Smart Eco Driving Assistant

Ion Cojocaru
University of Craiova
Craiova, Romania
cojocaru.ion.n8q@student.ucv.ro

Paul Stefan Popescu
University of Craiova
Craiova, Romania
stefan.popescu@edu.ucv.ro

Marian Cristian Mihaescu
University of Craiova
Craiova, Romania
cristian.mihaescu@edu.ucv.ro

ABSTRACT
This paper presents a system designed and implemented for eco-driving behaviour estimation using machine learning methods. The system is described from a high-level perspective, detailing the components and workflow, including data collection from an OBD-II adapter and smartphone sensors. The dataset used in this study was meticulously labelled and processed to train the machine learning model. We employed various clustering algorithms to automate the labelling process. The eco-driving estimation model operates in real-time, providing continuous feedback to the driver through notifications. Experimental results demonstrate the model's effectiveness in distinguishing eco-driving styles, with data metrics analysis confirming the robustness of the approach. This system aims to encourage more fuel-efficient driving habits by providing actionable insights to drivers.

Author Keywords
Machine learning, eco-driving, dataset.

DOI: 10.37789/ icusi.2024.29

INTRODUCTION
In recent years, the growing awareness of environmental issues and the urgent need to reduce greenhouse gas emissions have led to significant advancements in sustainable practices across various domains. One such domain is transportation, where ecological driving, commonly referred to as eco-driving, has emerged as a crucial strategy for mitigating the environmental impact of road vehicles. Ecological driving encompasses various driving techniques and behaviours designed to optimise fuel efficiency, improve fuel economy, and minimise emissions, contributing to environmental conservation and energy savings.

Integrating machine learning (ML) technologies into eco-driving practices represents a promising frontier in enhancing the effectiveness and adoption of sustainable driving behaviours. Machine learning, a subset of artificial intelligence, involves the development of algorithms that enable computers to learn from and make predictions or decisions based on data. By leveraging vast amounts of driving data, ML can provide drivers with personalised recommendations, real-time feedback, and predictive insights, significantly improving their ability to drive ecologically and maximise fuel economy.

A critical aspect of advancing the intersection of ecological driving and machine learning is the development of new, comprehensive datasets. High-quality datasets that capture diverse driving behaviours, vehicle types, road conditions, and environmental factors are essential for training robust ML models. Creating and utilising such datasets enable more accurate predictions and tailored eco-driving recommendations, enhancing the overall effectiveness of ML applications in this field.
Furthermore, deep learning algorithms, a subset of machine learning characterised by neural networks with many layers, offer advanced capabilities for processing complex and high-dimensional data. Deep learning can uncover intricate patterns and relationships within driving data that traditional ML methods might miss. This can lead to more precise and dynamic eco-driving recommendations and strategies, further boosting fuel efficiency and reducing emissions.

This paper explores the intersection of ecological driving and machine learning, highlighting the potential of ML and deep learning to revolutionise eco-driving practices and enhance fuel economy. We delve into the principles and techniques of ecological driving, examining how they can be augmented with machine learning and deep learning algorithms to achieve better fuel efficiency and lower emissions. Furthermore, we discuss various ML and deep learning approaches and their applications in eco-driving, such as supervised learning, reinforcement learning, unsupervised learning, and convolutional neural networks (CNNs), evaluating their effectiveness in real-world scenarios. Particular emphasis is placed on the importance of developing and utilising new datasets to drive innovation and improve the accuracy of these models in ecological driving.

Through this investigation, we aim to demonstrate how the synergy between ecological driving principles, machine learning, deep learning, and the development of new datasets can lead to more sustainable transportation systems with improved fuel economy. By providing a comprehensive overview of current research and advancements in this field, we seek to offer valuable insights for researchers, policymakers, and practitioners dedicated to promoting eco-friendly driving practices and reducing the carbon footprint of road transportation.

RELATED WORK
The problem of ecological driving is still exciting and has been tackled in many papers from a long time ago to the present. The approaches differ from one paper to another and are enhanced over time. One exciting paper is [1], which uses the OBD-II module and presents an eco-driving analysis system like ours. The authors researched and developed the

universal OBD-II module, adopted deep learning methods to evaluate fuel consumption, and proposed an intuitive driving graphic user interface design. In addition to using the universal module to obtain data on different CAN standards, this study used deep learning methods to analyse the fuel consumption of three other brands' vehicles on various road conditions. The accuracy was over 96%, thus validating the practicability of the developed system. This system will significantly benefit future OBD-II applications by collecting driving data from different car models. For example, it can be implemented to achieve eco-driving in bus driver training.

Another paper [2] refers to the same topic but from an electric vehicle perspective equipped with regenerative brakes. They state that the driver's braking style can make a significant difference and propose an approach which is based on the combination of big data and machine learning techniques to enhance the driver's braking style through visual elements (displayed in the vehicle dashboard, as a human–Machine Interface), actuating eco-driving behaviours. They designed and developed a system prototype, exploiting big data from an electric vehicle and a machine learning algorithm.

Paper [3] tackles the same problem but with a different approach as they develop a novel white-box evaluation model using machine learning for a manual transmission bus based on previous research about fuel consumption sensitivity to driving style. Using the proposed evaluation model, an algorithm for learning path planning (LPP) for a driving style is also proposed. The LPP method plans a step-by-step shortest learning path for different driving styles to achieve eco-driving. Simulation results based on vehicle and engine physical models show that the proposed evaluation model can be an alternative to the physical model for the eco-driving prompt strategy. The verification results show that the proposed strategy can improve fuel consumption by 6.25% with minimal changes to the driver's driving task and style. This approach was also close to paper [4], which tackles fuel economy in real-world traffic, but to simulate the context better designs a traffic flow prediction model based on deep learning regression machine and establishes a dynamic effective red-light duration model based on traffic flow queuing effect.

SYSTEM DESIGN

We must look at the entire system design to understand how eco-driving behaviour is estimated. The system includes several key components that work together to collect and process data from the OBD-II adapter, providing accurate assessments of driving Figure 1 presents the components of the system that integrate an Android application with an OBD-II adapter to facilitate the collection and analysis of data from a vehicle's Engine Control Unit (ECU). The system operates as follows: The vehicle's ECU serves as the primary source of diagnostic data. This data is accessed through an OBD-II adapter connecting to the vehicle's OBD-II port. The adapter wirelessly transmits the diagnostic data to the Android application via Bluetooth.

The Android application functions as the system's central hub, executing two crucial operations. Firstly, it incorporates an OBD
communication module that manages the interaction with the OBD-II adapter, ensuring seamless data collection from the vehicle's ECU. Secondly, it integrates a TensorFlow Lite (TFLite) [5] machine learning module, which processes the collected data for various diagnostic purposes.

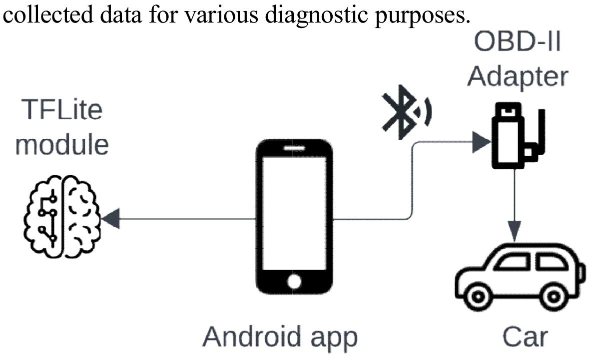


Figure 1. System Overview

Figure 2 reveals the operational workflow of the developed Android application. The process begins with the launch of the application, which then attempts to establish a Bluetooth connection with the OBD-II adapter. This connection is essential for the application's functionality, as without it, the system cannot proceed, and the application remains inoperative.
Once the connection is established, the application collects data from the vehicle's ECU. It decodes vehicle-specific details using the Vehicle Identification Number (VIN), which is used to ensure the accuracy and relevance of the data to the specific vehicle in use.

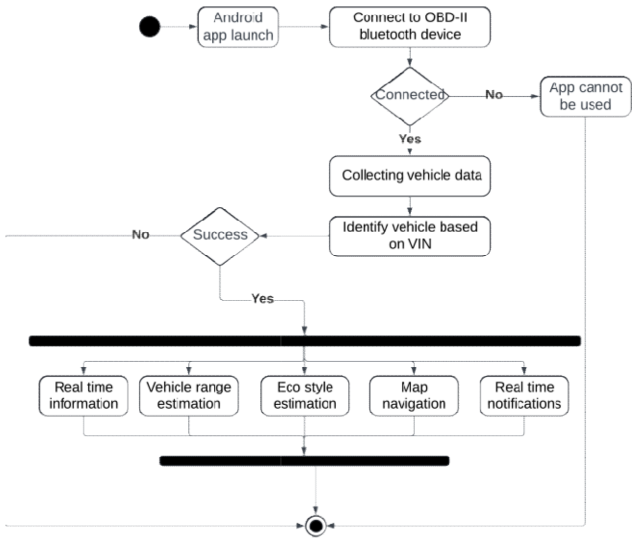


Figure 2. System flowchart

Upon successful data collection and vehicle identification, the system enables various functionalities that provide real- time information to the user. These functionalities include estimating vehicle range based on current fuel level, analysing driving eco style, and real-time notifications. These notifications are generated based on the collected data, offering insights into vehicle range and driving style.

Bluetooth communication with the OBD-II adapter

The communication with the OBD-II adapter begins by opening a Bluetooth socket connection. This connection allows the application to interact with the adapter through Parameter IDs(PID). Initially, the application identifies the adapter and sends the necessary PIDs to configure the communication channel. This configuration process includes resetting the adapter to its initial state, setting the timeout, turning off the echo, and disabling the line feed. These steps ensure a stable and efficient communication link.

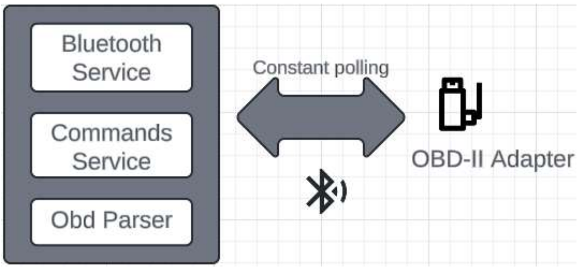


Figure 3. OBD-II communication component

Once the adapter is configured correctly, the application queries specific data from the vehicle's ECU using predefined PIDs. The adapter responds with the requested data, which the application parses using particular formulas for each PID. This parsing process converts the raw hex data into meaningful information that can be used for further analysis and display within the application.

Vehicle identification

Before accessing all functionalities, the application queries the VIN from the OBD-II adapter. The VIN is a unique code that provides detailed information about the vehicle. Once retrieved, the VIN is decoded with the help of a third- party API, which supplies the model, make, and specific details about the vehicle's configuration, including standard fuel consumption.

This detailed vehicle data is essential for adapting the eco-driving style estimation to the specific vehicle.

Different vehicles have varying fuel consumption rates, so understanding the exact configuration and specifications allows the system to provide more accurate and relevant eco-driving assessments, ensuring that the feedback given by the application is tailored to the unique characteristics of the vehicle in use.

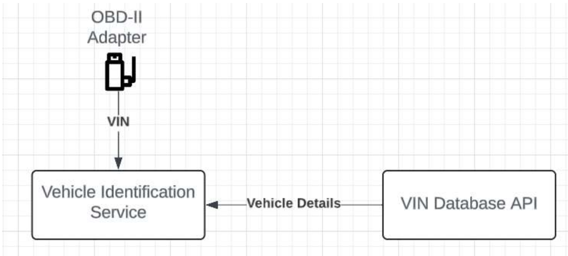


Figure 4. Vehicle identification component

Range estimation

Knowing the available vehicle range when driving to a destination can incredibly benefit drivers. It allows for better trip planning and reduces the risk of running out of fuel unexpectedly.

The application calculates the fuel use rate by continuously monitoring instant fuel consumption and vehicle speed. Combining this information with the current fuel level, the system can roughly estimate how far the vehicle can travel before refuelling.

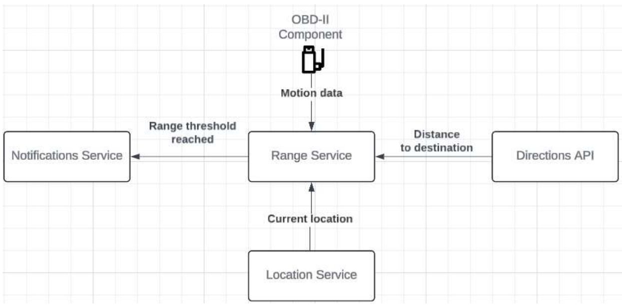


Figure 5. Range estimation component

Based on this range estimation, the application provides real-time feedback to the driver. Suppose the estimated range is insufficient to reach the destination. In that case, the driver may receive notifications to adopt an eco-friendly driving style, such as reducing speed or avoiding rapid acceleration, to extend the range. Alternatively, the application may suggest nearby gas stations where the driver can refuel, ensuring they do not run out of fuel unexpectedly.

Eco-driving style estimation

The eco-driving style estimation feature is the key component of the application, designed to promote more fuel-efficient driving habits. This feature operates by collecting data from both the OBD-II adapter, with PIDs that can be located in Table 1, and the device's sensors, such as the accelerometer, gyroscope and GPS. The collected data is fed into a machine learning model, classifying the driving behaviour into eco or non-eco categories.

PIDs	Description	Units
11	Throttle position	%
66	Mass airflow	g/s
61	Driver's demand engine torque	%
62	Actual engine per cent torque	%
46	Ambient air temperature	°C
49	Accelerator Pedal D	%
4A	Accelerator Pedal E	%
45	Relative throttle position	%
5E	Fuel rate	L/h
33	Barometric pressure	kPa
04	Engine load	%
0D	Speed	Km/h
0C	RPM	r/m

Table 1. PIDs for eco-driving estimation

This classification process happens in real-time through a background service that continuously collects and analyses data, even when the phone screen is off. The background service also periodically updates the associated notification, displaying the current eco-driving estimation and fuel consumption. This makes it easy for drivers to stay informed about their driving habits and fuel efficiency without opening the app.

In addition to real-time feedback, the application stores data such as fuel consumption rate, speed, and eco-driving style in a local SQLite database. This allows the driver to access meaningful metrics over a more extended period. By analysing this historical data, drivers can gain insights into their driving patterns and make adjustments to improve fuel efficiency and reduce their environmental impact.

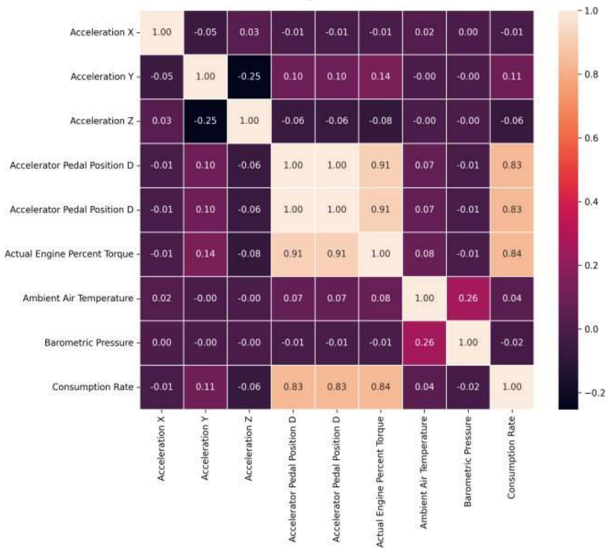


Figure 6. Correlation matrix (part 1)

Methods for estimating the eco-driving style

The dataset used in this application comprises inertial, spatial, and engine sensor data. Since this data lacked labels, the first step in training the eco-driving estimation models was to label the approximately 27,000 records in the dataset accurately. Manual labelling was not feasible, so we opted to apply unsupervised learning techniques, specifically clustering, to automate the labelling process.

Clustering algorithms

To automate the labelling process, we chose to use multiple clustering algorithms, including K-Means[6], DBSCAN [7], Gaussian Mixture[8], and Agglomerative Clustering[9]. The motivation behind choosing these clustering algorithms for our context lies in their complementary strengths in handling different aspects of the dataset. K-Means is efficient for large datasets and works well with spherical clusters, making it a good starting point. DBSCAN is robust to noise and can find arbitrarily shaped clusters, which helps identify outliers and complex driving patterns. Gaussian Mixture Models offer a probabilistic approach, allowing for the representation of clusters with varying shapes and densities. Agglomerative clustering, a hierarchical method, helps understand the nested structure of the data and does not require specifying the number of clusters in advance. Together, these algorithms provide comprehensive data analysis, enabling us to identify the most meaningful clusters for labeling the eco-driving style.

We evaluated the results of each algorithm using the Silhouette score, a metric that measures how similar an object is to its cluster compared to other clusters. By comparing the silhouette scores, we determined which algorithm provided the most accurate and meaningful clusters for labeling the data.

Feature selection

In this process, we will identify and select the most relevant features from the dataset that have the highest impact on the target variable, which is fuel consumption. This is the primary indicator of the generated CO2 emissions; thus, it is the most reliable indicator for determining eco-driving.

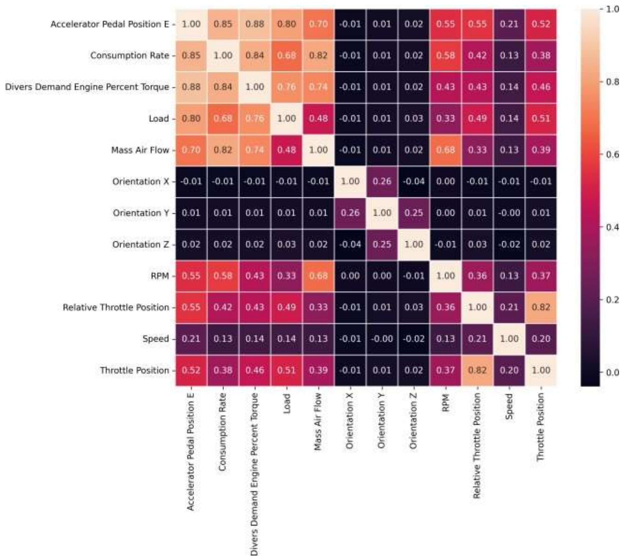


Figure 7. Correlation matrix (part 2)

Figures 6 and 7 were separated for readability and provide a detailed view of our dataset's relationships between various features. Based on the results, the gyroscope and accelerometer sensors did not yield meaningful data. However, considering that the core principle of eco-driving is minimizing carbon emissions, which is directly related to fuel consumption, we can identify the following features as having a high correlation with fuel consumption:

- Accelerator Pedal Position
- Engine Percent Torque
- Load
- Mass Air Flow
- RPM
- Throttle Position

Outliners removal

This step was performed to enhance the accuracy and reliability of our eco-driving estimation model. We identified a few instances of data that significantly deviated from the dataset mean, introducing noise and potentially skewing the model's performance. To address this, we automated the identification and removal of these outliers using the z-score technique. We effectively detected and eliminated anomalous records by calculating the z-score for all columns and comparing the absolute values with a predefined threshold.

Data dimensionality reduction

Data dimensionality reduction was performed to enhance our model by reducing the overall noise level, which is essential for improving accuracy and efficiency. Principal Component Analysis (PCA) was the perfect technique for this task, as it transforms the dataset into a set of orthogonal components, capturing the most significant variance in the data while discarding the less important, noisy dimensions. We simplified the dataset by applying PCA, retaining only the essential features that contribute most to distinguishing eco-driving styles.

Feature engineering

Vehicle specifications, especially fuel efficiency, vary significantly. Due to varying engine sizes and efficiencies, raw data such as consumption rate may not be effectively standardised across different vehicle types. We're creating new features to enhance our dataset's robustness to address this.

Feature	Formula	Definition
Instant Fuel Consumption per 100 km	Instant Fuel consumption = (Consumption Rate / Speed) * 100	Calculates the estimated fuel consumption per 100 kilometers based on the current Consumption Rate and Speed of the vehicle.
Fuel Efficiency Ratio	Ratio = Instant Fuel Consumption / Average Fuel Consumption	Normalises the Instant Fuel Consumption per 100 km against a reference average consumption rate (specified by the manufacturer).
RPM to Speed Ratio	Ratio = RPM / Speed	Helps in identifying instances where high RPM at low speeds may suggest inefficient driving practices, such as unnecessary engine strain or suboptimal gear usage.
Aggressive Acceleration	$70 \leq a \leq 100$	Indicates full or nearly full depression of the accelerator pedal.

Table 2. Engineered features

The newly engineered features in Table 2 should accentuate the driving efficiency more accurately and facilitate the clustering process by highlighting distinctive patterns characteristic of eco-friendly or non-eco-friendly driving behaviours.

RESULTS

Before presenting the actual results of the eco-driving system, we need to show the dataset used to train the model. The dataset is public and available on Kaggle[20] for downloading and contribution with notebooks. The whole system is available on GitHub, along with the application used to create the dataset [21].

Accelerator Pedal Position D	Accelerator Pedal Position E	Load	Relative Throttle Position	Actual Engine Percent Torque	Mass Air Flow	Throttle Position	Fuel Efficiency Ratio	RPM Speed Ratio	cluster
14.509804	14.509804	22.7	15.882353	6.0	7.220	24.7	0.119591	12.890411	eco
18.039216	21.568627	54.9	23.529412	0.0	7.520	24.3	0.499232	24.532258	eco
18.627451	14.509804	24.7	18.431373	8.0	7.790	27.5	0.173160	17.127273	eco
14.509804	14.509804	4.5	10.392157	0.0	5.800	21.6	0.086580	42.000000	eco
40.784314	41.176471	83.9	99.607843	43.0	24.270	83.1	4.504505	87.702703	non-eco
34.509804	34.313725	72.9	99.607843	31.5	19.050	47.8	1.619695	33.040816	non-eco
14.509804	14.509804	23.9	18.235294	7.0	7.660	26.7	0.125313	12.355263	eco
20.392157	22.352941	17.6	98.823529	2.5	6.815	21.4	0.760116	33.549296	eco
32.941176	32.745098	72.5	99.607843	25.5	19.565	83.1	0.669344	19.614458	non-eco
26.274510	25.882353	48.2	99.607843	17.0	12.220	83.1	0.497713	24.932203	non-eco

Figure 8. Eco-driving dataset sample

Figure 8 represents a sample of 10 instances from the dataset, along with the names of the features. Each instance was labelled according to the eco or non-eco cluster.

Clustering results

The clustering results were obtained by first determining the

[20] OBD-II dataset: https://www.kaggle.com/datasets/outofskills/obd-ii-dataset

[21] Advanced driver assistant app: https://github.com/OutofSkills/AdvancedDriverAssistant

optimal number of clusters using the elbow method. Based on the obtained optimal k, we then applied the Silhouette score to measure the cohesion and separation of the clusters.

Table 3. Clustering results on raw data

Algorithm	Silhouette score	
	2 clusters	3 clusters
K-means	0.44	0.45
DBSCAN	0.16	0.16
Agglomerative Clustering	0.45	0.46
Gaussian Mixture	0.45	0.47

Additionally, we analyzed visual plots to observe the distribution and boundaries of the clusters.

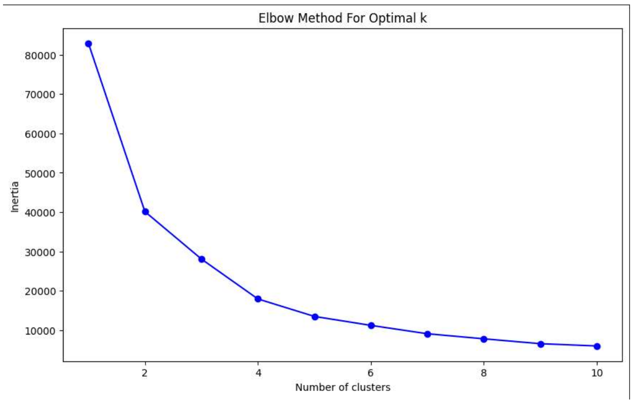


Figure 9. Optimal k number of clusters

Figure 9 presents the obtained optimal k number of clusters We can observe how the inertia decreases sharply from 1 to 2 clusters and continues to decrease at a slower rate as the number of clusters increases. The most noticeable change in the slope of the curve appears around 2 or 3 clusters.

Table 3 presents the results obtained on 2 and 3 clusters before applying PCA [10] or engineering new features. From the obtained Silhouette score, we can see that the results are almost identical.

Table 4. Clustering results with new features

Algorithm	Silhouette score	
	2 clusters	3 clusters
K-means	0.39	0.39
DBSCAN	-0.15	-0.15
Agglomerative Clustering	0.37	0.40
Gaussian Mixture	0.40	0.40

Figure 10 and Figure 11 present the clusters obtained with K-means. Based on these illustrations, we can understand the received Silhouette scores. Clusters overlap, and the "Non-Eco" cluster appears more dispersed and less dense. This variability within a cluster may reduce the silhouette score because the average intra-cluster distance increases.eco driving. For example, the fuel efficiency ratio is above 1.2, representing a higher-than-average consumption; in the other clusters, it is below 1.

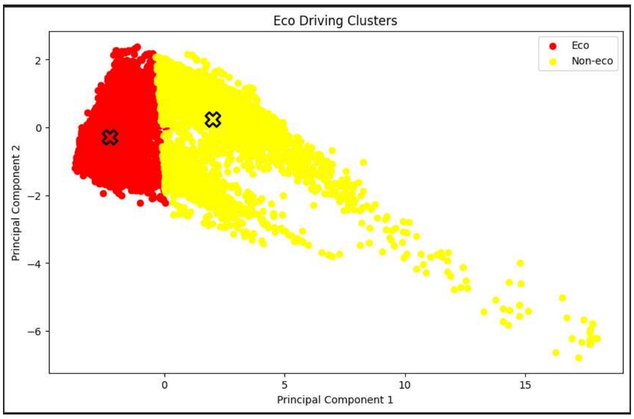


Figure 10. Raw data illustrated clusters for k=2

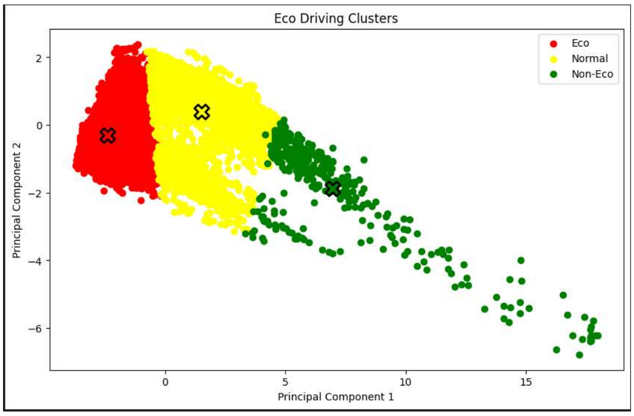


Figure 11. Raw data illustrated clusters for k=3

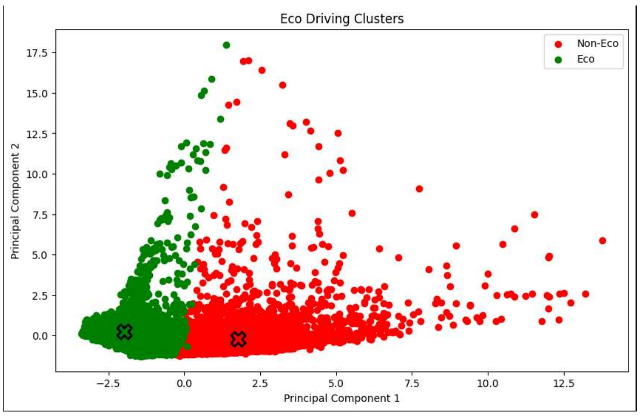


Figure 12. Clusters with new features for k=2

Table 4 presents the obtained Silhouette score after introducing new features from Table 2 in the dataset. A slight boost in scores can be observed for both approaches, with 2 and 3

clusters.
The 2 clusters obtained in Figure 12 still present a lot of dispersion for both clusters, especially the "Non-Eco" cluster, which may be a clear sign of outliners in most cases, but in our context, those points may contain significant information about the non-eco driving style.

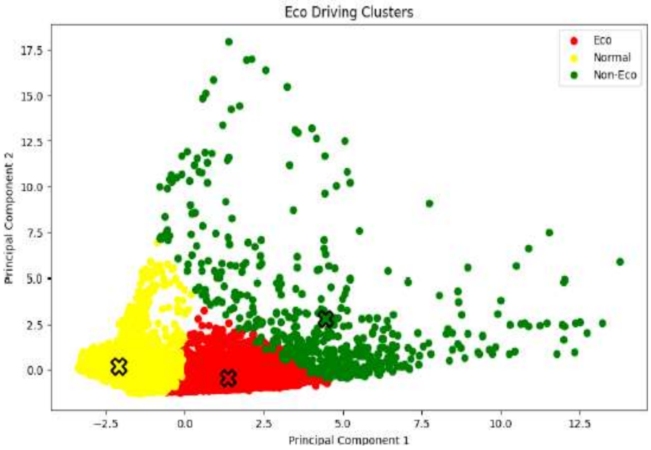


Figure 13. Clusters with new features for k=3

The newly obtained, non-eco cluster presented in Figure 13 seems highly dispersed, with a lot of noise, but the instances in this cluster have accentuated features of non-

Table 5. Clustering results with dimension reduction

Algorithm	Silhouette score	
	2 clusters	3 clusters
K-means	0.53	0.49
DBSCAN	0.51	0.51
Agglomerative Clustering	0.43	0.42
Gaussian Mixture	0.50	0.41

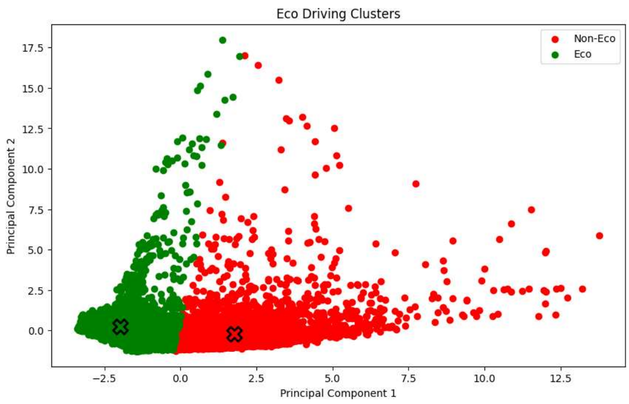


Figure 14. Clusters with less dimensions for k=2

Table 5 presents the results obtained after applying PCA to the dataset. The score was slightly improved for most of the algorithms.

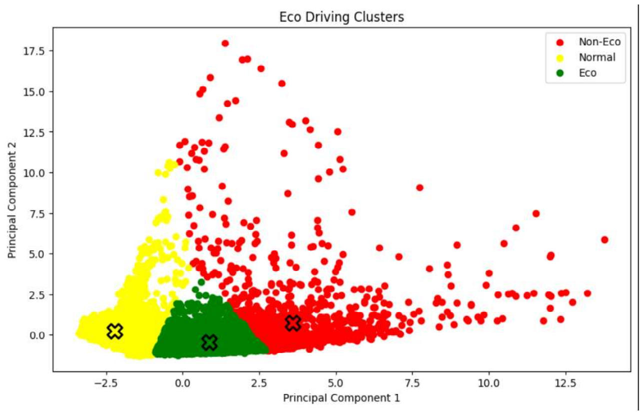
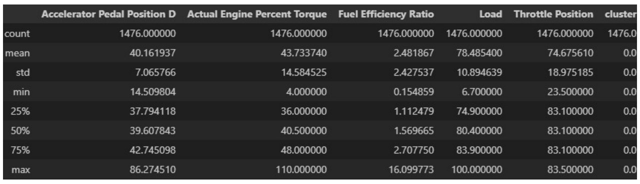


Figure 15. Clusters with less dimensions for k=3

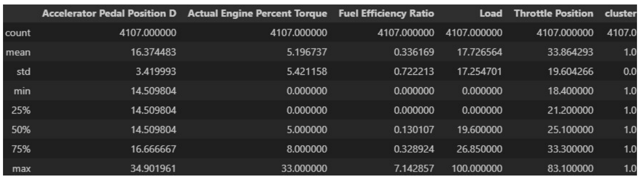
While PCA usually removes the noise in the feature reduction process from the dataset, it didn't significantly affect our case.



	Accelerator Pedal Position D	Actual Engine Percent Torque	Fuel Efficiency Ratio	Load	Throttle Position	cluster
count	1476.000000	1476.000000	1476.000000	1476.000000	1476.000000	1476.0
mean	40.161937	43.733740	2.481867	78.485400	74.675610	0.0
std	7.065766	14.584525	2.427537	10.894639	18.975185	0.0
min	14.509804	4.000000	0.154859	6.700000	23.500000	0.0
25%	37.794118	36.000000	1.112479	74.900000	83.100000	0.0
50%	39.607843	40.500000	1.569665	80.400000	83.100000	0.0
75%	42.745098	48.000000	2.707750	83.900000	83.100000	0.0
max	86.274510	110.000000	16.099773	100.000000	83.500000	0.0

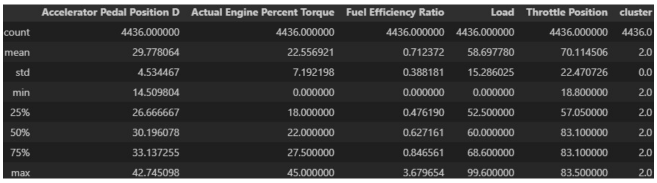
Figure 16. Non-eco cluster statistics

Figure 16 presents the non-eco-driving style resulting in cluster statistics. In these statistics, we can observe the fuel efficiency ratio, with a mean of 2.487 and a maximum value of 16.597, which indicates a poor fuel economy. This cluster has a higher average fuel efficiency ratio compared to eco-friendly cluster statistics presented in Figure 16.



	Accelerator Pedal Position D	Actual Engine Percent Torque	Fuel Efficiency Ratio	Load	Throttle Position	cluster
count	4107.000000	4107.000000	4107.000000	4107.000000	4107.000000	4107.0
mean	16.374483	5.196737	0.336169	17.726564	33.864293	1.0
std	3.419993	5.421158	0.722213	17.254701	19.604266	0.0
min	14.509804	0.000000	0.000000	0.000000	18.400000	1.0
25%	14.509804	0.000000	0.000000	0.000000	21.200000	1.0
50%	14.509804	5.000000	0.130107	19.600000	25.100000	1.0
75%	16.666667	8.000000	0.328924	26.850000	33.300000	1.0
max	34.901961	33.000000	7.142857	100.000000	83.100000	1.0

Figure 17. Eco cluster statistics



	Accelerator Pedal Position D	Actual Engine Percent Torque	Fuel Efficiency Ratio	Load	Throttle Position	cluster
count	4436.000000	4436.000000	4436.000000	4436.000000	4436.000000	4436.0
mean	29.778064	22.556921	0.712372	58.697780	70.114506	2.0
std	4.534467	7.192198	0.388181	15.286025	22.470726	0.0
min	14.509804	0.000000	0.000000	0.000000	18.800000	2.0
25%	26.666667	18.000000	0.476190	52.500000	57.050000	2.0
50%	30.196078	22.000000	0.627161	60.000000	83.100000	2.0
75%	33.137255	27.500000	0.846561	68.600000	83.100000	2.0
max	42.745098	45.000000	3.679654	99.600000	83.500000	2.0

Figure 18. Normal cluster statistics

Figure 17 and Figure 18 present the statistics of the eco and regular driving style clusters. We can see apparent differences between these two and the non-eco cluster. For example, the Load, Fuel Efficiency Rate, Engine Torque and Accelerator Pedal Position are much lower in the normal and eco clusters than in the non-eco clusters, indicating an evident success.

CONCLUSION

This paper presented an eco-driving behaviour estimation system using machine learning techniques and data collected from an OBD-II adapter and smartphone sensors. To enhance usability, the system was designed to provide real-time feedback and notifications to the driver, encouraging more fuel-efficient driving habits. Labelling the dataset was automated using various clustering algorithms, and feature selection was based on their correlation with fuel consumption, ensuring the most relevant data was used for model training.

Feature engineering was employed to create new features, significantly boosting clustering accuracy. Outlier removal further reduced noise, improving the robustness of the model. Splitting the data into 2 or 3 clusters was feasible using the elbow method. We validated these clusters by measuring performance using the Silhouette score, data plots, and metrics analysis.

The results showed that the system could accurately distinguish between eco and non-eco-driving styles, with clustering validation confirming the effectiveness of our approach. This system provides drivers with immediate feedback and stores historical data, allowing for long-term analysis and continuous driving behavior improvement. Our eco-driving estimation system demonstrates a promising step towards promoting more sustainable driving practices.

REFERENCES

1. Yen, M. H., Tian, S. L., Lin, Y. T., Yang, C. W., & Chen, C. C. (2021). Combining a universal OBD-II module with deep learning to develop an eco-driving analysis system. Applied Sciences, 11(10), 4481.
2. Delnevo, G., Di Lena, P., Mirri, S., Prandi, C., & Salomoni, P. (2019). On combining Big Data and machine learning to support eco-driving behaviours. Journal of Big Data, 6, 1-15.
3. Ma, H., Xie, H., & Brown, D. (2017). Eco-driving assistance system for a manual transmission bus based on machine learning. IEEE Transactions on Intelligent Transportation Systems, 19(2), 572-581.
4. Zhang, X., Zhang, T., Zou, Y., Du, G., & Guo, N. (2020). Predictive eco-driving application considering real-world traffic flow. Ieee Access, 8, 82187-82200.
5. Dai, J. (2020, November). Real-time and accurate object detection on edge device with TensorFlow Lite. In Journal of physics: conference series (Vol. 1651, No. 1, p. 012114). IOP Publishing..
6. Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k- means algorithm: A comprehensive survey and performance evaluation. Electronics, 9(8), 1295.
7. Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Transactions on Database Systems (TODS), 42(3), 1-21.
8. McLachlan, G. J., & Rathnayake, S. (2014). On the number of components in a Gaussian mixture model. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(5), 341-355.
9. Ackermann, M. R., Blömer, J., Kuntze, D., & Sohler, C. (2014). Analysis of agglomerative clustering. Algorithmica, 69, 184-215.
10. Maćkiewicz, A., & Ratajczak, W. (1993). Principal components analysis (PCA). Computers & Geosciences, 19(3), 303-342.