



# Driving Behavior Classification by Vehicle OBD data Using Machine Learning

Ms A JayaSmruthi<sup>1</sup>, J Salman<sup>2</sup>, K Santhosh<sup>3</sup>, N Sanjeev Kumar<sup>4</sup>

<sup>1</sup>Assistant Professor, Department of CSE, Sri Ramakrishna Institute of Technology, Coimbatore, Tamilnadu, India

<sup>2,3,4</sup>UG Students, Department of CSE, Sri Ramakrishna Institute of Technology, Coimbatore, Tamilnadu, India

**Abstract:** Driving behavior analysis is pivotal for enhancing road safety, optimizing fuel consumption, and understanding vehicle performance. The study focuses on developing a model to classify driving behavior using data collected from various vehicle parameters. The dataset comprises driving data from multiple drivers, encompassing features such as fuel consumption, engine speed, vehicle speed, and gear usage. The project is segmented into several phases. Initially, the dataset undergoes preprocessing, followed by the application of feature extraction techniques to derive meaningful features associated with driving behavior. Subsequently, the dataset is divided into training and testing sets, with the majority of drivers' data used for model training and a subset reserved for model evaluation. Various machine learning algorithms are evaluated for model development, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Gradient Boosting, Perceptron, and Stochastic Gradient Descent (SGD) classifiers. The performance of each model is assessed using accuracy metrics and classification reports. Additionally, oversampling techniques are employed to address class imbalance issues in the training dataset, ensuring robust model performance across different driving behaviors. The developed models demonstrate high accuracy in classifying driving behavior, with Gradient Boosting emerging as the top-performing algorithm. The model achieves accuracy scores ranging from 81.8% to 99.4% across different classifiers, showcasing its effectiveness in predicting driving behavior. Furthermore, the trained models are tested using data from new drivers, confirming their ability to generalize to unseen driving behavior patterns. The predictions provide valuable insights into the driving styles of the new drivers, facilitating personalized feedback and interventions for improving road safety and fuel efficiency. Overall, the study highlights the potential of machine learning techniques in analyzing driving behavior and underscores their significance in promoting safer and more efficient driving practices.

**Keywords:** OBD; Behavior; Accuracy; Google Colab; Grading Boost;

## I. INTRODUCTION

The increasing integration of technology into vehicles has paved the way for advanced monitoring and analysis of driver behaviour. One crucial aspect of this

technological integration is the use of on-board diagnostics (OBD) systems, which provide real-time data on various parameters related to vehicle performance and driver actions. Leveraging this wealth of data, the project endeavors to develop a predictive model capable of categorizing driver behaviour into distinct patterns such as aggressive, normal, and economic driving styles.

Utilizing the machine learning techniques, the project seeks to harness the insights hidden within OBD data to enhance driver safety, optimize fuel efficiency, and improve overall vehicle performance. Understanding driver behaviour patterns can aid in identifying risky driving habits, reducing the likelihood of accidents, and promoting safer driving practices.

The project's methodology involves several key steps, including data pre-processing to clean and prepare the OBD dataset, feature extraction to derive meaningful metrics from the raw data, model development using various machine learning algorithms, and model evaluation to assess performance metrics such as accuracy, precision, recall, and F1-score.

Through this endeavor, the project aims to contribute to the growing field of intelligent transportation systems by providing a robust framework for analyzing and predicting driver behaviour based on OBD data. The insights gained from this research can potentially inform the design of future vehicle safety systems and facilitate the development of personalized driving assistance technologies.

## II. SCOPE OF THE PROJECT

The scope of the project encompasses several key aspects, beginning with the collection and preprocessing of On-Board Diagnostics (OBD) data from vehicles equipped with OBD systems. This data includes parameters such as engine speed, vehicle speed, gear position, and brake usage. The collected data undergoes rigorous preprocessing to handle missing values, outliers, and inconsistencies, ensuring its quality and reliability.

Next, the project focuses on feature extraction, where relevant features are derived from the raw OBD data to capture essential aspects of vehicle behavior and driver performance. Feature extraction techniques are employed to transform and enhance the extracted features,

thereby improving the model's performance in classifying driving behavior accurately.

Furthermore, the project delves into model development, employing various machine learning algorithms such as Support Vector Machines (SVM), Gradient Boosting, and K-Nearest Neighbors (KNN) to develop predictive models for driving behavior classification. These models are trained on the extracted features and evaluated using performance metrics such as accuracy, precision, recall, and F1-score.

Additionally, the project aims to validate the developed models using real-world driving data, ensuring their robustness and generalizability across diverse driving scenarios. By testing the models on new datasets, the project seeks to assess their ability to accurately classify driving behavior and provide valuable insights into individual driving styles.

Overall, the scope of the project encompasses the entire process of analyzing driving behavior using machine learning techniques, from data collection and preprocessing to feature extraction, model development, and validation. Through this comprehensive approach, the project aims to contribute to the field of intelligent transportation systems by providing a reliable framework for analyzing and predicting driving behavior based on OBD data.

### III. PROPOSED SYSTEM

The proposed system endeavors to develop a predictive model for driver behavior classification based on data collected from On-Board Diagnostics (OBD) systems installed in vehicles. The system comprises several key components and methodologies:

#### A. Data Collection and Preprocessing:

Integration of OBD systems in vehicles to collect raw data on various parameters such as engine speed, vehicle speed, indication of brake switch ON/OFF, current gear, among others. Pre-processing tasks including handling missing values, eliminating duplicates, and addressing outliers or anomalies to ensure data quality and consistency.

#### B. Feature Extraction:

Application of feature extraction techniques to derive meaningful features from the raw OBD data. These features encompass crucial aspects of driver behavior and vehicle performance, including idle engine time, occurrences of high-speed braking, engine revving events, average speed, and gear utilization patterns.

#### C. Model Development:

Utilization of machine learning algorithms such as KNN, Support Vector Machine (SVM), and SGD classifier to develop a classification model for driver behaviour. Training the model using the extracted features and corresponding driver behaviour labels obtained from the pre-processed dataset.

#### D. Model Evaluation:

Evaluation of the trained model's performance using metrics such as accuracy, precision, recall, and F1-score to assess its effectiveness in predicting driver behaviour categories.

#### E. Real-World Applications:

Deployment of the developed model in real-world scenarios to classify driver behavior into categories such as Aggressive, Normal, and Economic. Potential applications of the model include driving support systems, insurance risk assessment, and remote claims processing.

## IV. ADVANTAGE OF THE PROPOSED SYSTEM

#### A. Improved Accuracy:

By leveraging real-time data processing techniques and advanced machine learning algorithms, the proposed system enhances the accuracy of driver behavior classification compared to the existing system. This leads to more reliable insights into driver behavior patterns and more accurate classification results.

#### B. Enhanced Robustness:

The proposed system's adaptability to emerging technologies and its scalability ensure that it remains robust and effective even as vehicle technologies evolve. This enables it to handle a wider range of vehicles, including hybrid and electric vehicles, and adapt to future advancements in vehicle diagnostics and data acquisition.

#### C. Reduced Latency:

With optimized data acquisition and storage mechanisms, the proposed system minimizes latency in processing OBD data, enabling real-time classification of driver behavior. This reduces delays in obtaining classification results, making the system more responsive and suitable for time-sensitive applications.

Overall, the proposed system offers a more accurate, robust, and user-friendly solution for driver behavior analysis and classification compared to the existing system, addressing its limitations while leveraging modern technologies and methodologies to deliver superior performance.

## V. LITERATURE SURVEY

Raman kumar and Anuj Jain, [1] The transportation sector's emphasis on enhancing efficiency and cutting costs has spurred the integration of IoT and machine learning technologies. Recognizing the link between driving behavior and fuel consumption/emissions, there's a growing need to categorize various driving patterns. Modern vehicles are equipped with sensors that capture extensive operational data, including speed, motor RPM, paddle position, and numerous other parameters via the OBD interface. Leveraging the OBD-II diagnostics protocol, real-time data related to engine operation are obtained from the car's communication port, aiding in both performance monitoring and fault detection. To address this need, machine learning techniques like SVM, AdaBoost, and Random Forest are utilized to classify driver behavior across ten categories, encompassing factors such as fuel consumption, steering stability,

velocity consistency, and braking habits. This approach offers an effective means to analyze driving behavior and recommend improvements for safer and more efficient driving practices. By extracting data directly from the engine's internal sensors through the OBD-II protocol, the reliance on additional sensors is eliminated, streamlining the data collection process. Employing supervised learning methods enables a thorough comparison across all driver classes. Achieving accuracy rates of 99% for SVM, 99% for AdaBoost, and 100% for Random Forest algorithms underscores the efficacy of the proposed model. In summary, this approach offers a practical solution for evaluating driving behavior and implementing measures to improve driving safety and efficiency.

Engr. Muhammad Kashif Shaikh, et al. [2] The development of an Android application aimed at tracking driving behavior and alerting users to any irregularities or anomalies represents a significant step towards promoting safer driving practices. By connecting to an OBD-II device via Bluetooth and Wi-Fi, the app efficiently records key metrics of driving behavior in real-time and post-trip for later analysis. This innovative approach not only enhances awareness of the potential dangers associated with reckless driving, over-speeding, and abrupt lane changes but also provides valuable insights to help prevent catastrophic accidents. The app's ability to offer timely alerts and access to comprehensive driving data underscores its value in fostering safer driving habits and reducing the risk of accidents and loss in Pakistan and beyond.

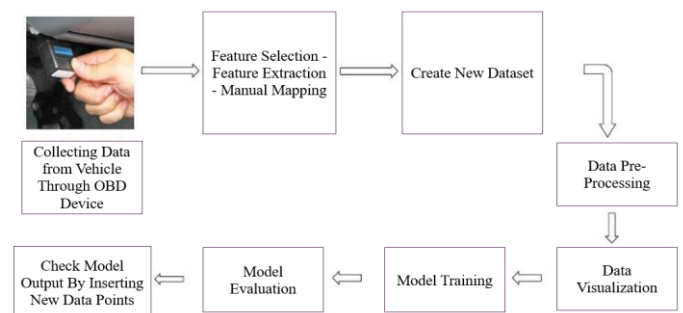
YINGFENG CAI, et al. [3] This paper aims to establish a driving style recognition method that is highly accurate, fast and generalizable, considering the lack of data types in driving style classification task and the low recognition accuracy of widely used unsupervised clustering algorithms and single convolutional neural network methods. First, we propose a method to collect the information on driver's operation time sequence in view of the imperfect driving data, and then extract the driver's style features through convolutional neural network. Then, for the collected temporal data, the Long Short Term Memory networks (LSTM) module is added to encode and transform the driving features, to achieve the driving style classification. The results show that the accuracy of driving style recognition reaches over 93%, while the speed is improved significantly.

Teck Kai Chan, et al., [4] Human factors are the primary catalyst for traffic accidents. Among different factors, fatigue, distraction, drunken ness, and/or recklessness are the most common types of abnormal driving behavior that leads to an accident. With technological advances, modern smartphones have the capabilities for driving behavior analysis. There has not yet been a comprehensive review on methodologies utilizing only a smartphone for drowsiness detection and abnormal driver behavior detection. In this paper, different methodologies proposed by different authors are discussed. It includes the sensing schemes, detection algorithms, and their corresponding accuracy and limitations. Challenges and possible solutions such as integration of the smartphone behavior classification system with the concept of context-aware, mobile crowdsensing, and active steering control are analyzed. The issue of model training and updating on the smartphone and cloud environment is also included.

Shuo Jia1 , Fei Hui1 , Shining Li1 , Xiangmo Zhao1 , Asad J. Khattak1,2, [5] Abnormal driving behaviours, such as rapid acceleration, emergency braking, and rapid lane changing, bring great uncertainty to traffic, and can easily lead to traffic accidents. The accurate identification of abnormal driving behaviour helps to judge the driver's driving style, inform surrounding vehicles, and ensure the road traffic safety. Most of the existing studies use clustering and shallow learning, it is difficult to accurately identify the types of abnormal driving behaviours. Aimed at addressing the difficulty of identifying driving behaviour, this study proposed a recognition model based on a long short-term memory network and convolutional neural network (LSTM-CNN). The extreme acceleration and deceleration points are detected through the statistical analysis of real vehicle driving data, and the driving behaviour recognition data set is established. By using the data set to train the model, the LSTM-CNN can achieve a better result.

### VI. SYSTEM ARCHITECTURE

1. The process starts with the Raw Dataset which is taken from the vehicle through the OBD device to predict the driving behaviour.
2. At first, the feature selection and extraction should be done to selected and extract the feature that is necessary and relevant to classify the driving behavior through Google Colab.
3. From the extracted feature, create new dataset and do manual labelling.
4. Next, the new dataset should be loaded for data pre-processing and data visualization.
5. Then the dataset is divided into 80:20 ratio for to train and test the Model.
6. After develop the model through different algorithms, select one of the algorithm which gives high accuracy.
7. At last, give new datapoint to that selected algorithm model and analyze the output.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Fuel_cons	Absolute_Engine_sp	Engine_co	Calculated	Activation	Current_V	Vehicle_sp	Accelerati	Indication	Steering_v	Steering_v	Time[s]	Class	
2	268.8	13.3	929	95	23.9	0	0	0	-8.5	1	0	-3.4	1 A	
3	243.2	13.7	726	95	30.6	0	0	0	0.1	1	0	-3.6	2 A	
4	217.6	13.7	685	95	31.8	0	0	0	0.1	1	0	-3.6	3 A	
5	204.8	13.3	675	95	29	0	0	0	0.1	1	0	-3.6	4 A	
6	217.6	13.7	716	95	30.2	1	0	0	0.1	1	0	-3.5	5 A	
7	243.2	13.7	740	95	31.4	1	0	0	0	2	0	-3.4	6 A	
8	217.6	13.7	682	94	31.8	1	14	0	0	2	0	-3.6	7 A	
9	294.4	14.5	670	94	41.2	1	14	0	0	2	0	-1.6	8 A	
10	332.8	14.5	655	94	49	1	14	0	0	2	8	7.9	9 A	
11	358.4	14.9	648	94	54.1	1	14	0	-0.1	2	0	7.7	10 A	
12	409.6	15.3	659	93	58	1	14	0	-0.2	2	0	5.9	11 A	
13	435.2	14.9	663	93	58	1	14	0	-0.1	2	0	4.8	12 A	
14	384	14.9	659	93	55.7	1	14	0	0	2	8	-7.3	13 A	
15	396.8	14.9	668	93	55.3	1	14	1	-0.1	2	0	-7.3	14 A	

## Figure 1: System Architecture for proposed system

### DATA COLLECTION

A Raw Dataset was collected from OBD device and stored as a .csv file format [1], that contains more than 20 parameters and then select and extract the features that are highly correlated to the labelled data.

## Figure 2: Dataset Collected from the OBD Device.

### DATASET DESCRIPTION

The dataset utilized in the project originates from OBD (On-Board Diagnostics) systems and encompasses a diverse array of parameters pertaining to vehicle performance and driver behaviour. Initially, the dataset comprises raw OBD data collected during vehicle operation, encompassing parameters like engine speed, vehicle speed, and indication of brake switch ON/OFF, current gear, among others. Following the data collection process, a series of pre-processing tasks are undertaken to ensure data quality, involving handling missing values, eliminating duplicates, and addressing outliers or anomalies.

Subsequently, feature extraction techniques are applied to derive insightful features from the raw OBD data. Key aspects of driver behaviour and vehicle performance, including metrics like idle engine time, occurrences of high-speed braking, engine revving events, average speed, and gear utilization patterns, are encapsulated within the features extracted. By extracting such features, the dataset is enriched with meaningful insights into driver behaviour, thereby enhancing its suitability for predictive modeling tasks.

The culmination of these processes yields the derived dataset, which comprises the extracted features alongside corresponding labels denoting driver behaviour categories (e.g., Aggressive, Normal, and Economic). Each instance in the derived dataset encapsulates a combination of extracted features and its associated driver behaviour category, forming a structured dataset conducive to the development and evaluation of machine learning models aimed at predicting driver behaviour based on OBD data.

### FEATURE EXTRACTION

Feature extraction is the process of transforming raw data into a set of meaningful features that can be used as inputs to a machine learning algorithm. Following are the steps happens during feature extraction:

#### A. Identifying Relevant Features:

First need to understand the data and identify which features are relevant for the task at hand. This may involve domain knowledge, data exploration, and understanding the problem and try to solve.

#### B. Data Pre-Processing:

Before extracting features, it is often necessary to pre-process the data. This can include handling missing values, scaling features to a similar range, encoding categorical variables, and removing noise or outliers.

#### C. Transforming Raw Data:

Once the data is pre-processed, you can start extracting features. This involves transforming raw data into a format that can be used by machine learning algorithms. For example, in the context of the OBD data mentioned earlier, calculate statistics such as mean, median, standard deviation, maximum, and minimum values for features like engine speed, vehicle speed, and braking events.

#### D. Creating New Features:

Sometimes, it's beneficial to create new features based on existing ones. For example, calculating the ratios or differences between different features, or derive new features based on domain knowledge.

#### E. Dimensionality Reduction:

In cases where the dataset has a large number of features, dimensionality reduction techniques like Principal Component Analysis (PCA) or feature selection methods may be applied to reduce the number of features while preserving most of the relevant information.

#### F. Feature and Scaling:

It is common to scale features to a similar range to ensure that they contribute equally to the model's performance. This can be done using techniques like min-max scaling or standardization.

Overall, feature extraction plays a crucial role in building effective machine learning models by transforming raw data into a meaningful representation that captures the important characteristics of the data.

### DERIVED PARAMETERS

Under the derived parameters topic, each individual parameter in the dataset is explained, along with its impact on adjusting the driver score:

#### A. IDLE\_ENGINE:

-Description: IDLE\_ENGINE represents the number of instances where the engine speed remains idle while the vehicle is stationary. It indicates periods of inactivity or prolonged idling of the engine.

-Impact on Driver Score: An increased number of IDLE\_ENGINE instances may suggest inefficient driving behavior, leading to higher fuel consumption and emissions. Therefore, a higher count of IDLE\_ENGINE instances can negatively adjust the driver score, reflecting poorer driving habits.

#### B. HIGH\_SPEED\_BRAKING:

-Description: HIGH\_SPEED\_BRAKING denotes the number of instances of high-speed braking events, indicating sudden or aggressive braking at high velocities.

-Impact on Driver Score: Frequent occurrences of HIGH\_SPEED\_BRAKING events may indicate

aggressive driving behaviour, posing safety risks and causing increased wear and tear on vehicle components. Consequently, a higher count of HIGH\_SPEED\_BRAKING instances can significantly adjust the driver score downwards.

**C. REVV\_ENGINE:**

-Description: REVV\_ENGINE represents the number of instances where the engine speed exceeds a predefined threshold, indicating aggressive acceleration or revving of the engine.

-Impact on Driver Score: Excessive revving of the engine may indicate aggressive driving behavior, leading to increased fuel consumption and potential damage to the engine. Thus, a higher count of REVV\_ENGINE instances can negatively adjust the driver score.

**D. AVG\_SPEED:**

-Description: AVG\_SPEED denotes the average vehicle speed during the recording period, calculated based on the speedometer readings.

-Impact on Driver Score: Higher average speeds may suggest faster driving, potentially leading to reduced fuel efficiency, increased risk of accidents, and higher emissions. Consequently, excessively high average speeds can negatively adjust the driver score.

**E. GEAR\_1, GEAR\_2, GEAR\_3, GEAR\_4:**

-Description: GEAR\_1, GEAR\_2, GEAR\_3, and GEAR\_4 represent the counts of gear shifts into the first, second, third, and fourth gears respectively, indicating gear usage patterns.

-Impact on Driver Score: Gear shifts reflect driver behaviour and driving conditions. Inefficient gear usage, such as frequently shifting gears at inappropriate times or excessive gear changes, can impact fuel efficiency and vehicle performance, thereby adjusting the driver score downwards.

Overall, each parameter in the dataset provides insights into specific aspects of driver behaviour and vehicle operation, allowing for precise adjustments to the driver score based on driving habits and performance metrics.

**MODEL DEVELOPMENT**

**A. K-NEAREST NEIGHBOR ALGORITHM (KNN):**

The K-Nearest Neighbors (KNN) algorithm is a simple and effective supervised machine learning algorithm used for classification and regression tasks. The algorithm memorizes the entire training dataset. In the case of classification, it stores all the data points along

with their corresponding class labels. When given a new, unseen data point, KNN looks at the 'k' closest data points (neighbors) to the new point based on some distance metric (usually Euclidean distance in most cases).

For classification tasks, KNN takes a majority vote among the 'k' neighbors to determine the class of the new data point. The class that appears most frequently among the neighbors is assigned to the new data point.

**Figure 3: Performance metrics of Model using KNN Algorithm**

**B. GRADIENT BOOSTING:**

Gradient Boosting combines multiple weak learners (typically decision trees) sequentially to build a strong learner. Each weak learner is trained to correct the errors made by the previous ones. Gradient Boosting builds a strong predictive model by iteratively improving upon the errors made by previous models. It is known for its high predictive accuracy and robustness against overfitting, although it can be computationally expensive and requires careful tuning of hyper parameters. Popular implementations include Gradient Boosting Machines (GBM), XGBoost, LightGBM, and CatBoost.

```

Accuracy (Gradient Boosting): 0.9943181818181818
Classification Report (Gradient Boosting):
      precision    recall  f1-score   support

 0         1.00      1.00      1.00         59
 1         1.00      0.98      0.99         60
 2         0.98      1.00      0.99         57

 accuracy         0.99         0.99         0.99         176
 macro avg        0.99         0.99         0.99         176
 weighted avg    0.99         0.99         0.99         176
    
```

**Figure 4: Performance metrics of Model using Gradient Boosting Algorithm**

**C. SUPPORT VECTOR MACHINE (SVM):**

SVM works by finding the best possible line or boundary (called hyperplane) that separates different classes in the feature space. For example, in a simple 2D space with two classes, the hyperplane is a line. SVM aims to maximize the margin, which is the distance between the hyperplane and the nearest data points from each class. These nearest data points are called support vectors. Maximizing the margin helps improve the generalization of the model and reduces the risk of overfitting. SVM is known for its ability to handle high-dimensional data and its effectiveness in cases where the number of features is greater than the number of samples. It's widely used in various applications such as text classification, image recognition, and bioinformatics.

```

Accuracy (SGD Classifier): 0.8181818181818182
Classification Report (SGD Classifier):
      precision    recall  f1-score   support

 0         0.82      1.00      0.90         59
 1         1.00      0.47      0.64         60
 2         0.75      1.00      0.86         57

 accuracy         0.82         0.82         0.82         176
 macro avg        0.86         0.82         0.80         176
 weighted avg    0.86         0.82         0.80         176
    
```

```

Accuracy (SVM): 0.8390804597701149
Classification Report (SVM):
      precision    recall  f1-score   support

 0         1.00      1.00      1.00         53
 1         0.69      1.00      0.82         63
 2         1.00      0.52      0.68         58

 accuracy         0.84         0.84         0.84         174
 macro avg        0.90         0.84         0.83         174
 weighted avg    0.89         0.84         0.83         174
    
```

### Figure 5: Performance metrics of Model using SVM Algorithm

#### D. STOCHASTIC GRADIENT DESCENT:

"Stochastic" means random. In SGD, instead of looking at the whole dataset to take a step (which can be slow for large datasets), it looks at just one random data point at a time. So, it's like randomly picking a spot on the hill, checking the slope there, and taking a step. SGD is fast and works well for large datasets because it updates the model's parameters frequently based on small batches of data.

However, it can be noisy because of the randomness, and it might take more steps to converge compared to traditional gradient descent, but it is often used because of its efficiency in training large-scale models.

Accuracy (K-Nearest Neighbors): 0.9942528735632183  
Classification Report (K-Nearest Neighbors):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	53
1	0.98	1.00	0.99	63
2	1.00	0.98	0.99	58
accuracy			0.99	174
macro avg	0.99	0.99	0.99	174
weighted avg	0.99	0.99	0.99	174

### Figure 6: Performance metrics of Model using SGD Classifier Algorithm

#### E. PERCEPTRON CLASSIFIER:

The perceptron classifier is one of the simplest forms of linear classifiers used in machine learning. The perceptron algorithm starts by initializing the weights and bias parameters. These parameters are used to define a hyperplane that separates the data into different classes. During the training phase, the perceptron algorithm iterates through the training data points one by one. For each data point, it computes the weighted sum of the input features using the current weights and bias. Once the perceptron is trained, it can be used to make predictions on new data points by computing the weighted sum of the input features and applying the activation function.

### Figure 7: Performance metrics of Model using Perceptron Algorithm.

## COMPARISON OF ALGORITHMS

### Figure 8: Comparison of all algorithms through accuracy

## VII. RESULT AND ANALYSIS

The classification results obtained from testing the trained models demonstrated promising performance in accurately classifying driving behavior. The models achieved high accuracy rates in distinguishing between Economic, Normal, and Aggressive driving behaviors, indicating their effectiveness in leveraging vehicle OBD data for driving behavior classification.

```
# Assuming you have a new data point stored in a variable named 'new_data_point'

# 1. Prepare the new data point
new_data_point = [37,33,2,20.91,0,0,5,0] # Replace with actual values

# 2. Use the trained Random Forest classifier to make predictions on the new data point
prediction_rf = gb_classifier.predict([new_data_point])

# 3. Map the predicted class label to a human-readable interpretation
class_mapping = {
    0: "Aggressive",
    1: "Normal",
    2: "Economic"
}

predicted_class = class_mapping[prediction_rf[0]]

# 4. Print or display the predicted class label
print("Gradient Boosting Prediction:", predicted_class)
```

### Figure 9: Giving New points to Gradient Boosting Model for Prediction

```
Gradient Boosting Prediction: Normal
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
warnings.warn(
```

### Figure 10: The Model using Gradient Boosting Algorithm Predict accurately that the given data will be a Normal Driving.

```
# Assuming you have a new data point stored in a variable named 'new_data_point'

# 1. Prepare the new data point
new_data_point = [595, 25, 481, 27.64, 0, 0, 0, 0] # Replace with actual values

# 2. Use the trained Gradient Boosting classifier to make predictions on the new data point
prediction_gb = gb_classifier.predict([new_data_point])

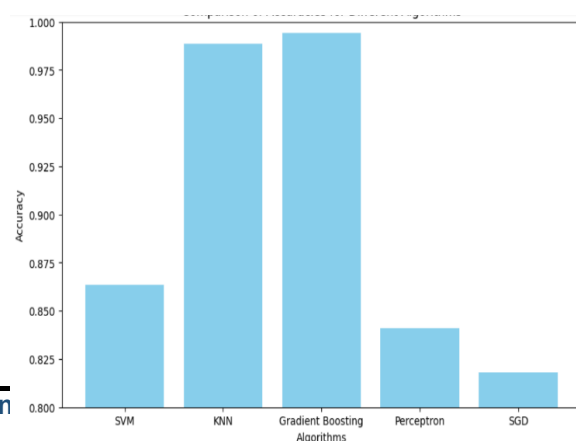
# 3. Map the predicted class label to a human-readable interpretation
class_mapping = {
    0: "Aggressive",
    1: "Normal",
    2: "Economic"
}

predicted_class = class_mapping[prediction_gb[0]]

# 4. Print or display the predicted class label
print("Gradient Boosting Prediction:", predicted_class)
```

### Figure 11: Giving another set of New Points to model to predict the behaviour.

```
Gradient Boosting Prediction: Aggressive
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
warnings.warn(
```



**Figure 12: Model Predicted accurately that the given set of data will be Aggressive Driving Behaviour.**

## VIII. CONCLUSION

In conclusion, the project successfully addresses the critical need for comprehensive analysis of driving behavior using machine learning techniques. By leveraging onboard diagnostics (OBD) data and advanced algorithms, the project develops predictive models capable of categorizing driving behavior into distinct patterns. The models offer valuable insights into individual driving styles, facilitating interventions to enhance road safety and optimize fuel efficiency. The methodology employed in the project, including data pre-processing, feature extraction, model development, and evaluation, demonstrates a systematic approach to tackling complex real-world problems. Moreover, the validation of the models with new data points underscores their robustness and generalizability across diverse driving scenarios. Overall, the project contributes to the advancement of intelligent transportation systems and holds promise for improving road safety and sustainability in the automotive industry.

## IX. FUTURE WORKS

In the future, the proposed system could be expanded and enhanced in several ways to further improve its functionality and effectiveness:

### A. Integration of Advanced Sensor Data:

Incorporating additional sensor data beyond OBD sensors, such as GPS, accelerometer, and camera data, could provide more comprehensive insights into driver behavior and vehicle performance.

### B. Real-Time Monitoring and Alerts:

Implementing real-time monitoring capabilities and instant alerts for critical events, such as harsh braking or acceleration, could enable immediate intervention and accident prevention.

### C. Adaptive Driver Feedback:

Designing personalized feedback mechanisms for drivers based on their behavior patterns could promote safer driving habits and improve overall driver performance.

### D. Data Privacy and Security Enhancements:

Implementing robust data privacy and security measures to protect sensitive driver and vehicle information from unauthorized access or misuse.

By pursuing these future avenues of research and development, the proposed system can continue to evolve and address emerging challenges in fleet management, driving safety, and vehicle maintenance.

## REFERENCES

[1] Raman kumar and Anuj Jain, "Driving behavior analysis and classification by vehicle OBD data using machine learning", The Journal of Supercomputing, Phagwara, India., vol. 79, sp. 18800, ep. 18819, May 2023.

- [2] Engr. Muhammad Kashif Shaikh, Prof. Dr. Sellappan Palaniappan, Engr. Fayyaz Ali, Engr. Dr. Muhammad Khurram, "Identifying Driver Behaviour Through OBD-II Using Android Application", vol. 17(7). University of Technology and Applied Sciences, 2020.
- [3] YINGFENG CAI, et al., "CNN-LSTM Driving Style Classification Model Based on Driver Operation Time Series Data", IEEE ACCESS – Open Access Journal, February 2023.
- [4] Teck Kai Chan, Cheng Siong Chin, Senior Member, IEEE, Hao Chen, Member, IEEE, and Xionghu Zhong, Member, IEEE, "A Comprehensive Review of Driver Behavior Analysis Utilizing Smartphones", IEEE Transactions On Intelligent Transportation Systems, 2019.
- [5] Shuo Jia<sup>1</sup>, Fei Hui<sup>1</sup>, Shining Li<sup>1</sup>, Xiangmo Zhao<sup>1</sup>, Asad J. Khattak<sup>1,2</sup>, "IET Intelligent Transport Systems", July 2019.
- [6] Shi-Huang Chen, Jeng-Shyang Pan, and Kaixuan Lu, "Driving Behavior Analysis Based on Vehicle OBD Information and AdaBoost Algorithms", Proceedings of the International Multi Conference of Engineers and Computer Scientists, Vol I, March 2015, HongKong.
- [7] Alexey Kashevnik, Igor Lashkov, and Andrei Gurtov, Valley, "Methodology and Mobile Application for Driver Behavior Analysis and Accident Prevention", Ieee Transactions On Intelligent Transportation Systems.
- [8] Malintha Amarasinghe<sup>1</sup>, Sasikala Kottegoda<sup>2</sup>, Asiri Liyana Arachchi<sup>3</sup>, Shashika Muramudalige<sup>4</sup>, H. M. N. Dilum Bandara<sup>5</sup>, and Afkham Azeez, "Cloud-Based Driver Monitoring and Vehicle Diagnostic with OBD2 Telematics", International Conference on Advances in ICT for Emerging Regions (ICTer), 2015.
- [9] Meenakshi Malik, Rainu Nandal, "A framework on driving behavior and pattern using On-Board diagnostics (OBD-II) tool", Department of Computer Science and Engineering, U.I.E.T, Maharshi Dayanand University, Rohtak 124001, India.
- [10] SOOKSAN PANICHPAPIBOON, (Senior Member, IEEE), AND PUTTIPONG LEAKKAW, "Lane Change Detection With Smartphones: A Steering Wheel-Based Approach", Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand, May 2020.
- [11] Yu-Jen Pan<sup>1, a</sup>, Te-Cheng Yu<sup>1, b</sup> and Rung-Shiang Cheng<sup>2, c</sup>, "Using OBD-II Data to Explore Driving Behavior Model", IEEE International Conference on Applied System Innovation, 2017.
- [12] Aurélie Pirayre, Pierre Michel, Sol Selene Rodriguez, and Alexandre Chasse, "Driving Behavior Identification and Real-World Fuel Consumption Estimation With Crowdsensing Data", Ieee Transactions On Intelligent Transportation Systems, VOL. 23, NO. 10, October 2022.