Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Classification of driving behaviour using a smartphone

Project Studies on Contemporary Topics

Name of the Study Programme

The Master's Programme Professional IT Business and Digitalization

from

Yash Revannavar

Nikhil Amin

Sevinj Abdullazade

Date:

Berlin, 24.04.2023

Supervisor: Prof. Frank Fuchs-Kittowski

1. Introduction

As a group of students, our part of the problem in this project is to investigate how the driving behaviour can be classified based on the data collected from the sensors within a smartphone. We classify driving behaviour into 3 categories such as calm, normal, and aggressive behaviour. Specifically, we aim to explore the relationship between differences in velocity, time, and accelerometer data with respect to the aforementioned driving behaviours. To do so, we tried and developed various data analysis methods on the collected data. From the data, we tried to identify specific driving manoeuvres, such as strong or weak acceleration, braking, and steering, to differentiate between the three types of driving behaviours.

Drivers differ in the way they steer, accelerate, brake, shift gears, how much distance they keep to the vehicle in front and whether they keep to the speed limits. (Mantouka et al. 2020). By identifying the differences in the behaviours between calm, normal, and aggressive driving, we can develop interventions and strategies in the future to promote safer driving practices and reduce the incidence of road accidents. In addition to the road conditions, the kind of driving behaviour also has an influence on the quality of the transported goods, especially which are highly sensitive. Hence, it would help transport such goods in a safe and reliable manner.

We are excited to be participating in this project and eager to explore the scientific question at hand. We believe that delving into this topic with the help of Machine Learning will not only deepen our understanding of the subject matter, but also allow us to engage with important and relevant research in our field. By working together and sharing our knowledge and expertise, we hope to make significant progress towards answering this question and contributing to the advancement of our discipline.

1.1 Structure of Paper

In this paper, we present a study on the detection of driving behaviour using machine learning. The paper is structured as follows.

1. Introduction:
   We provide an overview of the problem of detecting driving behaviour and its importance for road safety. We also outline the contributions and organisation of the paper.
   1.1 Structure of Paper:
      We provide a brief summary of the sections of the paper.

2. Detection of Driving Behavior:
   We review related work in the literature on the detection of driving behaviour using Machine Learning. We discuss the methods and techniques used by other researchers, and identify the strengths and weaknesses of these approaches.

3. Methodology:
   We describe the approach we took to detect driving behaviour, including the data collection and preprocessing, feature extraction, algorithm selection, and model evaluation. We also explain the statistical analysis and visualisation techniques used to gain insights into the data.

4. Implementation:
   We provide details on the implementation of the methodology, including the programming languages, libraries, and tools used. We also discuss the availability of the code and data.

5. Results:
   We present the results of our study, including the accuracy and other performance metrics of the algorithms tested, as well as the interpretation of the features and their importance for the task.

   5.1 Charts:
   We include a set of charts to illustrate the data and the results, including scatter plots, bar graphs, and box plots.

   5.2 Conclusion:
   We summarise the main findings of the study and draw conclusions about the effectiveness of the approach. We also discuss the limitations and future directions of the research.

   5.3 Further work:
   We suggest some avenues for future work, including the use of more advanced machine learning techniques, the extension of the study to other contexts and datasets, and the exploration of real-time detection of driving behaviour using sensors and other devices.

2. Detection of Driving Behaviour

This subsection is dedicated to reviewing various research papers aimed at identifying driving behaviour using different algorithms and then in the later part, we discuss the Machine Learning algorithms that we used in this project. We reviewed the following three scientific papers.

1. Smartphone sensing for understanding driving behaviour: Current practice and challenges (Mantouka et al. 2020)

2. Driving Behavior Classification and Sharing System Using CNN-LSTM Approaches and V2X Communication (Kwon et al. 2021)
3. Driver Action Prediction Using Deep (Bidirectional) Recurrent Neural Network (Olabiyi et al. 2017)

The research and study of the three scientific papers have provided insights into the relevant implementation decisions for driving behaviour analysis. These papers emphasise the importance of selecting appropriate sensors, communication infrastructure, and machine learning algorithms for accurate and reliable driving behaviour analysis. Based on the review of three scientific papers related to driving behaviour, the following implementation decisions were found to be relevant.

- The first paper (Mantouka et al. 2020) highlighted the challenges of smartphone sensing for understanding driving behaviour and emphasised the need for more accurate and reliable sensor data. The paper establishes an extensive step-by-step framework to define the process from data collection to informed decision-making. The researchers then discuss various other relevant papers and the steps used in those to identify driving behaviours. They discuss the use of Statistical methods in brief. Next, they go on to discuss Machine Learning (ML) techniques used in various other relevant research papers, few noteworthy are listed below:-
    - (Hong, Margines, and Dey 2014) mentions Naïve Bayes classifier to identify aggressive driving styles with an accuracy of 90%.
    - (Fazeen et al. 2012) mentions the three-axis accelerometer to analyze driving behaviour and detect road anomalies.
    - (Saiprasert, Pholprasit, and Thajchayapong 2015) mentions fuzzy logic and a rule-based algorithm to detect driving behaviour.
    - In (Koh and Kang 2015) researchers used Gaussian Mixture Model (GMM) with periodogram method.
    - In (Vlahogianni and Barmpounakis 2017) the MODLEM algorithm achieved the maximum accuracy compared to other classification algorithms.

- The second paper (Kwon et al. 2021) proposed a CNN-LSTM approach combined with V2X communication to classify and share driving behaviour, which requires a well-designed communication infrastructure.

- The third paper (Olabiyi et al. 2017) focused on predicting driver action behaviour using a timeseries anomaly prediction problem. The data collection was done using camera-based knowledge of the driving environment and the driver themselves, along with the usual vehicle dynamics. It then used a deep bidirectional recurrent neural network (DBRNN) which is a key component in this paper. DBRNN is used to learn the correlation between sensory inputs and impending driver behaviour to achieve precise prediction.

In our project, we have used three machine learning algorithms for classifying the driving behaviour: Support Vector Classification (SVC), Decision Tree, and Random Forest. These algorithms were implemented using Python's Scikit-learn library.

Support Vector Classification is a supervised learning algorithm that separates data points using a hyperplane. The algorithm aims to find the best hyperplane that divides the data points into their respective classes. (“sklearn documentation”, n.d.). To comprehend in detail we read the paper “Support Vector Machines for Classification and Regression” (Rice and Galbraith 2008). Here SVMs are gaining popularity due to many attractive features and promising empirical performance. The formulation embodies the Structural Risk Minimization (SRM) principle, which has been shown to be superior to traditional Empirical Risk Minimization (ERM) principle employed by conventional neural networks. SRM minimises an upper bound on the expected risk, as opposed to ERM that minimises the error on the training data. It is this difference which equips SVM with a greater ability to generalise, which is the goal in statistical learning. The report then considers the problem of regression and provides illustrative examples to show the properties of the techniques. (Rice and Galbraith 2008)

Decision Tree is also a supervised learning algorithm that constructs a tree-like structure to represent decisions and their possible consequences. The tree is constructed by recursively splitting the data based on the most important feature until a stopping criterion is reached. (“sklearn documentation”, n.d.). For further reference, we studied “An introduction to decision tree modeling.” (Myles and Feudale, n.d.). It emphasises techniques to introduce decision tree modelling for classification with emphasis on adaptations to the traditional decision tree design that make the methodology more useful to chemical and biochemical applications. The paper discusses traditional decision tree implementation, graphical representation and terminology, and alternative uses for decision tree modelling. The paper also discusses decision tree modelling for pattern recognition, classification, and ensemble modelling. (Myles and Feudale, n.d.)

Random Forest is an ensemble learning algorithm that utilises multiple Decision Trees to make predictions. Each Decision Tree in the forest is trained on a different subset of the data, and the final prediction is the majority vote of all trees. (“sklearn documentation”, n.d.)
We dig down to better understand the Random Forest by reading the paper “An assessment of the effectiveness of a random forest classifier for land-cover classification” (Rice and Galbraith 2008) which evaluates the performance of Random Forest (RF) classifier for land cover classification of a complex area. The evaluation was based on several criteria: mapping accuracy, sensitivity to data set size and noise. Results show that the Random Forest algorithm yields accurate land cover classifications, with 92% overall accuracy and a Kappa index of 0.92. Random Forest is robust to training data reduction and noise because significant differences in kappa values were only observed for data reduction and noise addition values greater than 50 and 20%, respectively. Additionally, variables that Random Forest identified as most important for classifying land cover coincided with expectations. A McNemar test indicates an overall better performance of the Random Forest model over a single Decision Tree at the 0.00001 significance level. (Rice and Galbraith 2008)

Random Forest has several advantages when it comes to the Driving Behaviour project. (Rice and Galbraith 2008) First, it requires a smaller amount of data and computational resources for training as compared to the Neural Network. In most of the papers on this topic, Neural Networks are commonly used, which require more data and computational resources. (Kwon et al. 2021) Second, Random Forest is relatively easy to use and can handle a large number of input features without overfitting.

Additionally, Random Forest can handle missing data and outliers. Missing data is a common issue in any dataset and being able to handle this is crucial for obtaining accurate results. Outliers are also common in the Driving Behaviour dataset, and Random Forest's ability to handle these makes it a suitable algorithm for this project.

However, Random Forest also has some disadvantages when it comes to the Driving Behaviour project. One disadvantage is that this algorithm is trained only on the data set that is collected by driving on the same path or the road. This means that the algorithm may not generalise well to different roads or paths. (Revannavar 2023) Another disadvantage is that Random Forest may not perform well with highly imbalanced datasets.

In summary, the advantages of Random Forest such as its ability to handle a large number of input features without overfitting, handling missing data and outliers, and its relative ease of use make it a suitable algorithm for the Driving Behaviour project. However, its limitations such as being trained only on the same path or road and its performance with highly imbalanced datasets should also be considered.

3. Methodology

The driving behaviour project involves the use of machine learning algorithms, specifically Support Vector Classification (SVC), Decision Tree, and Random Forest. The data set used for this project was obtained from HTW Berlin University of Applied Science and contained 11 calm, 10 normal, and 10 aggressive driving behaviour samples in CSV format. The dataset consists of columns including ‘line number’, ‘accel x’, ‘accel y’, ‘accel z’, ‘velocity’, and ‘real-time’, from which we created a new column using the Euclidean distance formula to compare all the accel x, y, and z data. The architecture/structure of our solution involves the training of these algorithms on the dataset and then testing their accuracy in predicting driving behaviour.

In this study, we utilised the csv data collected from calm, normal, and aggressive driving behaviour to analyse the acceleration of the x, y, and z axes. However, after observing that there is a significant amount of noise in the accel z data, we decided to exclude this feature from our analysis. To gain a better understanding, we created several charts that helped us explore the behaviour. However, these charts were not sufficient to provide us with an accurate understanding of the data.To further refine our analysis, we created a new column using the Euclidean distance formula on accel x, accel y, and accel z. This helped us gain a more precise understanding of the data. We also created charts to perform a statistical analysis of the accel x, accel y, accel z, and velocity data. By calculating basic statistics such as mean, median, standard deviation, maximum, and minimum, we gained a significant understanding of the features.

Furthermore, we observed that accel_y is the most relevant and important feature for our analysis. Finally, to gain a better understanding of the data, we plotted box plots of only the first acceleration, second breaking, and u-turn. We selected these variants with the help of the existing column labelled "label." We created arrays of box plots for each accel x, accel y, accel z, and velocity, and colour-coded them with blue for calm, green for normal, and red for aggressive driving behaviour.(refer section 5)

As we began working on the driving behaviour project, one of the critical decisions was selecting the appropriate algorithm. We conducted extensive research to understand the pros and cons of various algorithms that could potentially be used. After considering several options, we finally decided to use Support Vector Classifier (SVC), decision tree, and random forest for our analysis.(refer section 2)

4. Implementation

To ensure reproducibility, all the code used in this study is available on an open-source GitHub repository. The repository includes a detailed readme.md file with installation instructions, requirements, and guidelines for contributing to the project.

The code is organised into two Jupyter notebooks in the scripts directory: dataAnalysis.ipynb and Modeling.ipynb. The dataAnalysis.ipynb notebook contains all the code for the data preprocessing and exploratory data analysis discussed in this paper, including the creation of charts and graphs. The Modeling.ipynb notebook contains the code for the implementation of the chosen algorithms, namely SVC, decision tree, and random forest. The notebook also includes the corresponding scores and evaluations for each algorithm.

To ensure efficient and organised code, we utilised popular Python libraries such as Pandas, Numpy, Matplotlib, Scikit-learn, and Seaborn. These libraries provide easy-to-use and robust tools for data analysis, modelling, and visualisation.

- Python 3.10.6v
- Scikit-learn 1.2.2v

We used a 64-bit machine running Ubuntu 22.04.2 LTS x86_64 with an I11th Gen Intel i5-1135G7 (8) @ 4.200GHz CPU and 16GB of RAM for running the code.

5. Results

5.1 Charts

The first chart, which plots the values of Accel x, Accel y, and Accel z with respect to the index, provides insights into the magnitude and direction of the Accel values recorded during the driving activity. These Accel values can be indicators of driving behaviour. For example, sudden changes in Accel values in the x, y, or z direction can indicate the gyroscopic changes of the car, while high Accel values in the y direction could indicate swerving or abrupt U-turn. By plotting these values for calm, normal, and aggressive driving, we can compare and analyse the Accel patterns for different driving behaviours.
The values of Accel x, Accel y, and Accel z with respect to the index. The three axes are represented with different colours (blue for Accel x, green for Accel y, and red for Accel z).

The purpose of this chart is to visualise the data and identify any patterns or trends in the acceleration values for the different driving behaviours (calm, normal, and aggressive). The y-axis range is set from +8 to -8 for all three axes.

Figure 1: Accel x, Accel y, and Accel z with respect to the index

The second chart, which plots the values of Accel x and Accel y with respect to the index, was created after determining that the values of Accel z were irrelevant due to the high level of noise in that dimension (as per figure 1). By analysing the Accel patterns in the x and y dimensions, we can identify patterns that may be associated with different driving behaviours. For example, high Accel values in both x and y dimensions could indicate aggressive driving, while lower values could indicate calm or normal driving. The values of Accel x and Accel y with respect to the index. The three axes are represented with different colours (blue for Accel x and green for Accel y).

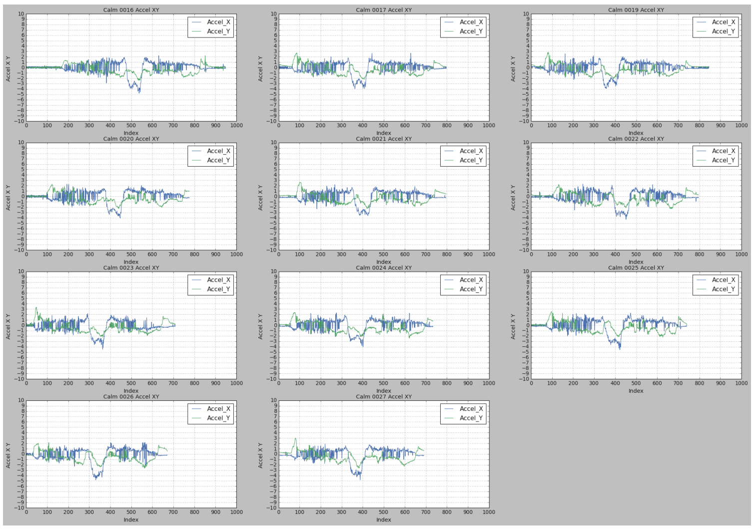

Figure 2: Accel x and Accel y with respect to the index for all calm dataset.

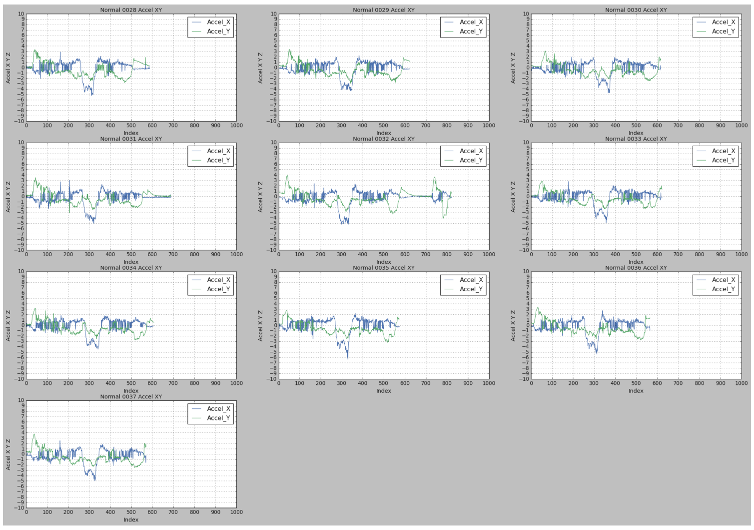

Figure 3: Accel x and Accel y with respect to the index for all normal dataset.

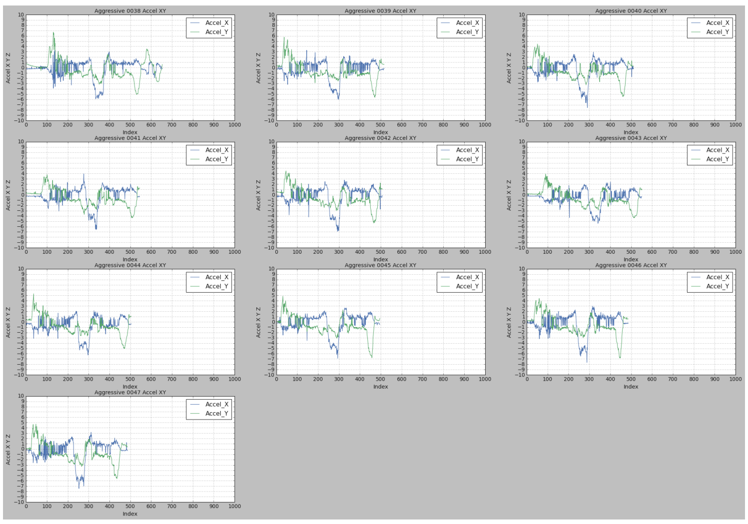

Figure 4: Accel x and Accel y with respect to the index for all aggressive dataset.

The third chart which plots the values of Euclidean distance calculated from Accel x, Accel y, and Accel z with respect to the index, provides a more comprehensive analysis of the acceleration patterns during the driving activity. The Euclidean distance represents the magnitude of the acceleration values in all three dimensions and can help identify patterns that may be associated with different driving behaviours. By plotting this distance for calm, normal, and aggressive driving, we can compare and analyse the acceleration patterns across all three dimensions for different driving behaviours. The chart plots the Euclidean distance values against the index and helps in understanding the overall acceleration patterns for the different driving behaviours. The y-axis range is set from 0 to 35.

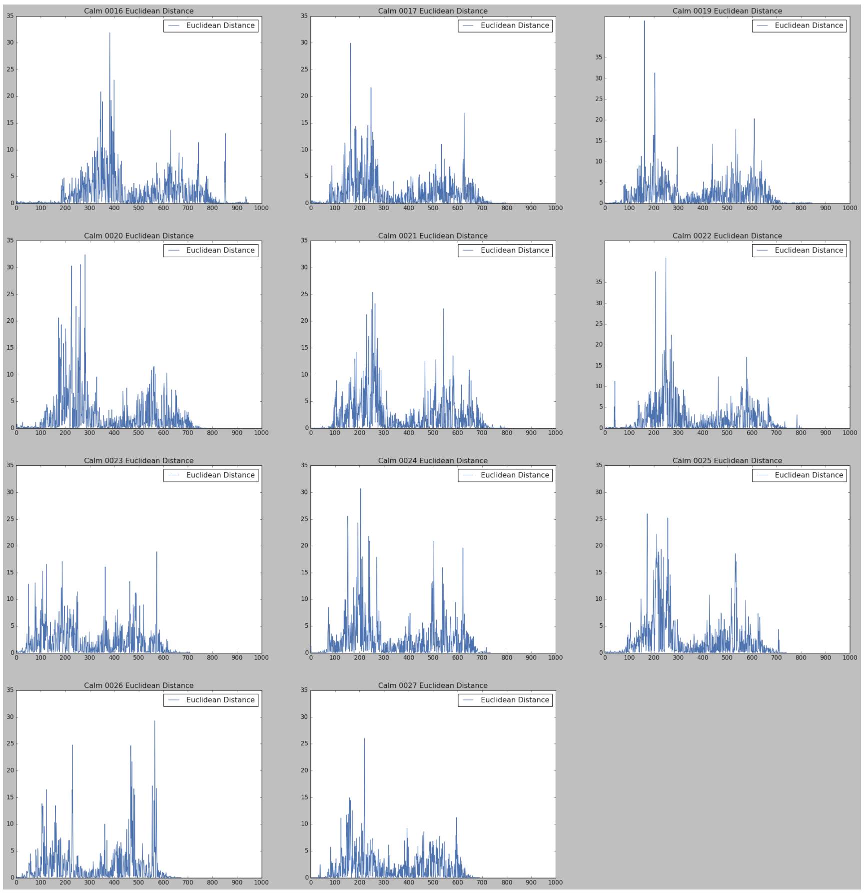

Figure 5: euclidean distance with respect to the index for all calm dataset.

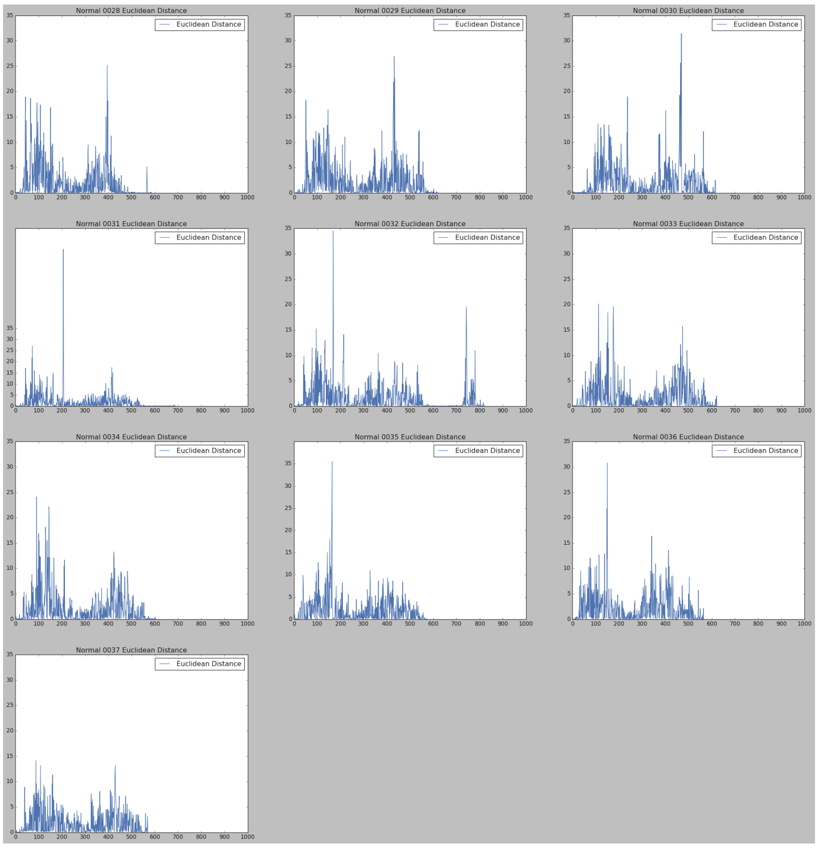

Figure 6: euclidean distance with respect to the index for all normal dataset.

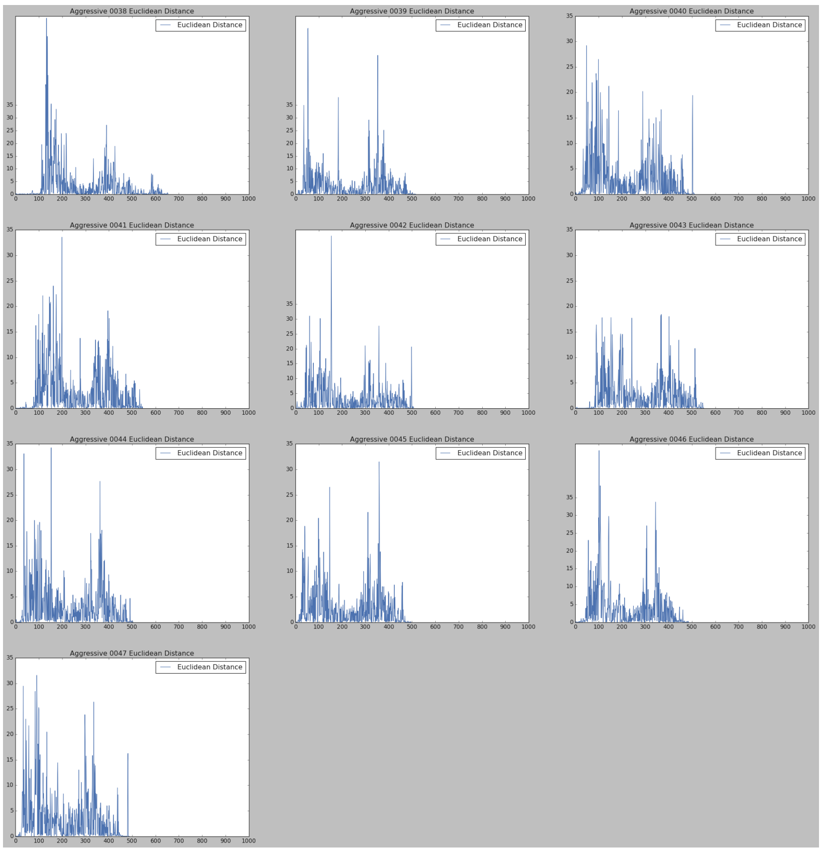

Figure 7: euclidean distance with respect to the index for all aggressive dataset.

In our study, we have developed a function to generate statistical data for the accelerometer and velocity data collected from drivers. The function, named "statsDf", takes in a pandas dataframe as input and returns an array of statistical data that includes mean, median, maximum, minimum, standard deviation, and count for the Accel_X, Accel_Y, Accel_Z, and Velocity parameters.

The "statsDf" function is crucial for understanding the driving behaviour of the participants in our study. The accelerometer data provides insights into the level of acceleration or deceleration applied by the driver, which can be indicative of aggressive driving behaviour. The velocity data provides insights into the speed of the vehicle, which is an essential parameter for evaluating driving behaviour.

By calculating the statistical data for these parameters, we can identify patterns and trends in driving behaviour. For example, if we observe a high standard deviation in the Accel_Y parameter for a particular driver, it may suggest that the driver is prone to sudden or erratic movements while driving, indicating a higher likelihood of aggressive driving behaviour. This function plays a critical role in our analysis of driving behaviour and provides a comprehensive understanding of the patterns and trends observed in our data.

The statsDf function presented here serves as a powerful tool for analysing driving behaviour data. This function calculates and returns the mean, median, maximum, minimum, standard deviation, and count for the accelerometer data in the x, y, and z directions, as well as the velocity data. To visualise the statistical data obtained from the statsDf function, we generated bar graphs comparing the values for calm, normal, and aggressive driving behaviours. The x-axis of these graphs shows the different statistics, including mean, median, maximum, and minimum values for Accel x, Accel y, Accel z, and velocity. The bar graphs generated from this data show distinct differences between the three driving styles for Accel_X, Accel_Y, and Velocity. Specifically, the mean and maximum values for Accel_X and Accel_Y are significantly higher in the aggressive driving style than in the calm and normal driving styles. This suggests that aggressive drivers tend to accelerate more rapidly and frequently than calm and normal drivers. In contrast, the velocity data shows the highest values in the normal driving style, indicating that normal drivers tend to maintain higher average speeds than calm and aggressive drivers.

However, it is important to note that the Accel_Z data did not show any meaningful differences between the three driving styles. This could suggest that vertical acceleration does not play a significant role in distinguishing driving behaviour between these three styles.

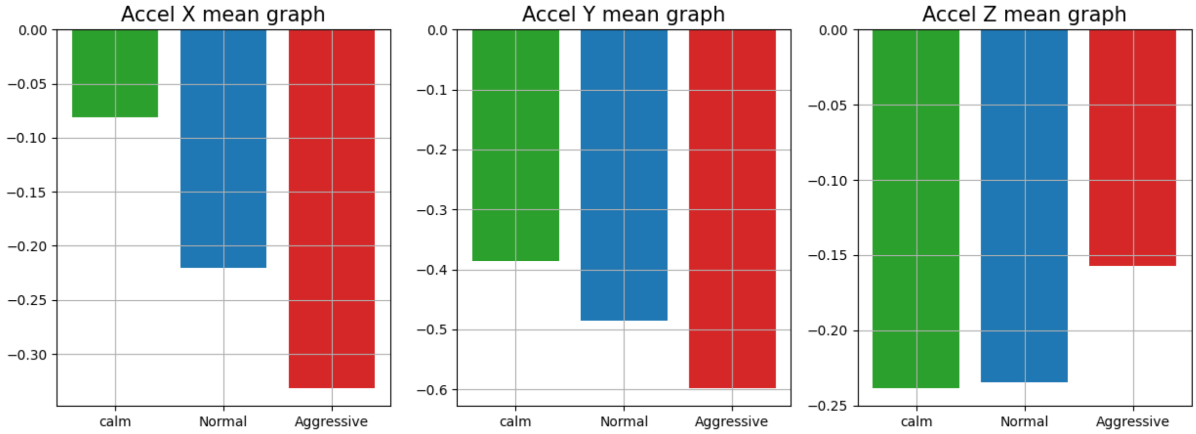

Figure 8: Comparing Accel X, Y and Z means.

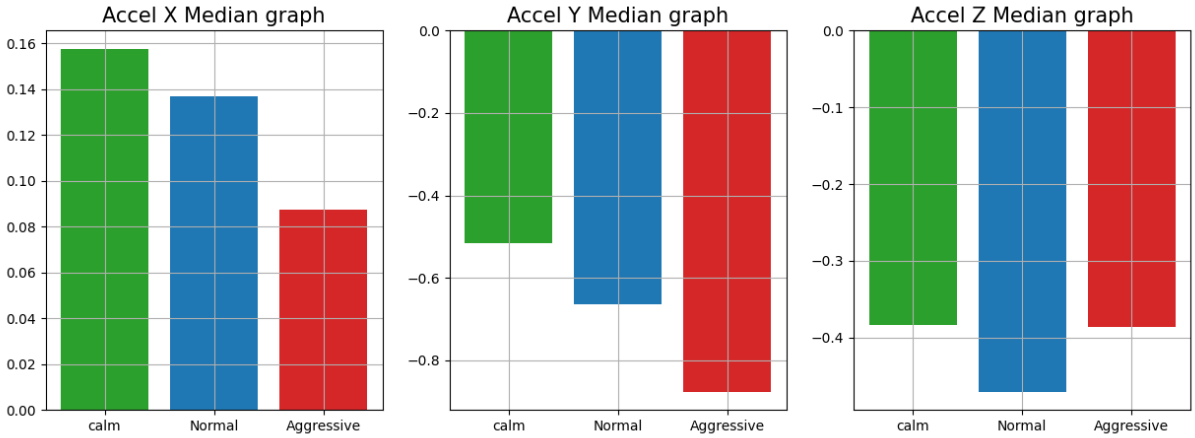

Figure 9: Comparing Accel X, Y and Z medians.

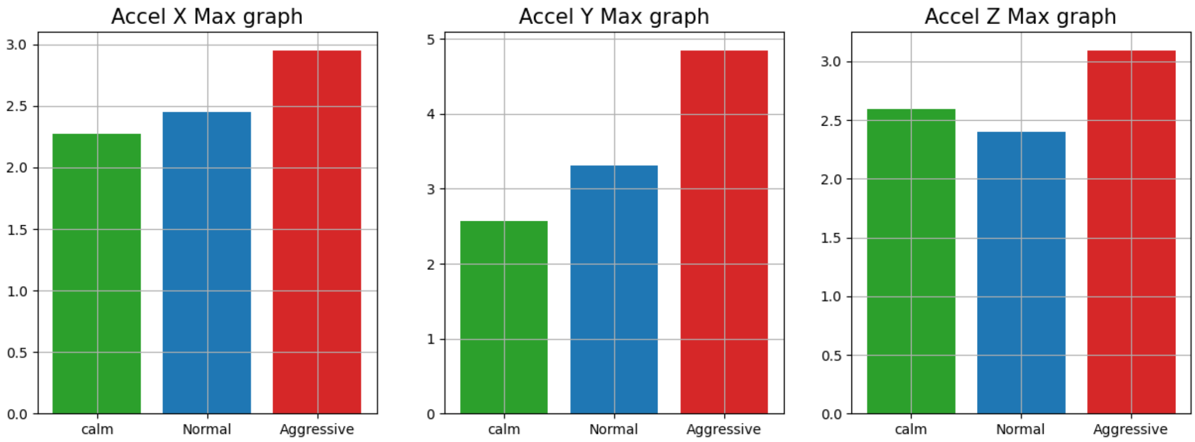

Figure 10 : Comparing Accel X, Y and Z Max.

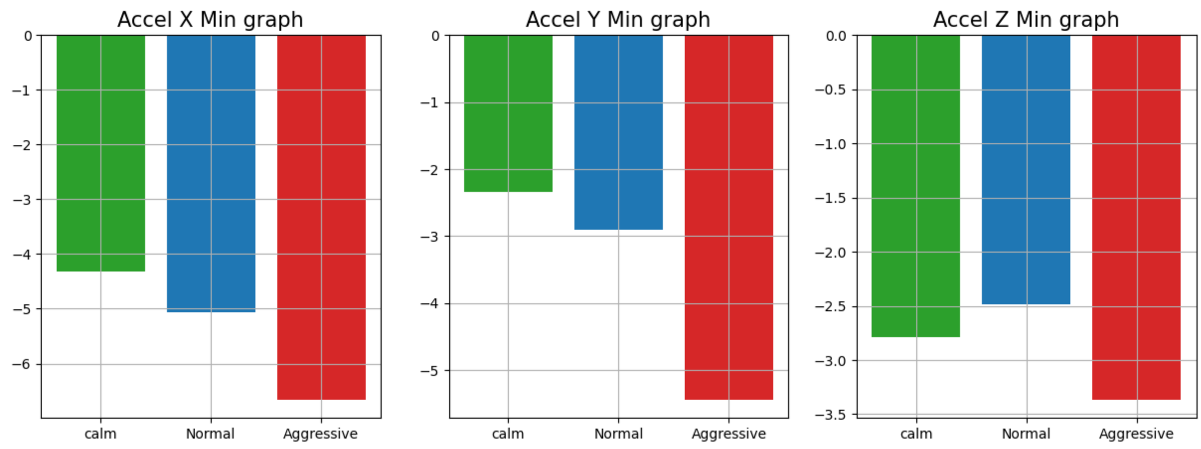

Figure 11: Comparing Accel X, Y and Z Min.

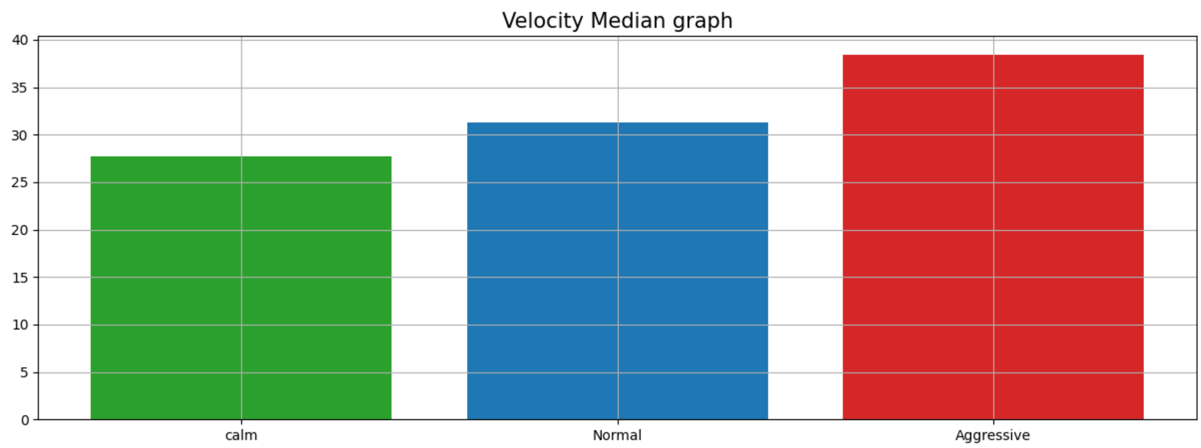

Figure 12: Comparing Accel X, Y and Z velocity.

Box plots are an important tool for visualising the distribution of data. In our driving behaviour project, we plotted four arrays of box plots to better understand the characteristics of our datasets. The first three arrays correspond to the three dimensions of the accelerometer data: accel_x, accel_y, and accel_z. The fourth array represents the velocity data.

From these box plots, we observed that there are no significant differences in the distributions of the First acceleration accel_x, First acceleration accel_z, and First acceleration velocity arrays across all variants of datasets. However, we found a notable difference in the First acceleration accel_y array. The interquartile range of the calm dataset is the smallest, while the normal dataset has a medium-sized interquartile range. On the other hand, the aggressive dataset has the largest interquartile range.

Figure 13: Box plot for First acceleration

In the second box plot, which also includes the second breaking and U-turn data, we observe similar characteristics as in the previous box plot.

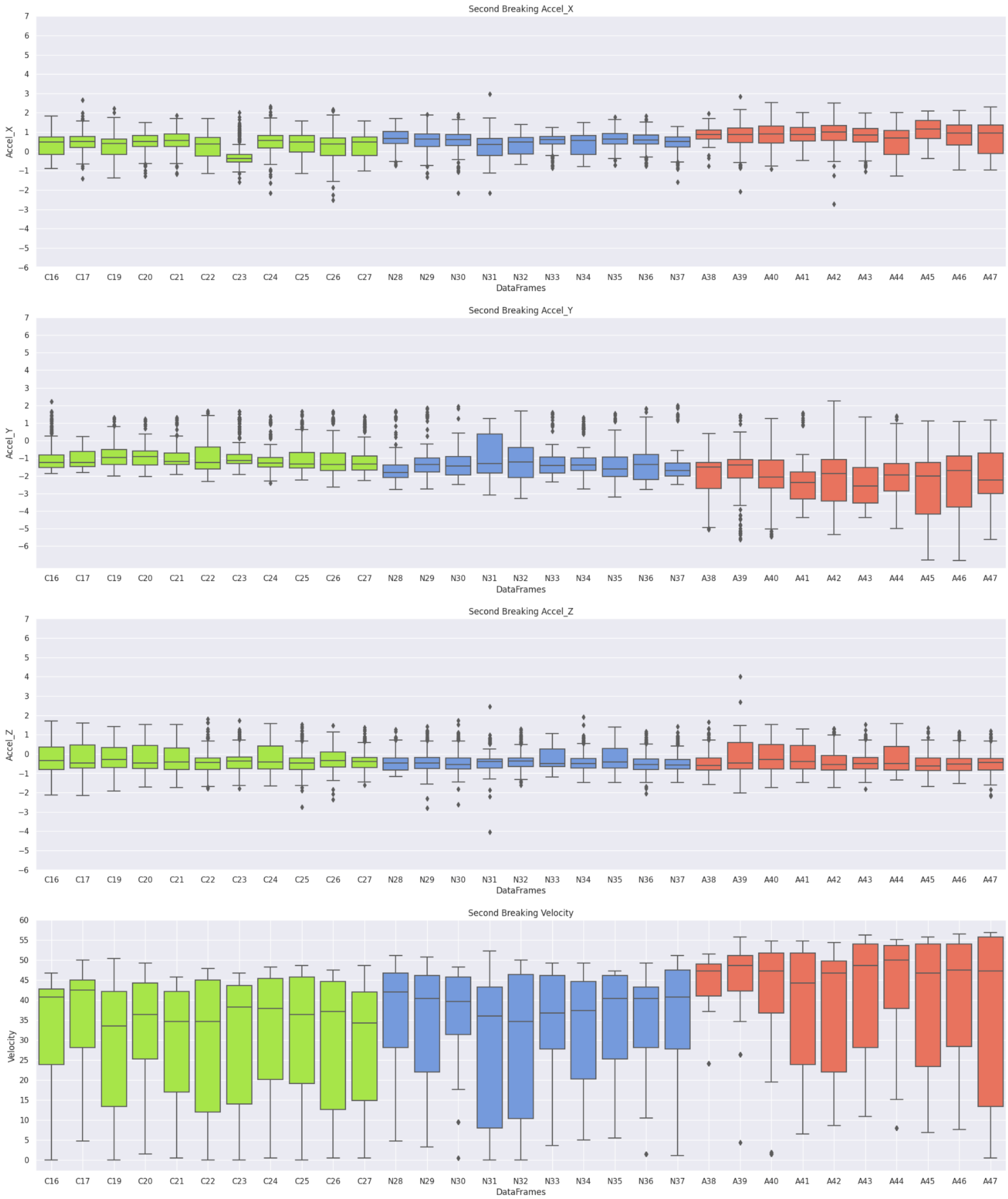

Figure 14: Box plot for Second Breaking

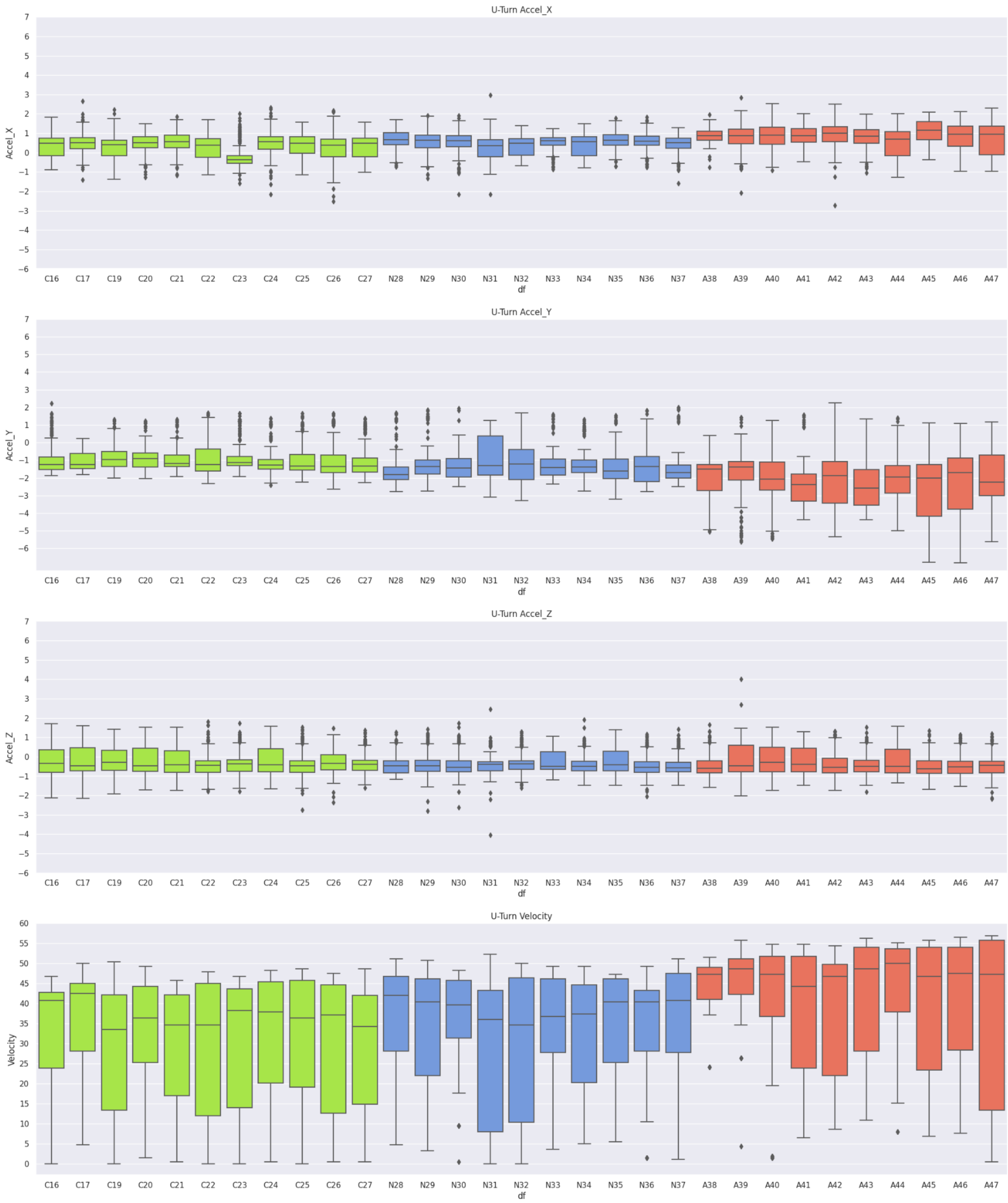

Figure 15: Box plot for U-turn acceleration

5.2 Conclusion

Overall, these box plots provide valuable insights into the distributions of our datasets, particularly in identifying differences between the three driving behaviour categories. By examining the box plots and statistical bar graphs in our analysis of the driving behaviour project, we have found that accel_y can be the most important feature for machine learning.
These findings suggest that our machine learning algorithm can differentiate between calm, normal, and aggressive driving behaviours based on accel data in the y direction.

The results show that the Decision Tree and Random Forest algorithms outperformed the Support Vector Classifier (SVC) in all metrics. Specifically, the Decision Tree and Random Forest models achieved higher accuracy, precision, recall, and F1 scores compared to the SVC model.

The Cross Validation Scores further confirm the superiority of Decision Tree and Random Forest models, as both models achieved significantly higher scores compared to the SVC model. The Cross Validation Scores are as follows:

- SVC: 0.4298
- Decision Tree: 0.9211
- Random Forest: 0.9262

To better illustrate the performance of the models, we present the results in the following table:

Model	Accuracy	Precision	Recall	F1 Score
SVC	42.98	18.473	42.98	25.84
Decision Tree	98.85	98.75	98.46	98.93
Random Forest	97.229	97.231	97.229	97.228

Our analysis suggests that both the Decision Tree and Random Forest algorithms are suitable for predicting driving behaviour. While the Decision Tree model achieved the highest performance across all metrics, the Random Forest model may be preferred in certain scenarios due to its ability to reduce overfitting and improve generalisation performance.

5.3 Further work

In the future, we could use the most recent deep learning based systems, algorithms and techniques for the detection of Driving Behavior by classifying it into other major categories such as Drowsiness and Fatigue. (Alkinani, Khan, and Arshad 2020)

(Ed-doughmi and Idrissi 2019) have proposed an RNN (recurrent neural network) based driver drowsiness detection technique for road safety. The LSTM (long short-term memory) algorithm using a video public dataset can be employed for the training and validation of the proposed driver drowsiness technique.

(Zhou et al. 2020) have proposed driver fatigue detection method using driver's eyes blinking duration and sequences. In the proposed detection method, for training and testing of driver's eyes sequence through images the CNN (convolution neural networks) with LSTM can be employed. The eye regions extracted from videos using multi-task framework based on deep cascading and using deep CNN and LSTM spatial features of driver's eyes are learned. Finally, the driver fatigue is detected using the duration and sequences of his/her eyes.

References

1. Alkinani, Monagi H., Wazir Z. Khan, and Quratulain Arshad. 2020. “Detecting Human Driver Inattentive and Aggressive Driving Behavior Using Deep Learning: Recent Advances, Requirements and Open Challenges.” ieeexplore.ieee.org. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9107077.
2. Anjum, Mohd., Sana Shahab, George Dimitrakopoulos, and Habib F. Guye. 2023. “An In-Vehicle Behaviour-Based Response Model for Traffic Monitoring and Driving Assistance in the Context of Smart Cities.” www.mdpi.com. https://www.mdpi.com/2079-9292/12/7/1644/html.
3. Barmpounakis, Emmanouil. 2019. “On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment.” *Transportation Research Part C: Emerging Technologies* 111, no. 1 (July): 50-71. 10.1016.
4. Ed-doughmi, Younes, and Najlae Idrissi. 2019. “Driver Fatigue Detection using Recurrent Neural Networks.” ResearchGate. https://www.researchgate.net/publication/333253805_Driver_Fatigue_Detection_using_Recurrent_Neural_Networks.
5. Fazeen, Mohamed, Brandon Gozick, Ram Dantu, Moiz Bhukhiya, and Marta C. González. 2012. “Safe Driving Using Mobile Phones.” Researchgate. https://www.researchgate.net/publication/260543538_Safe_Driving_Using_Mobile_Phones/link/57e7f93908ae9e5e4558bf7d/download.
6. Hong, Jin-Hyuk, Ben Margines, and Anind K. Dey. 2014. “A Smartphone-based Sensing Platform to ModelAggressive Driving Behaviors.” ResearchGate. https://www.researchgate.net/publication/266655676_A_smartphone-based_sensing_platform_to_model_aggressive_driving_behaviors.

7. Koh, Dong-Woo, and Hang-Bong Kang. 2015. “Smartphone-based modeling and detection of aggressiveness reactions in senior drivers.” Researchgate. https://www.researchgate.net/publication/308810779_Smartphone-based_modeling_and_detection_of_aggressiveness_reactions_in_senior_drivers.
8. Kwon, Seong K., Ji H. Seo, Young Yun, and Kyoung Kim. 2021. “Driving Behavior Classification and Sharing System Using CNN-LSTM Approaches and V2X Communication.” MDPI. https://www.mdpi.com/2076-3417/11/21/10420.
9. Mantouka, Eleni, Emmanouil Barmpounakis, Eleni Vlahogianni, and John Golias. 2020. “.,.” Smartphone sensing for understanding driving behaviour: Current practice and challenges. https://www.sciencedirect.com/science/article/pii/S2046043020300460.
10. “Matplotlib documentation — Matplotlib 3.7.1 documentation.” n.d. Matplotlib. Accessed April 14, 2023. https://matplotlib.org/stable/index.html.
11. Myles, Anthony J., and Robert N. Feudale. n.d. “An introduction to decision tree modeling.” analyticalsciencejournals.onlinelibrary.wiley.com. Accessed April 16, 2023. https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/epdf/10.1002/cem.873.
12. “NumPy Documentation.” n.d. NumPy. Accessed April 14, 2023. https://numpy.org/doc/.
13. Olabiyi, Oluwatobi, Eric Martinson, Vijay Chintalapudi, and Rui. 2017. “Home.” YouTube. https://arxiv.org/pdf/1706.02257.pdf.
14. “pandas documentation — pandas 2.0.0 documentation.” n.d. Pandas. Accessed April 14, 2023. https://pandas.pydata.org/docs/.
15. “Python Documentation.” n.d. Python.org. Accessed April 14, 2023. https://www.python.org/doc/.
16. Revannavar, Yash. 2023. *Driving Behavior*. Berlin, Germany: Github. https://github.com/YashRevannavar/DrivingBehavior.

17. Rice, Damien, and Matt Galbraith. 2008. “An assessment of the effectiveness of a random forest classifier for land-cover classification.” www.sciencedirect.com. https://www.sciencedirect.com/science/article/abs/pii/S0924271611001304.
18. Rice, Damien, and Matt Galbraith. 2008. “Support Vector Machines for Classification and Regression.” see.xidian.edu.cn. https://see.xidian.edu.cn/faculty/chzheng/bishe/indexfiles/new_folder/svm.pdf.
19. Saiprasert, Chalermpol, Thunyasit Pholprasit, and Suttipong Thajchayapong. 2015. “Detection of Driving Events using Sensory Data on Smartphone.” ResearchGate. https://www.researchgate.net/publication/282544185_Detection_of_Driving_Events_using_Sensory_Data_on_Smartphone.
20. “scipy documentation.” n.d. scipy. Accessed April 14, 2023. https://docs.scipy.org/doc//scipy/index.html.
21. “seaborn documentation.” n.d. seaborn: statistical data visualization — seaborn 0.12.2 documentation. Accessed April 14, 2023. https://seaborn.pydata.org/.
22. “sklearn documentation.” n.d. scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation. Accessed April 14, 2023. https://scikit-learn.org/stable/index.html.
23. Vlahogianni, Eleni I., and Emmanouil N. Barmpounakis. 2017. “Driving analytics using smartphones: Algorithms, comparisons and challenges | Request PDF.” ResearchGate. https://www.researchgate.net/publication/315984585_Driving_analytics_using_smartphones_Algorithms_comparisons_and_challenges.
24. Zhou, Nana, Zuopeng Zhao, Lan Zhang, Hualin Yan, Yi Xu, and Zhongxin Zhang. 2020. “Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN.” Hindawi. https://www.hindawi.com/journals/cin/2020/7251280/.