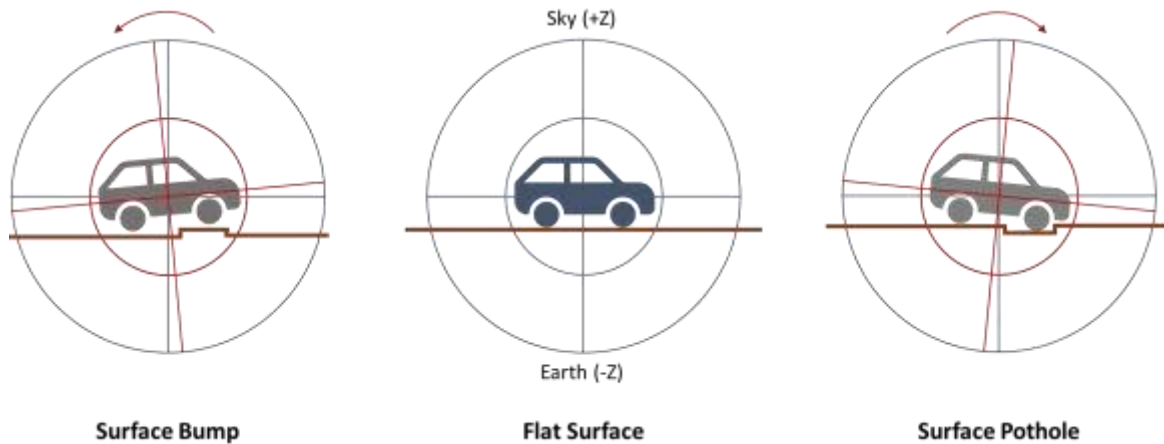


Figure 6: Pitch Rotation (on X-Axis) when encountering a surface bump or pothole due to the rocking action. The event would also cause a spike in the Z-axis acceleration due to the jerk caused. It should be noted that travelling uphill or downhill will cause a change in the Z-axis acceleration, however, such a change is expected to be much more gradual in comparison.

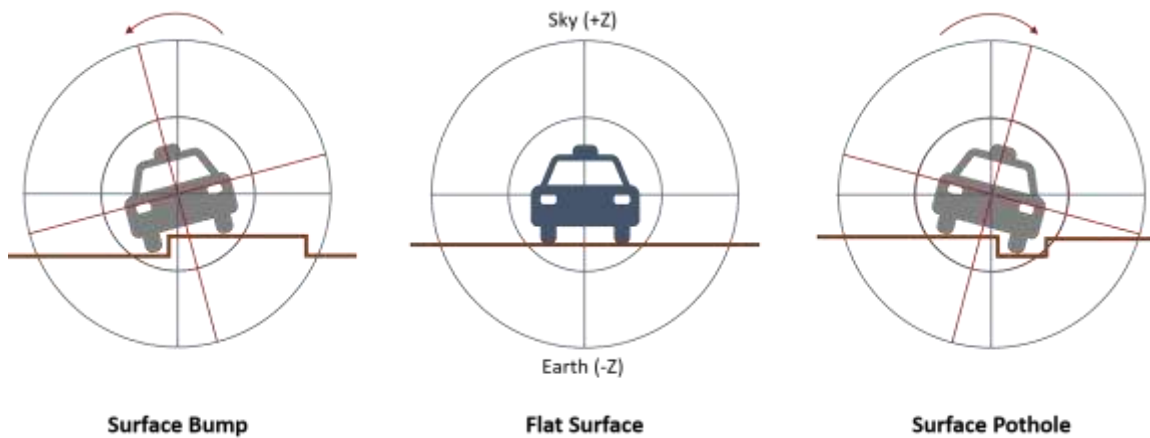


Figure 7: For non-single tracked vehicle, encountering a pothole or surface bump will result in a roll rotation of the Y-Axis. If only 1 tire is affected, a twisting action may be observed (rotation on the x and y axes). It should be noted that Y-Axis rotation does not apply to single-tracked vehicles like bicycles due to the wobbling and leaning action when balancing and turning.

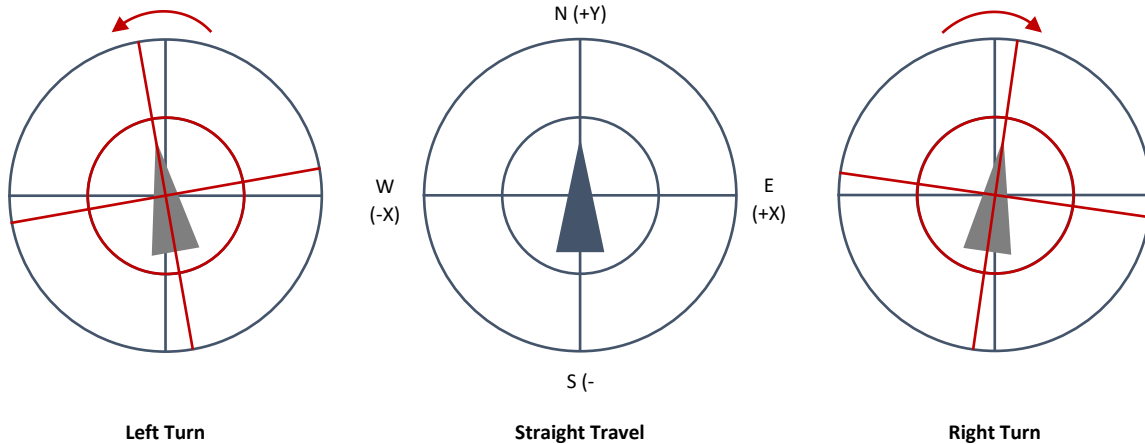


Figure 8: Yaw rotation on the Z-axis. Turning action due to drivers avoiding major road anomalies and normal travel would be impossible to distinguish apart based on accelerometer and gyroscope readings alone. Hence acceleration on the X-Y axes and rotation on the Z-axis is unlikely to be useful for our analysis. It should be noted that changes in the horizontal speed of the vehicle will also cause spikes in the acceleration on the X and Y axes e.g. through braking or accelerating from a stop.

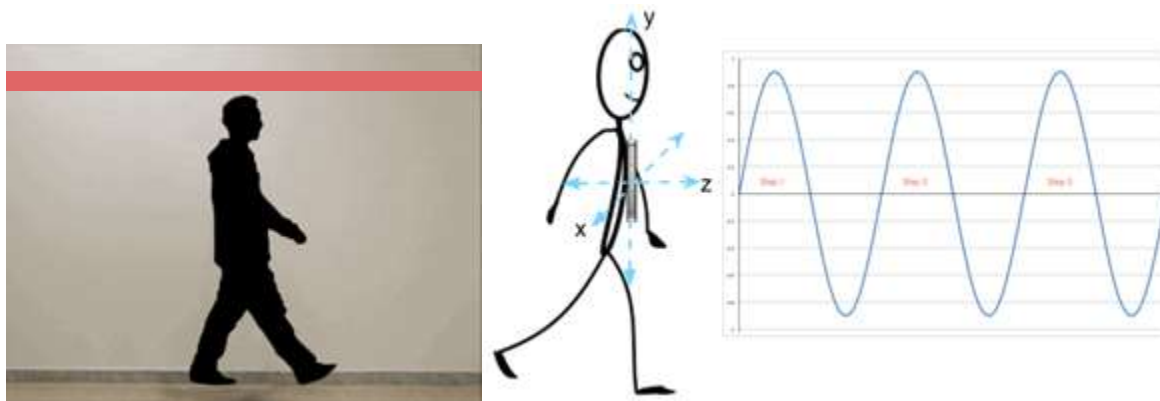


Figure 9: The bobbing motion caused by walking and running would produce a sine wave of sorts oscillating at different frequencies. Identifying this signature would allow one to identify walk or running patterns.

Figure 4 to Figure 9, illustrates the applications of the accelerometer and gyroscope. Changes to X and Y axes rotations are indicative of potential potholes and surface defect for vehicles travelling on the road; whilst patterns and changes in the Z-axis linear acceleration may be used for transport mode detection or road surface anomalies. However, it should be noted that all the sensor readings are taken based on the phone's current orientation (LePage, 2020) as seen in Figure 10, hence without proper orientation of the sensor readings to a common plane of reference such as the world coordinates, comparison of results would be extremely difficult if the phone's position is not locked consistently throughout all the recordings. Therefore, it would be imperative that one use the quaternion units provided by the phone OS to reorient the sensor readings (especially for acceleration values) to a common reference frame. Otherwise, the next best alternative would be to take the magnitude of the total linear acceleration for analysis, which loses any direction information on the forces applied to the different axes. Another challenge would be dealing with the effect of angular acceleration due to rotational forces when the vehicle/phone sensor is spinning or turning at high speeds. For safety and simplicity sake only readings from the linear motion will be taken, thus avoiding issues of angular acceleration.

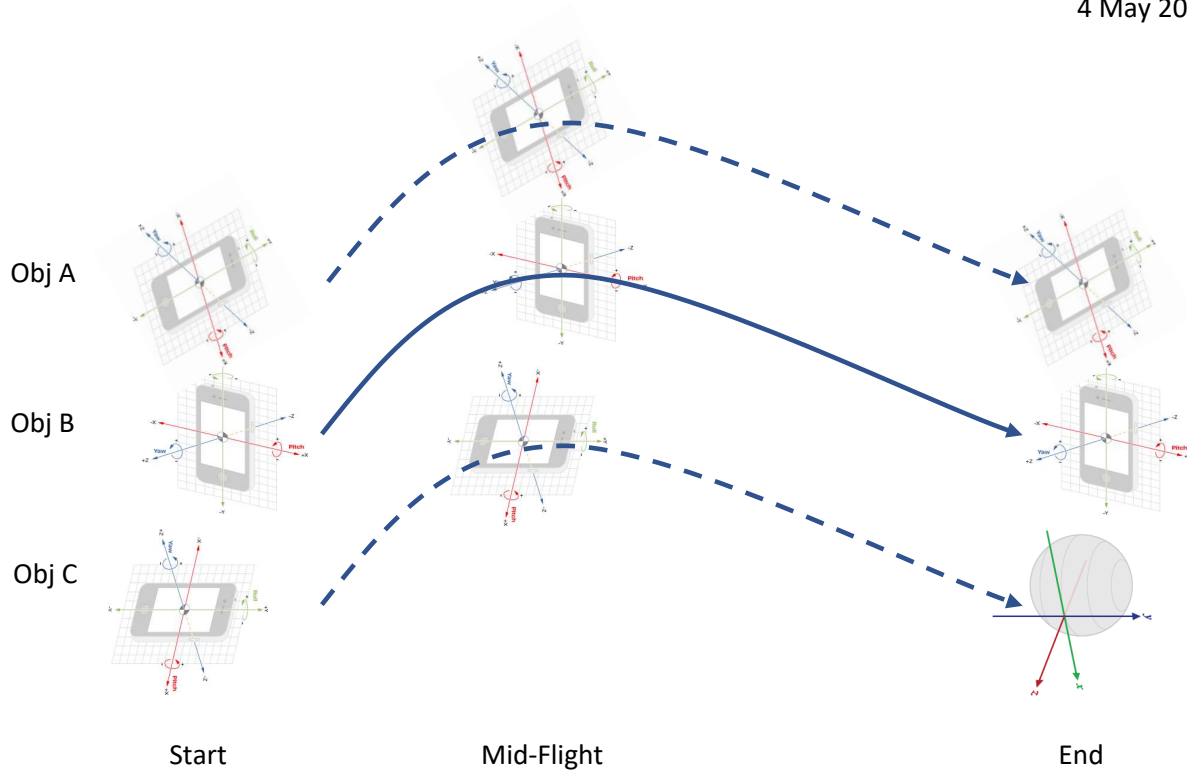


Figure 10: Without proper sensor reorientation, the sensor values will be off alignment on the true 3-axis state (as seen in Obj B), since readings are taken based on the phone's current orientation, thus making comparisons difficult across different recording events. If the phone is not placed in a locked position when the readings are taken, spin rotation may also be present on all 3-axis making standardized measurements even harder, especially with the introduction of angular acceleration.

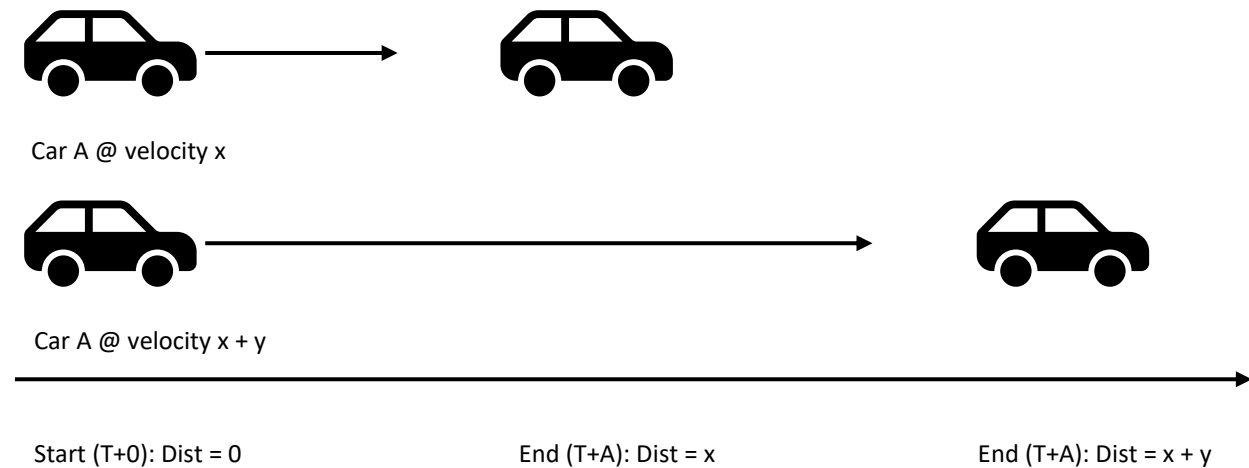


Figure 11: Amount of horizontal displacement is dependent on the velocity of the vehicle

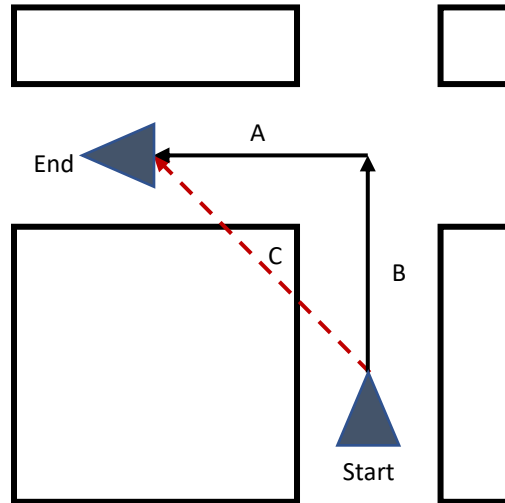


Figure 12: The use of Euclidean Distance for distance estimation (C) tends to underestimate the velocity travelled especially if there are a lot of turns and if the sampling frequency is too small; where the actual distance covered is $A + B$.

Another major sensor used would be the GPS sensor, especially for the tagging of road anomalies and speed (or velocity) estimation. It primarily works by getting a line-of-sight signal lock of four or more GPS⁸ satellites and other forms for telecommunications networks, to triangulate its location. In general, the more signal lock points acquired, the better the GPS accuracy. GPS would not be able to work well indoors, underground, underwater and shadow areas of valleys (natural or man-made) where GPS signals are obscured. The limited accuracy of GPS of estimated $\pm 10\text{m}$ would also limit one's ability to discern which lane or direction the road anomaly is found, requiring actual site surveys to verify the location. Also, one's displacement between the sampling intervals must exceed a minimum of 10m to reliably register any appreciable change in location by the GPS. This means that a longer sampling interval is required to capture displacement for slower moving objects.

Firstly, the GPS sensor is highly resource-intensive in terms of computing power and battery consumption, thus limiting continuous high-frequency sampling of GPS coordinates for long periods. Furthermore, activation of the phone's GPS would prevent the phone from putting other background applications to a power saver mode, resulting in even higher power consumption. As such, any long-term high-frequency sampling of GPS would likely require a connection to a power supply, though such a requirement would also limit the take-up rate in any actual app deployed, even if it is designed as an in-vehicle application with an additional power source

In the final deployment of the app, a low sampling frequency for the GPS used would have to be used to save power; and only activated upon pothole detection. However, the accuracy of such a reading is dependent on the time taken⁹ to compute whether a road anomaly exists, and the velocity of the vehicle. As seen in Figure 11, by the time the next reading is taken at time $T+A$ caused by the lag, the displacement would be equivalent to the velocity of the vehicle. Hence, there is a strong need to minimize any lag time

⁸ The GPS sensor may use networks of various geo-positioning satellite constellations such as GPS (USA), GNSS (EU), GLONASS (Russia) and Beidou (China) to get a better signal lock and location triangulation

⁹ This is a combination of the computational time required; and the amount of waiting time to get the minimum amount of observations in a time window before any calculation can be made.

in order of the GPS readings to be as accurate as possible. Below are some of the estimated displacement per second at some common vehicle velocity experienced during peak hour in Singapore:

1. Average speed on arterial roads (25km/hr): 6.94m/s
2. Average speed on expressways (65km/hr): 18.06m/s

GPS coordinates can also be used to estimate the average velocity of an object using the Euclidean Distance between two coordinates. The velocity information is useful, as previous studies have shown that the velocity of a vehicle traversing through a road anomaly or if the person is running vs walking, there will be an increase in the amplitude of the linear acceleration values. However, the use of Euclidean Distance has a limitation in urban environments where there are a lot of potential turning points since it will always underestimate the distances, taking distance C instead of distance A + B, as seen in Figure 12. One method to overcome it would be to use shorter sampling intervals to minimize the error, though computational power and power consumption will be of concern. Another method would be to combine velocity derived from Euclidean distance with acceleration information for more accurate velocity estimation, though such accuracy would not be needed for our project, hence this method combining acceleration data for velocity estimation was not explored, with the Euclidean Distance method being used instead. It should be noted that the Ellipsoids method is preferred over the Great-Circle method in terms of calculating geographical distances, as the former accounts for the ellipsoidal shape of the Earth. However, if one uses the phone's inbuilt function to extract speed based on GPS, one may sidestep the above issues, though it should be noted that GPS reception can vary, even with open urban spaces in Singapore.

Data Preparation and Processing Overview

Between the three modules being developed (TMD, TPD and RAD), we would be deploying largely the same methods to process the data since we are using the same datasets, though the differences in applications would affect how we finetune our data preparation steps and our end-stage feature engineering and modelling. The proposed process overview have been documented in Figure 13.

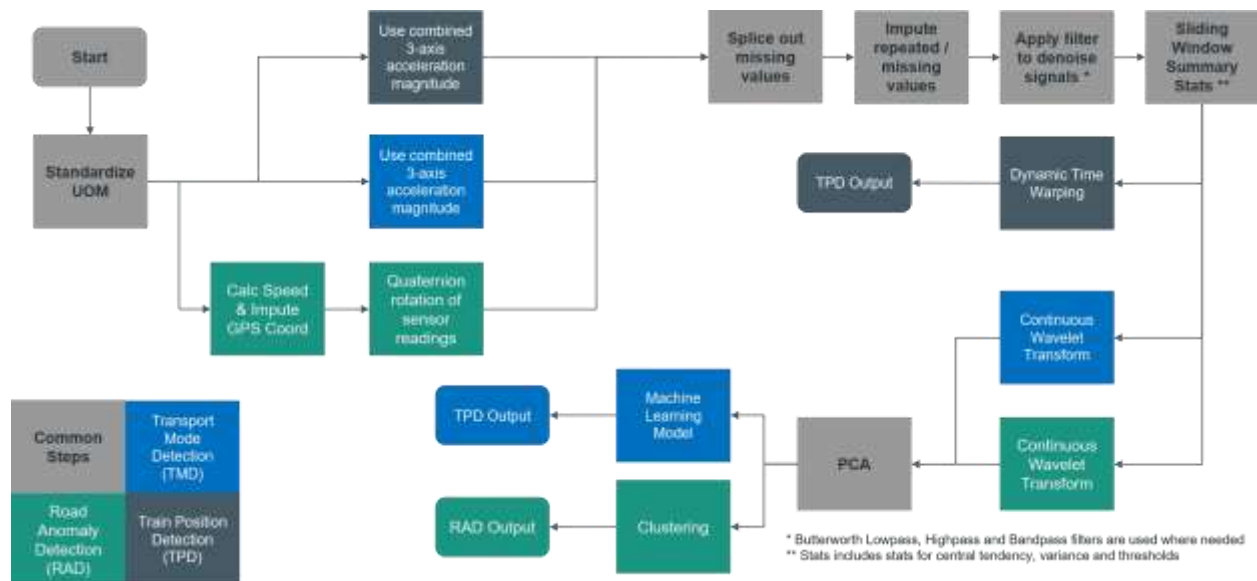


Figure 13: Proposed data processing and modelling overview. This process is subject to change as the project evolves.

It should be noted that for any signal processing work, the sensor records must be taken with approximately consistent timing intervals. Any missing values need to be imputed accordingly. Furthermore, the sampling frequency must follow the Nyquist Rate, whereby the sampling frequency must be at least more than two times higher than the frequency of the target wave pattern, to properly capture the waveform. Likewise, in the selection of the cut off frequencies for the application of frequency filters, the cut-off frequencies must be between $0 < x < \text{Nyquist Frequency}^{10}$.

Depending on the application, the choice of filter design and filter type would also differ. For instance, in the RAD project, an Infinite Impulse Rate (IIR) filter was chosen since due to the use of digital signals with discrete values for our analysis. A 6th order Butterworth design was also chosen¹¹, as it provides a relatively sharp cut-off transition with minimal rippling in the stopbands and passbands as compared to other filter designs such as a Bessel and Chebyshev. The choice of a lowpass, high-pass or bandpass filter is then dependent on the region of the target frequency and the expected regions where signal noise is found. Signal noise can come in the form of jerky braking and acceleration from regular driving, background engine vibration or the user fidgeting with their phone etc. A zero-phase filter was chosen, to enable the filter results to be synchronized in time consistently, however that will come at the expense of additional computing power as compared to a forward phase filter. Regardless of filter phase types, both methods would require a lag time to wait for additional forward values up to time $T+x$, before it can generate a result at time T , which may be an issue for real-time applications.

The following sliding window descriptive stats of the acceleration magnitude were chosen¹²:

1. Central tendency
2. Variance
3. Deviation from Threshold Point

The window size used is dependent on the following factors:

1. A lower sampling frequency would require a long window time
2. Estimated time for the target wavelet to render from start to end

Lastly, any use of PCA would primarily be for feature reduction or for transforming colinear variables only. Depending on whether these conditions are met, PCA may or may not be ultimately used.

Data Collection, Processing, Analysis and Findings

Transport Mode Detection

Data Collection

The Mobility Sensing (MS) App to collect smartphone sensor data of the various experimenters going about their usual routines in an unconstrained fashion¹³ with a manual classification tagging of common transport modes used, e.g. idle, walking, bus, MRT. The sampling rate used was 1Hz. Additional field studies were made to collect some specific transport mode data on purpose e.g. taking assorted

¹⁰ The Nyquist Frequency is $0.5 * \text{Sampling Frequency}$

¹¹ We used a 3rd order Second Order Series (SOS) filter for additional robustness; but the outputs are approximately equal to a convention 6th order filter ($n * 2$ order)

¹² Actual stats created during feature engineering may vary depending on application

¹³ No deliberate restrictions imposed to restrict phone movement of any sort

smartphone sensor readings on trains and cycling across different terrain to acquire better quality and varied data.

Table 2: Definition of different transportation modes

Mode	Definition
Idle	User is stationary; inclusive of waiting for transport and when the user is in a stationary vehicle
Walking	User is walking about
Running	User is running about
Bus	User is on the bus
MRT / LRT	User is on the train
Bicycle	User is on the bicycle
PMD	User is on the Personal Mobility Device (PMD) / personal motorised transport

Data Processing

Key data processing steps:

1. Filling those missing values using the values in the next or last row.
2. Using Timestamp to calculate Date and Time for indexing later.
3. Drop duplicate data points at the same time and just retain the first ones.
4. Eliminating feature dependence on the coordinate system
 - a. Accelerometer, Gyroscope and Magnetometer all have 3-Axis which are strongly related to the orientation of the device and hence, they may interfere modelling.
 - b. Without the quaternion units¹⁴, reorientation of those values to a common coordinate system would be impossible, hence the magnitude of those three features ($\sqrt{x^2 + y^2 + z^2}$) was used as it is an orientation-independent metric for feature engineering.
5. Removal of the effect of gravity
 - a. Since the raw acceleration does not make adjustments for gravity, an adjustment needs to be made for it.
 - b. In this case, it was found that the gravitational effect had a very high amplitude in the 0Hz in the frequency domain, masking the signals from the other frequencies as shown in Figure 14. Hence, a use Butterworth high pass filter to filter out the effect of gravity first. The filtered pattern is shown in Figure 15.

¹⁴ The MS app is does not provide the quaternion units for reorienting the sensor readings it its current version

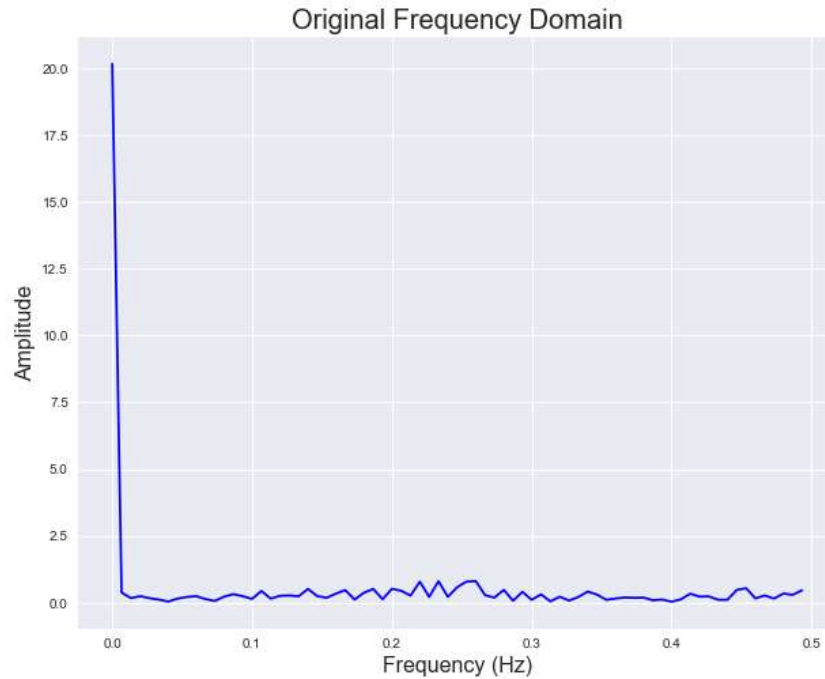


Figure 14: Original amplitude of accelerometer in the frequency domain

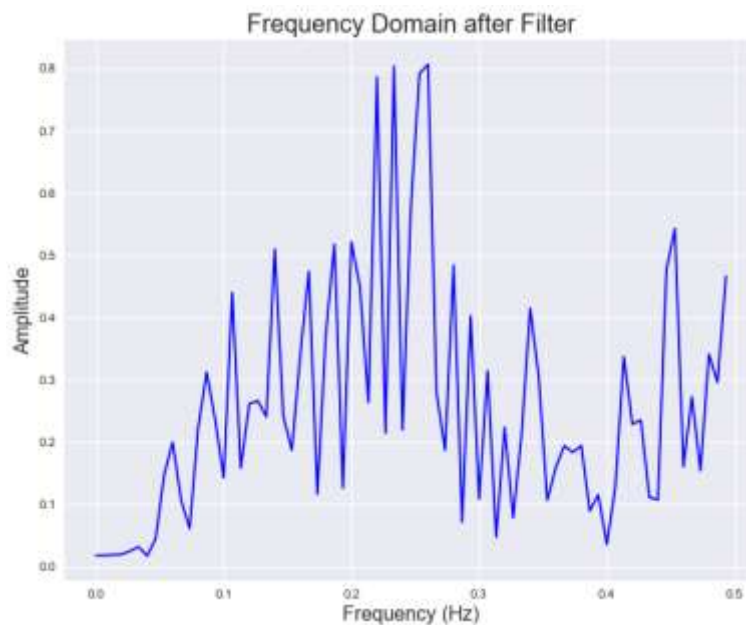


Figure 15: Filtered amplitude of accelerometer in the frequency domain

Feature Engineering

Taking reference from previous studies, a sliding window was applied, as shown in Figure 16, using the accelerometer, magnetometer and gyroscope sensor data. Trial and error were used to determine the impact window's overlap percentage and window size on the modelling results. Feature extraction was performed on the following domains: time, frequency, and wavelet.

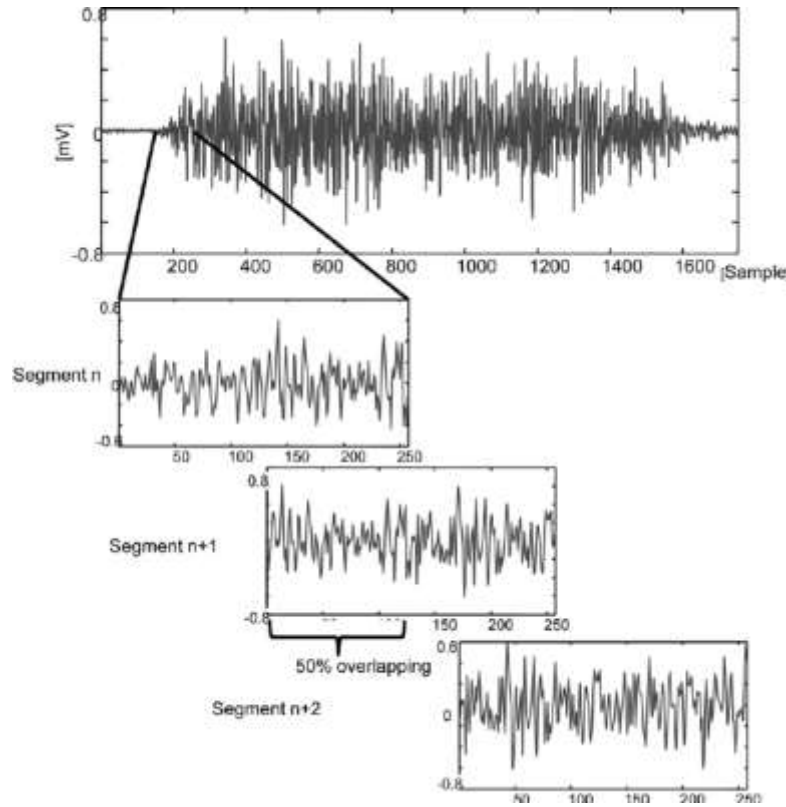


Figure 16: Sliding window example

As for time domain, the X-Axis is time, Y-Axis is the magnitude of features, as shown in Figure 17.

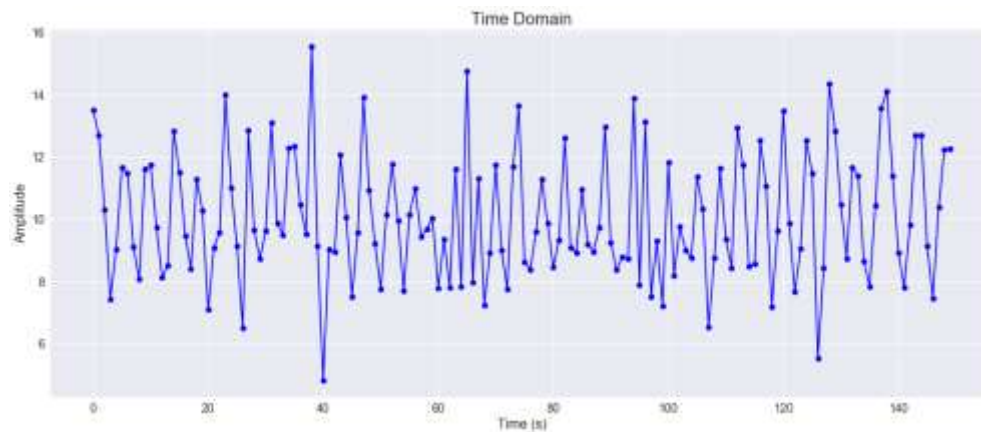


Figure 17 Acceleration in Time Domain

Descriptive stats were extracted on the time domain, as shown in Table 3.

Table 3: Time-domain Features Extracted

New Feature	Description
Min	The minimum sample value of the frame
Max	The maximum sample value of the frame
Medium	The middle sample value when samples are ordered as their values

Mean	The mean value of all samples in the frame
Range	The difference between min and max features
RMS	The value after taking the squares of each sample value, summing them all and dividing them to sample size of the frame and taking the squared value
STD	The measure that is used to find the amount dispersion of a set of data values according to mean
Variance	The sum of squared deviations of samples according to mean
Skewness	The measure of the skewness of samples according to mean
Kurtosis	The measure of the thickness or heaviness distribution of samples
Energy	The summation of each squared sample values and dividing the frame sample size

As for the frequency domain, Fourier Transform was used to transform the signal into the frequency domain. the X-Axis is frequency, Y-Axis is the magnitude of features, as shown in Figure 15. The following features were then extracted as shown in Table 4.

Table 4: Frequency domain features

New Feature	Description
Max Amplitude	The highest amplitude of frame in the frequency domain
Max Frequency	The frequency which has the highest amplitude of frame in the frequency domain
Ratio	The ratio between the highest amplitude of frame and the second-highest amplitude of frame in the frequency domain
Skewness	The measure of the skewness of samples against the window frame mean in the frequency domain
Kurtosis	The measure of the thickness or heaviness distribution of samples of the window frame in the frequency domain
Entropy	The Entropy of the frame's power spectral density in the frequency domain

Continuous Wavelet Transform was used to transform the acceleration signal into the wavelet domain, namely the wavelet domain coefficients (amplitude) split by wavelet scale over time as shown in Figure 18. Smaller scales correspond to high-frequency signals, whilst larger scales correspond to lower frequency signals. Since the coefficients in wavelet domain are 3-dimensional as shown in Figure 19, Principal Component Analysis (PCA) is applied for dimension reduction (the first Principal Component is retained) shown in Figure 20 before modelling is performed.

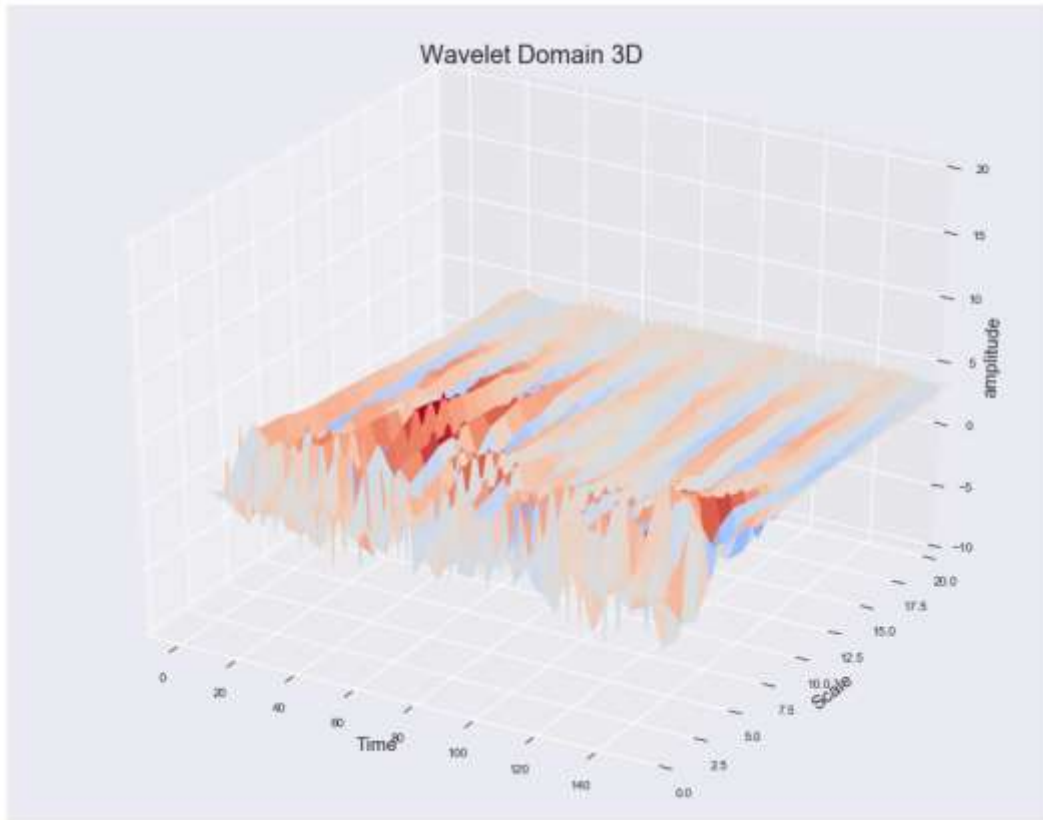


Figure 18: Wavelet domain coefficients

	Time:0	Time:1	Time:2	Time:3	Time:4	Time:5	Time:6	Time:7	Time:8	Time:9
Scale:2	0.617552	-0.811131	0.481634	0.372034	-1.038034	1.093405	-0.786944	0.297983	0.224813	-0.445863
Scale:3	-0.528094	-0.403459	0.921514	0.495581	-0.810323	-0.457934	0.172036	0.501110	0.252426	-0.409119
Scale:4	-0.784219	0.511799	1.552414	0.672351	-1.226919	-1.666302	-0.105339	1.369088	1.055057	-0.280598
Scale:5	0.135911	1.432353	1.684119	0.254898	-1.623350	-1.966440	-0.681842	1.195675	1.769759	0.756230
Scale:6	0.998716	1.695703	1.241012	-0.182764	-1.672247	-1.941092	-0.963225	0.706242	1.632524	1.450260
Scale:7	1.367839	1.411422	0.684675	-0.487520	-1.515271	-1.660119	-1.052052	0.219767	1.198429	1.481763
Scale:8	1.226685	0.982828	0.271101	-0.617443	-1.283026	-1.414248	-0.947567	-0.100730	0.738793	1.208365
Scale:9	0.987354	0.620365	0.006601	-0.675745	-1.055134	-1.153639	-0.826690	-0.320490	0.436322	0.905224
Scale:10	0.681685	0.337488	-0.162615	-0.609647	-0.833577	-0.971534	-0.698075	-0.321613	0.184379	0.546181
Scale:11	0.429881	0.177199	-0.196777	-0.541696	-0.745913	-0.845943	-0.594096	-0.317768	0.069921	0.386241
Scale:12	0.221401	0.113169	-0.275483	-0.510160	-0.535615	-0.700251	-0.471670	-0.339884	-0.041646	0.297471
Scale:13	0.170944	0.037714	-0.230235	-0.411218	-0.531540	-0.542888	-0.456090	-0.261516	-0.063774	0.227708
Scale:14	0.096703	-0.144676	-0.152682	-0.313887	-0.533094	-0.373661	-0.432072	-0.207046	-0.045037	0.008328

Figure 19: Original wavelet domain coefficients

	Scale:2	Scale:3	Scale:4	Scale:5	Scale:6	Scale:7	Scale:8	Scale:9	Scale:10	Scale:11	Scale:12	Scale:13	Scale:14
1st_Component	-1.995045	-1.145935	0.754346	2.27961	2.485418	1.779925	0.879905	0.114086	-0.501682	-0.820153	-1.122905	-1.247114	-1.480457

Figure 20: Wavelet domain features

Motorized Transportation Mode Detection (1Hz)

For the initial modelling, a sampling frequency of 1Hz was used, a sliding window with a 60s window size and 1 step overlap was used. Only time-domain features based on sensor readings from the 3-axis accelerometer, 3-axis rate of rotation and 3-axis magnetometer were used to create a binary classification model: motorized and non-motorised. The detection results are shown in Figure 21.

All the modelling methods used performed well, possibly since it was a simple binary classification model and the features in the time domain are enough to differentiate them. Random Forest, Extremely Randomized Trees, XGBoost models performed best, hence, future work would be focusing on bagging and boosting type models.

Model \ Metric	Accuracy	Precision	Recall	F1
Naïve Bayes	0.8233	0.84	0.82	0.82
Logistics Regression	0.8413	0.85	0.84	0.84
Decision Tree	0.9464	0.95	0.95	0.95
Random Forest	0.9720	0.97	0.97	0.97
Extra Trees	0.9719	0.97	0.97	0.97
GBDT	0.9178	0.92	0.92	0.92
XGBoost	0.9656	0.97	0.97	0.97
Neural Network	0.9136	0.92	0.91	0.91
SVM	0.8527	0.86	0.85	0.85
Adaboost	0.9020	0.90	0.90	0.90
Vote	0.9402	0.94	0.94	0.94

Figure 21: Initial modelling results

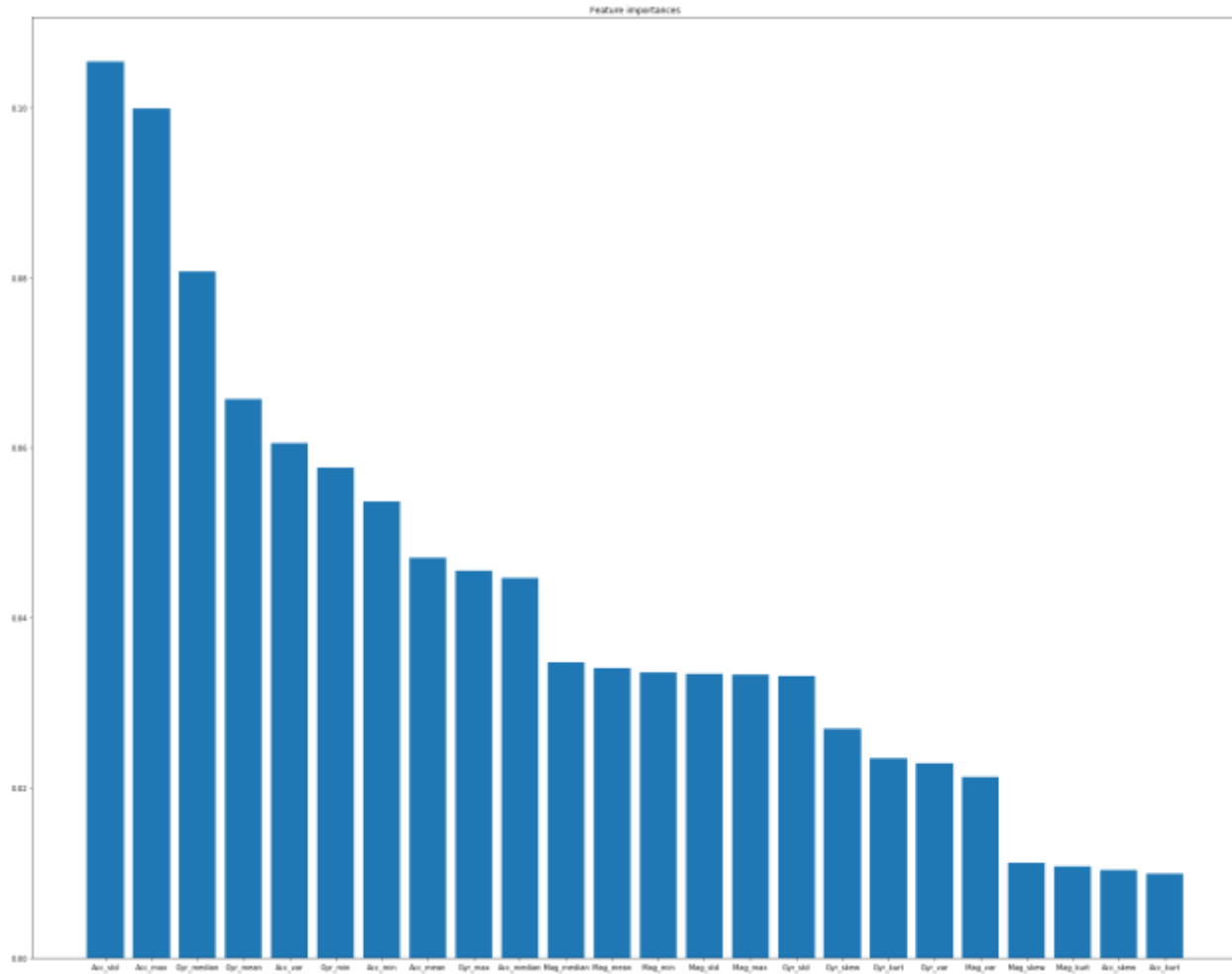


Figure 22: Feature importance

Based on the feature importance plot, the minimum, maximum, median, mean, variance, standard deviance of accelerometer and gyroscope were found to be the most useful features to differentiate between Motorised and Non-Motorised transportation modes; whilst skewness, kurtosis and all of the magnetometer features are deemed as less important. Magnetometer features are not deemed important as they only provide the device orientation in relation to the Earth's magnetic North pole, whereas transportation mode is direction independent, and hence unrelated.

Multi-Class Transportation Mode Detection (1Hz)

For the second model, a sampling window of 1Hz, with a sliding window with a 60s window size and 1 step overlap was used again. Features were extracted of accelerometer and gyroscope readings in the time and frequency domains to build multi-class classification model for the following classes as seen in Figure 23: Idle, Walking, Cycling, MRT, Car, Bus.

Ensemble models were found to perform better than classical models with higher metric scores; and LightGBM model being the best with the highest metric scores (Accuracy 0.7854, Precision 0.79, Recall 0.78, F1 0.78). The confusion matrix of LightGBM model is shown in Figure 24. We could find the model

was good at detecting all modes except Bus, which would mistakenly identify the Bus mode as the MRT mode. Maybe it because Bus and MRT have similar transport patterns.

Model \ Metric		Accuracy	Precision	Recall	F1
Classical	Naïve Bayes	0.5857	0.6200	0.5900	0.5700
	Logistics Regression	0.5970	0.6100	0.6000	0.5900
	Decision Tree	0.7289	0.7500	0.7300	0.7200
	Neural Network	0.6857	0.7200	0.6900	0.6700
	SVM	0.6476	0.6500	0.6400	0.6400
Ensemble	Random Forest	0.7713	0.7700	0.7600	0.7600
	Extra Trees	0.7720	0.7700	0.7600	0.7600
	Hist-GBDT	0.7747	0.7700	0.7700	0.7600
	XGBoost	0.7629	0.7500	0.7500	0.7400
	LGBM	0.7854	0.7800	0.7800	0.7800
	Vote	0.7887	0.7900	0.7800	0.7800
	Stacking	0.7534	0.7400	0.7300	0.7300

Figure 23 Second Models' Results (Trained in 1 Hz data on window size 60, and overlap 1 step)

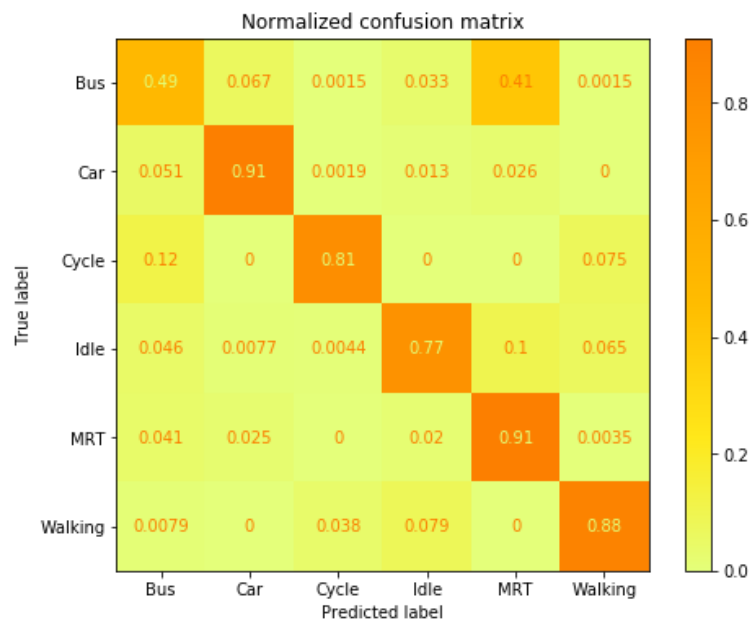


Figure 24 LightGBM Model's Confusion Matrix

However, when the LightGBM model tested with unseen test data, its performance was not stable as shown in Figure 25 and Figure 26.

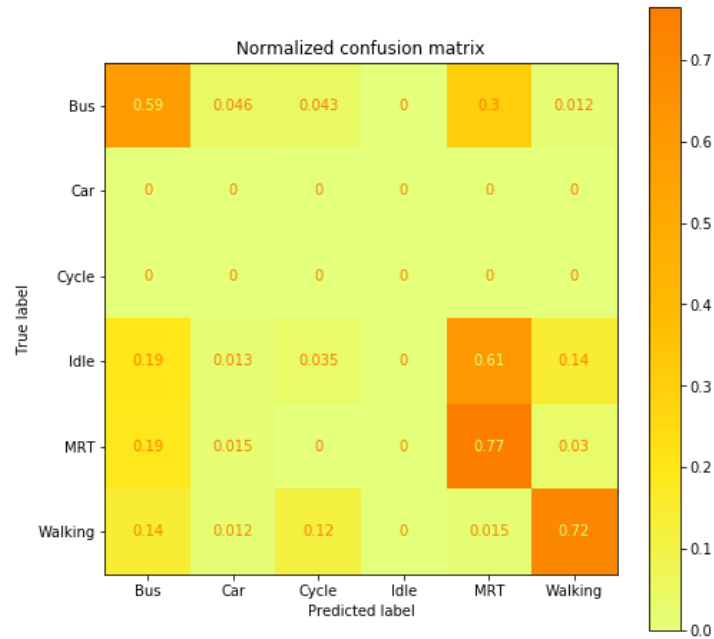


Figure 25 Confusion Matrix on Unseen Data 1

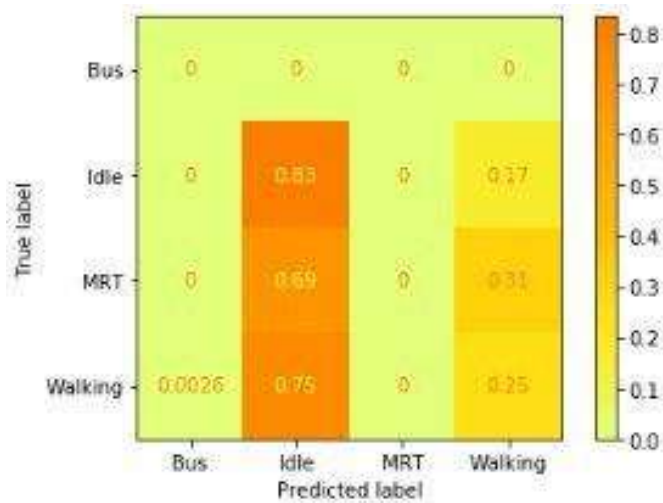


Figure 26 Confusion Matrix on Unseen Data 2

This could be due to the sampling rate not being high enough to satisfy the Nyquist Rate requirements. Hence, the sampling rate of the MSP app was increased 50Hz to catch more granular activity information that not previously captured at 1Hz.

Multi-Class Transportation Mode Detection (50Hz)

Despite increasing the sampling rate to 50Hz (with the same methodology applied), the prediction results remained unstable, as shown in Figure 27, Figure 28 and Figure 29.

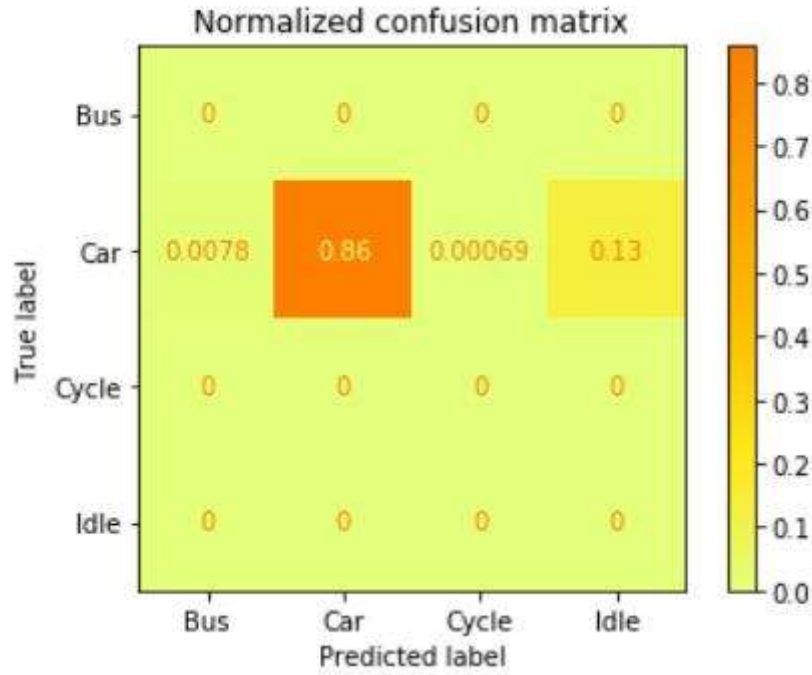


Figure 27 Test 1: User 9 Car Data

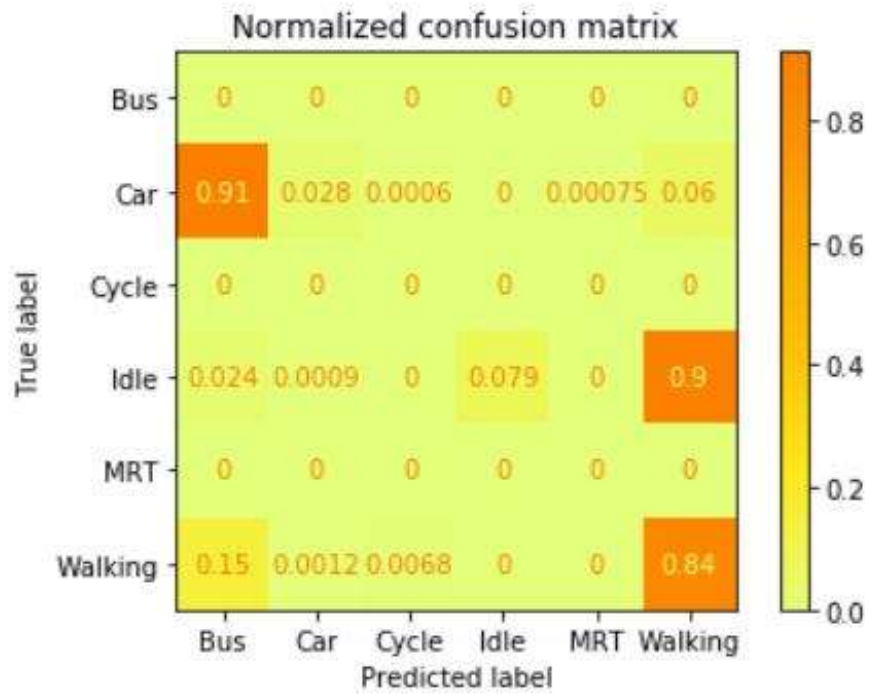


Figure 28 Test 2: User 9 Car, Walk, Idle Data

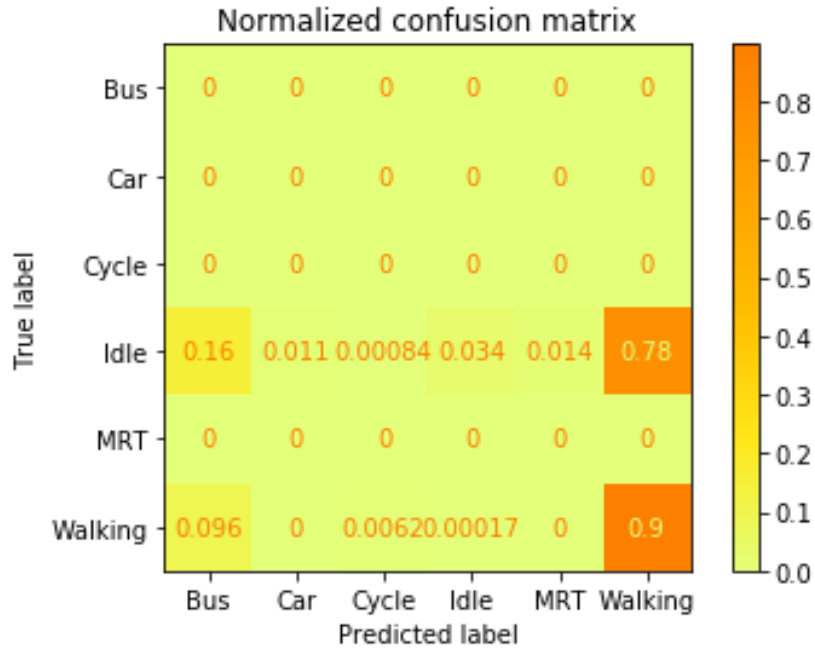


Figure 29 Test 3: User 45 Walk, Idle Data

As for the result of Test 1, it is suspected that since the car data in the initial model was largely contributed by User 9 resulting in a bit of overfitting. As for the result of Test 3, I think it because User 45 did not contribute walk and idle data in the initial model, and hence the model was overfitted to the other users. Essentially, more data needs to be collected to allow the model to generalise better to other users.

Multi-Class Transportation Mode Detection (50Hz + Differing Sliding Window Settings)

For this test, the same settings are used as the Multi-Class Transportation Mode Detection model with a sampling 50Hz sampling frequency. But this time, different sliding window sizes and overlap percentages are used to investigate our they impact the model's accuracy. The candidates are shown in Figure 30.

Candidate			
Window Size	1s	1.5s	2s
Overlap Percent	50%	70%	90%

Figure 30 Candidate Parameter

Key considerations:

1. Larger window size provides more information
2. Smaller Overlap Percent improves calculation speed, and produce more diverse feature values as shown in the comparison in Figure 31 with Figure 32
3. Larger window and smaller overlap will result in a fewer number of samples.

Hence needs to perform this test to determine the optimum trade-off between accuracy and computational speed. Currently, due to the limitations of small data size¹⁵, a larger window size (> 2s) and smaller overlap percentage (< 50%) cannot be selected as it will result in too few samples for building machine learning models.

Acc_Mag_Time_Max	Acc_Mag_Time_Min	Acc_Mag_Time_Mean	Acc_Mag_Time_Median
14.60517	6.846441	9.595267	9.341045
14.60517	6.846441	9.549934	9.316164
14.60517	6.323796	9.522882	9.316164
14.60517	6.323796	9.428885	9.282014
14.60517	6.323796	9.367624	9.282014
14.60517	6.323796	9.411463	9.316164

Figure 31 Overlap 1 Step

Acc_Lin_Mag_Time_Max	Acc_Lin_Mag_Time_Min	Acc_Lin_Mag_Time_Mean	Acc_Lin_Mag_Time_Median
16.344998	0.396832	2.684319	2.112793
16.344998	0.238000	1.584998	0.833678
4.603701	0.238000	1.241438	0.822982
6.821385	0.341033	1.804549	1.540366
6.821385	0.293309	1.619955	1.372488
5.437729	0.115935	1.284010	1.092606

Figure 32 Overlap 50%

Similar to the previous models, features were extracted from the accelerometer and gyroscope sensors in the time and frequency domain for building a multi-class classification model to determine the following transportation modes: Idle, Walking, Cycling, MRT, Car, Bus. The detection results are shown in Figure 33.

¹⁵ The COVID-19 circuit breaker lockdown measures in place from April to June 2020 during the experiment period severely limited ability of the team to collect fresh transportation data at 50Hz

Parameter		Window 1s Overlap 50%	Window 1s Overlap 70%	Window 1s Overlap 90%	Window 1.5s Overlap 50%	Window 1.5s Overlap 70%	Window 1.5s Overlap 90%	Window 2s Overlap 50%	Window 2s Overlap 70%	Window 2s Overlap 90%
Model										
Classical	Naïve Bayes	0.4800	0.4768	0.4847	0.4854	0.4834	0.4831	0.4930	0.4829	0.4922
	Logistics Regression	0.5021	0.4926	0.5109	0.5081	0.4713	0.4943	0.4688	0.4467	0.4894
	Decision Tree	0.5573	0.5455	0.5529	0.5636	0.5609	0.5631	0.5605	0.5966	0.5699
	Neural Network	0.5352	0.5461	0.5554	0.5800	0.5368	0.6131	0.5242	0.5594	0.6307
	SVM	0.4401	0.4859	0.4808	0.4073	0.4555	0.4579	0.3633	0.4084	0.4317
Ensemble	Random Forest	0.5955	0.5929	0.5799	0.6090	0.6046	0.6072	0.6211	0.6256	0.6096
	Extra Trees	0.5904	0.5944	0.5907	0.6040	0.5888	0.6108	0.6107	0.5873	0.6241
	GBDT	0.5726	0.5883	0.5768	0.5889	0.6031	0.6160	0.6141	0.6049	0.6158
	XGBoost	0.5811	0.5980	0.5809	0.6090	0.6099	0.6112	0.6089	0.6215	0.6003
	Adaboost	0.2650	0.3025	0.4802	0.2572	0.2823	0.5600	0.4394	0.4508	0.4100
	Vote	0.5590	0.6006	0.5698	0.6052	0.6114	0.6018	0.6245	0.5832	0.5644

Figure 33 Different Models' Results(Trained in 50 Hz Data on Different Window Sizes and Overlap Percents)

It was found that the Neural Network model in window 2s and overlap 90% and Random Forest model in window 2s overlap 70% performed best; their classification reports are shown in Figure 34 and Figure 35.

	precision	recall	f1-score	support
Bus	0.57	0.92	0.70	626
Car	0.54	0.40	0.46	792
Cycle	0.06	0.13	0.09	15
Idle	0.50	0.51	0.51	372
MRT	0.89	0.82	0.85	826
Walking	0.53	0.23	0.32	264
accuracy			0.63	2895
macro avg	0.51	0.50	0.49	2895
weighted avg	0.64	0.63	0.62	2895

Figure 34 Neural Network Model's (in Window 2s and Overlap 90%) Classification Report

	precision	recall	f1-score	support
Bus	0.57	0.73	0.64	210
Car	0.59	0.65	0.62	265
Cycle	0.00	0.00	0.00	5
Idle	0.29	0.19	0.23	123
MRT	0.82	0.86	0.84	276
Walking	0.58	0.20	0.30	88
accuracy			0.63	967
macro avg	0.48	0.44	0.44	967
weighted avg	0.61	0.63	0.61	967

Figure 35 Random Forest model's (in Window 2s Overlap 70%) Classification Report

Both models performed well in detecting MRT since the collected MRT data is very clean while they performed poorly in detecting Cycle mode. This is since there was insufficient cycling data which cause data imbalance problem, where the models deemphasised the Cycle mode class.

Comparing the performance of ensemble models with the Neural Network model with different window size and overlap percentage settings, it was found that:

1. For the same overlap percentage and different window size situation, having a larger window size tends to result in higher accuracy
2. For the same window size and different overlap percentage situation, different overlap percentage have little impact on accuracy for Ensemble models, whilst Neural Network only performed well when overlap percentage is high.

Overall, Ensemble Models was the best performer, balancing accuracy and computational speed.

Limitations

3 key limitations were identified, primarily due to the data collection process, namely:

1. Due to COVID-19 circuit breaker lockdown measures during the experiment period, limiting data collection, insufficient data was collected for building a sufficient robust model, only several hours each mode was collected at 50Hz, which is much lower than 1Hz where a lot more data has been amassed before the circuit measures in place. Hence, the models built based on 50Hz data performed worse than the original models built based on 1Hz data.
2. Only a few experimenters were involved in the data collect, resulted in the data overfitted to their movement patterns
3. Data collected in an unconstrained fashion i.e. not mounted to a stable stationary object, has a significant amount of noise, obscuring target signals, making modelling harder

Future Work

Now that the modelling method has been proven to work, the following aspects could be further explored:

1. More data needs to be collected for different transport modes with a larger base of experimenters collecting data, to create a more robust model for can be generalised

2. Use of quaternion rotation to orientate sensor values to world coordinates to retain the original vector information for modelling purposes
3. Calculating Velocity Based on Linear Acceleration to add another robust attribute for mode detection. Velocity estimation based on the formula $V_t = V_{t-1} + a\Delta t$ was attempted, but the results in Y-Axis and Z-Axis are not reliable. As shown in Figure 36, Figure 37 and Figure 38. The X-Axis is speed (m/s), Y-Axis is frequency. More research would be needed to test the viability of Inertial Navigation System (INS) technology on smartphone sensors, though it is likely that the precision of existing smartphone sensors would be a key limiting factor for now.

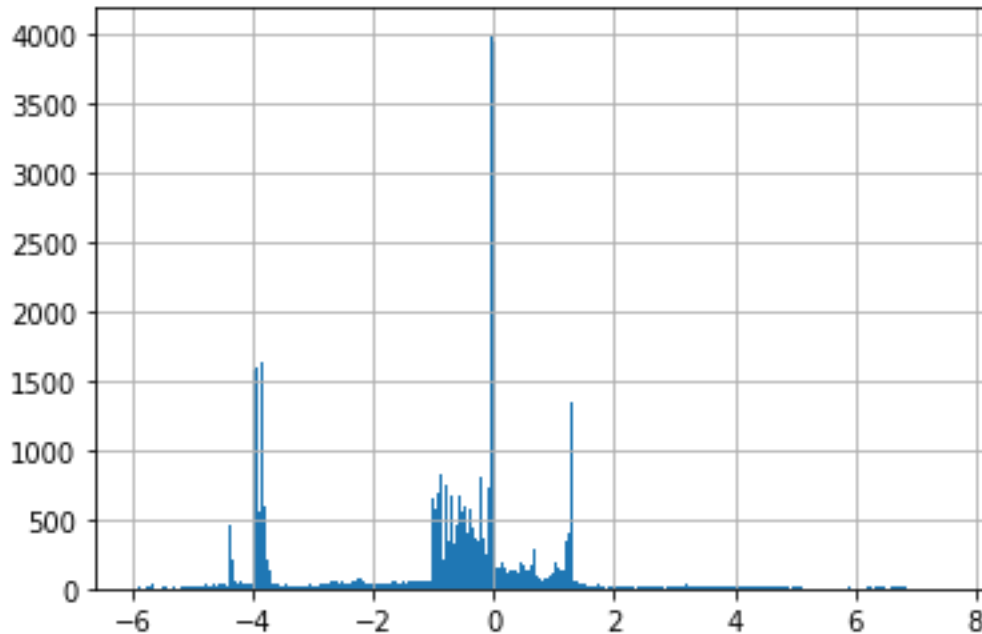


Figure 36 Velocity in X-Axis

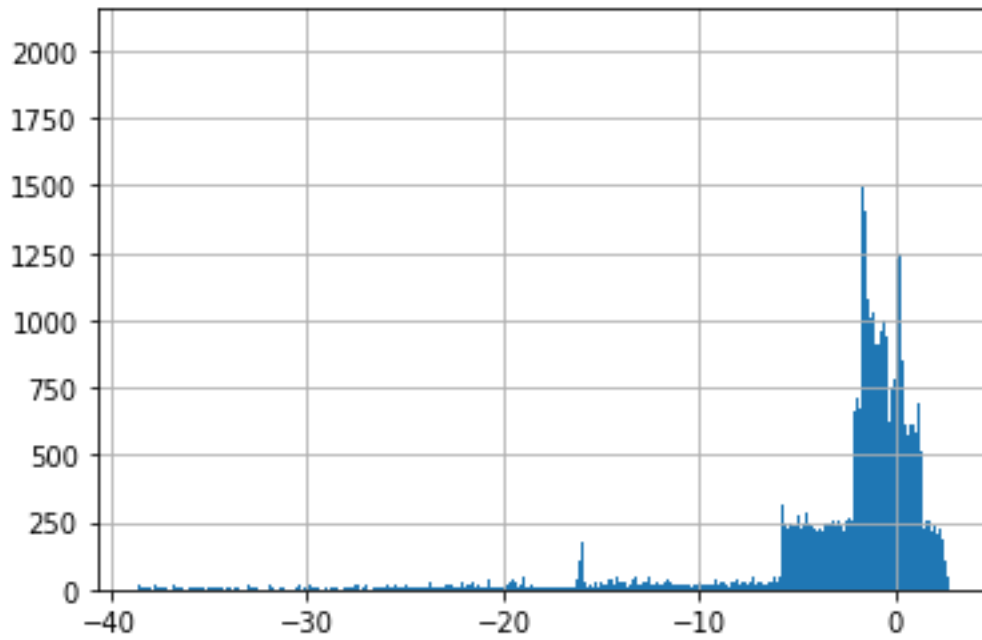


Figure 37 Velocity in Y-Axis

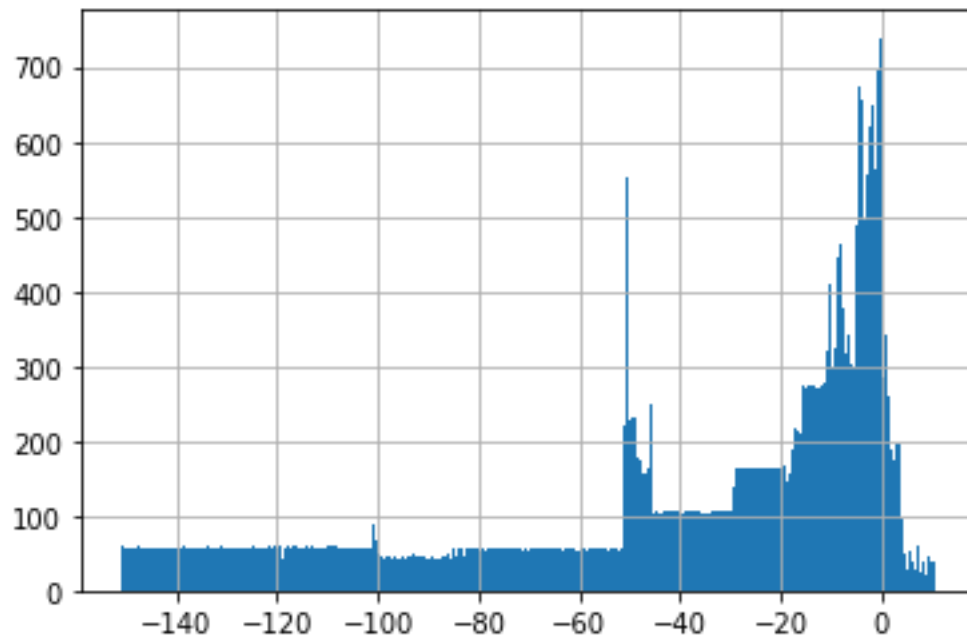


Figure 38 Velocity in Z-Axis

Train Position Detection

Data Collection

In this first phase of the project, data was collected by travelling on driverless MRT line, particularly, on the Downtown Line and NorthEast Line. Data was collected by travelling through 10 stations from Bedok North to Chinatown on the Downtown Line. The following are station-to-station journeys taken to record accelerometer data on smartphones:

Table 5: Station-to-Station journeys (Bedok North to Chinatown)

Journeys	Start	End
1	Bedok North	Kaki Bukit
2	Kaki Bukit	Ubi
3	Ubi	Macpherson
4	Macpherson	Mattar
5	Mattar	Geylang Bahru
6	Geylang Bahru	Bendemeer
7	Bendemeer	Jalan Besar
8	Jalan Besar	Bencoolen
9	Bencoolen	Fort Canning
10	Fort Canning	Chinatown

To confirm that the DDTW technique can indeed differentiate the same routes and different routes, data was also obtained from travelling on another train line, North-East Line from Chinatown to Little India. Differentiable scores generated for same and different route comparison would signal DDTW's ability to discriminate between routes. The stations of the journey on the North-East Line that act as a different route for comparison are as follows:

Table 6: Station-to-Station journeys (Chinatown – Little India)

Journeys	Start	End
1	Chinatown	Clarke Quay
2	Clarke Quay	Dhoby Ghaut
3	Dhoby Ghaut	Little India

Subsequently, with the Mobility Sensing App that was provided by LTA, accelerometer data was collected by on various smartphones from travelling on the Downtown Line between Bedok North station and Chinatown station. As the shortest route anyone can take on the MRT is one-stop, the experiment was done on station-to-station journeys between Bedok North and Chinatown. This is to see if the collection frequency was sufficient to collect key acceleration signal patterns to be matched with DDTW and successfully identify where the passenger is on the train line.

Key Technique Used

Derivative Dynamic Time Warping is applied here in the matching of cleaned acceleration signals to determine the level of similarity between two signals.

Dynamic Time Warping is traditionally used for speech recognition as it can determine the level of similarity between two audio clips. The following diagram illustrates two different audio clips of the same sentence spoken by two different people. Dynamic Time Warping can measure the distance between every point of the signal and produce a score where the lower the score indicates a higher level of similarity between the two signals.¹⁶

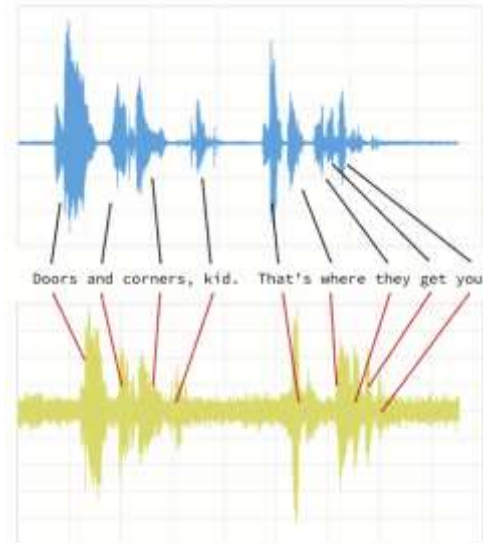


Figure 39: Illustration of Dynamic Time Warping on audio clips for speech recognition

Dynamic Time Warping can be applied to acceleration signals collected from smartphones. By matching the acceleration signals collected on the train journey with the acceleration signals taken by mounting the smartphone on the train to collect pure train acceleration signals, the user's location on the train track can be determined by comparing the similarity scores produced. Using DDTW as suggested by the research paper (Khuong An Nguyen, 2019), normalised scores will be produced. The more similar the two signals, the lower normalised DDTW score. If the DDTW algorithm can help one differentiate same and different routes, same route comparisons should produce consistently lower normalised DDTW scores as compared to different route comparisons.

According to the same research paper written in London (Khuong An Nguyen, 2019), the team attempted to apply Dynamic Time Warping to match acceleration signals, but it did not work well as Dynamic Time Warping tends to over-warps certain points where the variability on Y-axis is higher on the corresponding signal. Therefore, a specific Dynamic Time Warping technique known as Derivative Dynamic Time Warping was used instead.

¹⁶ Source: <https://pages.databricks.com/rs/094-YMS-629/images/dynamic-time-warping-background.html>

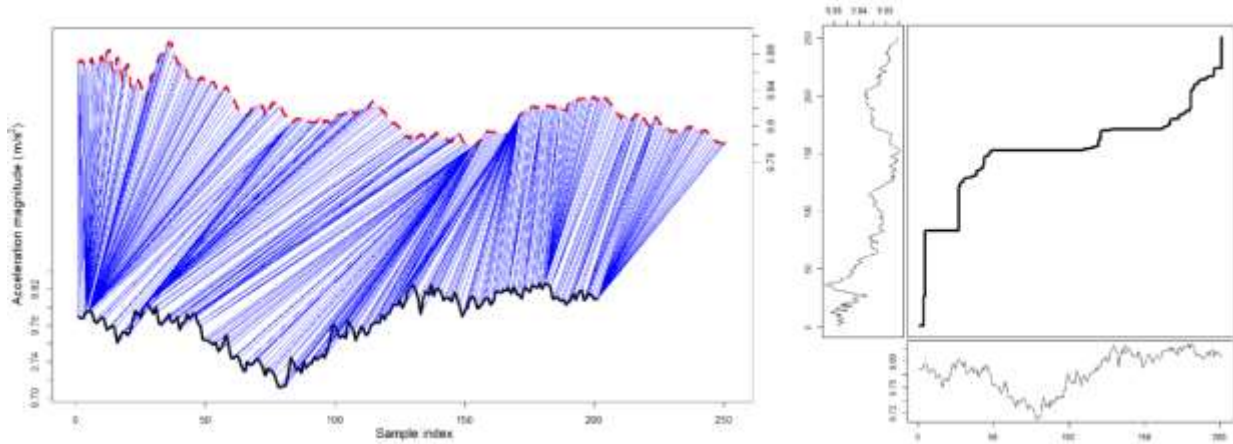


Figure 40: Illustration of Dynamic Time Warping on Train Acceleration Signals

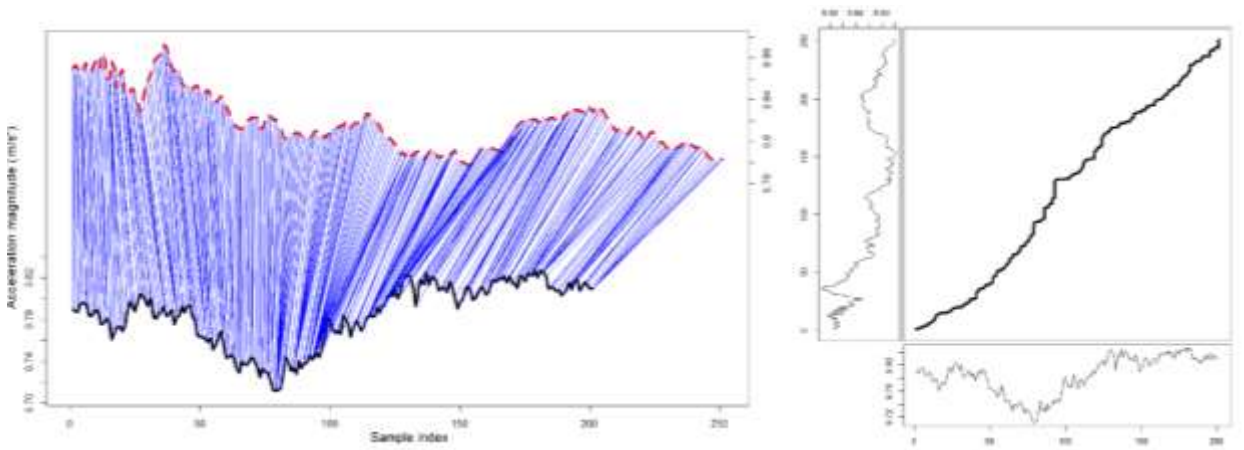


Figure 41: illustration of Derivative Dynamic Time Warping on Train Acceleration signals

While Dynamic Time Warping technique does not produce an ideal diagonal warping path, the Derivative Dynamic Time Warping technique helps to overcome the over-warping issue of Dynamic Time Warping producing an ideal diagonal warping path.

Key Assumption - One-stop or station-to-station Journeys Duration Stability

One of the key assumptions for using this pattern matching method on acceleration signals to identify the location in the train track is that the train is driver-less and hence, exhibits a certain consistent and predictable pattern of accelerations and decelerations. To determine if the train is indeed moving consistently, time was taken for one-stop or station-to-station journeys were recorded over 6 days and observations can be visualized in a graph of multiple boxplots as shown:

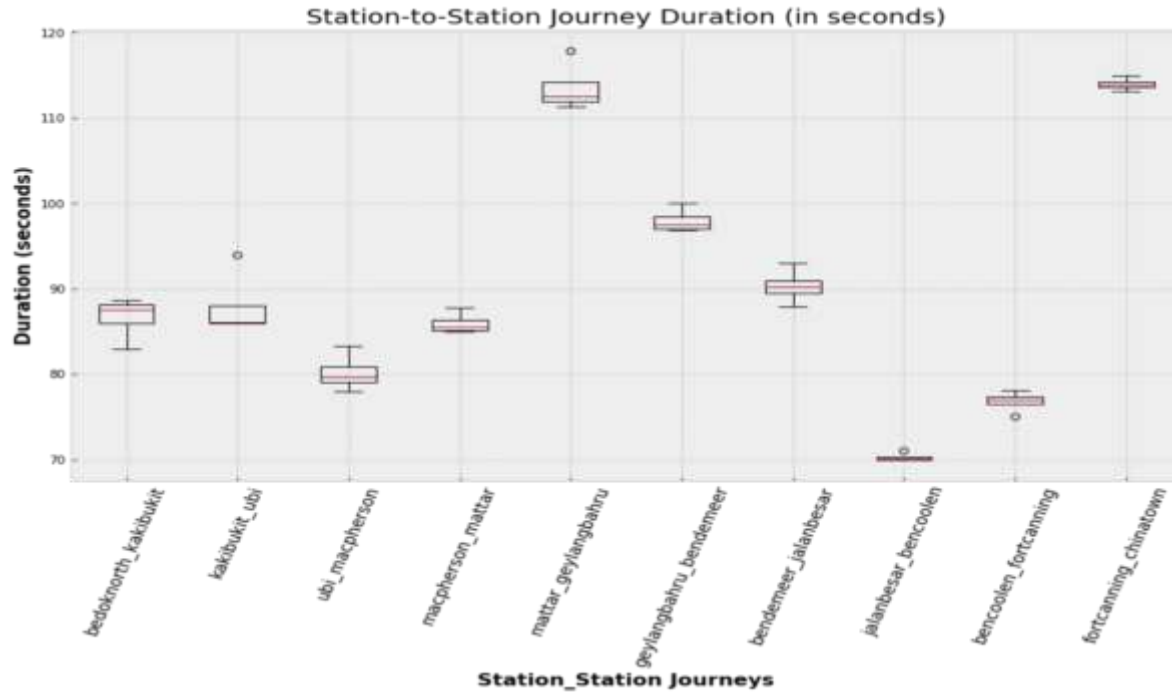


Figure 42: Multiple Boxplot Graph showing Journey Duration recorded over 6 different days

From the graph, factoring the hardware lags and human error, the train took about similar durations to reach each station across 6 days.

Overview of Data Processing

There were a few steps to the data cleaning and processing as follows:

1. Converting timestamp values to readable date-time format
2. Total acceleration computed from raw triaxial acceleration by the square root of the sum of squares of the triaxial acceleration
3. Total acceleration values were then filtered with a low-pass filter to remove as much human noise as possible
4. Acceleration values passed by human noise filter are then standardised and normalised
5. With the normalised values, higher-level features were computed with equations as stated in Section 4.5 of the paper (Khuong An Nguyen, 2019)

Higher-level features for Derivative Dynamic Time Warping

In TPD, the use of Derivative Dynamic Time Warping technique requires pure acceleration signals collected by smartphones to be standardised, normalised and pre-processed.

Human Noise Filter

As smartphones measure acceleration data based on devices' X, Y and Z axis, the human movements have to be smoothened out before matching of signals can happen. A low-pass filter can be applied to smooth out the human noise on the assumption that the trains do not stop or increase speed abruptly, and swift human gestures tend to cause sudden peaks in acceleration recorded (i.e. picking up the phone, or rotating the phone). The following shows the low-pass filter formula, where an alpha value of 0.35 which was

determined by (Khuong An Nguyen, 2019) as optimal is adopted and the effects of smoothing are also shown in the graph displayed below :

$$a'_i = a'_{i-1} + \alpha \times (a_i - a'_{i-1})$$

Figure 43: Human Noise Filter

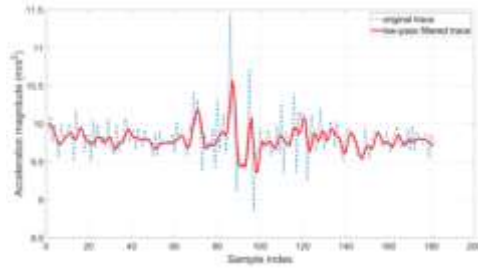


Figure 44: Illustration of the effects of the human noise filter on acceleration signals

Standardisation, Normalisation of acceleration values and computing higher-level features

For Derivative Dynamic Time Warping to be applied, the acceleration signal has to be standardised and normalised after filtering out human noise. Thereafter, higher-level features are computed with formulas as shown below, where a and b are cleaned, standardized and normalized acceleration signals obtained from two separate journeys, and i and j are the number of data points on each signal:

$$a'_i = \frac{a_i - a_{i-1} + ((a_{i+1} - a_{i-1})/2)}{2}, (1 < i < m)$$

$$b'_j = \frac{b_j - b_{j-1} + ((b_{j+1} - b_{j-1})/2)}{2}, (1 < j < n)$$

Figure 45: Higher-Level Features computations

Test 1

Data collected at 1Hz frequency on the Mobility Sensing App (Continuous collection across multiple stations)

In the first test, continuous train journey acceleration data were collected over two days on the Downtown Line between Bedok North station and Chinatown station, producing signals 1 and 2. Also, acceleration data was collected on train journeys between Chinatown station and Little India station on the North-East Line, producing signals 3 and 4.

Signals 1 and 2 were matched in the DDTW algorithm, followed by signals 3 and 4 in the same manner. This way two scores for the same route comparison were obtained. For different route comparisons, DDTW was run on signal 1 with signals 3 and 4, and signal 2 was matched with signals 3 and 4 with DDTW algorithms as well. This will, in turn, produce four scores for different route comparisons.

Test 1 Observations

Then, DDTW was performed onto the final cleaned acceleration signals. The two processed signals from the journey between Bedok North station and Chinatown station on the Downtown Line on two separate days are shown below.

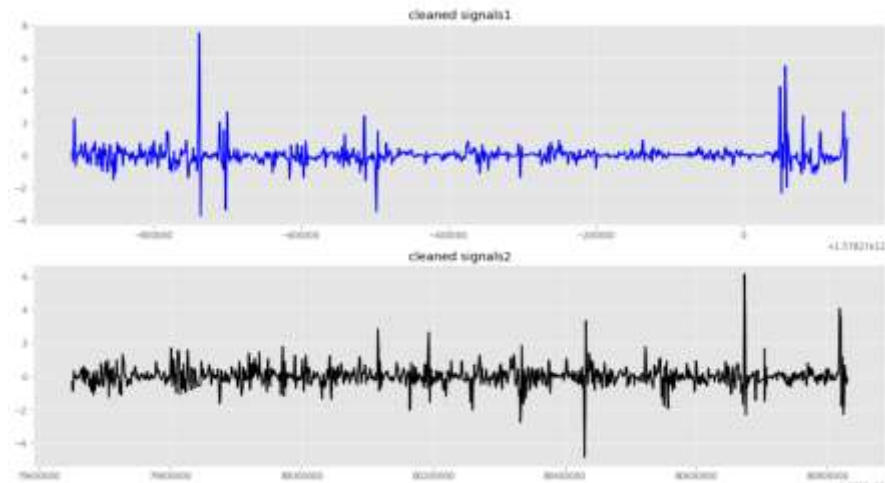


Figure 46: Cleaned Signals from two journeys from Bedok North to Chinatown on DownTown Line

Processed acceleration signals from journeys between Chinatown station to Little India station on a different train line (North-East Line) are shown below.

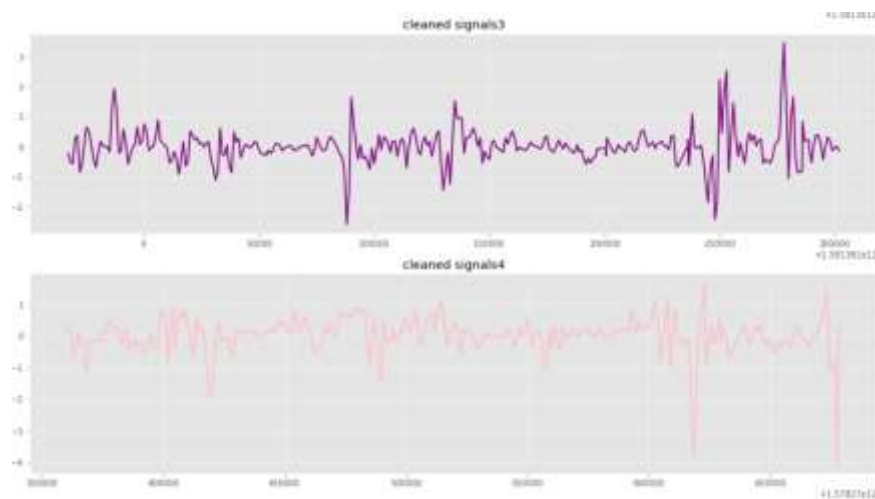


Figure 47: Cleaned Signals from two journeys from Chinatown to Little India

The following shows the resulting scores obtained from the various comparison with DDTW:

Table 7: DDTW Scores

Same Route Comparisons	
Signals compared	DDTW Scores
Signal 1 vs Signal 2	0.42689660702709403
Signal 3 vs Signal 4	0.3540037626005349
Different Route Comparisons	
Signals compared	DDTW Scores
Signal 1 vs Signal 3	1.0033068651823023
Signal 1 vs Signal 4	1.057522207488681
Signal 2 vs Signal 3	1.1229805405653814
Signal 2 vs Signal 4	1.158519661985768

With the scores computed from the comparison between different routes being about two to three times higher than the scores computed from the comparison between same routes, more tests can be done to experiment the method in the paper (Khuong An Nguyen, 2019).

Test 2

Data collected at 1Hz frequency on the Mobility Sensing App (Collection of accelerations based on each station-to-station journeys)

In the second test, train journey acceleration data were taken for each station-to-station journey (ie. one-stop journey). This was done by starting to record on a smartphone when the train closed its doors just before it moved and stopped recording when the train stopped moving before the train opened its doors for commuters to alight.

As the shortest journey any commuter can take is a one-stop journey, a test was run on whether the DDTW algorithm can be used to differentiate processed acceleration signals collected from one station-to-station journey from another station-to-station journey effectively.

Firstly, training dataset was taken by physically placing the phone on the train to take the pure train movement (fully constrained with no other external interference). Then, testing sets was taken by taking the train while doing the regular phone checking and rotations to simulate movements on smartphones commonly done by commuters as they use their phones on commute.

For same route comparisons, ten station-to-station journeys between Bedok North and Chinatown on the Downtown Line were taken on 6 different days at 1Hz recording frequency. For different route comparisons, the same ten station-to-station journeys between Bedok North and Chinatown on the Downtown Line were taken on 6 different days and compared to the station-to-station journeys taken on the reverse route, which is from Chinatown to Bedok North.

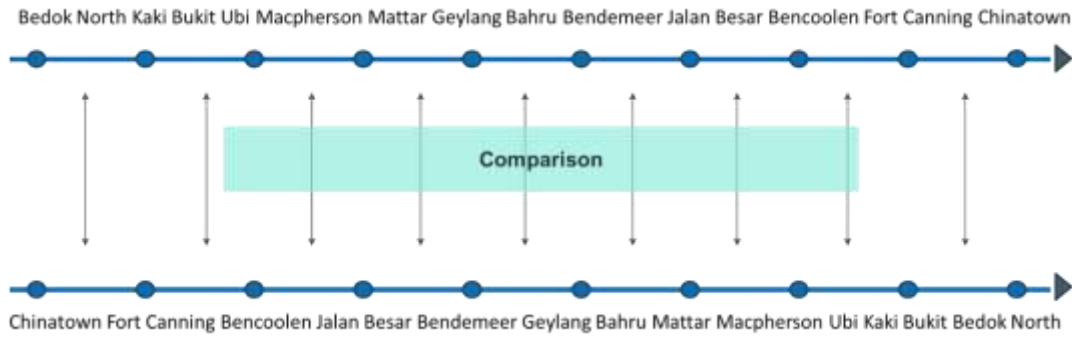


Figure 48: Different route comparison (station-to-station journeys)

Test 2 Observations

After applying the DDTW algorithm prepared and tested in Test 1 onto the station-to-station journey 1Hz acceleration dataset, the following are the observations about DDTW scores obtained for each station-to-station journey plotted on graphs with multiple boxplots as shown:

Same Station-to-Station Journeys Comparison – 1Hz Dataset

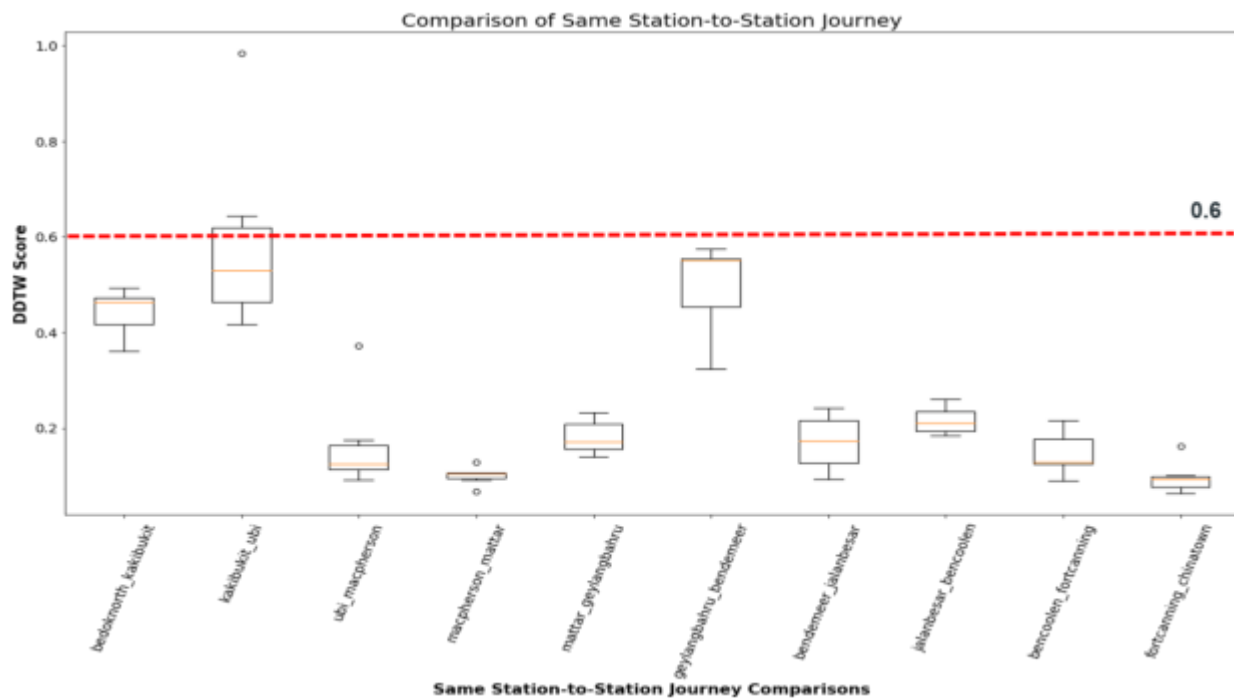


Figure 49: Same Station-to-Station Journeys Comparison – 1Hz Dataset

Different Station-to-Station Journeys Comparison – 1Hz Dataset

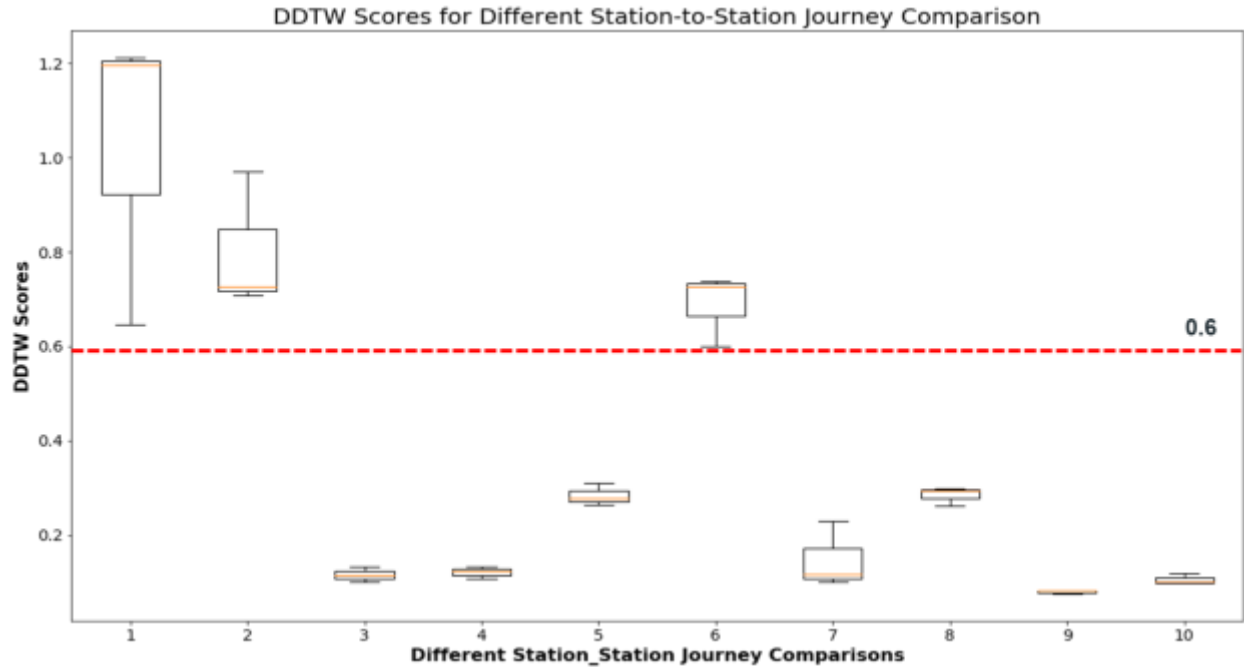


Figure 50: Different Station-to-Station Journeys Comparison – 1Hz Dataset

From the two plots, it was observed that when comparing the same station-to-station journeys, DDTW scores obtained are mostly under 0.6. Therefore, checks have to be done to check if the different station-to-station journey comparisons were showing scores that were consistently above 0.6. However, it was observed that for the different station-to-station journey comparisons, most DDTW scores were below 0.6 as well, except for different journeys comparison 1,2 and 6.

Test 3

Same and different route comparisons with DDTW algorithm were also done in Test 3, except for the fact that the dataset was collected at a higher recording frequency of 50Hz instead of 1Hz. This is done because correspondence with researchers that wrote the paper points us to recording data at higher frequencies. Upon knowing that tests on 1Hz station-to-station journey comparisons did not successfully yield differentiable scores, one of the original researchers suggested that data be collected at a recording frequency of 50Hz instead of 1Hz. 1Hz recording frequency might not be able to record the accelerations and decelerations that were happening on the train due to some bumps and curves, which might form important changes in acceleration values crucial in helping DDTW identify differences between signals taken on different station-to-station journeys.

Hi Pei Wen,

We sampled our accelerometer at 50 Hz. 1 Hz may be too low, depending on how fast the train moves, you may miss some important changes (i.e. bumps, curves).

Yes, the London underground has many routes with similar travelling time, but it has plenty of environmental differences to compensate for it.
I am not sure how smooth and straight the Singaporean network is ?

Best regards
Khuong

Figure 51: Email correspondence with one of the researchers

However, the 50Hz dataset was not collected on Mobility Sensing App, it was collected on a paid app on iPhone known as *Sensor Play*. This is because the Mobility Sensing app was still under development and have not been able to collect data at 50Hz collection frequency on the iPhone that was used to collect train acceleration data. The Sensor Play had good reviews on the iOS AppStore and was able to collect data at 50 Hz, so it was used instead of the Mobility Sensing App.

The 50Hz Dataset was also recorded on 3 separate journeys on one day, as during this period of the experiment, the Circuit Breaker measures implemented by the Singapore government and our supervisor advised on not going to take the train too often for safety reasons.

Test 3 Observations

After applying the DDTW algorithm onto the 50Hz datasets, the following shows observations about DDTW scores obtained for station-to-station journey comparisons plotted on graphs with multiple boxplots:

Same Station-to-Station Journeys Comparison – 50Hz Dataset

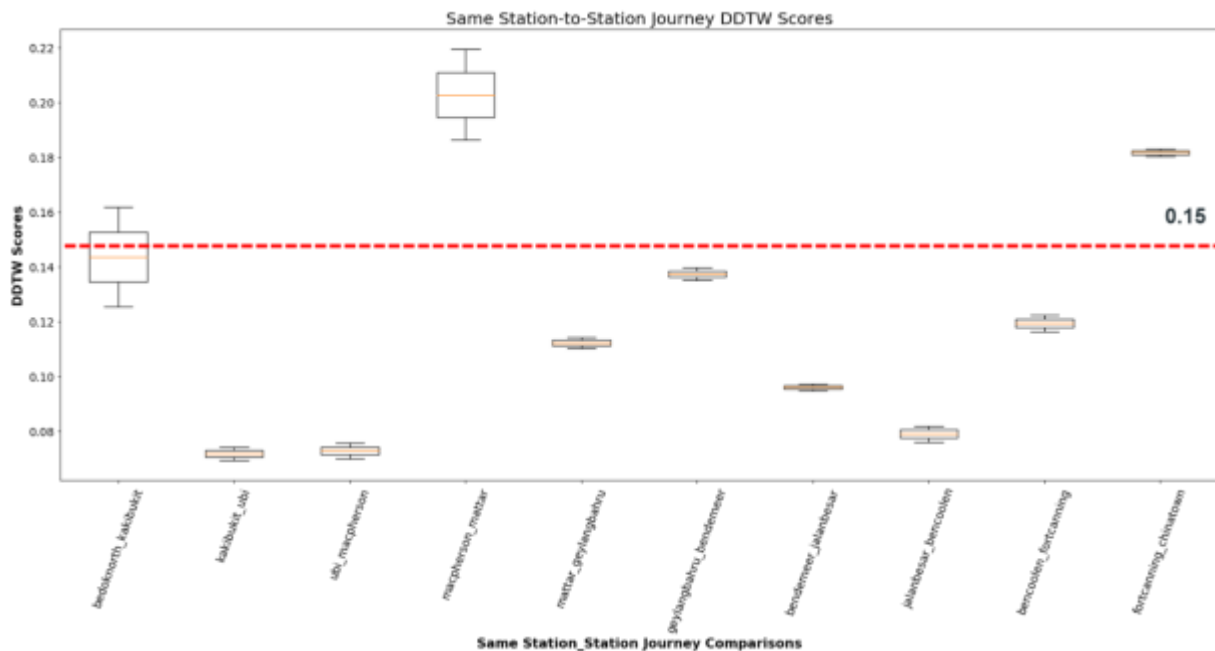


Figure 52: Same Station-to-Station Journeys Comparison – 50Hz Dataset

Different Station-to-Station Journeys Comparison – 50Hz Dataset

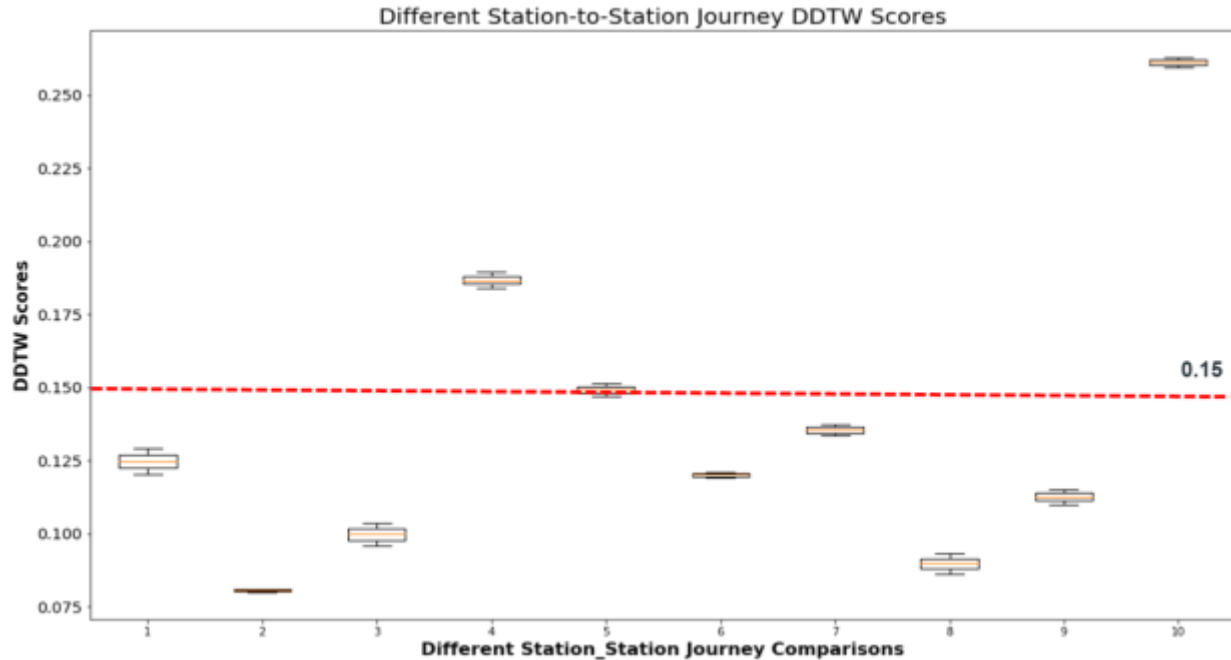


Figure 53: Different Station-to-Station Journeys Comparison – 50Hz Dataset

As observed, DDTW scores for same station-to-station journey comparison are mostly below 0.15 except for the station-to-station journeys between MacPherson and Mattar, and Fort Canning to Chinatown. Likewise, the DDTW scores for different station-to-station journey comparison are also mostly below 0.15, except for comparison number 4.

Therefore, the results showed that at 50Hz recording frequency, differentiable scores between same and different station-to-station journey comparisons with DDTW could not be achieved. Furthermore, this exercise was computationally intensive on regular machines/ computers as each 10 station-to-station journey comparison lasted for at least 2.5 hours. This still happened even when hardcoding of the DDTW functions was done, without downloading any Dynamic Time Warping packages on Python.

Test 4

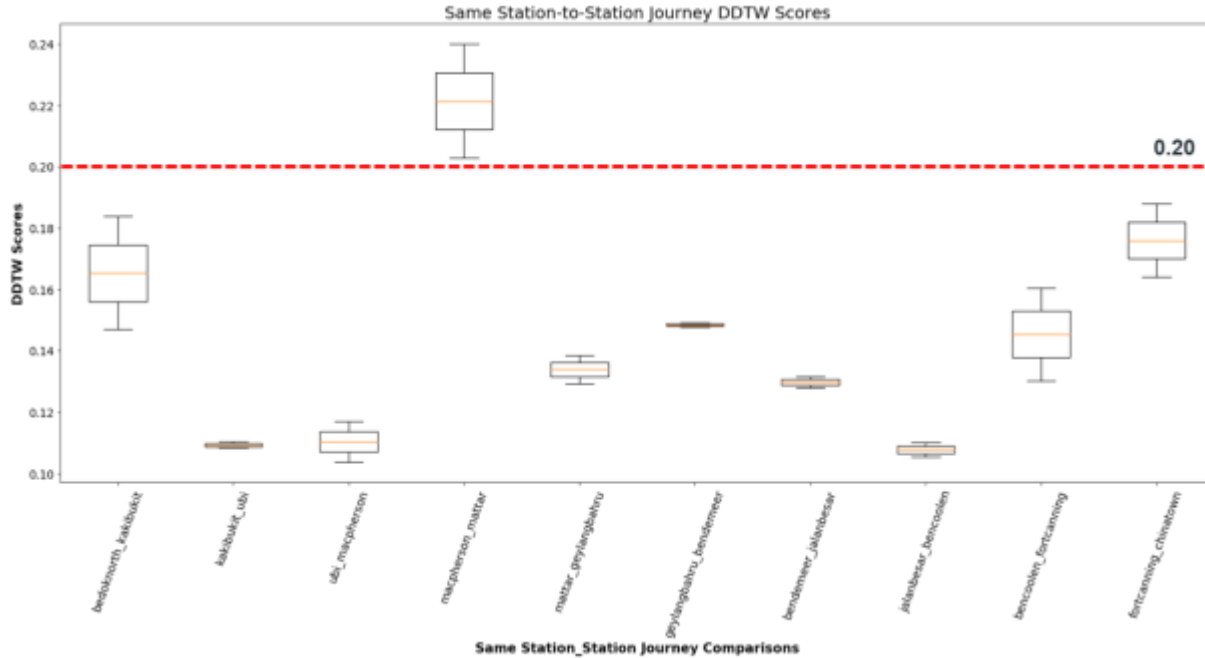
As stated in the research paper (Khuong An Nguyen, 2019), the researchers achieved optimal results at 15Hz recording frequency, 50Hz dataset was down-sampled to 25Hz, which is a recording frequency closer to 15Hz.

Due to Circuit Breaker measures, more data could not be collected so downsampling was done on the 50Hz dataset by taking every alternate data point. Then the same DDTW algorithm was also applied to compare between same and different station-to-station journeys.

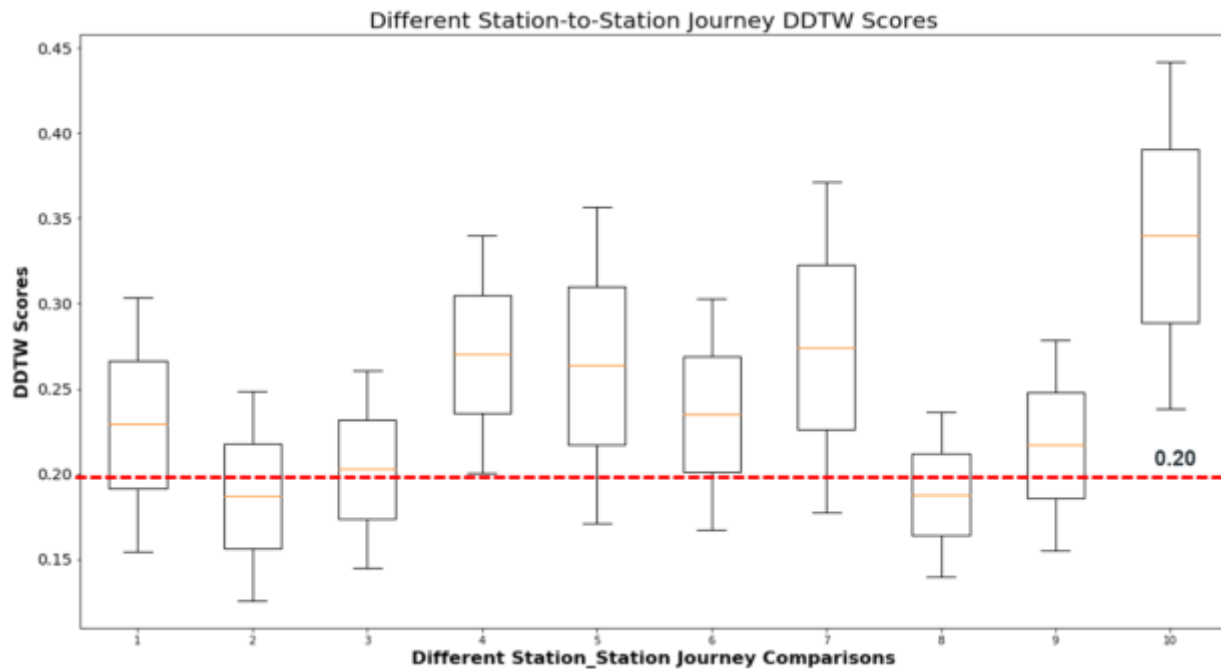
Test 4 Observations

After applying the DDTW algorithm prepared onto the 25Hz datasets obtained from down-sampling, the following shows observations of DDTW scores obtained for each station-to-station journey plotted on graphs with multiple boxplots:

Same Station-to-Station Journeys Comparison – 25Hz Dataset



Different Station-to-Station Journeys Comparison – 25Hz Dataset



As observed, most of the same station-to-station journey comparisons were giving scores below 0.20 except for the station-to-station journey comparison between MacPherson and Mattar. As for different station-to-station journey comparisons, the scores were mostly above 0.20. Though some scores for different station-to-station journey comparisons were falling below 0.20, there were 7 out of 10 different station-to-station journeys that achieved scores above 0.20. This means that there were more instances

where the DDTW produced differentiable scores for same station-to-station journey comparisons and different station-to-station journey comparisons.

Test 5

Thereafter, downsampling on the same 50Hz dataset was done to obtain 10Hz dataset, by taking every 5th data point in the dataset. This is because at 25Hz, running the DDTW algorithm was still quite computationally intensive. It took about an hour to complete the DDTW on comparing signals of two separate journeys of ten stations.

Test 5 Observations

After performing DDTW on the 10Hz dataset obtained from downsampling, the following shows observations of DDTW scores obtained for each station-to-station journey plotted on graphs with multiple boxplots:

Same Station-to-Station Journeys Comparison – 10Hz Dataset

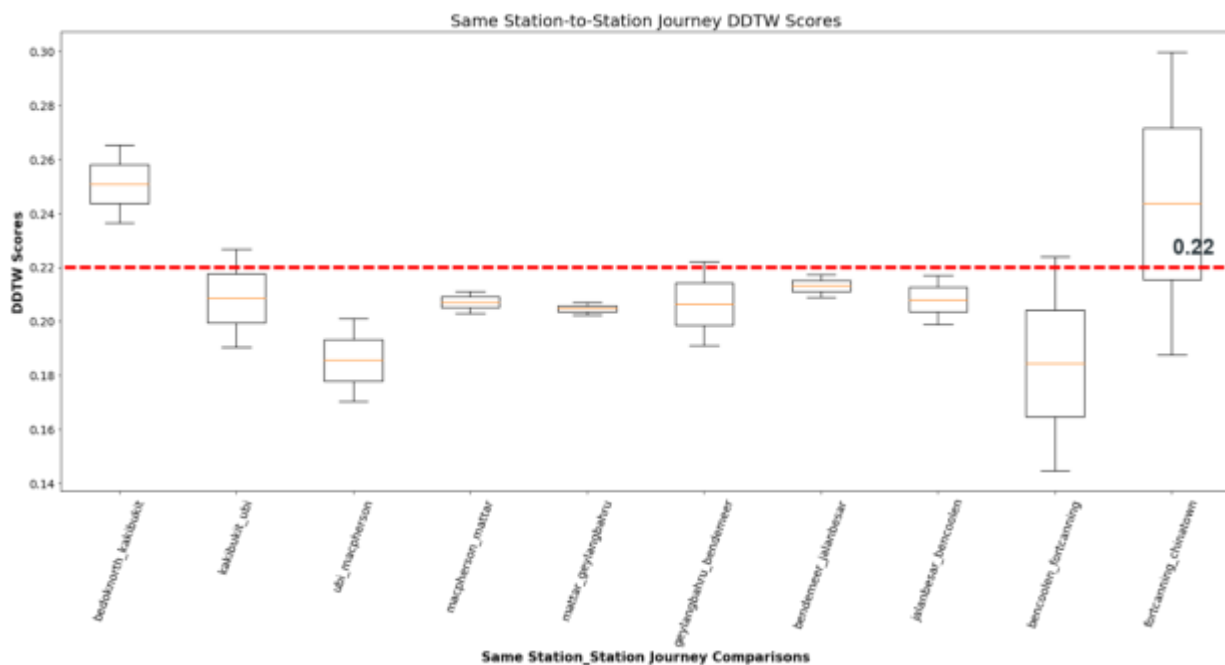


Figure 54: Same Station-to-Station Journeys Comparison – 10Hz Dataset

Different Station-to-Station Journeys Comparison – 10Hz Dataset

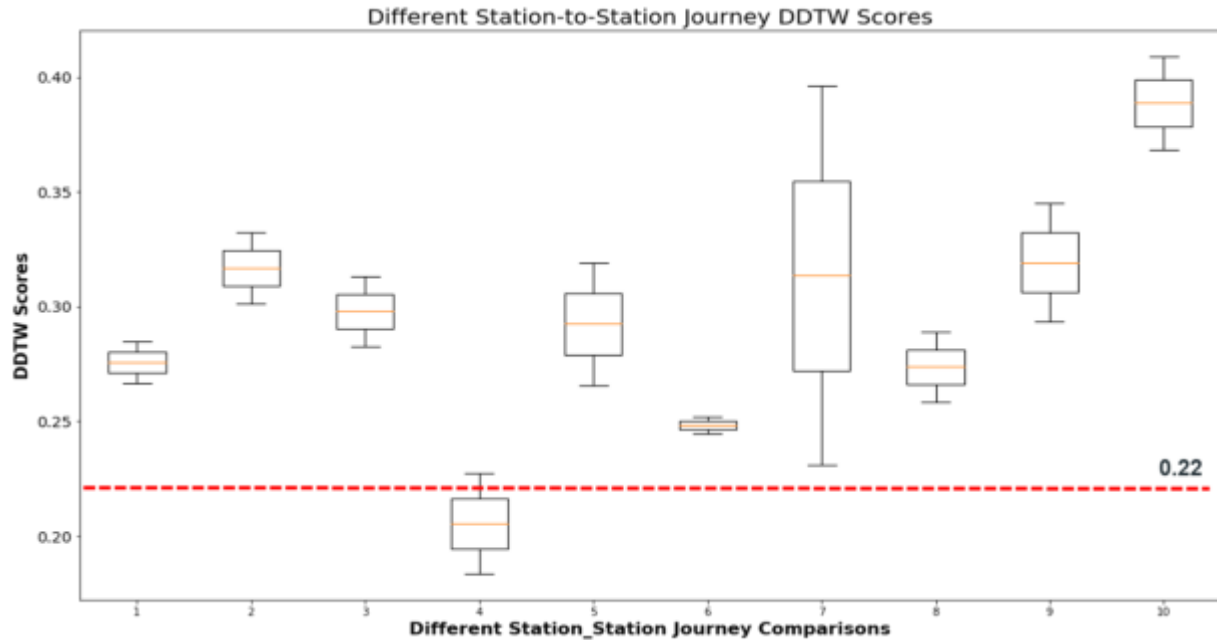


Figure 55: Different Station-to-Station Journeys Comparison – 10Hz Dataset

As observed, most of the scores obtained from the same station-to-station journey comparisons gave scores that were below 0.22, while most scores obtained from the different station-to-station journey comparisons gave scores that were above 0.22.

Summary of Observations

The following table shows the summary of the different tests and the observations:

Tests	Recording Frequency	App	Observations
Test 1	1Hz (Long journey)	LTA Mobility Sensing App	DDTW can produce differentiable scores between same and different station-to-station journey comparisons
Test 2	1Hz (Station-to-station)	LTA Mobility Sensing App	DDTW was only able to produce differentiable scores between same and different station-to-station journey comparison on 3 out of 10 station-to-station journey comparisons
Test 3	50Hz (Station-to-station)	Sensor Play (paid app from iOS)	DDTW produced non-differentiable scores between same and different station-to-station journey comparisons
Test 4	25Hz (Station-to-station)	Sensor Play (paid app from iOS) Down-sampling from 50Hz	DDTW produced reasonably differentiable scores between same and different station-to-station journey comparison on 5 out of 10 station-to-station journey comparisons
Test 5	10Hz (Station-to-station)	Sensor Play (paid app from iOS)	DDTW produced reasonably differentiable scores between same and different station-to-station journey

		Down-sampling from 50Hz	comparison on 7 out of 10 station-to-station journey comparisons
--	--	-------------------------	--

According to the research paper by (Khuong An Nguyen, 2019), the optimal recording frequency that was obtained by their experiment was 15Hz. Here, our 10Hz dataset obtained from down-sampling of 50Hz has also obtained a more satisfactory result in producing reasonably differentiable scores between same and different station-to-station journey comparisons.

Limitations

There were a few limitations to this experiment. Firstly, results were based on 6 ten-station journeys for 1Hz data, and 3 ten-station journeys for 50Hz data. The 25Hz ten-station journeys and 10Hz ten-station journeys were down-samplings of the 50Hz ten-station journeys data. More experiments on more journey data have to be done to confirm results.

Changes in railway track smoothness due to maintenance have not been factored in. This means that it is two to three months later when some track maintenance happened, it is unclear if acceleration signals produced by a train travelling on the same train line be recording the same bumps that identify any particular train station-to-station journey acceleration signal.

Most data of the 50Hz dataset taken during Circuit Breaker, when the train has a lesser load and it is uncertain if it will affect accelerometer readings.

DDTW comparison is computationally intensive. For instance, it took about 2.5 hours to run each ten-station journeys comparison at 50Hz (dataset size is 50 times bigger than previous 1Hz dataset). 10Hz comparison also took about 0.5 hours and 25Hz took about 1.5 hours to complete running.

Future Work

The experiment done is only the beginning and there is some future work to do before DDTW can be adopted as a method for passenger tracing on Singapore MRT and thereby improving the MyTransport.SG application.

Firstly, tests on the algorithm need to be done on more datasets, especially after the Circuit Breaker when train load returns to a normal level.

Instead of comparing reverse route data, a test should be done on different station-to-station journeys to ensure that algorithm can produce differentiating scores when the different station-to-station data are compared (i.e. Bedok North-Kaki Bukit versus Kaki Bukit-Ubi).

Down-sampling of data to lower frequencies can be done to determine the optimal frequency that would allow us to balance between accuracy and algorithm running time.

Taking known journeys made by a passenger as a reference point, an inertial tracking module can be built in to estimate where a passenger is at. This, in turn, estimates the increased duration that is needed to complete his journey should train stop in the middle of two stations

Road Anomaly Detection

As this is a POC test, the goal is primarily to validate the science and methods for pothole detection. Hence, there may be some deviation in the data collection and data analysis methods for a full-scale study with motorized vehicles.

Data Collection

In terms of raw variables, we used the variables listed in Table 1, collected using the AndroSensor app (v1.9.6.3). For initial testing, the Samsung Galaxy S7 installed with Android OS 8.0.0 and the Samsung Galaxy S20+ installed with Android OS 10.0.0 were used as sensor logger devices sampling at 20Hz. Different app combinations and sampling frequencies were attempted, but the above combo had the most stable usable readings at the point of experimentation hence the selection. It should be noted that the speed estimate is only used as a reference reading for tagging the different event types, as it is not reliable enough for continuous high-frequency speed estimation.

In the choice of vehicles, a variety of vehicle types was tested such as buses, taxis and vans to have a gauge of what sort of vehicles are suitable for data collection. Generally, it was found that heavier vehicles such as buses and even vans tend to have much more violent changes to acceleration when moving and have higher engine vibration, which would obscure key target signals, though this may be mitigated by signal filtering. It should be noted that currently, the road inspection vehicles used by LTA contractors are vans, and they often travel at a slow constant speed of roughly 20km/h, to facilitate roving visual inspection of the roads. However, to have better control of the pothole surveying and data collection process¹⁷ and the difficulty in requisitioning a motorised vehicle for extended use, a bicycle was ultimately used for the POC study. This resulted in several key issues in the parameters that would differ from using non-single tracked vehicles like cars:

1. Road safety
2. Difficulty in maintaining a constant speed
3. Slower speeds
4. Increased sensitivity to smaller road defects
5. Y-Axis gyroscope rotation can no longer be used

Table 8: Summary of variables that may affect the experiment results

SN	Variable	Effect	Variable Class
1	Tire pressure	The higher the pressure, the more sensitive to road anomalies	Vehicle
2	Tire dimensions	The larger the dimensions, the less sensitive to road anomalies	
3	Tire rigidity (rubber hardness)	The more rigid the tire, the more sensitive to road anomalies	
4	Suspension stiffness	The stiffer the suspension, the more sensitive to road anomalies; Less stiff suspensions would have more rocking motion	
5	Brake responsiveness	More responsive brakes would have smoother deceleration (less violent jerks and false-positive events)	
6	Vehicle weight	The heavier the vehicle, the less sensitive to road anomalies; but inertia forces when turning and acceleration/deceleration would be higher	
7	Engine vibration	The more engine vibration the noisier the signal	
8	Attainable speed	The higher the speed, the more sensitive to road anomalies	
9	Road roughness	The higher the road roughness, the noisier the signal	Road

¹⁷ Field recce sessions needed to be conducted to hunt for potholes around Singapore; and repeated readings have to be taken too, especially if one missed the pothole. Typically, at least 3 sets of readings were taken for each zone being sampled.

SN	Variable	Effect	Variable Class
10	Pothole size	The larger the pothole, the higher the signal amplitude; however if the pothole is too big, there is a good chance that drivers may actively avoid it resulting in false-negative events	
11	Road unevenness	The higher the road unevenness, the noisier the signal	
12	Presence of road debris	The higher the amount of road debris, the noisier the signal	
13	Miscellaneous road features (e.g. drain covers, expansion gaps, humps, speed strips etc.)	Encounter with such events may result in false-positive events due to the similar motion captured	
14	Speed consistency	More irregular speeds would result in more variation in the signal amplitudes observed	Driver
15	Actual speed travelled	The higher the speed, the more sensitive to road anomalies	
16	Changes in acceleration / deceleration	Sudden changes to acceleration/deceleration can result in more noise	
17	Turning action	Faster turning action across a steeper angle would result in more noise from angular acceleration	
18	Sensor model	Lower end sensors would have lower precision and accuracy	Sensor
19	Sensor logger model	Lower CPU power and memory limitations may result in missing values; unoptimised codes may have problem recording and synchronising readings from multiple sensors at higher frequencies	
20	Rigidity of sensor mount	The more rigid the sensor, the less noise from inertia forces	
21	Proximity of the sensor mounted to the vehicle's centre of gravity (CG)	The closer the sensor to the vehicle's centre of mass, the less noise from inertia forces	
22	GPS Availability	The more satellites the GPS can lock on to, the more accurate the GPS readings; GPS is largely unavailable at indoor, underground and valley shadow areas	
23	GPS Power Consumption	Higher GPS power consumption would limit continuous extended usage	

Nonetheless, as phones were mounted directly on to the bicycle frame with a phone holder for maximum stability when taking readings. In terms of the areas chosen for sampling potholes, industrial areas in western Singapore was chosen due to the higher prevalence of potholes in those areas. A pothole-ridden stretch of park connectors along Mandai Road and Mandai Avenue was chosen too, as it was paved with asphalt (analogous to roads), suffered from severe potholes and was much safer to sample on a bicycle since there was no need to watch for motorised traffic. Other non-pothole events (normal roads, speed strips and humps) were also sampled at a Yishun neighbourhood as a reference for baseline comparison, especially when trying to remove false-positive events. Ideally, when taking readings, multiple sensor loggers should be used as redundancy and also for results comparison.

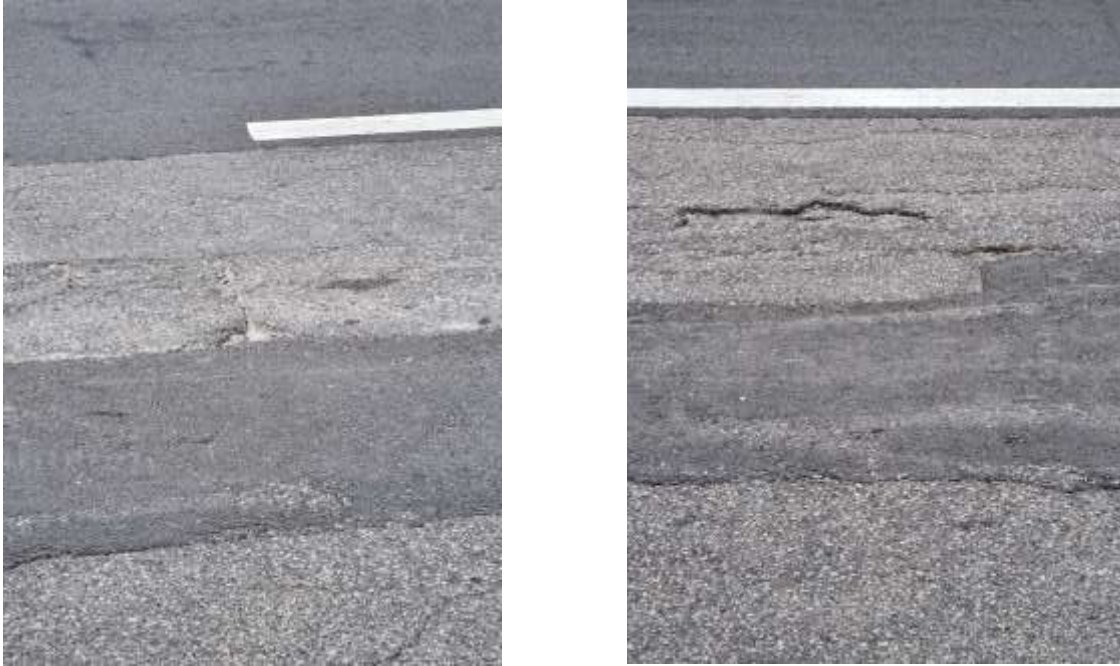


Figure 56: Sample images of potholes surveyed

It should be noted that whilst every effort has been made to control the variables, there are still some confounding factors which cannot be fully controlled as seen in Table 8. In this case, only a single experimenter, bike and app were used for the final readings taken, though 2 different phones were used on different occasions for the final readings.

Additional Notes

Current road inspections by LTA surveyors are primarily visual-based, with photos taken on the move to document the anomaly. As such, one could potentially use image recognition technologies to tag such items with a dashboard camera mounted similar to a project by CarVi (Davies, 2019) – assuming with we could acquire a large enough repository of labelled images. Doing so will overcome the major limitation of using accelerometer and gyroscope data to detect road anomalies – namely that drivers must ride over the road anomalies with their vehicles directly over it to register these anomalies. Furthermore, drivers are likely to avoid more serious road anomalies e.g. larger potholes to avoid risking damage to their vehicles, resulting in false negative events by our current modelling method. However, such a video analytics based set up would require the use of additional hardware which could make mass adoption harder to the public.

Data Processing

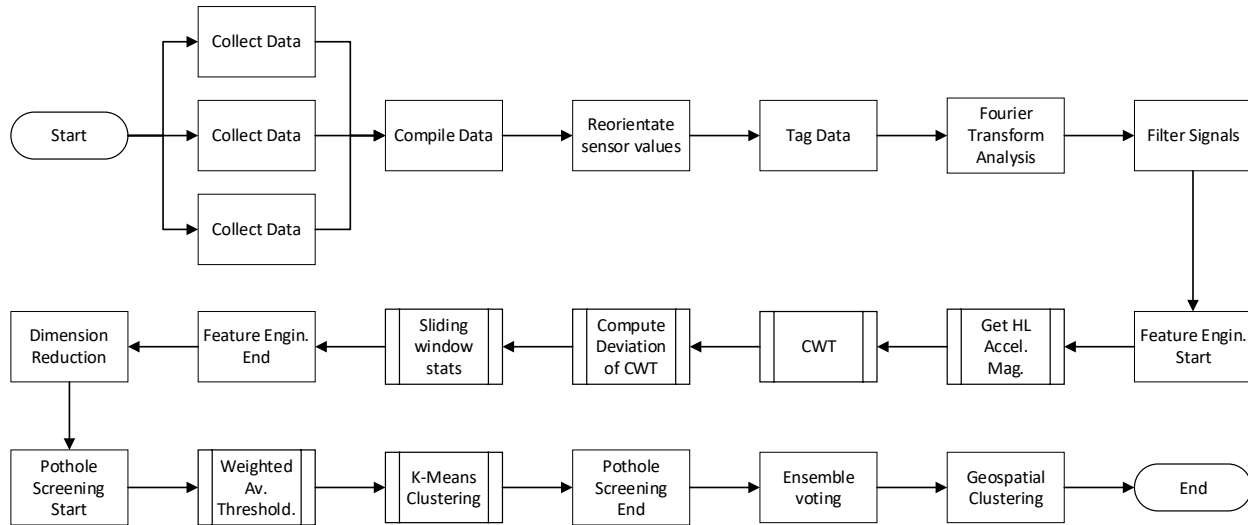


Figure 57: Overview of data collection, processing, and analysis steps

The raw data collected from AndroSensor was exported as separate CSV files and often comes with significant noise at the tail ends of each recording session, due to the operator having to remove the phones from the mount to start and end each recording session as seen in Figure 59. This is dependent on how accessible the mounting location is to the experimenter. Nonetheless, these tail ends need to be removed, before compiling all the different session readings as a single continuous record. It should be noted that since all the sensor readings have been taken at the same frequency and synchronised¹⁸ within the AndroSensor application, additional synchronising and interpolation of values is not required. Upon compilation of the signal data, reorientation of the 3-axis linear acceleration and gyroscope rotation values via quaternion rotation would be needed. However, since AndroSensor does this within the application itself, this step was skipped. The next step would be manually tagging the signal data into the following zones:

1. Event 0: Stationary
2. Event 1: Potholes
3. Event 2: Speed strips
4. Event 3: Normal cycling
5. Event 4: Road humps
6. Event 9: Uneven terrain & uncategorised noise (e.g. pavements and roadside grass patches etc.)

¹⁸ All the sensors fire and record data at different timestamps; hence all observations with full sensors values from multiple timestamps are in fact only loosely synchronised to the same timestamp.

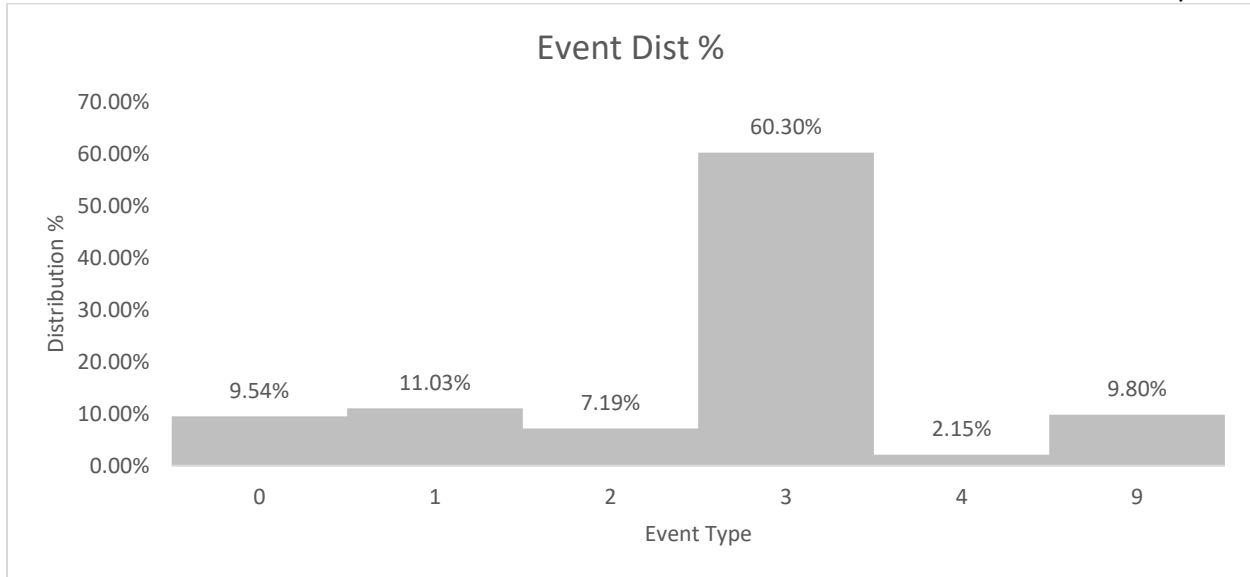


Figure 58: Distribution of Data Collected by Event Tag

Tagging the data will greatly assist in the analysis of the data when comparing pothole regions with non-pothole regions. However, such methods of tagging is less useful when potholes appear in close physical proximity with each other or when there is a lot of noise from other vibration sources, making accurate tagging of the wavelet signals difficult. Such a method can only provide a rough gauge at best, and is may not be suitable for generating clean training data for supervised learning on its own.

Also, since this is done manually in post-data-collection with significant signal noise present, data tagging can be very labourious. Ideally, a second experimenter should be used to tag the timestamp and GPS coordinates of the potholes encountered with the data matched back to the main accelerometer, gyroscope and GPS readings from a separate logger, to make the tagging of clean data more accurate and less tedious. This method would work best of vehicles where having a pinion rider or multiple passengers is possible

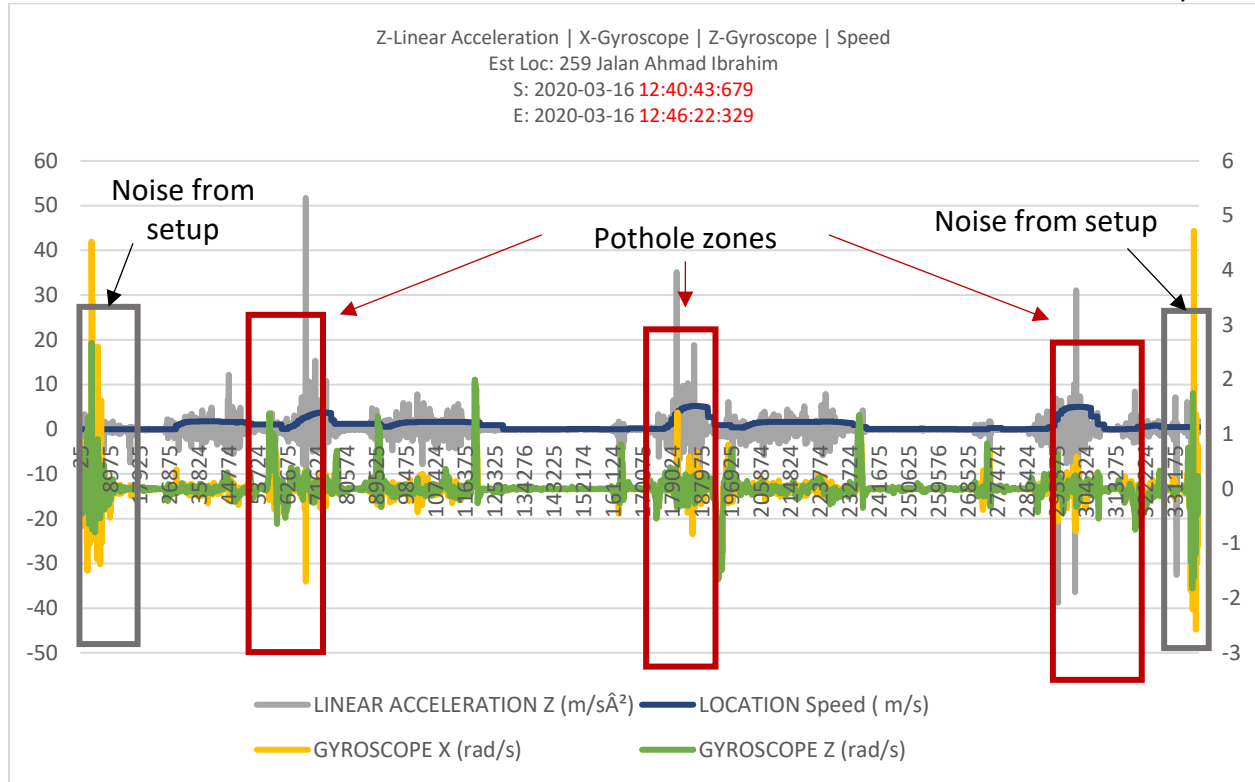


Figure 59: Sample raw data recording

In the next step, Fourier transform is applied on the signal data to identify key frequencies with high power signal regions of the pothole zones compared with minimal overlap with non-pothole regions. Using the identified frequencies, a bandpass filter would then be applied to filter the signal. It should be noted that some trial and error is needed to estimate the cutoff frequencies.

Table 9: Identified passband frequency ranges

Cutoff Points	Linear Acceleration	Gyroscope Rotation
Low-Frequency Cutoff (Reject lower frequencies)	2.5Hz	3 Hz
High-Frequency Cutoff (Reject higher frequencies)	5 Hz	5 Hz

Since digital signals are being processed, an Infinite Impulse Response (IIR) filter was chosen, of which a Butterworth bandpass variant was used. A Butterworth design was chosen as it can produce a relatively sharp frequency band transition of the pass and stop bands with minimal aliasing, compared to other IIR filter designs such as Bessel and Chebyshev as seen in Figure 60. For this application, a filter order of 6 was chosen, as that provided the filter properties sought as seen in Figure 61 (refer to *B02 – Filtered Signal Data* in the Appendix for the filtered results).

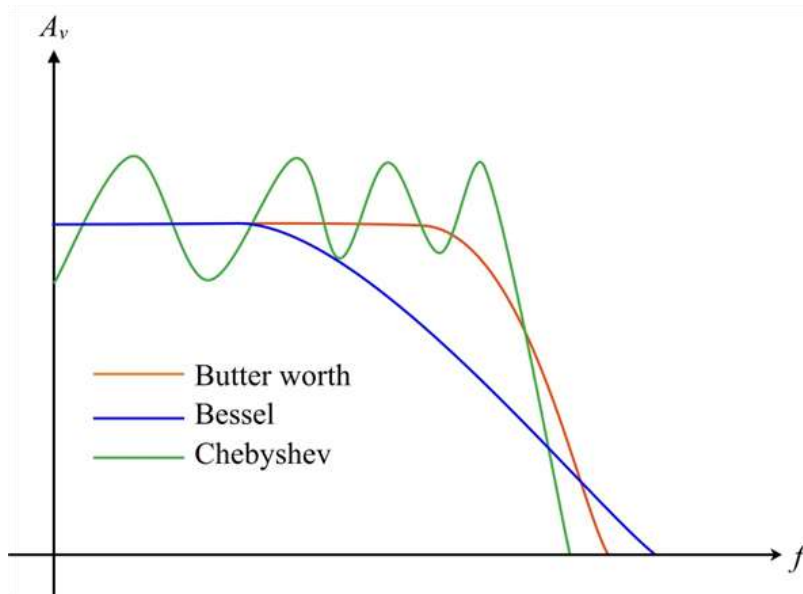


Figure 60: Frequency band transition between the pass and stop bands of different IIR filter designs

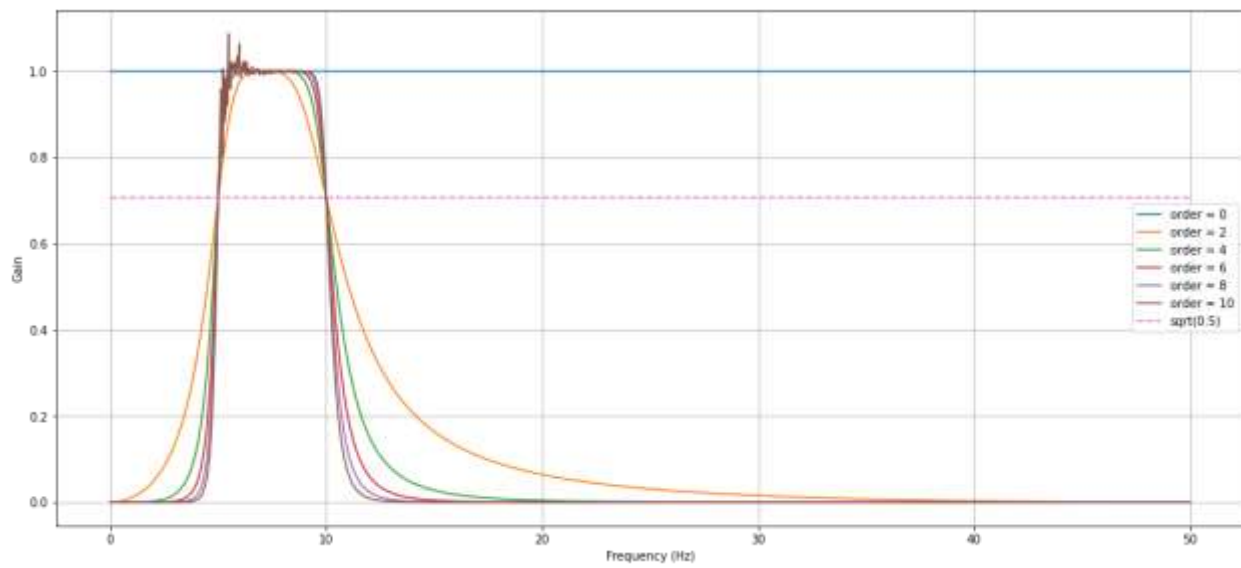


Figure 61: Butterworth Bandpass Filter in different order settings. Severe aliasing starts appearing at orders 10 and above.

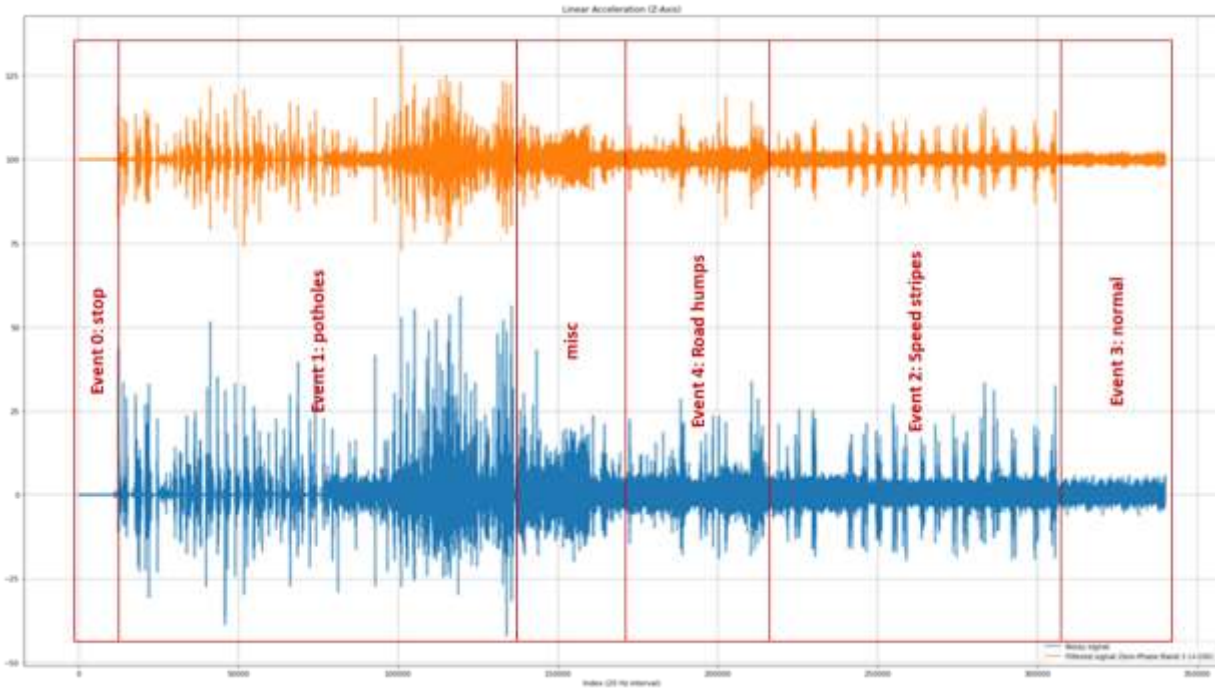


Figure 62: Filtered signal (orange) and unfiltered data (blue) of linear acceleration (Z-Axis data), joint together as a continuous times series grouped by event. Knowing the approximate zones and their corresponding event tags would be useful in analyzing the continuous wavelet transform scalogram plots later, as one is better able to identify the key signals.

Feature Engineering

Previous studies have indicated that during a pothole encounter, acceleration of all 3 axes will dip to near 0 for that instant (Mednis, Strazdins, Zviedris, Kanonirs, & Selavo, 2011). Hence, a similar methodology was adopted without compromising the isolation of the Z-axis acceleration, by computing the horizontal linear acceleration via the formula below.

$$\text{Horizontal Linear Acceleration} = \sqrt{LA_{x-Axis}^2 + LA_{y-Axis}^2}$$

Where, LA = Filtered Linear Acceleration.

Following which, Continuous Wavelet Transform (CWT) is applied to the following variables to generate the respective wavelet coefficients:

1. Horizontal Linear Acceleration
2. Filtered Linear Acceleration (Z-Axis)
3. Filtered Gyroscope Rotation (X-Axis)

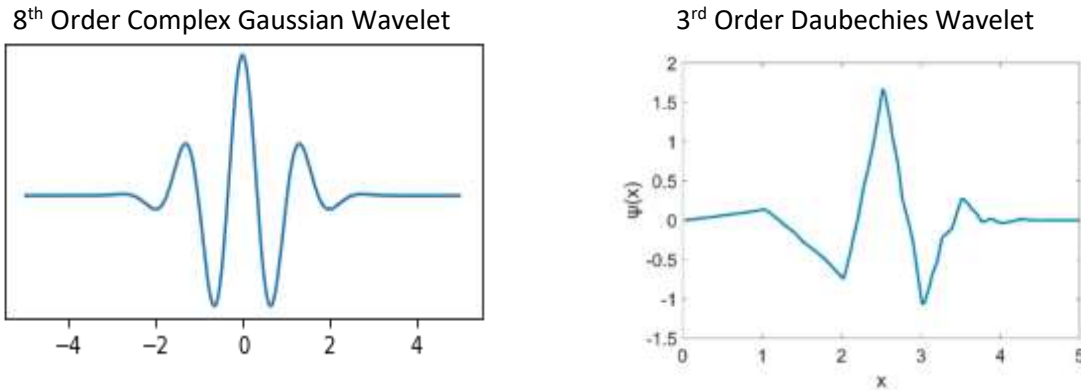


Figure 63: Mother wavelet used for Continuous Wavelet Transformation (CWT)

An 8th order complex gaussian wavelet was chosen as the mother wavelet due to its similarity with the target wavelet which had a similar shape to a 3rd order Daubechies wavelet¹⁹ which was previously identified for being able to model the wavelet properties of potholes fairly well (Wei, Fwa, & Zhe, 2005). Other wavelet orders from the complex gaussian and gaussian wavelet families were experimented on too, though the 8th order complex gaussian wavelet was proven to have the best signal response. The initial coefficients generated were unable to discern the changes in amplitude very well, hence the following additional steps were deployed to clean up the signal:

1. Log transform of the absolute values of the signal amplitudes
2. Thresholding to suppress signals with weaker amplitudes (Log scale)
 - a. Horizontal Linear Acceleration: suppress values lower than -2
 - b. Filtered Linear Acceleration (Z-Axis): suppress values lower than 0.5
 - c. Filtered Gyroscope Rotation (X-Axis): suppress values lower than -5

For the 3 sets of CWT coefficients, only coefficients from the first 10 scales were used, as they generally capture fast-moving changes indicative of the pothole encounter; excludes non-pothole events; and they capture the highest power (amplitude) regions best, except for the horizontal linear acceleration signal data. Given the inconsistent nature of the coefficient amplitudes at higher scales for horizontal linear acceleration, it is likely that the higher amplitudes registered were caused by inertial forces (e.g. jerks from the suspension and rattling of objects mounted on the bike) at work during the sudden change of acceleration when encountering a pothole and other compensatory action by the rider to maintain balance and steering of the vehicle.

¹⁹ At the point of writing, only the PyWavelets python package used does not support the use of Daubechies wavelets for continuous wavelet transform, hence other alternative wavelets needed to be sought.

Table 10: CWT wavelet frequencies at different scales for Linear Acceleration (Z-Axis) and Horizontal Linear Acceleration

Scale	Wavelet Frequency (Hz)	Scale	Wavelet Frequency (Hz)	Scale	Wavelet Frequency (Hz)	Scale	Wavelet Frequency (Hz)
1	14	11	1.273	21	0.667	31	0.452
2	7	12	1.167	22	0.636	32	0.437
3	4.667	13	1.077	23	0.609	33	0.424
4	3.5	14	1	24	0.583	34	0.412
5	2.8	15	0.933	25	0.56	35	0.4
6	2.333	16	0.875	26	0.538	36	0.389
7	2	17	0.824	27	0.519	37	0.378
8	1.75	18	0.778	28	0.5	38	0.368
9	1.556	19	0.737	29	0.483	39	0.359
10	1.4	20	0.7	30	0.467		

Table 11: CWT wavelet frequencies at different scales for Gyroscope Rotation (X-Axis)

Scale	Wavelet Frequency (Hz)	Scale	Wavelet Frequency (Hz)	Scale	Wavelet Frequency (Hz)	Scale	Wavelet Frequency (Hz)
1	10	11	0.91	21	0.48	31	0.32
2	5	12	0.83	22	0.45	32	0.31
3	3.33	13	0.77	23	0.43	33	0.3
4	2.5	14	0.71	24	0.42	34	0.29
5	2	15	0.67	25	0.4	35	0.29
6	1.67	16	0.62	26	0.38	36	0.28
7	1.43	17	0.59	27	0.37	37	0.27
8	1.25	18	0.56	28	0.36	38	0.26
9	1.11	19	0.53	29	0.34	39	0.26
10	1	20	0.5	30	0.33		

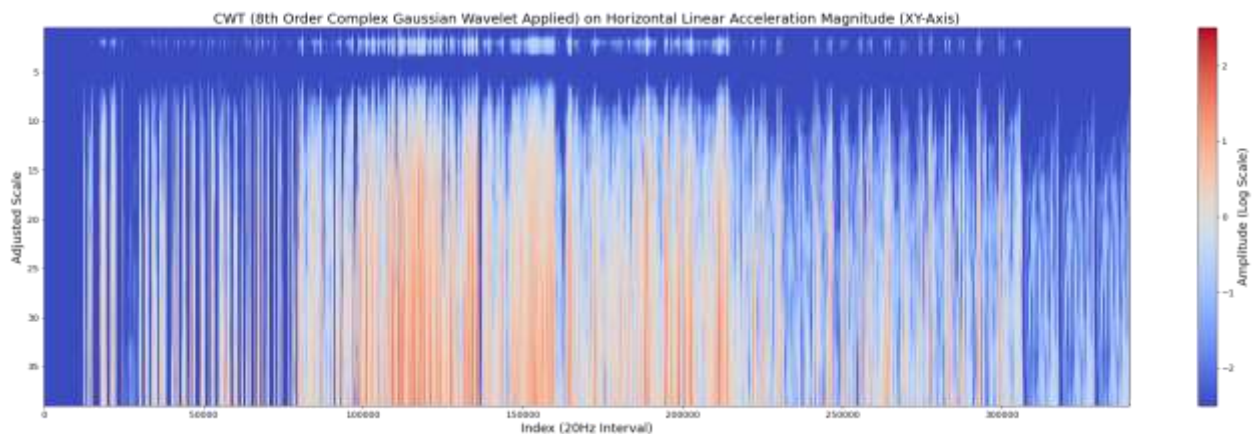


Figure 64: CWT Scalogram Plot for Horizontal Linear Acceleration (Filtered); all scales shown. Note that low energy events such as normal cycling have been filtered out at the lower scales.

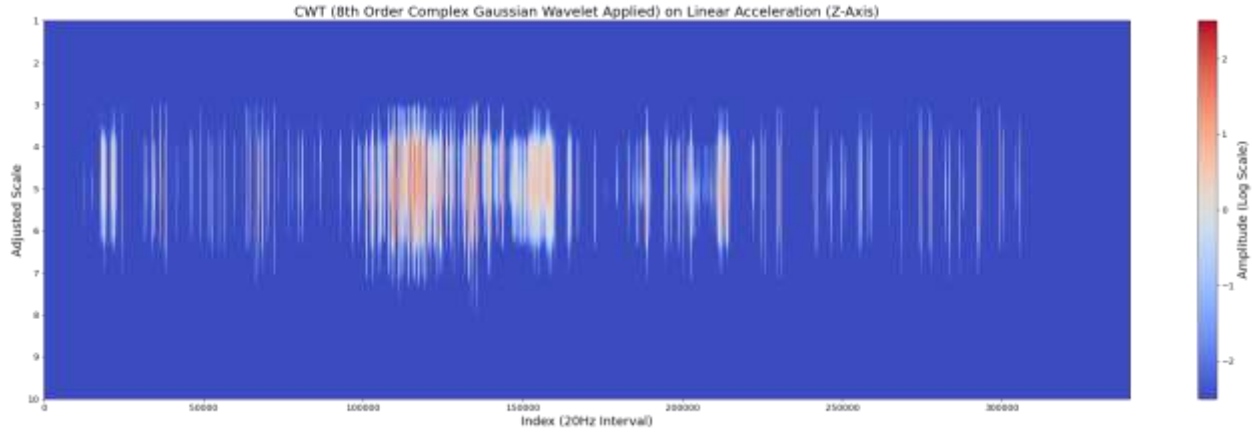


Figure 65: CWT Scalogram Plot for Z-Axis Linear Acceleration (Filtered); first 10 scales shown (as the other scales were primarily low energy regions). Note that low energy events such as normal cycling have been filtered out.

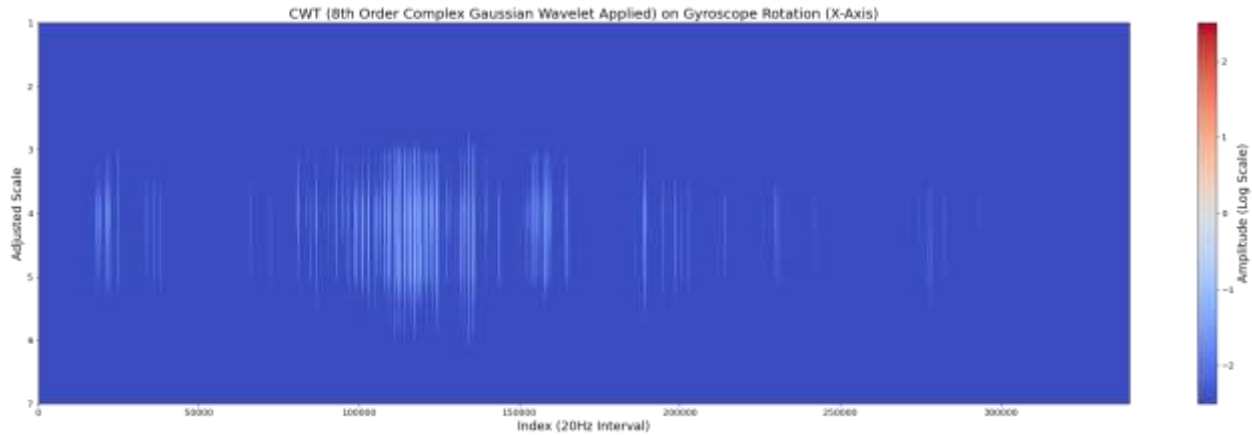


Figure 66: CWT Scalogram Plot for X-Axis Gyroscope Rotation (Filtered); first 10 scales shown (as the other scales were primarily low energy regions). Note that low energy events such as normal cycling and speed strips have been filtered out.

Additional features were generated from the CWT coefficients by measuring the amount of deviation each CWT coefficient from the 95th percentile and Upper Outlier Boundary²⁰ of their respective scales. The general notion is that potholes tend to generate a stronger amplitude change, compared to non-pothole events, hence, a more positive deviation from this extreme threshold points, the more likely the observation is a pothole. Also, since potholes are considered to be outlier events that produce more extreme changes in the signal, this is to be expected too. Tests on the distribution of the tagged data between different event types also proved this hypothesis as evidence by the scalogram plots and Table 12 where the vast majority of the non-potholes having weaker amplitudes and hence suppressed.

Table 12: Selected Descriptive Stats of CWT Coefficients (Scale 5)

	Horizontal Linear Acceleration (Scale 5)		Z-Axis Linear Acceleration (Scale 5)		X-Axis Gyroscope Rotation (Scale 5)	
Event	Upper Bound	P95	Upper Bound	P95	Upper Bound	P95
0	-20.000	-20.000	-20.000	-20.000	-20.000	-20.000
1	26.362	-0.304	33.630	2.531	-0.839	-1.493

²⁰ The upper outlier boundary is taken as the median + 1.5 * Interquartile Range (IQR)

Event	Horizontal Linear Acceleration (Scale 5)		Z-Axis Linear Acceleration (Scale 5)		X-Axis Gyroscope Rotation (Scale 5)	
	Upper Bound	P95	Upper Bound	P95	Upper Bound	P95
3	7.663	-0.716	33.131	2.223	-1.563	-2.193
4	-20.000	-1.311	10.832	1.426	19.841	-2.800
5	26.476	-0.237	33.295	2.281	-1.046	-1.899
9	-20.000	-1.227	10.906	1.474	19.770	-2.796

Descriptive stats of the filtered linear acceleration and gyroscope data, via a sliding window of 30 observations (1.5 secs @ 20Hz sampling frequency), with a shift of 1 observation. This is meant to capture the sharp changes in amplitude when a road bump, whereby the following variables would see changes:

1. A positive shift in central tendency
2. Larger spread and deviation of the observations
3. More positive deviation from defined cut off points; typically, more extreme values are expected

Table 13: Descriptive stats generated via a sliding window

SN	Descriptive Stats Generated
1	Signal Magnitude
2	Moving Average (absolute values used for comparison)
3	Moving Median (absolute values used for comparison)
4	Moving 5 th Percentile
5	Moving 95 th Percentile
6	Moving Range
7	Moving Percentile Range (95 th percentile – 5 th percentile)
8	Moving Standard Deviation (SD)
9	Moving Inter Quartile Range (IQR)
10	Deviation from 2 SD from Moving Average
11	Deviation from Moving Median + 1.5 * Moving IQR
12	Deviation from 95 th Percentile
13	Deviation from fixed cut-off point (absolute values used for comparison) <ol style="list-style-type: none"> 1. Horizontal Linear Acceleration cut-off point: 10 m/s² 2. Filtered Linear Acceleration (Z-Axis) cut-off point: 7 m/s² 3. Filtered Gyroscope Rotation (X-Axis) cut-off point: 0.2 rad/s

At this point, a total of 134 different features has been generated, as seen in Table 14, which can be too computationally intensive for modelling purposes, hence variable reduction is needed. In this case, Principal Component Analysis (PCA) is applied to reduce the number of variables down to 26 Principal Components (PC) which would be used for modelling purposes as seen in Table 15. Selection of the PCs was based on PCs having eigenvalues more than 1; and a total cumulative explained variance of 70% or more.

Table 14: Summary of all features generated for modelling

Variables	No. of Features Generated
Linear Acceleration (Filtered) – 3 Axis	3
Horizontal Linear Acceleration	1
Gyroscope Rotation (Filtered) – X-Axis	1
CWT Coefficients (Scales 1-10) <ol style="list-style-type: none"> 1. Horizontal Linear Acceleration 2. Filtered Linear Acceleration (Z-Axis) 	10 * 3 = 30

3. Filtered Gyroscope Rotation (X-Axis)	
CWT Deviation from Threshold Points (Scales 1-10)	$2 * 10 * 3 = 60$
1. Horizontal Linear Acceleration	
2. Filtered Linear Acceleration (Z-Axis)	
3. Filtered Gyroscope Rotation (X-Axis)	
Sliding Window Descriptive Stats on Filtered Signals	$13 * 3 = 39$
1. Horizontal Linear Acceleration	
2. Filtered Linear Acceleration (Z-Axis)	
3. Filtered Gyroscope Rotation (X-Axis)	
Total	134

Table 15: Generated Principal Components and their explained variances

PC No.	Explained Variance (EV) %	Cumulative Explained Variance (CEV) %	Eigenvalues	Cumulative Eigenvalues
1	39.35	39.35	51.549	51.549
2	9.468	48.818	12.403	63.952
3	4.937	53.755	6.468	70.42
4	4.305	58.06	5.64	76.06
5	3.484	61.544	4.564	80.624
6	2.968	64.512	3.888	84.512
7	2.536	67.048	3.323	87.835
8	2.276	69.324	2.982	90.817
9	2.074	71.398	2.717	93.534
10	2.026	73.424	2.654	96.188
11	1.801	75.225	2.359	98.547
12	1.734	76.959	2.272	100.819
13	1.633	78.592	2.139	102.958
14	1.48	80.072	1.939	104.897
15	1.41	81.482	1.847	106.744
16	1.254	82.736	1.643	108.387
17	1.189	83.925	1.557	109.944
18	1.158	85.083	1.517	111.461
19	1.091	86.174	1.43	112.891
20	1.056	87.23	1.384	114.275
21	0.96	88.19	1.258	115.533
22	0.893	89.083	1.17	116.703
23	0.871	89.954	1.141	117.844
24	0.831	90.785	1.088	118.932
25	0.813	91.598	1.065	119.997
26	0.796	92.394	1.043	121.04

Modelling

On inspection of the PC plots, it was found that most of the outlier / extreme values tended to be zones previously marked as potential pothole areas (Zone 1), particularly for potholes 1-10, opening up the

possibility of using various rule-based or unsupervised learning methods to detect these potholes. Refer to *B03 – Principal Components* in the Appendix for more details on the plots. For our modelling approach, we would be taking a semi-supervised learning approach for pothole classification, as the data has already been tagged (albeit not very clean) in the initial stage. This would for verification of the observations with some level of certainty whether they have been tagged correctly.

The first approach taken would be the use of weighted average thresholding of the PC generated to determine the probability of whether an observation is a pothole or not. The general formula is given as:

$$\text{Weighted Average Probability of Pothole (WAPP)} = \sum_{i=1}^n (EV_n \times TC_n)$$

Where:

- EV = Explained Variance of PC_n
- TC = Meets Threshold Condition of PC_n
 - If the threshold condition is met (considered an extreme value), the value = 1
 - If the threshold condition fails (not considered an extreme value), the value = 0
- Maximum WAPP value = Cumulative Explained Variance (CEV) of PC 1-26 = 92.394%

Table 16: WAPP Threshold Settings Used

PC	WAPP 1	WAPP 2
1 – 2	Top 0.2% most extreme values	Top 1% most extreme values
3 – 26	Top 0.1% and bottom 0.1% most extreme values	Top 0.5% and bottom 0.5% most extreme values
WAPP Cut-off for Pothole Detection	WAPP >= 70%	WAPP >= 80%

Varying level of sensitivity was used to generate the 2 WAPP prediction as seen in Table 14. Determining the cut-off point and whether a 1-tail or 2-tail approach should be taken in the selection of the extreme values was primarily done via observing the PC plots and determining which cut off points resulted in the least amount of overlap with non-pothole events. A WAPP cut off value of 70% and 80% were used for WAPP 1 and 2 respectively for pothole determination. WAPP 2 was assigned a higher cut off value to help compensate for its potentially lower specificity caused by the lower PC threshold conditions used.

K-Means clustering was also used for pothole detection. For modelling purposes, only the first 10 PC were used, as they provided the best separation of the tagged pothole observations – the inclusion of the other PC may dilute the results since such clustering methods do not support a differential weighting of input variables. Furthermore, these PC contained the highest explained variance, with a CEV of 73.42%; using lesser number of variables would reduce the computational load required, given that there are over 340K observation used. The latter point is also more important for other clustering methods which are even more computationally expensive. In determining the K- value, a K value of 5 was chosen as it provided an adequate split of the data into the smaller clusters desired. Furthermore, since there were only 6 major event tags created, one would expect a similar number of clusters only, as confirmed by the elbow plot in Figure 74. However, one could theoretically push the number up to 9 clusters where the elbow plot fully

tapers off. The average silhouette value of the resulting K-means cluster (K=5) was 0.348, indicating that there is still a significant amount of overlapping clusters, especially for the non-pothole clusters. But since the focus is primarily on pothole detection, this is not a major issue as long as a single cluster is found to be able to discriminate between a pothole and non-pothole events well. In this case, cluster 3 was shown to be able to discriminate between non-pothole and pothole events as seen in Figure 68 and Figure 69 respectively, satisfying this requirement. A train-test split done with the K-means clustering settings reapplied to both train and test data, resulting in similar results, thus validating the clusters (refer to *B04 – K-Means Clustering* in the Appendix for more info). Hence, for modelling purposes, we would primarily treat cluster 3 as suspected potholes.

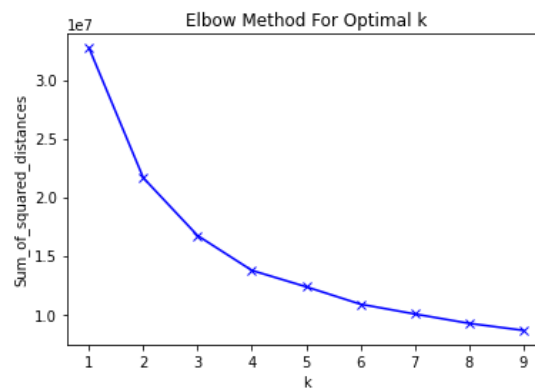


Figure 67: Elbow plot for determining optimal K value for K-Means clustering

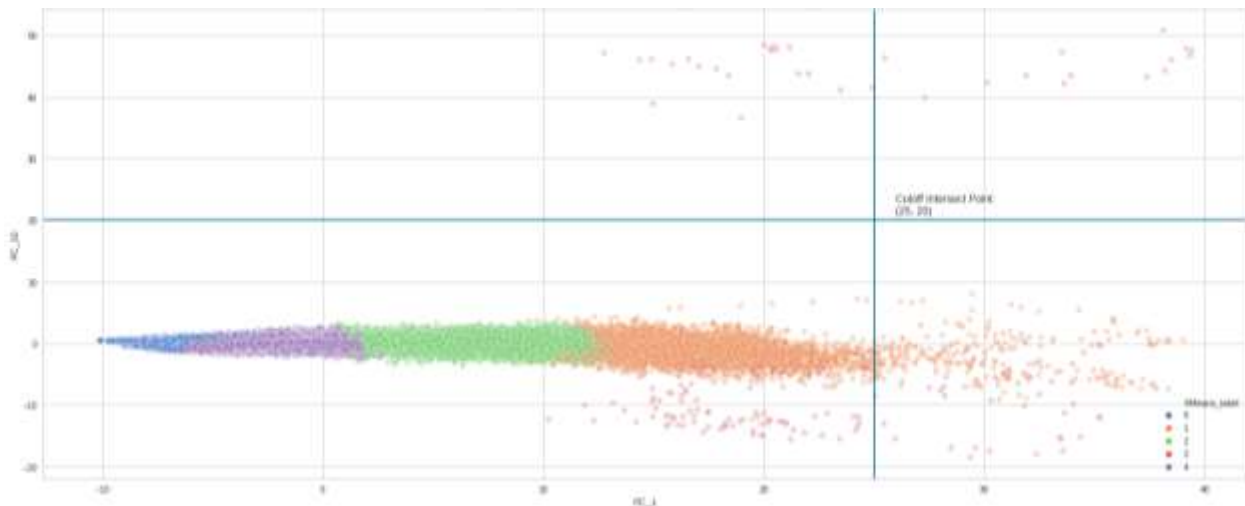


Figure 68: PC Plot of K-Means Cluster Labels, filtered by Tagged Non-Pothole Events. Cluster 3 (red) identified as a suspected pothole due to relative absence in the tagged non-pothole events plot compared to tagged pothole events.

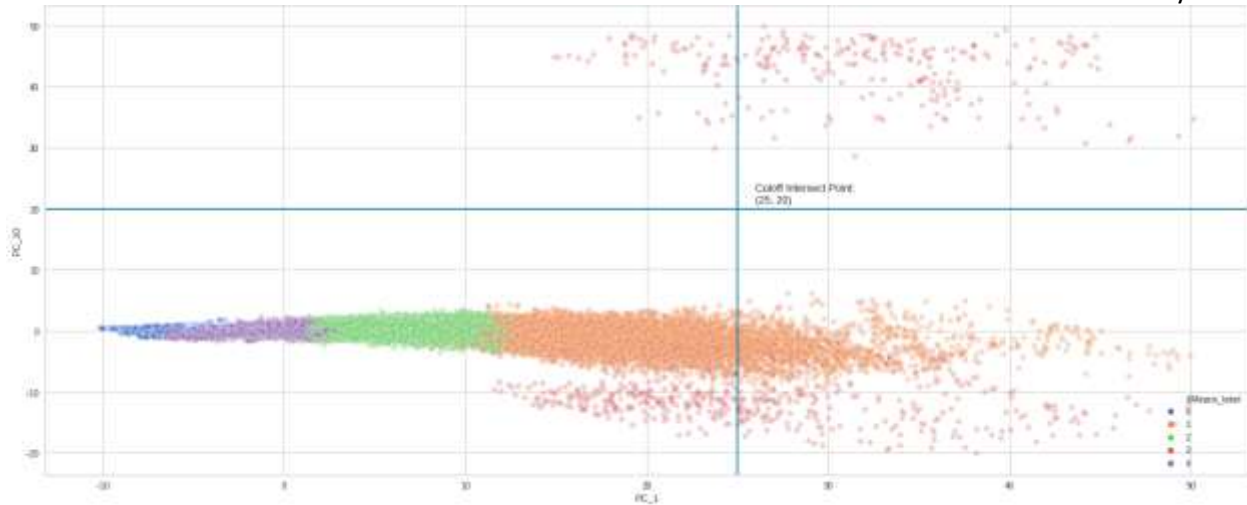


Figure 69: PC Plot of K-Means Cluster Labels, filtered by Tagged Pothole Events. Cluster 3 (red) identified as a suspected pothole due to relative absence in the tagged non-pothole events plot compared to tagged pothole events.

Other unsupervised learning methods such as Local Outlier Factor (LOF), Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) were attempted too, but they were unable to provide good discrimination in pothole identification, hence their cluster results would not be used. More details can be found in *B05 – Results from Other Clustering Methods in the Appendix*. It should also be noted that these methods are significantly more resource-intensive compared to K-means clustering, making them unsuitable for clustering large amounts of data (especially with high-frequency sampling) from multiple users.

With the results from WAPP1, WAPP2 and K-Means Clustering (Cluster 3), an ensemble voting method is then applied to determine whether an event is a pothole by having a minimum of 2 or more votes as a suspected pothole. This narrowed the suspected pothole observations down to 622 observations.

Table 17: Breakdown Stats of Suspected Potholes

Breakdown	Alpha (2 Or More Votes)	Beta (3 Votes)
Count	622	298
Percentage against all observations	0.183%	0.088%

With the 622 suspected potholes identified, HDBSCAN is then applied to these observations using geospatial clustering of their GPS coordinates – this was applied twice, once on observations with “2 Or More Votes” and another on observations with “3 Votes”. This would serve as a crowdsensing module, where observations from multiple pooled together. The general premise goes that potholes would create hot zones with concentrated clusters of suspected potholes, whereas false-positive spurious events would appear in clusters with smaller concentrations which can then be filtered out. Also, since GPS coordinates only have limited precision, clustering is required to compensate for it. For the 622 identified observations, the median GPS accuracy was $\pm 4\text{m}$.

Key Settings used:

- Input data: GPS Coordinates of suspected potholes
 - Alpha: Observations with 2 or more votes
 - Beta: Observations with 3 votes
- GPS Accuracy = pegged to median GPS Accuracy of suspected potholes = $\pm 4\text{m} = \pm 0.004\text{km}$
- Earth Radius = 6378km
 - pegged to equatorial radius since Singapore (where the measurements are taken) is in the equatorial region; else the average radius may be used instead
- $\text{Epsilon}^{21} = \text{GPS Accuracy (in km)} \div \text{Earth Radius}$
- Minimum Cluster Size = 5
- Distance metric: Haversine
- Cluster selection method = Excess of Mass (EOM)

The results are then compared against the ground of truth readings to test the accuracy of the algorithm in detecting potholes correctly. Unfortunately, due to road safety issues, it was not possible to take GPS readings directly on the potholes for more accurate results. Furthermore, it was found that many of the potholes were found in physical clusters near each other i.e. each zone may have multiple potholes. This is likely due to potholes naturally forming as the original zone of weakness fractures and spreads along fault lines (transitions in road surfaces between lanes, trench works, drains etc.). The subsequent COVID-19 circuit breaker (semi-lockdown) measures from April to June 2020 during the experiment period also made subsequent remeasurements impossible. Hence to compensate of such experimental error, a larger radius of 15m is chosen instead of 10m used in previous studies.



Figure 70: Overview of potholes identified via HDBSCAN Clustering 1 plotted via Google Maps. Ground of Truth Points marked by images in circular frames; and suspected potholes are the coloured location markers.

²¹ Epsilon = distance between observations to be considered within the same cluster



Figure 71: Overview of potholes identified via HDBSCAN Clustering 2 plotted via Google Maps. The Ground of Truth Points marked by images in circular frames; and suspected potholes are the coloured location markers.

Table 18: Ground of Truth Zones (Inclusive of Null Cases) Correctly Predicted (22 Zones). This test indicates the overall accuracy and sensitivity of the prediction.

Zone	Road / PCN	Road Artifact Truth	Pothole / Nil	Alpha Pred.	Beta Pred.	Alpha-w/o HDBSCAN Pred.	Beta-w/o HDBSCAN Pred.	Alpha-Pred. Remarks	Beta-Pred. Remarks
Tuas Ave 1	Road	Pothole	Pothole	Pothole	Nil	Pothole	Nil		
Tuas Ave 3	Road	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		
Jalan Ahmad Ibrahim (near FMC Tech)	Road	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		
Jalan Ahmad Ibrahim (near MC Packaging)	Road	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		
Jalan Bahar (Jurong West)	Road	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		
Jalan Bahar (NTU Entrance)	Road	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		
Old Chua Chu Kang Rd (Brickland Rd junction)	Road	Pothole	Pothole	Pothole	Nil	Pothole	Nil		
Mandai Road	PCN	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		
Mandai Road (Opp Lor Lada Hitam)	PCN	Speed Strips	Nil	Pothole	Nil	Pothole	Nil		
Mandai Road (to SLE)	PCN	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		
Mandai Road (Mandai Ave Junction) (Zone A)	PCN	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		

Zone	Road / PCN	Road Artifact Truth	Pothole / Nil	Alpha Pred.	Beta Pred.	Alpha-w/o HDBSCAN Pred.	Beta-w/o HDBSCAN Pred.	Alpha-Pred. Remarks	Beta-Pred. Remarks
Mandai Road (Mandai Ave Junction) (Zone B)	PCN	Pothole	Pothole	Nil	Nil	Pothole	Nil	Pothole - Rejected by HDBSCAN Algo	
Mandai Ave (Nee Soon Camp Side Entrance)	PCN	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		
Sembawang Road	Road	Normal Road	Nil	Nil	Nil	Nil	Nil		
Springside Road	Road	Hump	Nil	Pothole	Nil	Pothole	Pothole		Pothole - Rejected by HDBSCAN Algo
Springside View A	Road	Hump	Nil	Nil	Nil	Nil	Nil		
Springside Ave	Road	Hump	Nil	Pothole	Nil	Pothole	Pothole		Pothole - Rejected by HDBSCAN Algo
Springside View B	Road	Pothole	Pothole	Pothole	Pothole	Pothole	Pothole		
Springside Drive	Road	Hump	Nil	Nil	Nil	Nil	Nil		
Nee Soon Road	Road	Hump	Nil	Nil	Nil	Pothole	Nil	Pothole - Rejected by HDBSCAN Algo	
Lower Seletar Reservoir Park (Zone A)	PCN	Speed Strips	Nil	Pothole	Nil	Pothole	Nil		
Lower Seletar Reservoir Park (Zone B)	PCN	Speed Strips	Nil	Pothole	Nil	Pothole	Nil		
Accuracy				72.73%	86.36%	72.73%	77.27%		

Table 19: Accuracy Against Known Pothole Ground of Truth (GOT) Points containing potholes (33 Points). This test indicates the specificity of the prediction.

Percentage of Suspected Potholes Near Tagged Ground of Truth (GOT) Potholes (33 Points)	Alpha w/o HDBSCAN	Alpha	Beta w/o HDBSCAN	Beta
Count	622	589	298	267
Percentage of Suspected Points on Target ($\pm 15m$)	77.01%	76.74%	80.201%	84.27%
Percentage of Suspected Points on Target ($\pm 10m$)	65.434%	65.195%	69.463%	74.906%

Findings & Conclusion

Analysing the results, in Table 18 and Table 19, the prediction outcomes from Beta is superior to Alpha, producing a higher accuracy and specificity in every metric. However, it is much less sensitive than Alpha, especially at detecting milder pothole events. Overall, when compared against the ground of truth points, it can successfully predict the existence of potholes correctly between 69 – 86% of the time minimally. It was also observed that the use of HDBSCAN can achieve up to at 14% improvement in prediction accuracy by filtering out more spurious observations, as was the case for Beta in Table 18. However, setting the minimum cluster size (that affects the filtering process) should also take into account the amount of traffic

in the area, especially if there is a road artefact that may produce similar signals to potholes such as road humps. As by the virtue having more people travel past the same area, the algorithm may be tricked into accepting such spurious observations that made it past the ensemble voting checks previously, as legitimate potholes, resulting in false-positive predictions. On the flip side, if the road experiences little traffic, there is also a chance that legitimate potholes may be excluded by the HDBSCAN algorithm since it fails to meet the minimum required number of observations to be considered a pothole cluster.

Overall, the combination of HDBSCAN and ensemble voting of WAPP and K-Means clustering has been proven to be moderately effective in the detection of potholes as a proof of concept experiment to use smartphone sensors to detect potholes, with minimal false-positive events.

Limitations

Overall, there are 3 key limitations of the method deployed, namely:

1. The vehicle's tires must physically make contact for the pothole to be detected
2. For real-time applications, unsupervised learning methods such as K-means clustering cannot be used as the detection and tagging of potholes needs to be performed in real-time
3. For real-time applications, extended high-frequency sampling of the GPS needs to be performed due to power consumption issues

As mentioned in the previous sections, since detection is motioned primarily based on the use of accelerometer and gyroscope sensors, direct vehicle contact of the road artefact is required to generate a signal to be matched against. Given that only a small portion of the vehicle's surface is in contact with the road (typically just the tires) – typically just the edges of each lane for non-single tracked vehicles e.g. cars – only a small portion of the road surface is being sampled, potentially leading to false-negative events. However, the use of crowdsensing to pool results from multiple users (such as members of the public) could potentially mitigate this somewhat for roads with heavier traffic. But for larger potholes, drivers would typically avoid the pothole to prevent damage to their vehicles when presented the option to do so, resulting in false-negative events which crowdsensing is less able to compensate for.

The second and third points are related to the deployment of the model for real-time applications would impose several restrictions due to the computational speed and power requirements. Unsupervised learning methods generally require a longer period as sufficient data must first be collected before analysing in post. To overcome the power drain of the GPS, the GPS should be used only on an on-demand basis, upon pothole detection. To do this, fast signal processing and supervised learning methods would be required to trigger the activation of the GPS and tag the pothole, with minimum displacement error. However, it should be noted that the displacement error is also a function of the vehicle's speed and direction of travel, hence this must be taken into account for the minimum speed required.

Future Work

Now that the proof of concept test has been proven to be successful, this could be adapted for motorised vehicles instead. However, a higher sampling rate may be needed to better capture the signal pattern given that such vehicles are travelling at much higher speeds. Also, this could be used to augment the data collection process for the application of supervised learning techniques for pothole detection, especially since good quality clean data is required for training to get the best prediction results. Regardless, it would also be interesting to see if such classifier techniques can also be used to estimate the extent of road damage and classify different types of road surfaces through signal analysis too. The latter would also

require testing against more types of road surfaces such as expansion gaps, drain covers and patches from past trench works to make the model more robust. Further work could also be done to finetune the input variables and features generated further, reducing the amount of noise and processing required.

Lastly, to overcome the limitations of the need for physical contact of the road artefact for pothole detection, image recognition techniques could also be considered to assist in pothole detection. Especially since LTA already has an archive of road defect images taken by road inspectors on their smartphones as part of their routine road surveys. Hence compiling these images and tagging them could open up new methods of detecting such road defects and other road artefacts not directly on the tarmac surfaces, such as paint weathering of traffic signs, damaged traffic fences etc.

Refinement of the above-mentioned techniques could then be deployed in an app for the general public such as LTA's *MyTransport.SG* app as a new module or as an entirely new app to capitalised on the analytics capabilities developed.

Appendix A – Brand Share by Sales Volume of Smartphone Market

Figure 72: Brand Share by Sales Volume of Singapore Smartphone Market (2014-2019) from Euromonitor

Brand Name	2014	2015	2016	2017	2018	2019
iPhone (Apple Inc)	37.5	38.7	39.1	36.5	34.5	33.1
Samsung (Samsung Corp)	29.0	29.0	29.2	30.8	32.7	32.9
Xiaomi (Xiaomi Corp)	7.6	6.3	3.8	4.6	5.2	5.6
Huawei (Huawei Technologies Co Ltd)	3.7	4.1	4.9	5.4	5.7	4.3
Oppo (Guangdong OPPO Mobile Telecommunications Corp Ltd)	2.3	2.7	3.2	3.5	3.8	4.0
Sony (Sony Corp)	5.0	4.7	2.6	2.6	2.3	2.1
LG (LG Corp)	1.2	1.4	1.0	1.2	0.8	0.6
Nokia (Nokia Corp)	5.3	3.4	1.8	1.0	0.7	0.5
HTC (HTC Corp)	2.5	1.7	1.5	0.9	0.6	0.4
Blackberry (BlackBerry Ltd)	-	-	-	-	-	-
Blackberry (Research in Motion Ltd)	-	-	-	-	-	-
Motorola (Google Inc)	-	-	-	-	-	-
Motorola (Motorola Inc)	-	-	-	-	-	-
Motorola (Motorola Mobility Inc)	-	-	-	-	-	-
Sony Ericsson (Sony Ericsson Mobile Communications AB)	-	-	-	-	-	-
Others	6.0	8.1	12.8	13.5	13.6	16.4
Total	100.0	100.0	100.0	100.0	100.0	100.0

Appendix B – Road Anomaly Detection

B01 – Fourier Transform Results

Linear Acceleration (20Hz)

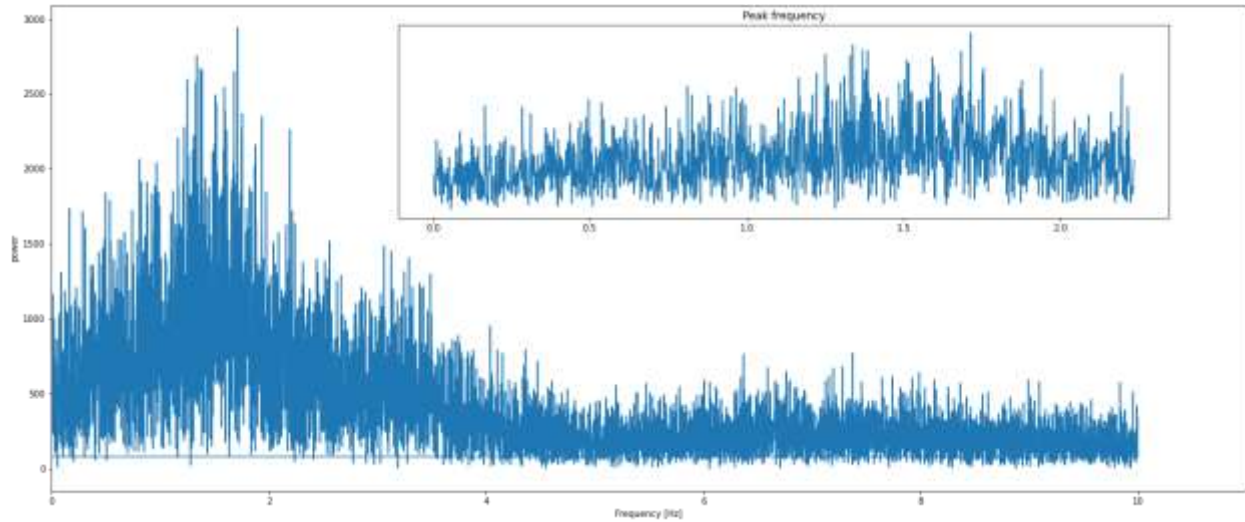


Figure 73: Fourier transform on normal cycling event Z-axis linear acceleration signal

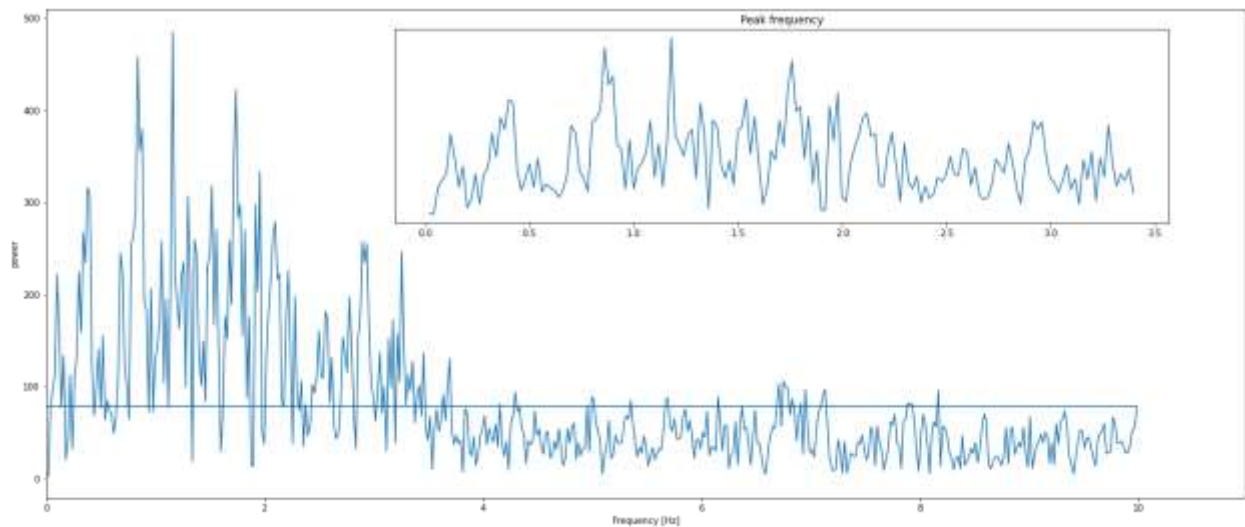


Figure 74: Fourier transform on road hump event Z-axis linear acceleration signal

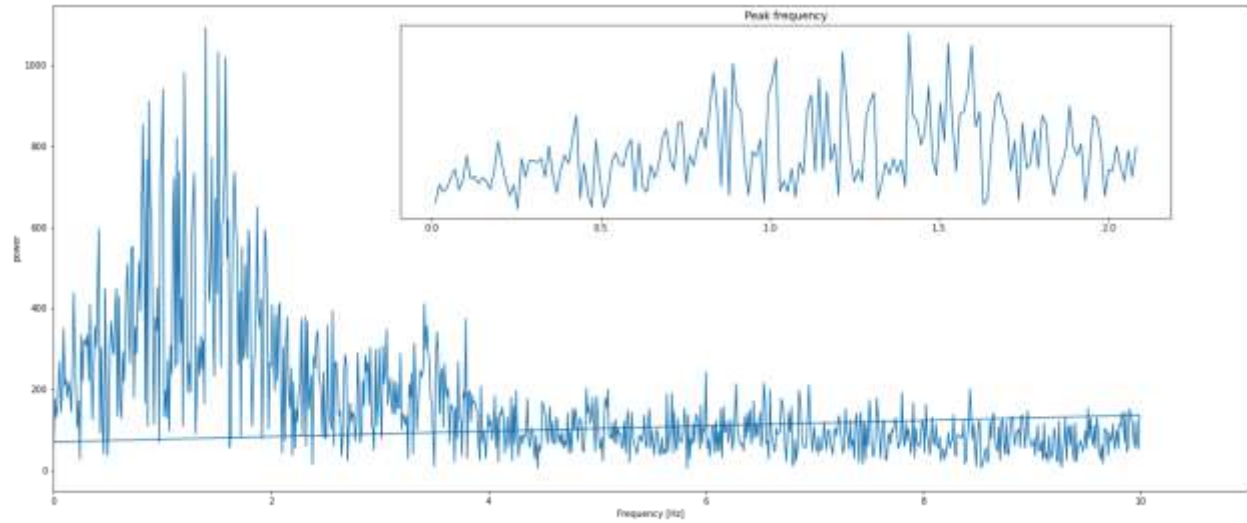


Figure 75: Fourier transform on speed strips event Z-axis linear acceleration signal

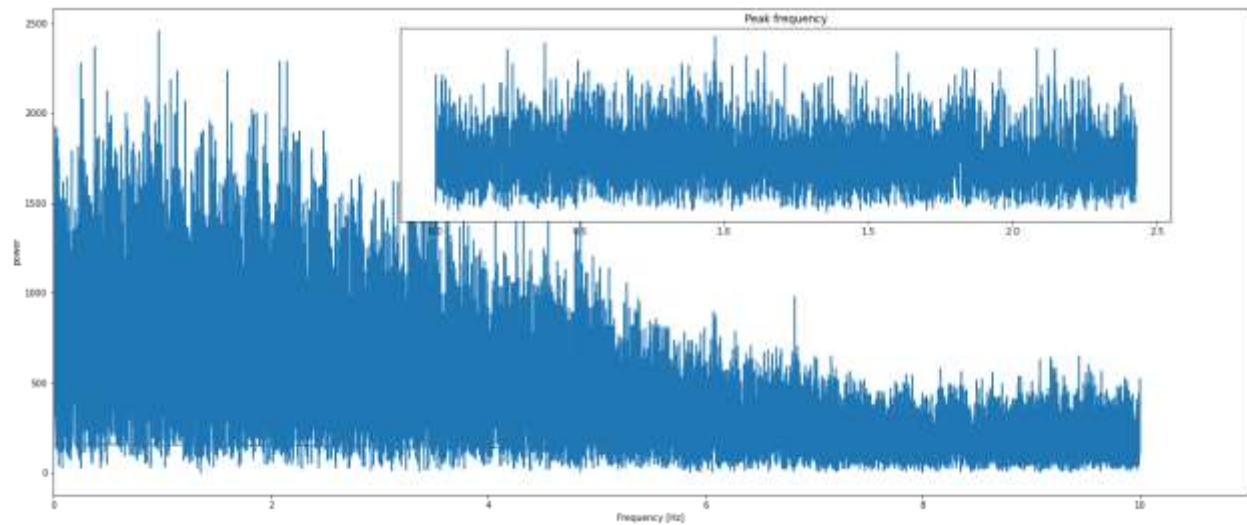


Figure 76: Fourier transform on pothole event Z-axis linear acceleration signal

Gyroscope Rotation (20Hz)

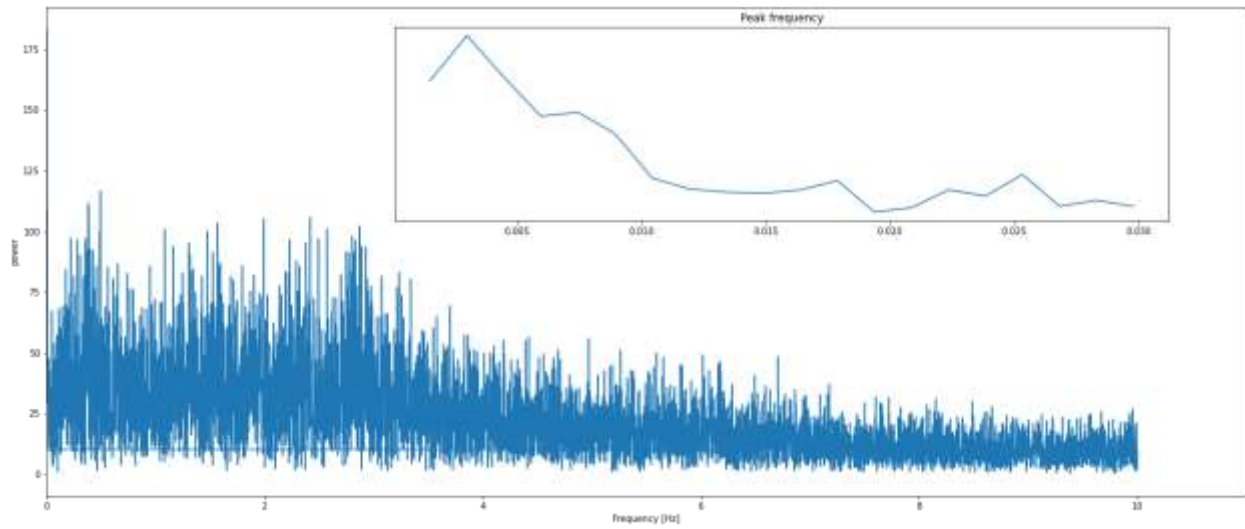


Figure 77: Fourier transform on normal cycling event on X-axis gyroscope rotation signal

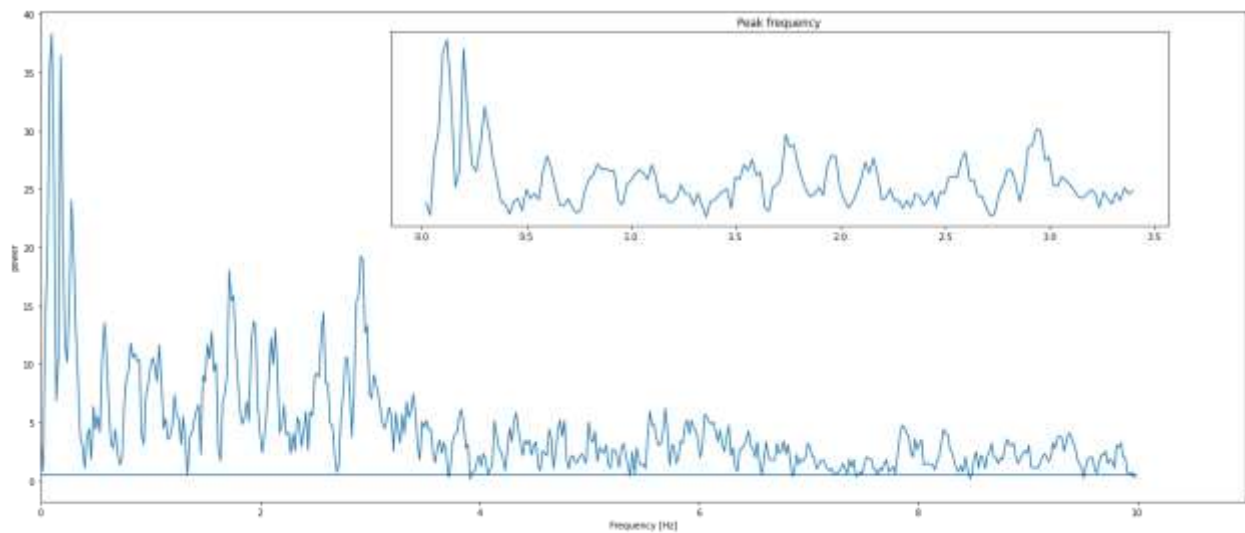


Figure 78: Fourier transform on road hump event on the X-axis gyroscope rotation signal

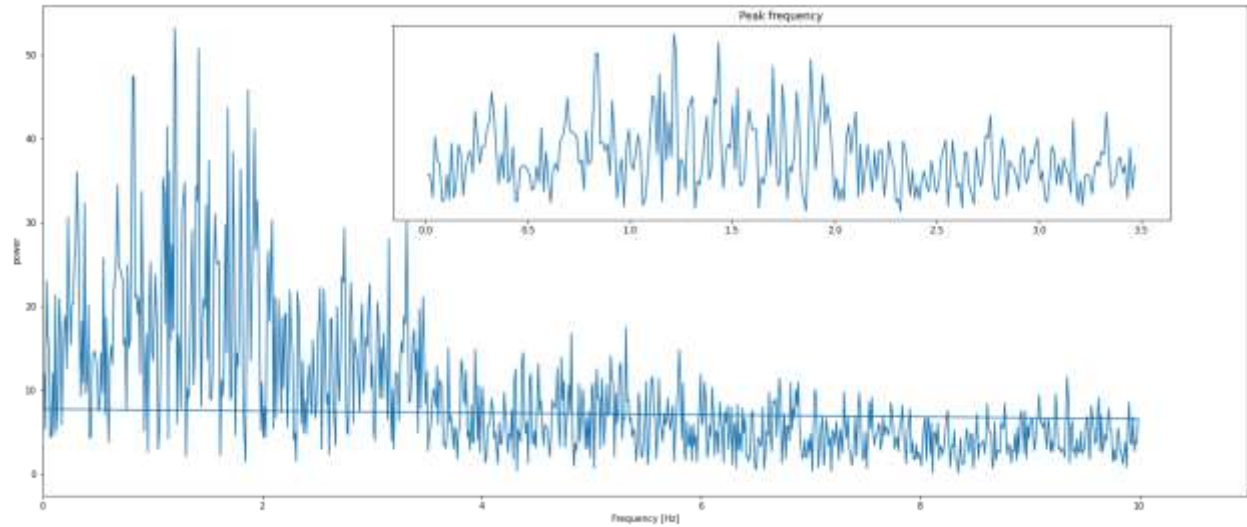


Figure 79: Fourier transform on speed stripe event on the X-axis gyroscope rotation signal

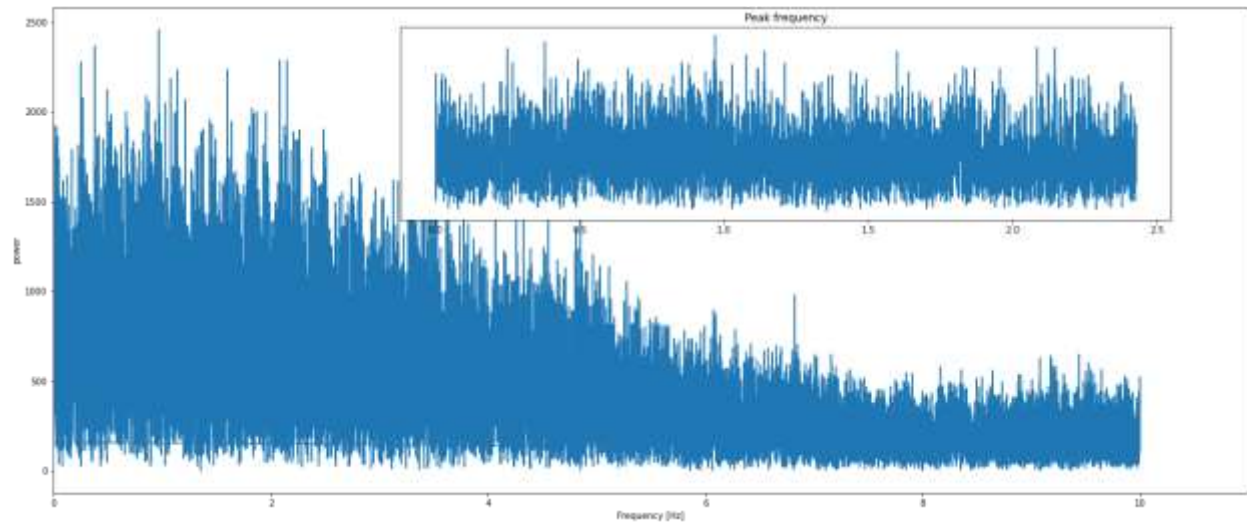


Figure 80: Fourier transform on pothole event on X-axis gyroscope rotation signal

B02 – Filtered Signal Data
Linear Acceleration (20Hz)

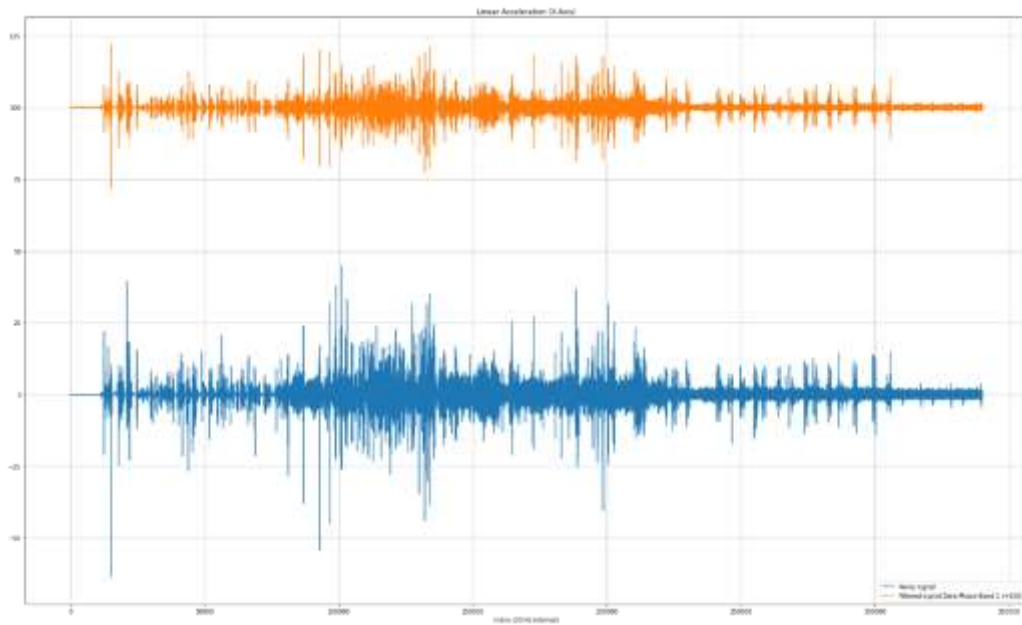


Figure 81: Filtered X-Axis Linear Acceleration

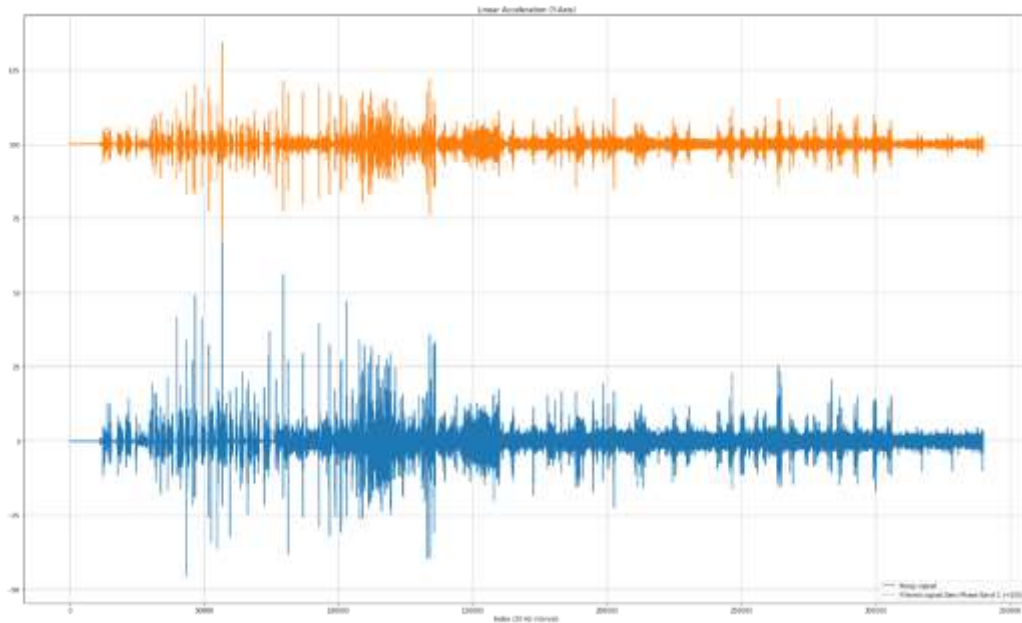


Figure 82: Filtered Y-Axis Linear Acceleration

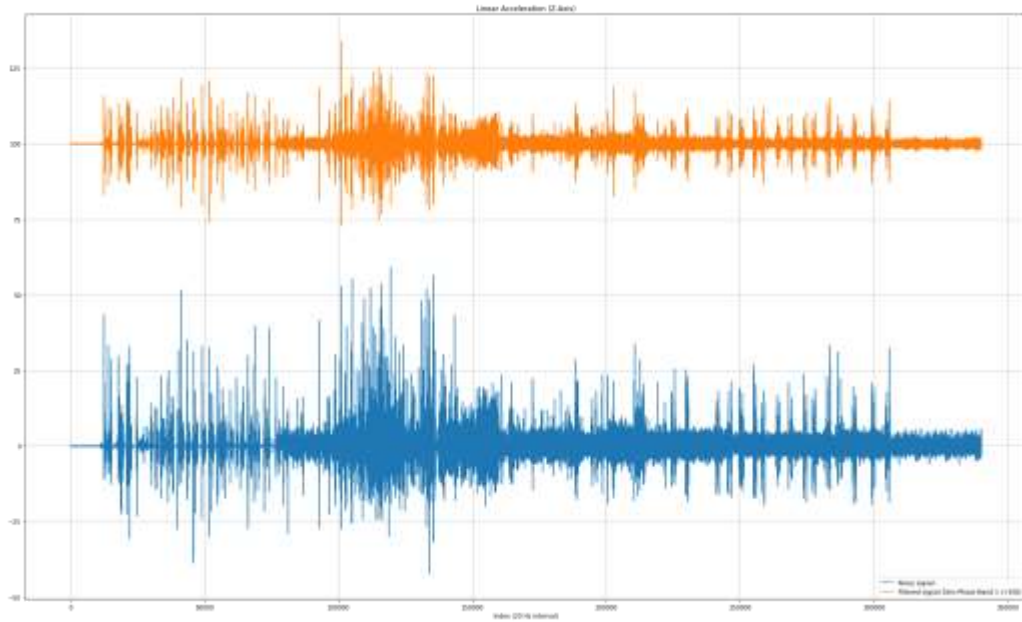


Figure 83: Filtered Z-Axis Linear Acceleration

Gyroscope Rotation (20Hz)

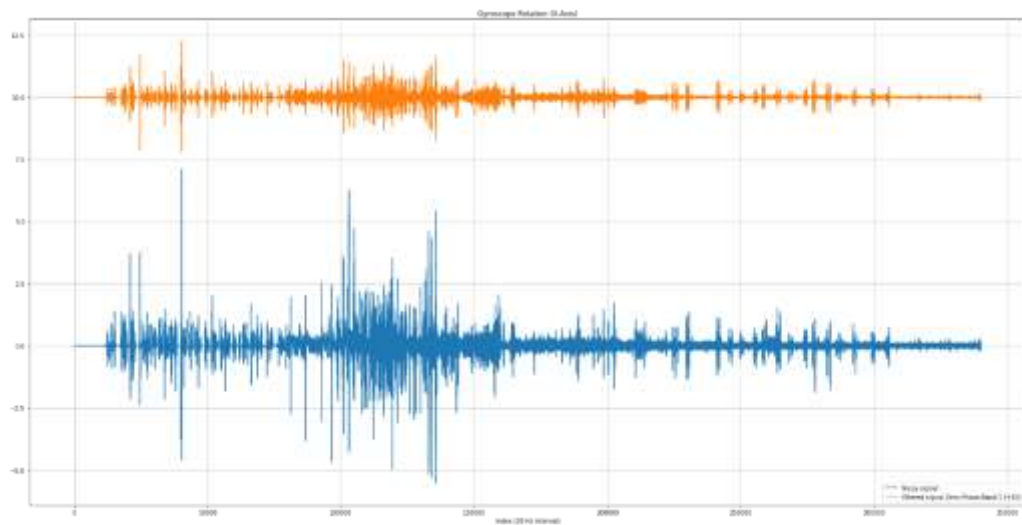
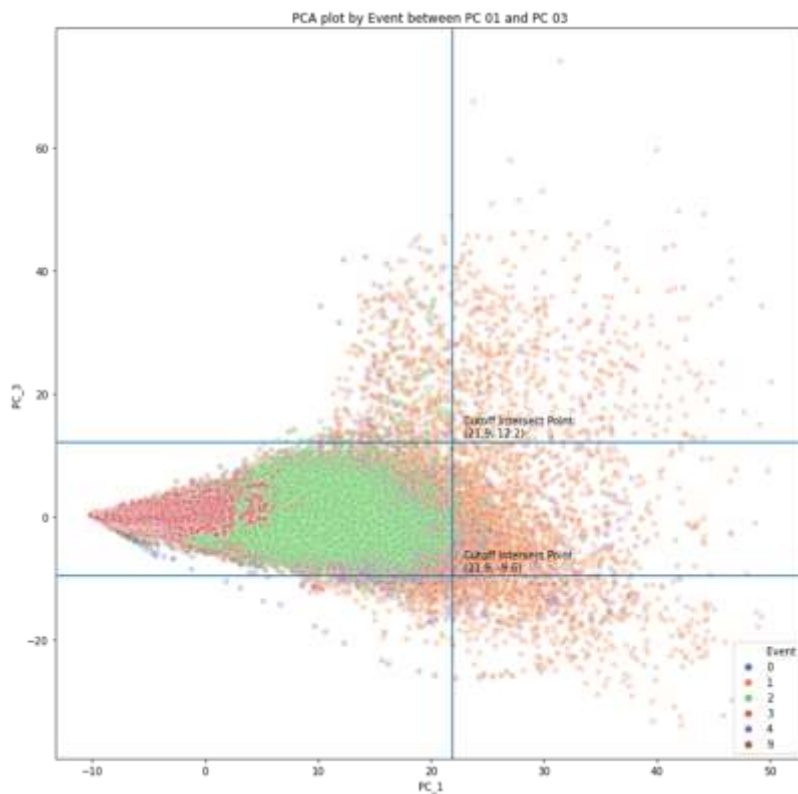
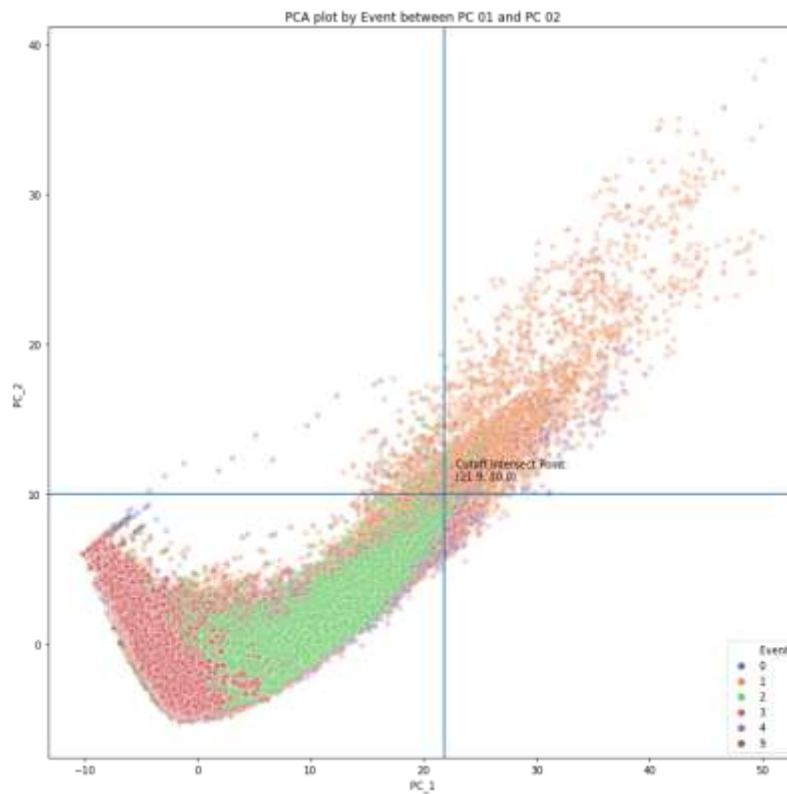
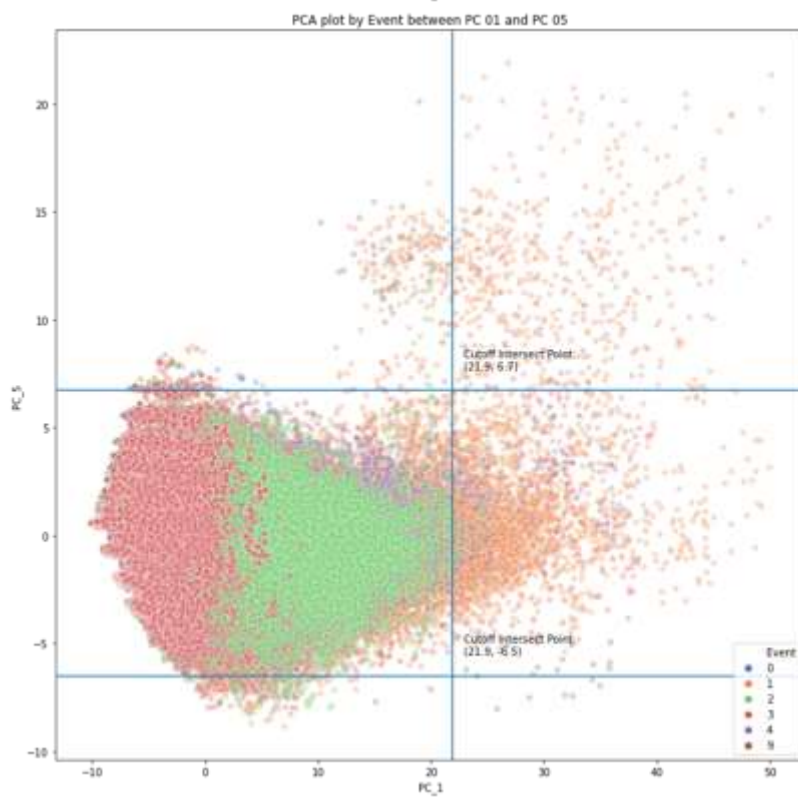


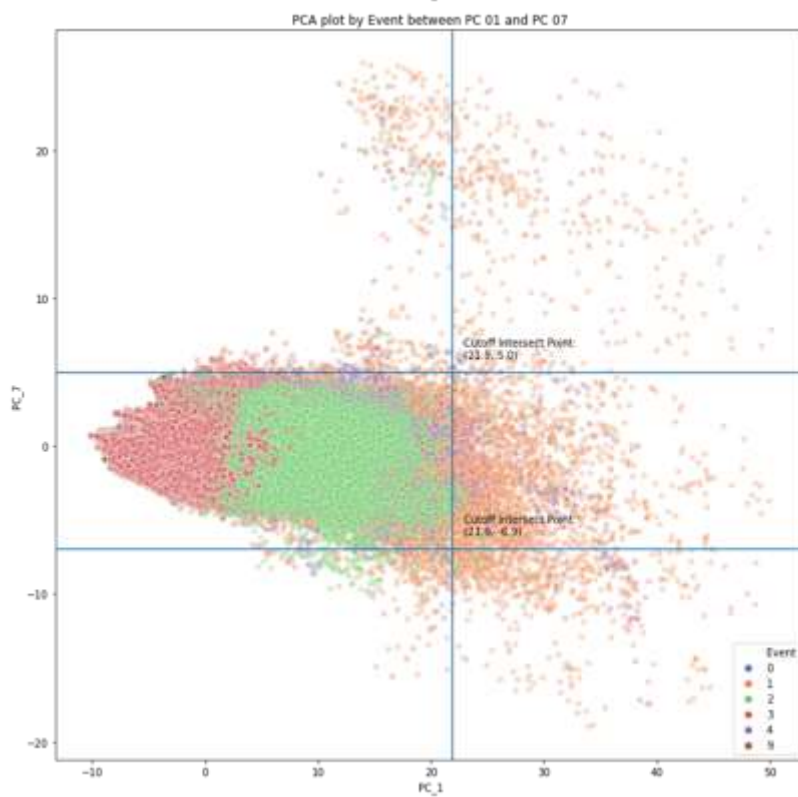
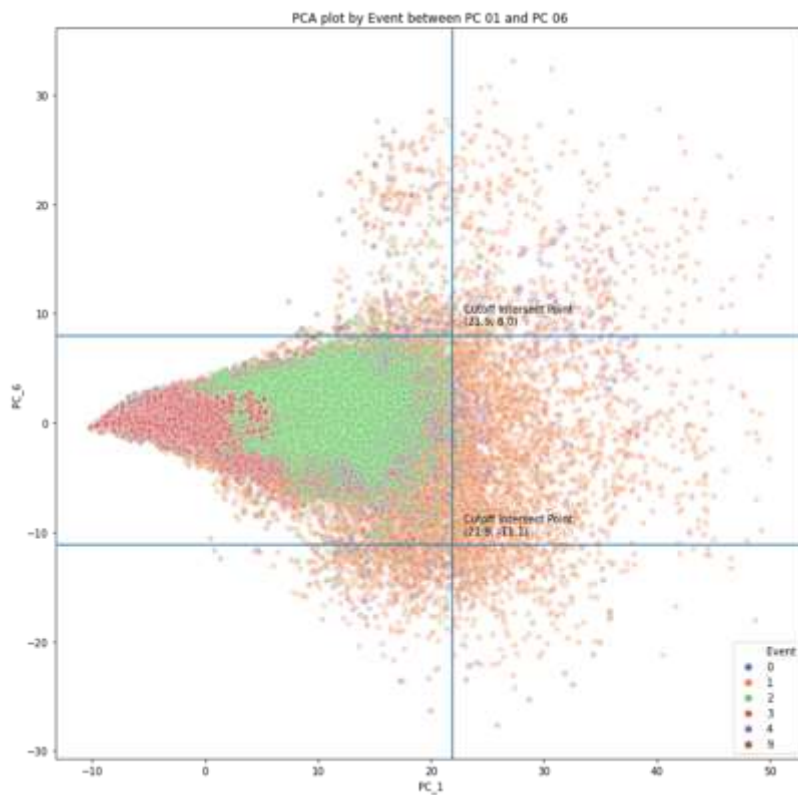
Figure 84: Filtered X-Axis Gyroscope Rotation

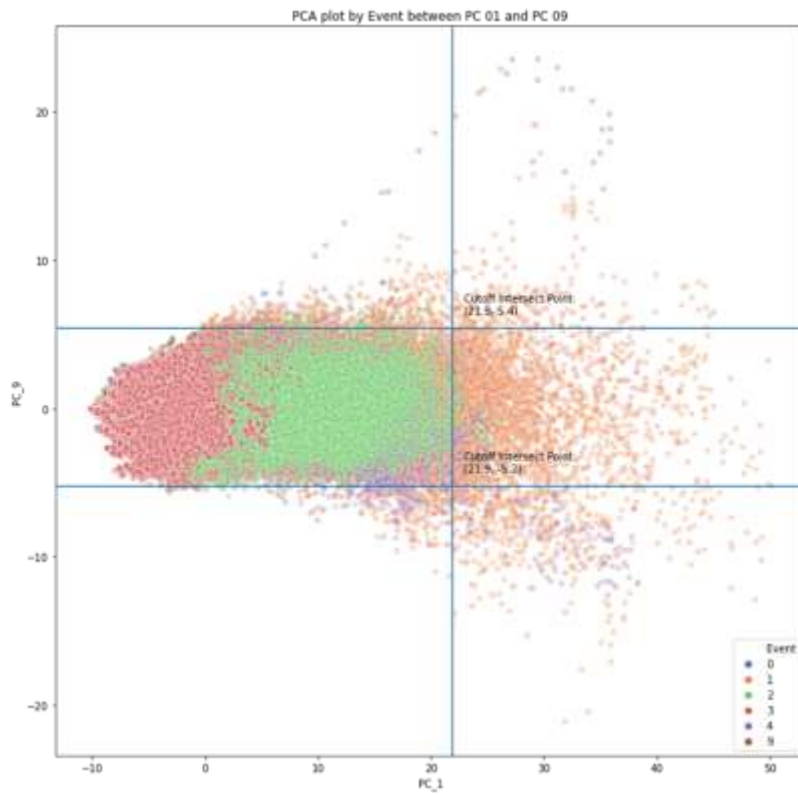
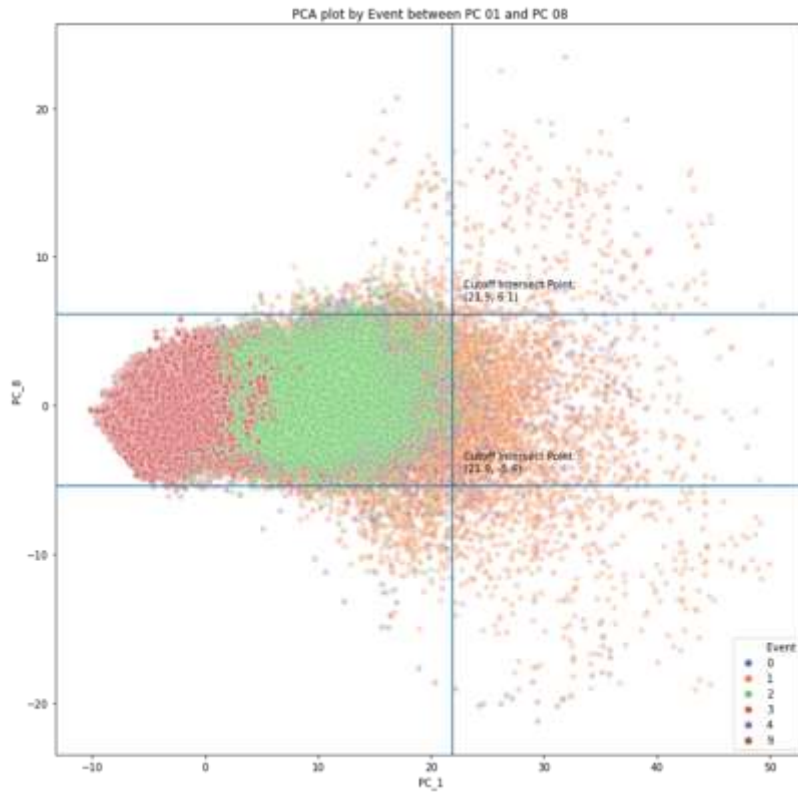
B03 – Principal Components

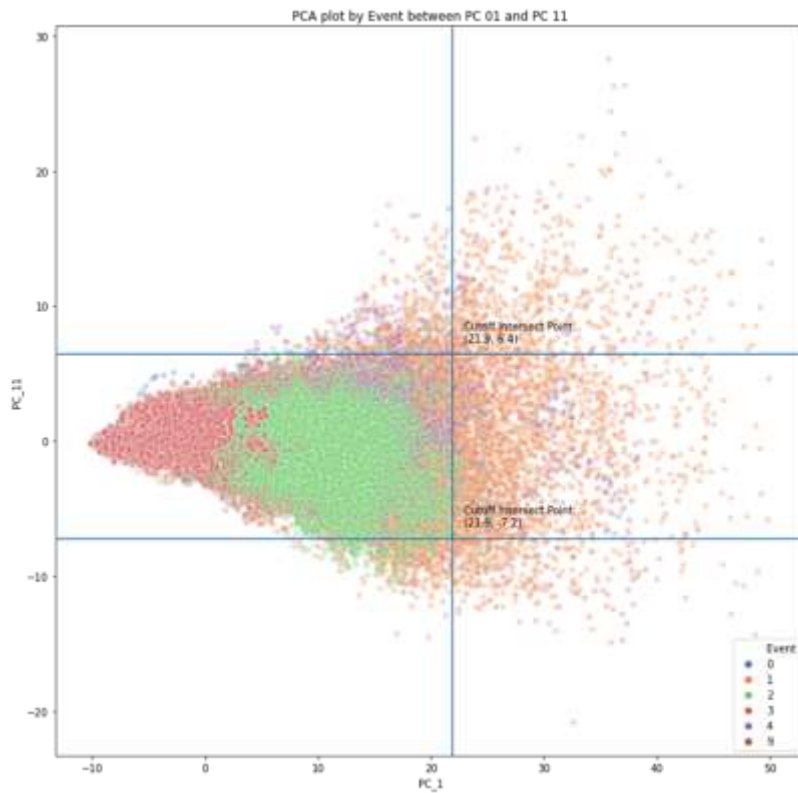
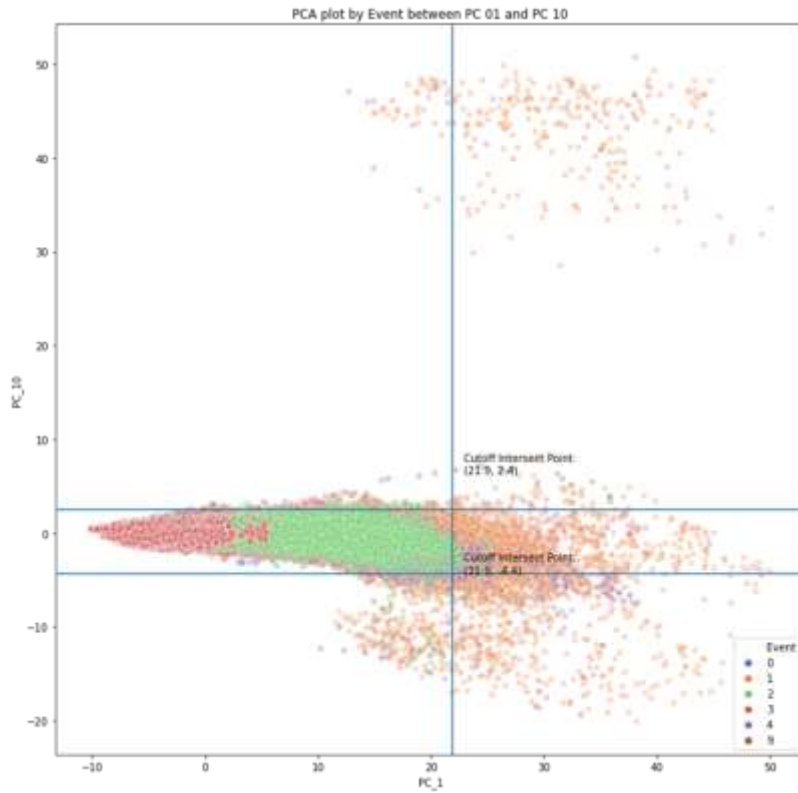
Threshold points of the most extreme 0.2% values (single: PC 1-2; and two-tail: PC 3-26) plotted.

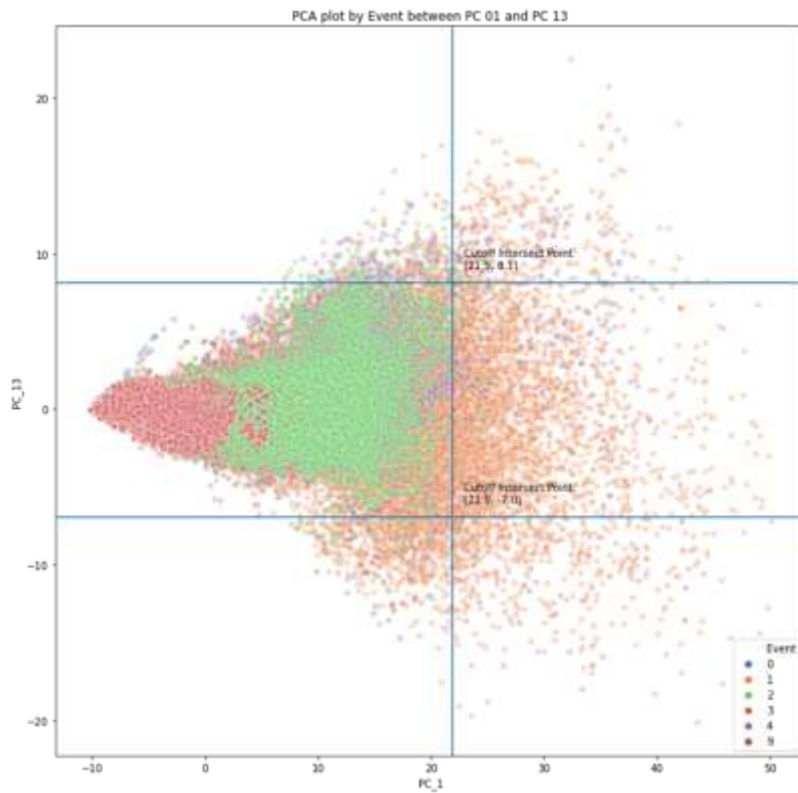
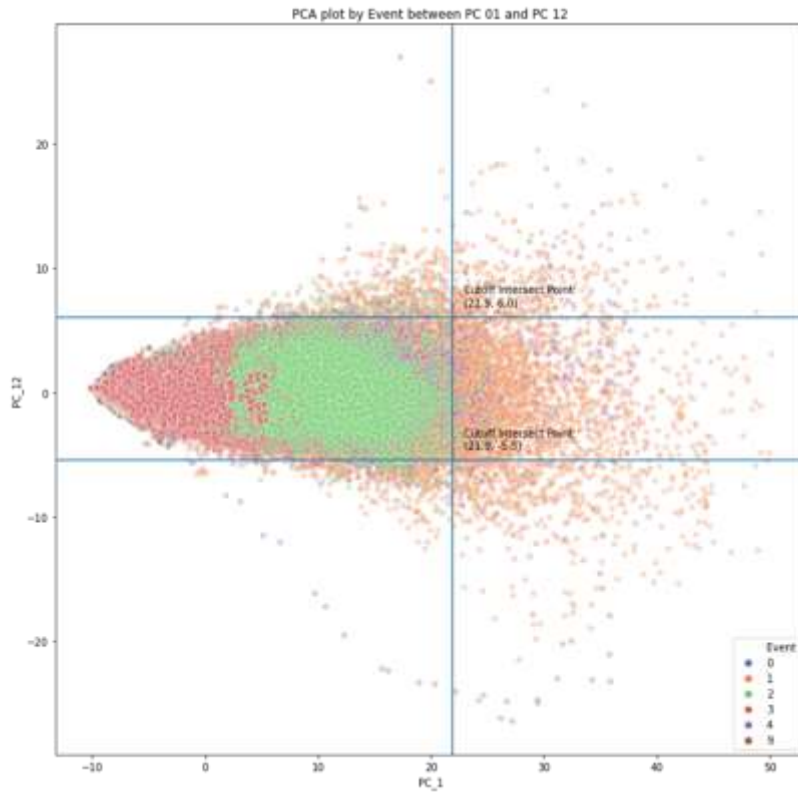


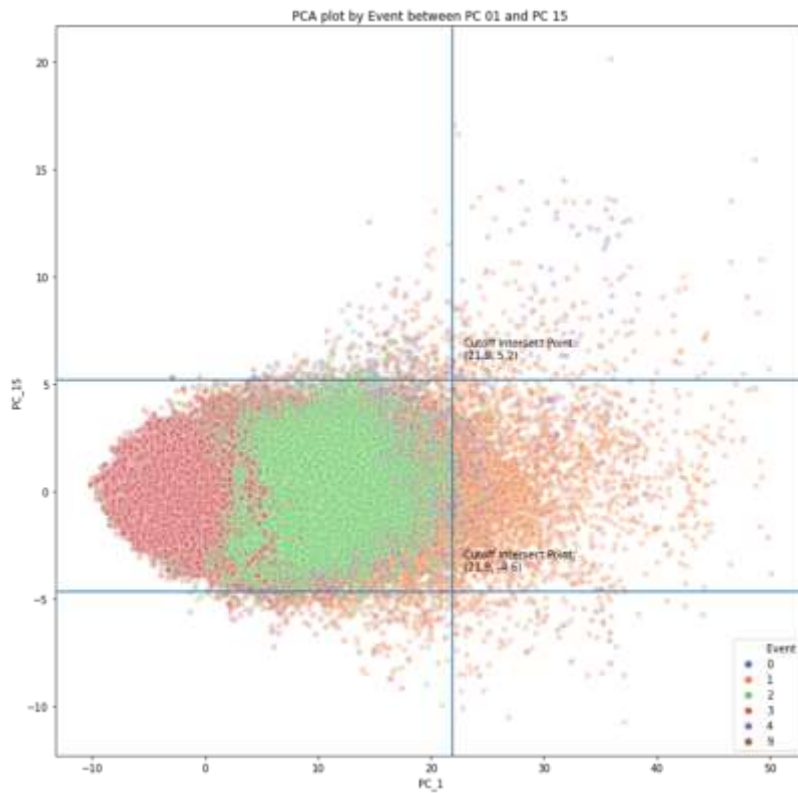
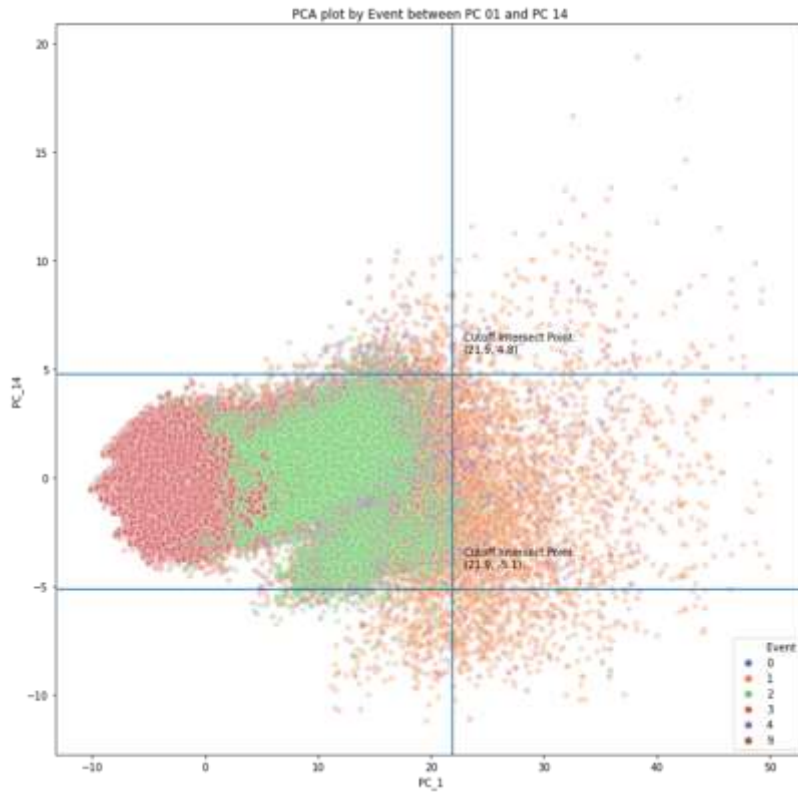


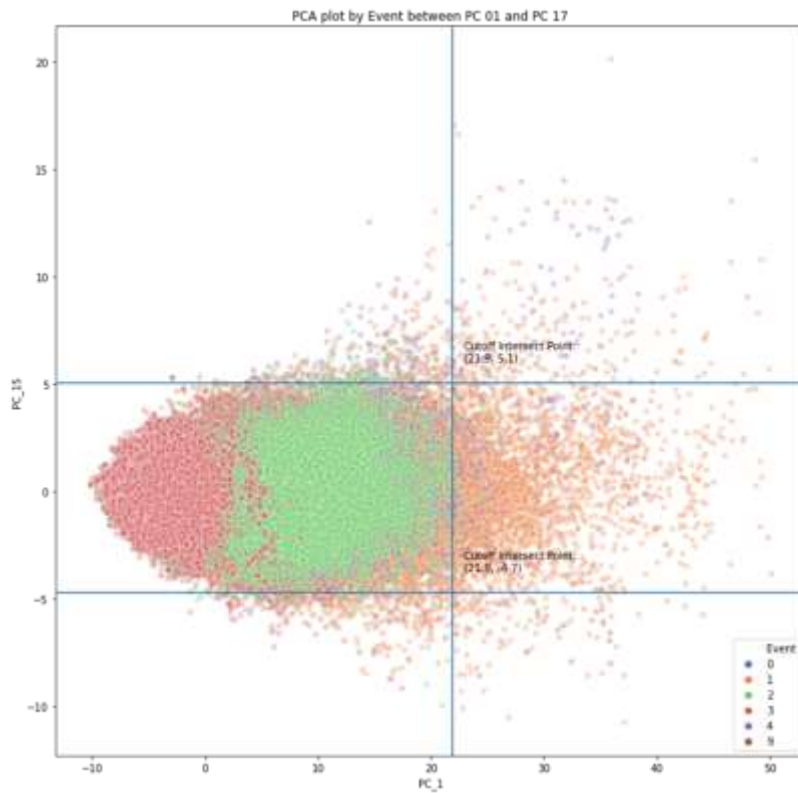
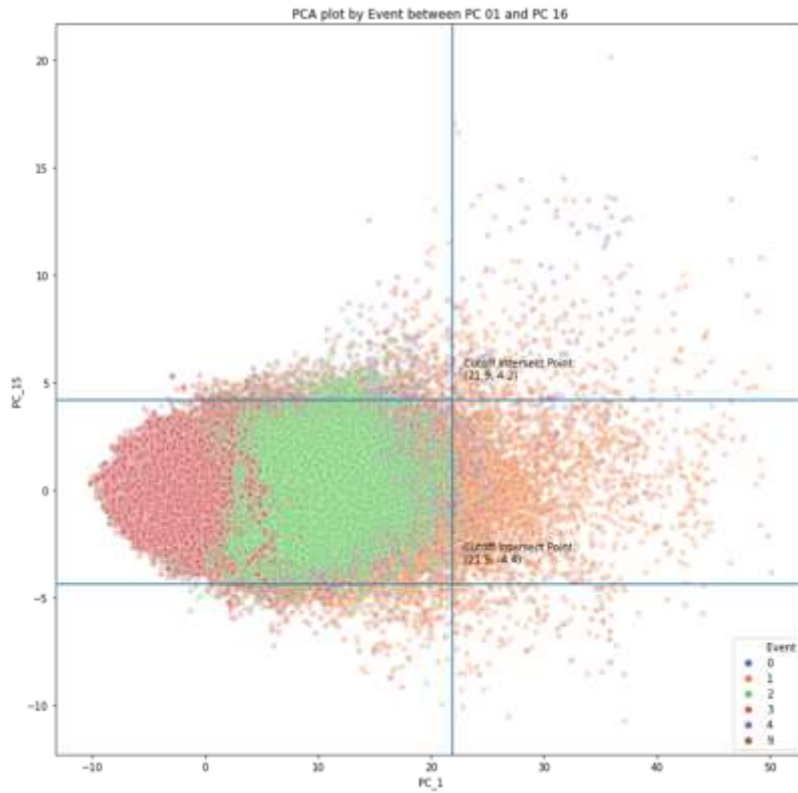


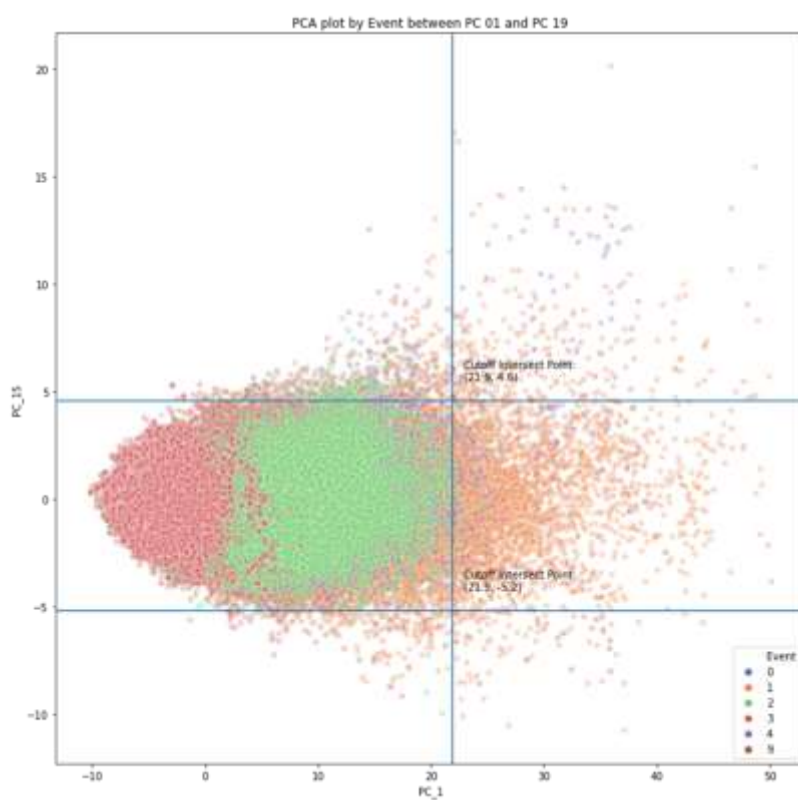
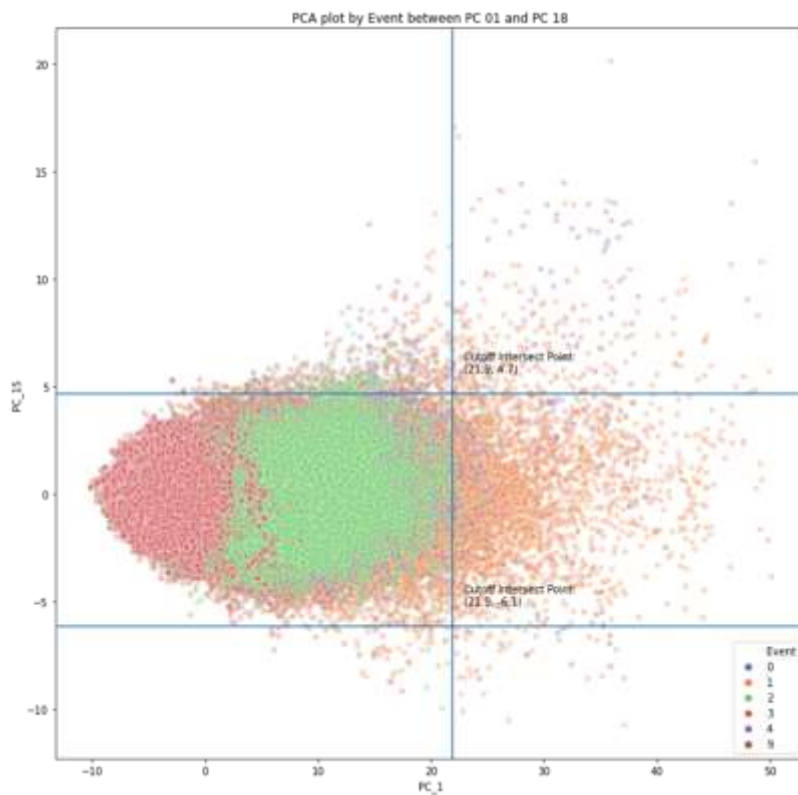


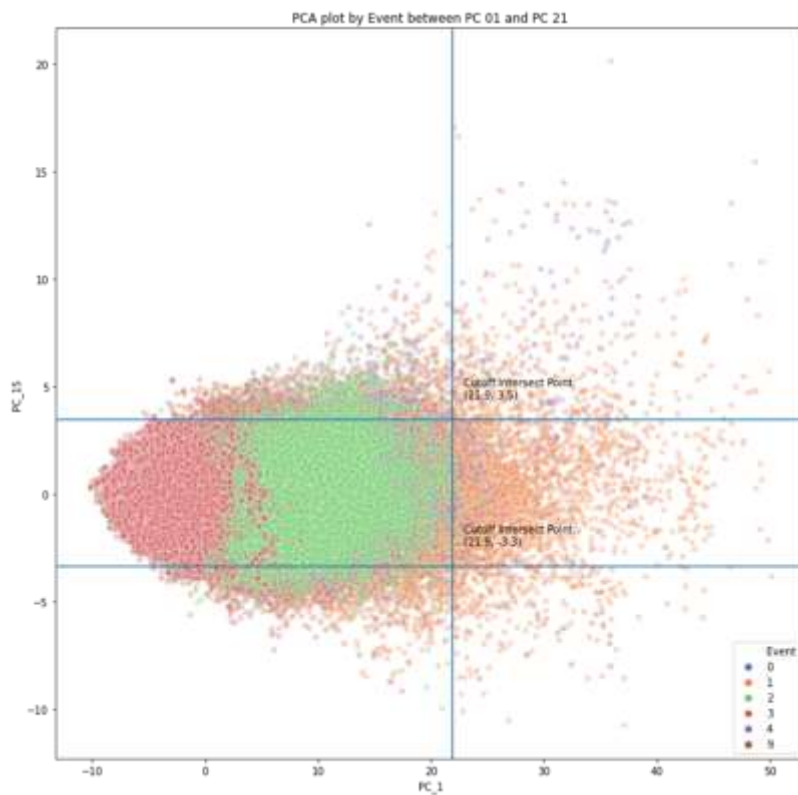
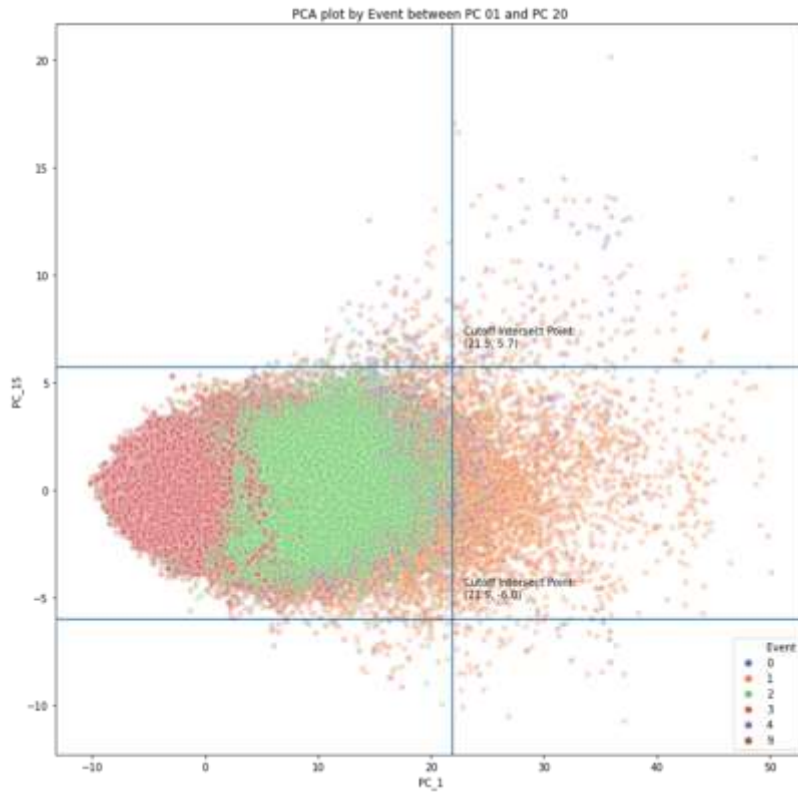


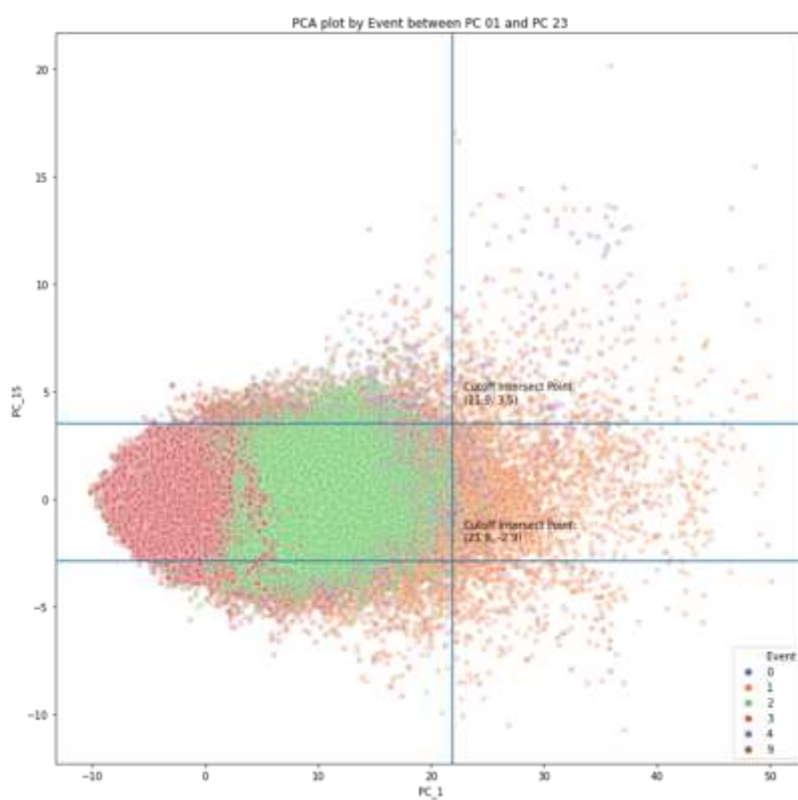
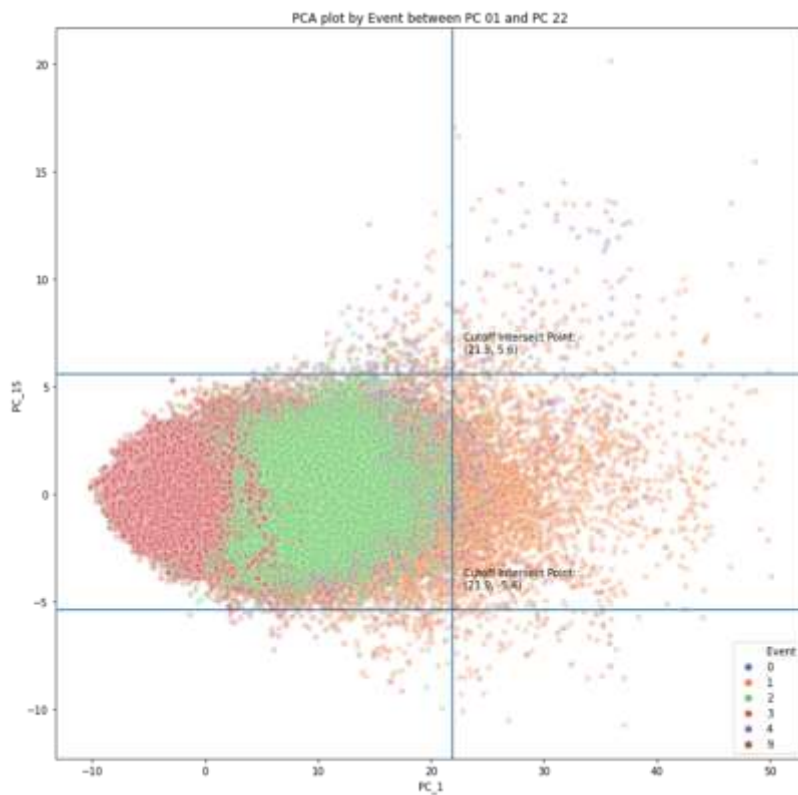


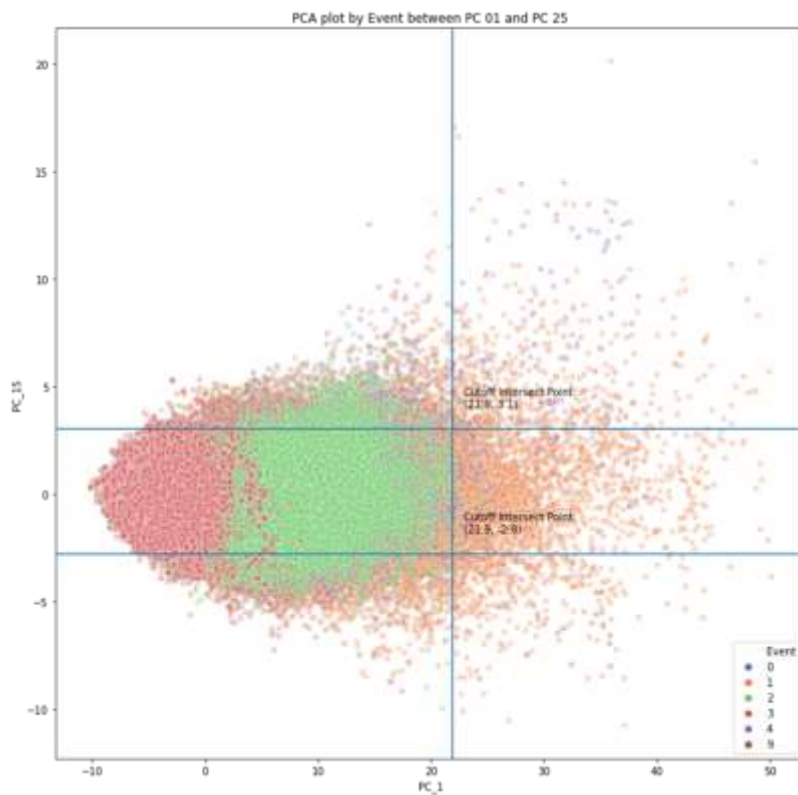
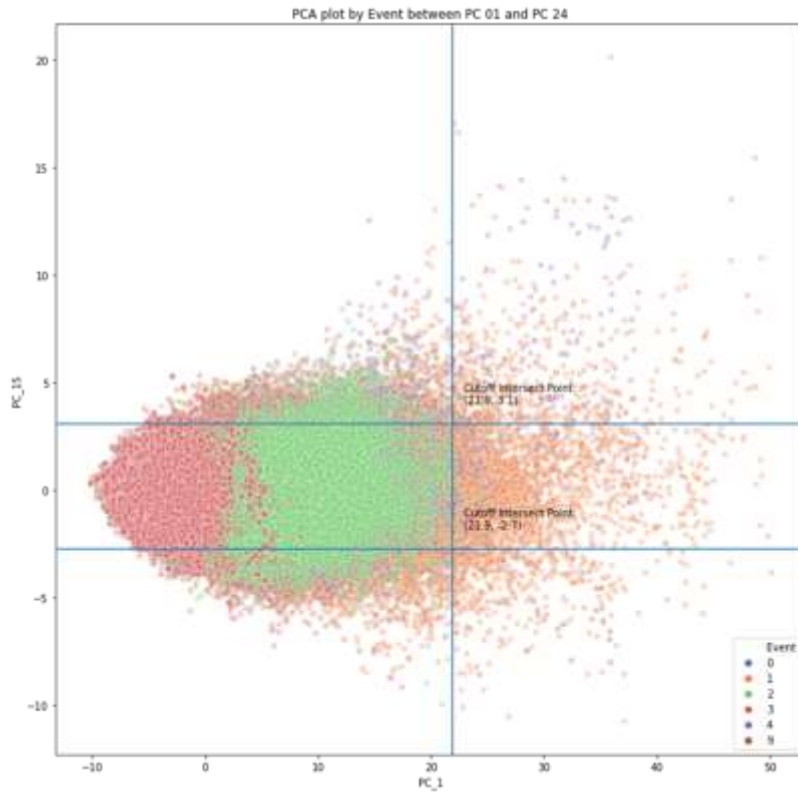


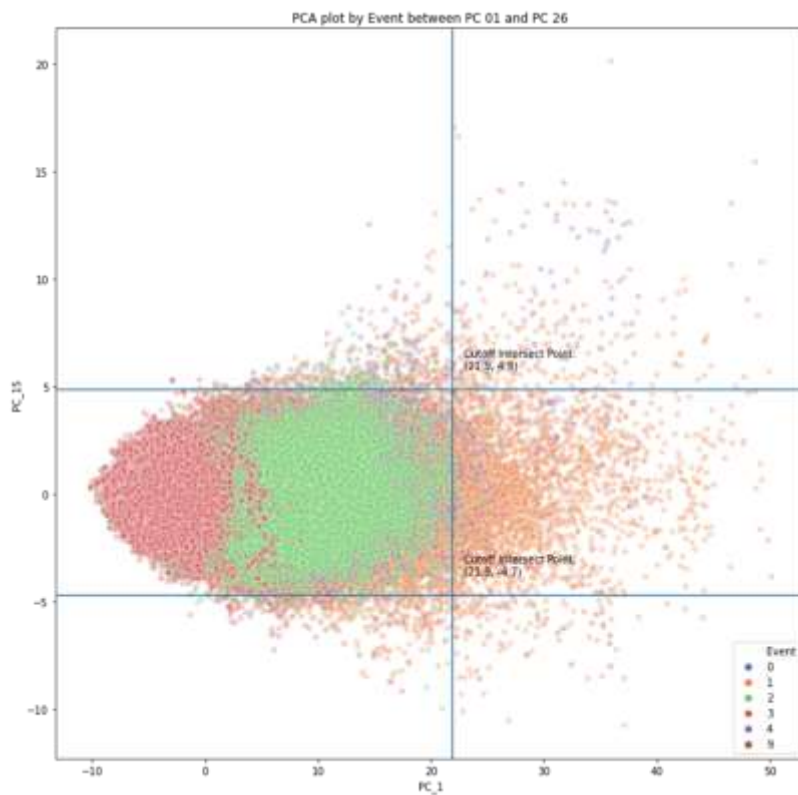












B04 – K-Means Clustering

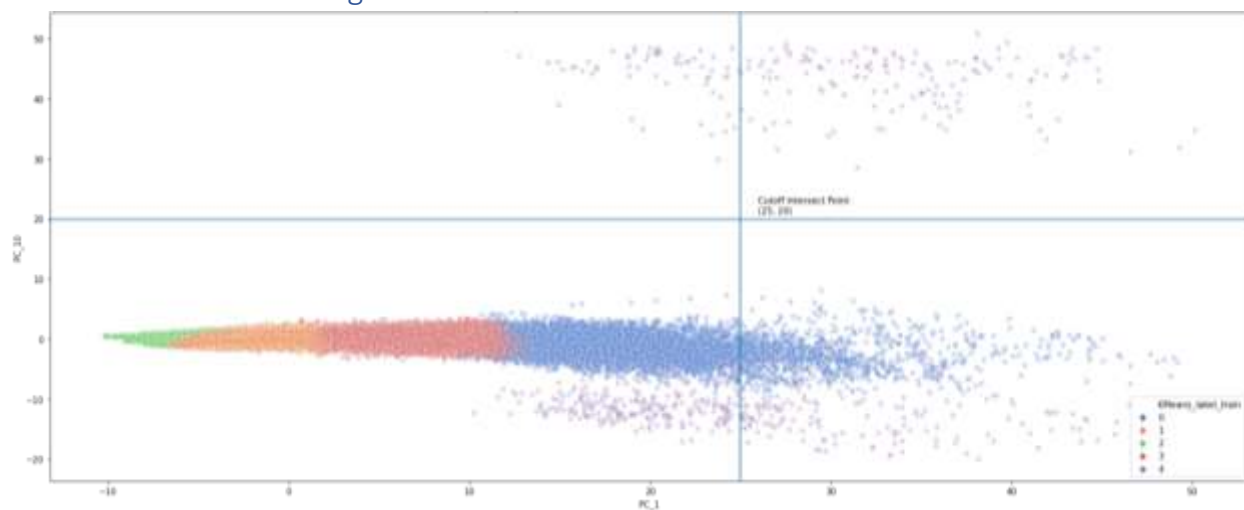


Figure 85: PC Plot of K-Means Cluster Labels (Train Data)

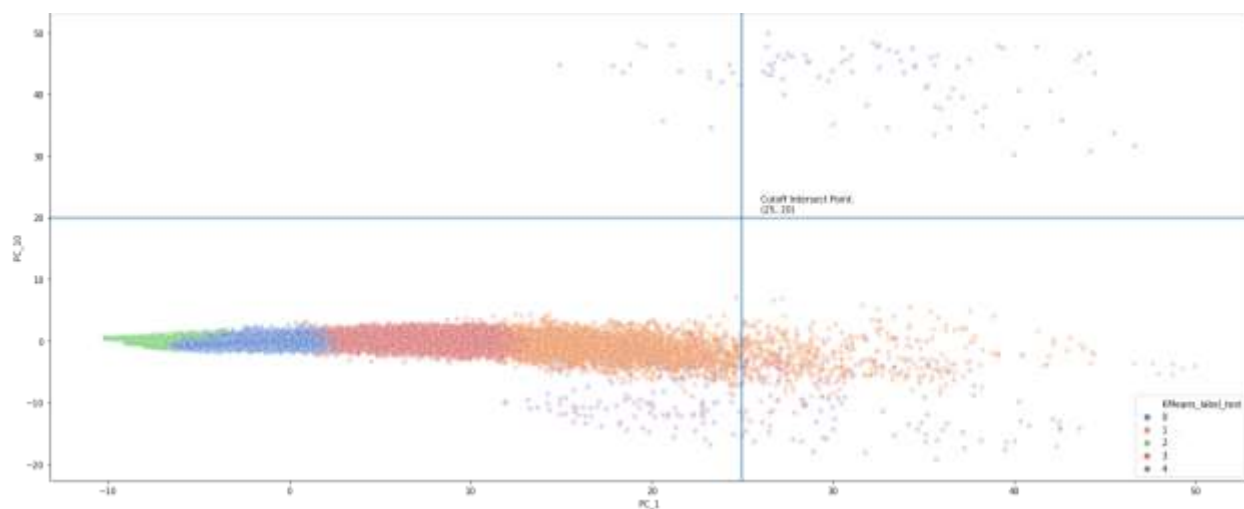


Figure 86: PC Plot of K-Means Cluster Labels (Test Data)

B05 – Results from Other Clustering Methods

Local Outlier Factor (LOF) Method

Potholes are generally treated as an outlier event, hence outlier detection methods like LOF may be used to detect such events. However, poor discrimination of pothole event was observed instead, making it not suitable for road anomaly detection. This could be due to the similarity of pothole events with non-pothole events (which is a large bin of different sub-events) and that insufficient normal cycling event data was collected to make the density clustering more pronounced.

Key Settings Used:

1. Input variables: PC 1-10
2. N-neighbours: 100
3. Contamination: 0.1

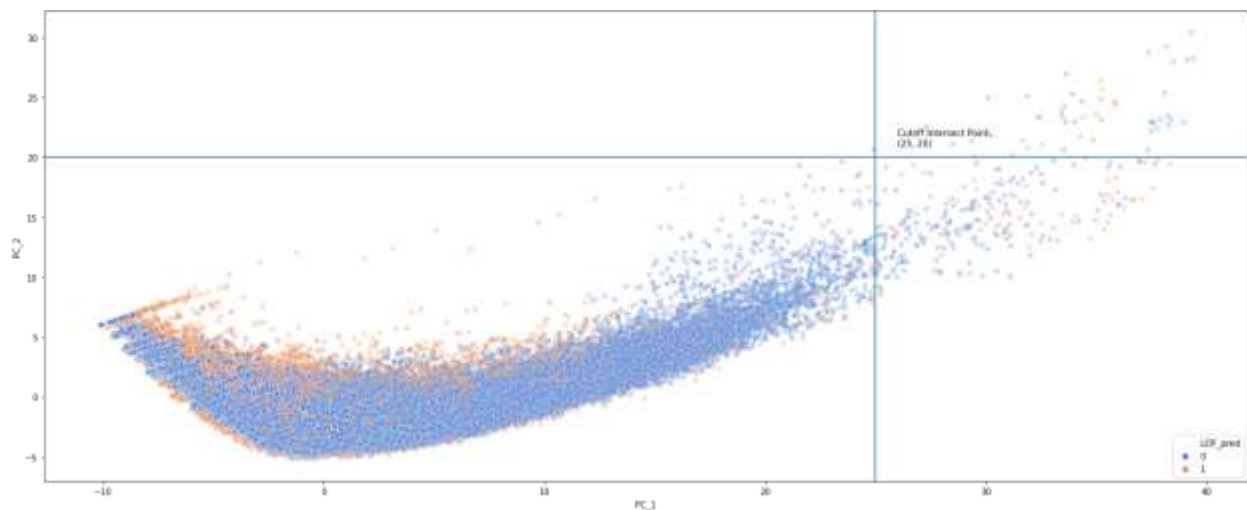


Figure 87: PC Plot of LOF Clusters; filtered by tagged non-pothole events. Outliers (potential potholes) are tagged as 1 in orange.

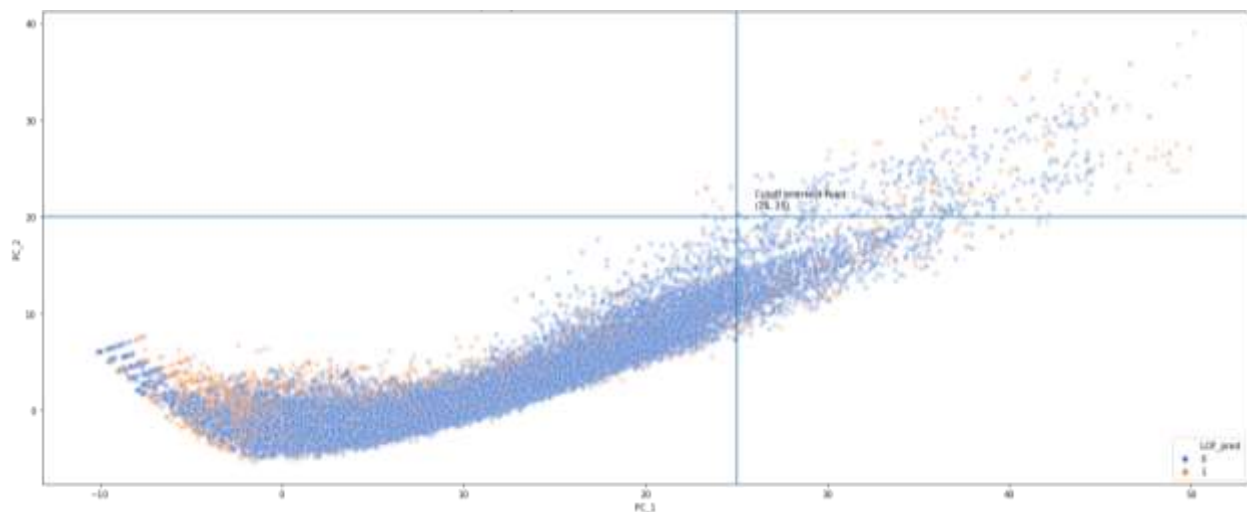


Figure 88: PC Plot of LOF Clusters; filtered by tagged pothole events. Outliers (potential potholes) are tagged as 1 in orange.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Poor discrimination of pothole events observed, making it unsuitable for road anomaly detection. This is likely caused the insufficient spread of the true pothole observations from the non-pothole observations, as either an outlier or a separate cluster. This is partly due to various road anomalies producing similar signals to potholes, confusing the clustering algorithm.

Key Settings Used:

1. Input variables: PC 1-10
2. Epsilon value²²: 2
3. Minimum cluster size: 100

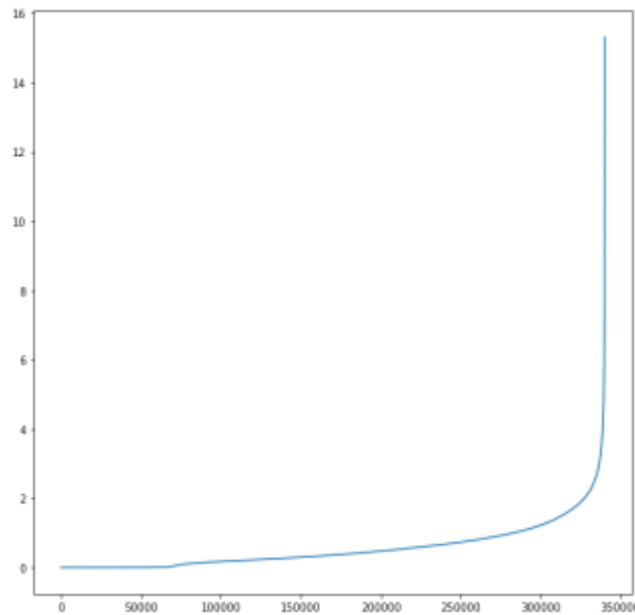


Figure 89: Elbow plot to determine optimum epsilon value. The X-axis refers to the number of observations; Y-Axis refers to the epsilon value. The epsilon value is taken based on the inflexion point which is around a value of 2.

²² Epsilon = distance between observations to be considered within the same cluster

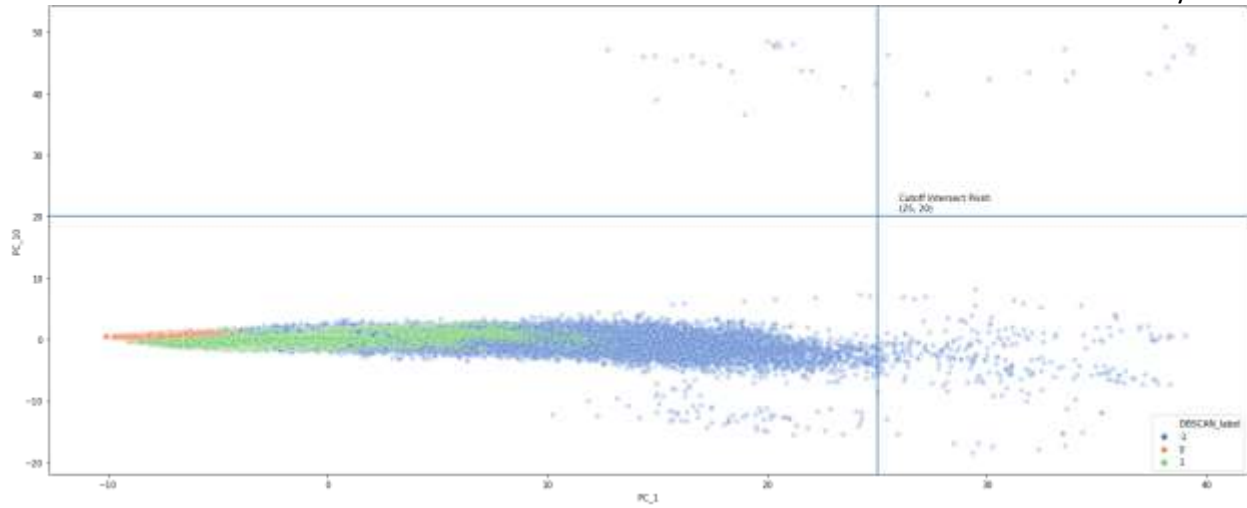


Figure 90: PC Plot of DBSCAN Clusters; filtered by tagged non-pothole events. Outliers (potential potholes) labelled as -1 in blue.

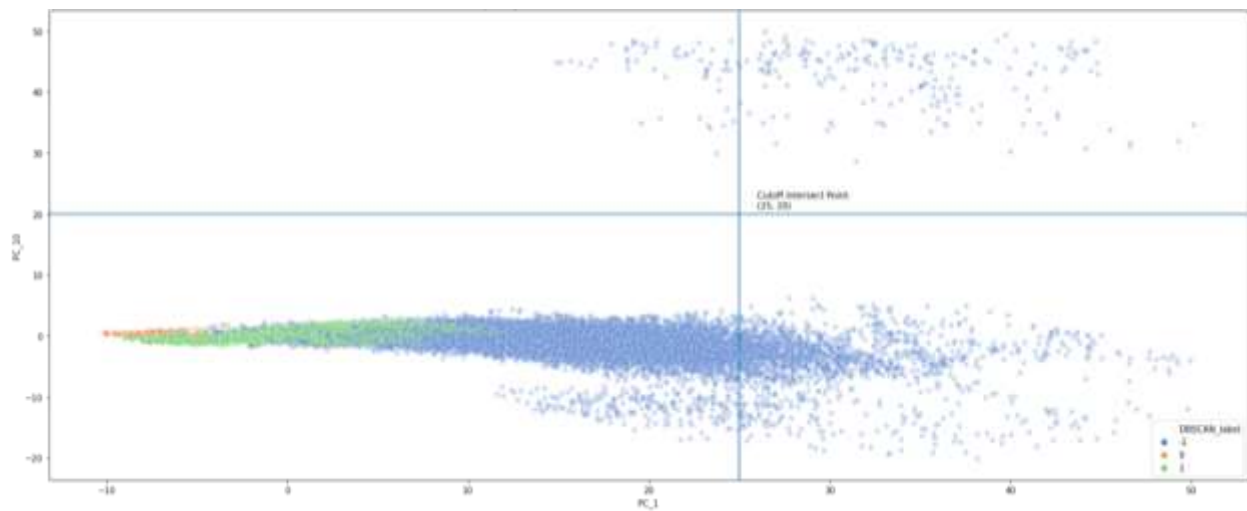


Figure 91: PC Plot of DBSCAN Clusters; filtered by tagged pothole events. Outliers (potential potholes) labelled as -1 in blue.

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

Like DBSCAN which it is derived from, poor discrimination of pothole events was observed, making it unsuitable for road anomaly detection. This is likely caused the insufficient spread of the true pothole observations from the non-pothole observations, as either an outlier or a separate cluster. This is partly due to various road anomalies producing similar signals to potholes, confusing the clustering algorithm.

Key Settings Used:

1. Input variables: PC 1-10
2. Minimum cluster size: 1000

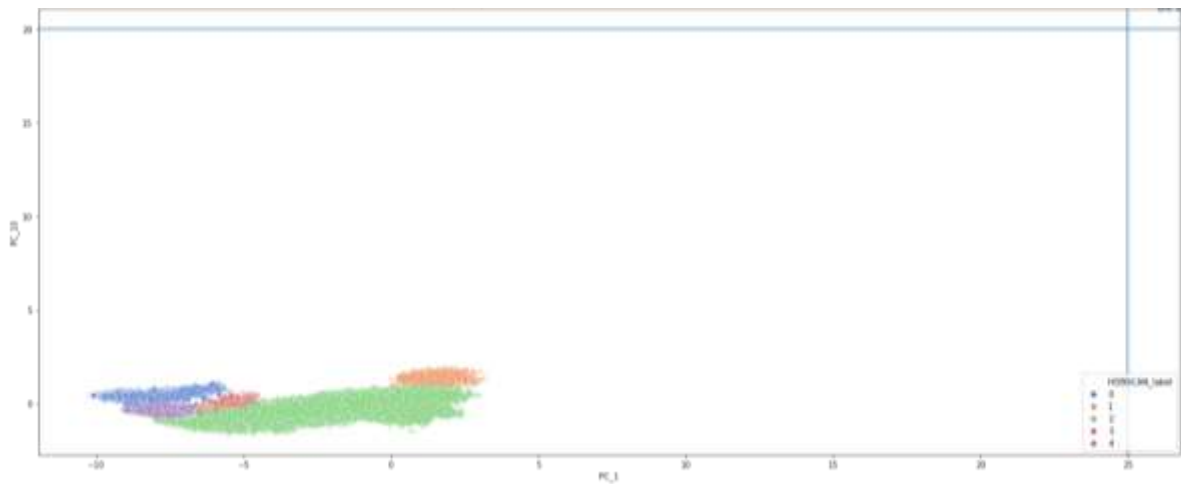


Figure 92: PC Plot of HDBSCAN Clusters; filtered by tagged non-pothole events. Outliers (potential potholes) labelled as -1 in blue.

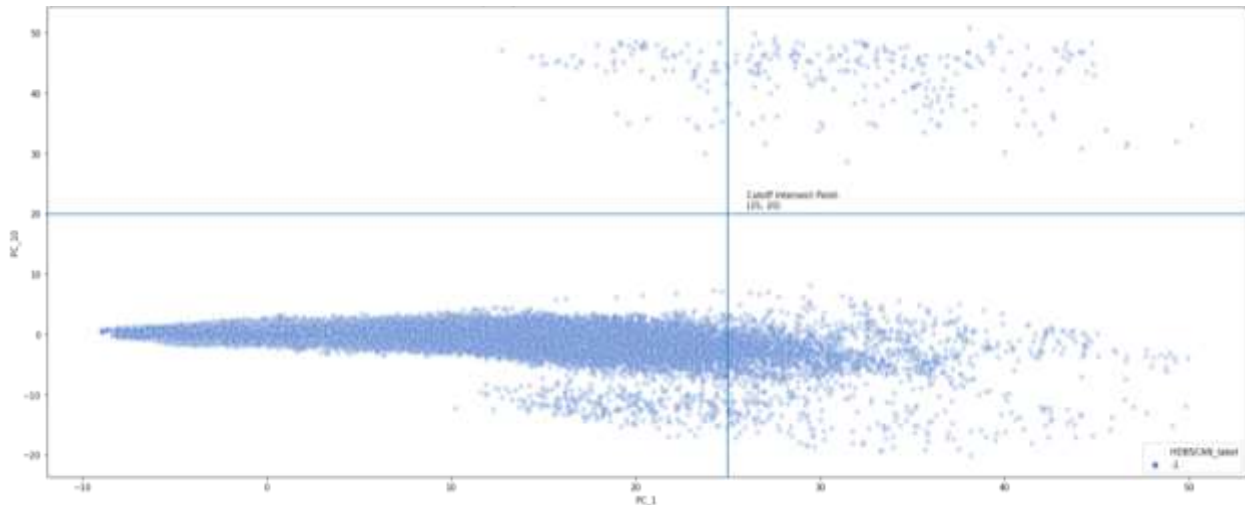


Figure 93: PC Plot of HDBSCAN Clusters; filtered by tagged pothole events. Outliers (potential potholes) labelled as -1 in blue.

Table of Figures

Figure 1: Transport Mode Classification	2
Figure 2: Classification of Different Road Surface Defects	4
Figure 3: Sensors details (from Smartphone-based Human Activity Recognition by Jorge Luis Reyes Ortiz)	7
Figure 4: Neutral orientation of a smartphone aligned to world coordinates. Positive and negative values indicate the direction of the acceleration (g-force) applied. For android acceleration and rotation is measured in m/s^2 and degrees respectively, but in iOS g-force and radians are used, respectively.....	8
Figure 5: Neutral orientation of the smartphone superimposed over an object it is placed on, aligned to world coordinates.	8
Figure 6: Pitch Rotation (on X-Axis) when encountering a surface bump or pothole due to the rocking action. The event would also cause a spike in the Z-axis acceleration due to the jerk caused. It should be noted that travelling uphill or downhill will cause a change in the Z-axis acceleration, however, such a change is expected to be much more gradual in comparison.	9
Figure 7: For non-single tracked vehicle, encountering a pothole or surface bump will result in a roll rotation of the Y-Axis. If only 1 tire is affected, a twisting action may be observed (rotation on the x and y axes). It should be noted that Y-Axis rotation does not apply to single-tracked vehicles like bicycles due to the wobbling and leaning action when balancing and turning.	9
Figure 8: Yaw rotation on the Z-axis. Turning action due to drivers avoiding major road anomalies and normal travel would be impossible to distinguish apart based on accelerometer and gyroscope readings alone. Hence acceleration on the X-Y axes and rotation on the Z-axis is unlikely to be useful for our analysis. It should be noted that changes in the horizontal speed of the vehicle will also cause spikes in the acceleration on the X and Y axes e.g. through braking or accelerating from a stop.....	10
Figure 9: The bobbing motion caused by walking and running would produce a sine wave of sorts oscillating at different frequencies. Identifying this signature would allow one to identify walk or running patterns.....	10
Figure 10: Without proper sensor reorientation, the sensor values will be off alignment on the true 3-axis state (as seen in Obj B), since readings are taken based on the phone's current orientation, thus making comparisons difficult across different recording events. If the phone is not placed in a locked position when the readings are taken, spin rotation may also be present on all 3-axis making standardized measurements even harder, especially with the introduction of angular acceleration.....	11
Figure 11: Amount of horizontal displacement is dependent on the velocity of the vehicle	11
Figure 12: The use of Euclidean Distance for distance estimation (C) tends to underestimate the velocity travelled especially if there are a lot of turns and if the sampling frequency is too small; where the actual distance covered is A + B.	12
Figure 13: Proposed data processing and modelling overview. This process is subject to change as the project evolves.	13
Figure 14: Original amplitude of accelerometer in the frequency domain	16
Figure 15: Filtered amplitude of accelerometer in the frequency domain	16
Figure 16: Sliding window example	17
Figure 17 Acceleration in Time Domain	17
Figure 18: Wavelet domain coefficients	19
Figure 19: Original wavelet domain coefficients	19
Figure 20: Wavelet domain features	19

Figure 21: Initial modelling results.....	20
Figure 22: Feature importance	21
Figure 23 Second Models' Results (Trained in 1 Hz data on window size 60, and overlap 1 step)	22
Figure 24 LightGBM Model's Confusion Matrix.....	22
Figure 25 Confusion Matrix on Unseen Data 1	23
Figure 26 Confusion Matrix on Unseen Data 2	23
Figure 27 Test 1: User 9 Car Data.....	24
Figure 28 Test 2: User 9 Car, Walk, Idle Data.....	24
Figure 29 Test 3: User 45 Walk, Idle Data	25
Figure 30 Candidate Parameter	25
Figure 31 Overlap 1 Step.....	26
Figure 32 Overlap 50%	26
Figure 33 Different Models' Results(Trained in 50 Hz Data on Different Window Sizes and Overlap Percents)	27
Figure 34 Neural Network Model's (in Window 2s and Overlap 90%) Classification Report	27
Figure 35 Random Forest model's (in Window 2s Overlap 70%) Classification Report.....	28
Figure 36 Velocity in X-Axis	29
Figure 37 Velocity in Y-Axis	30
Figure 38 Velocity in Z-Axis	30
Figure 39: Illustration of Dynamic Time Warping on audio clips for speech recognition.....	32
Figure 40: Illustration of Dynamic Time Warping on Train Acceleration Signals.....	33
Figure 41: illustration of Derivative Dynamic Time Warping on Train Acceleration signals.....	33
Figure 42: Multiple Boxplot Graph showing Journey Duration recorded over 6 different days	34
Figure 43: Human Noise Filter	35
Figure 44: Illustration of the effects of the human noise filter on acceleration signals.....	35
Figure 45: Higher-Level Features computations.....	35
Figure 46: Cleaned Signals from two journeys from Bedok North to Chinatown on DownTown Line	36
Figure 47: Cleaned Signals from two journeys from Chinatown to Little India	36
Figure 48: Different route comparison (station-to-station journeys).....	38
Figure 49: Same Station-to-Station Journeys Comparison – 1Hz Dataset.....	38
Figure 50: Different Station-to-Station Journeys Comparison – 1Hz Dataset	39
Figure 51: Email correspondence with one of the researchers.....	40
Figure 52: Same Station-to-Station Journeys Comparison – 50Hz Dataset.....	40
Figure 53: Different Station-to-Station Journeys Comparison – 50Hz Dataset	41
Figure 54: Same Station-to-Station Journeys Comparison – 10Hz Dataset.....	43
Figure 55: Different Station-to-Station Journeys Comparison – 10Hz Dataset	44
Figure 56: Sample images of potholes surveyed	48
Figure 57: Overview of data collection, processing, and analysis steps	49
Figure 58: Distribution of Data Collected by Event Tag	50
Figure 59: Sample raw data recording.....	51
Figure 60: Frequency band transition between the pass and stop bands of different IIR filter designs....	52
Figure 61: Butterworth Bandpass Filter in different order settings. Severe aliasing starts appearing at orders 10 and above.	52

Figure 62: Filtered signal (orange) and unfiltered data (blue) of linear acceleration (Z-Axis data), joint together as a continuous times series grouped by event. Knowing the approximate zones and their corresponding event tags would be useful in analyzing the continuous wavelet transform scalogram plots later, as one is better able to identify the key signals.	53
Figure 63: Mother wavelet used for Continuous Wavelet Transformation (CWT)	54
Figure 64: CWT Scalogram Plot for Horizontal Linear Acceleration (Filtered); all scales shown. Note that low energy events such as normal cycling have been filtered out at the lower scales.	55
Figure 65: CWT Scalogram Plot for Z-Axis Linear Acceleration (Filtered); first 10 scales shown (as the other scales were primarily low energy regions). Note that low energy events such as normal cycling have been filtered out.	56
Figure 66: CWT Scalogram Plot for X-Axis Gyroscope Rotation (Filtered); first 10 scales shown (as the other scales were primarily low energy regions). Note that low energy events such as normal cycling and speed strips have been filtered out.	56
Figure 67: Elbow plot for determining optimal K value for K-Means clustering	60
Figure 68: PC Plot of K-Means Cluster Labels, filtered by Tagged Non-Pothole Events. Cluster 3 (red) identified as a suspected pothole due to relative absence in the tagged non-pothole events plot compared to tagged pothole events.	60
Figure 69: PC Plot of K-Means Cluster Labels, filtered by Tagged Pothole Events. Cluster 3 (red) identified as a suspected pothole due to relative absence in the tagged non-pothole events plot compared to tagged pothole events.	61
Figure 70: Overview of potholes identified via HDBSCAN Clustering 1 plotted via Google Maps. Ground of Truth Points marked by images in circular frames; and suspected potholes are the coloured location markers.	62
Figure 71: Overview of potholes identified via HDBSCAN Clustering 2 plotted via Google Maps. The Ground of Truth Points marked by images in circular frames; and suspected potholes are the coloured location markers.	63
Figure 72: Brand Share by Sales Volume of Singapore Smartphone Market (2014-2019) from Euromonitor.....	I
Figure 73: Fourier transform on normal cycling event Z-axis linear acceleration signal	II
Figure 74: Fourier transform on road hump event Z-axis linear acceleration signal.....	II
Figure 75: Fourier transform on speed strips event Z-axis linear acceleration signal	III
Figure 76: Fourier transform on pothole event Z-axis linear acceleration signal.....	III
Figure 77: Fourier transform on normal cycling event on X-axis gyroscope rotation signal	IV
Figure 78: Fourier transform on road hump event on the X-axis gyroscope rotation signal	IV
Figure 79: Fourier transform on speed stripe event on the X-axis gyroscope rotation signal	V
Figure 80: Fourier transform on pothole event on X-axis gyroscope rotation signal	V
Figure 81: Filtered X-Axis Linear Acceleration	VI
Figure 82: Filtered Y-Axis Linear Acceleration	VI
Figure 83: Filtered Z-Axis Linear Acceleration	VII
Figure 84: Filtered X-Axis Gyroscope Rotation	VII
Figure 85: PC Plot of K-Means Cluster Labels (Train Data)	XXI
Figure 86: PC Plot of K-Means Cluster Labels (Test Data).....	XXI
Figure 87: PC Plot of LOF Clusters; filtered by tagged non-pothole events. Outliers (potential potholes) are tagged as 1 in orange.....	XXII

Figure 88: PC Plot of LOF Clusters; filtered by tagged pothole events. Outliers (potential potholes) are tagged as 1 in orange.	XXII
Figure 89: Elbow plot to determine optimum epsilon value. The X-axis refers to the number of observations; Y-Axis refers to the epsilon value. The epsilon value is taken based on the inflexion point which is around a value of 2.	XXIII
Figure 90: PC Plot of DBSCAN Clusters; filtered by tagged non-pothole events. Outliers (potential potholes) labelled as -1 in blue.	XXIV
Figure 91: PC Plot of DBSCAN Clusters; filtered by tagged pothole events. Outliers (potential potholes) labelled as -1 in blue.	XXIV
Figure 92: PC Plot of HDBSCAN Clusters; filtered by tagged non-pothole events. Outliers (potential potholes) labelled as -1 in blue.	XXV
Figure 93: PC Plot of HDBSCAN Clusters; filtered by tagged pothole events. Outliers (potential potholes) labelled as -1 in blue.	XXV

References

- Astarita, V., Caruso, M. V., Danieli, G., Festa, D. C., Giofrè, V. P., Iuele, T., & Vaiana, R. (2012). A mobile application for road surface quality control: UNIQALroad. *Procedia - Social and Behavioral Sciences*, 1135 – 1144. doi:<https://doi.org/10.1016/j.sbspro.2012.09.828>
- CEA. (30 Oct, 2015). *Street Bump: Crowdsourcing Better Streets, but Many Roadblocks Remain*. Retrieved 01 Mar, 2020, from Harvard Business School - Digital Innovation and Transformation: <https://digital.hbs.edu/platform-digit/submission/street-bump-crowdsourcing-better-streets-but-many-roadblocks-remain/>
- Claudia Carpineti, V. L. (2018, March). Custom Dual Transportation Mode Detection By Smartphone Devices Exploiting Sensor Diversity. *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*.
- Databricks. (n.d.). Dynamic Time Warping Background. Retrieved from <https://pages.databricks.com/rs/094-YMS-629/images/dynamic-time-warping-background.html>
- Davies, A. (30 Jan, 2019). *A Windshield-Mounted Cure for the Common Pothole*. Retrieved 01 Mar, 2020, from Wired: <https://www.wired.com/story/carvi-pothole-detection/>
- Device Atlas. (17 Sep, 2019). *The most popular smartphones in 2019*. Retrieved 01 Mar, 2020, from Device Atlas: <https://deviceatlas.com/blog/most-popular-smartphones#sg>
- Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., & Balakrishnan, H. (2008). The pothole patrol: using a mobile sensor network for road surface monitoring. *Proceedings of the 6th international conference on Mobile systems, applications, and services*, 29-39.
- Euromonitor International. (04 Feb, 2020). *Brand share of Smartphone Market in Singapore (2014-2019)*. Retrieved from Passport GMID: <https://go.euromonitor.com/passport.html>
- Google. (27 Dec, 2019). *Sensors Overview*. Retrieved 01 Mar, 2020, from Android Developers: https://developer.android.com/guide/topics/sensors/sensors_overview
- Guyen Asci, M. A. (11-15 March, 2019). A Novel Input Set for LSTM-based Transport Mode Detection. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*.
- Highways Department (Hong Kong). (Jan, 2013). *Guidance Notes (RD/GN/015B - Catalogue of Road Defects)*. Retrieved from Highways Department (Hong Kong): https://www.hyd.gov.hk/en/publications_and_publicity/publications/technical_document/guidance_notes/index.html
- Khuong An Nguyen, Y. W. (01 09, 2019). Realtime Tracking of Passengers on the London. *Sensors (Basel, Switzerland)*, 4184. doi:10.3390/s19194184
- LePage, P. (12 Feb, 2020). *Device Orientation & Motion*. Retrieved 01 Mar, 2020, from Google Developers: <https://developers.google.com/web/fundamentals/native-hardware/device-orientation>

- Li, X., Huo, D., Goldberg, D. W., Chu, T., Yin, Z., & Hammond, T. (2019). Embracing Crowdsensing: An Enhanced Mobile Sensing Solution for Road Anomaly Detection. *International Journal of Geo-Information*. doi:<https://doi.org/10.3390/ijgi8090412>
- Luca Bedogni, M. D. (19 July, 2016). Context-aware Android applications through transportation mode detection techniques. *Wiley Online Library*.
- Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G., & Selavo, L. (2011). Real Time Pothole Detection Using Android Smartphones with Accelerometers. *IEEE*, 1-6. doi:10.1109/DCOSS.2011.5982206
- Meng-Chieh Yu, T. Y.-C.-J. (2014). Big data small footprint: the design of a low-power classifier for detecting transportation modes.
- Rishiwal, V., & Khan, H. (30 May, 2016). Automatic pothole and speed breaker detection using android system. *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1270-1273. doi:10.1109/MIPRO.2016.7522334
- Sattar, S., Li, S., & Chapman, M. (2018). Road Surface Monitoring Using Smartphone Sensors: A Review. *PubMed Central*. doi:10.3390/s18113845
- Simon, P. (Mar, 2014). *Potholes and Big Data: Crowdsourcing Our Way to Better Government*. Retrieved 01 Mar, 2020, from Wired: <https://www.wired.com/insights/2014/03/potholes-big-data-crowdsourcing-way-better-government/>
- Wei, L., Fwa, T. F., & Zhe, Z. (01 Feb, 2005). Wavelet Analysis and Interpretation of Road Roughness. *Journal of Transportation Engineering*, 120-130. doi:[https://doi.org/10.1061/\(ASCE\)0733-947X\(2005\)131:2\(120\)](https://doi.org/10.1061/(ASCE)0733-947X(2005)131:2(120))