Smart System for Driver Behavior Prediction

Hajar LAZAR, Zahi JARIR
Computer Science Engineering Laboratory, Cadi Ayyad University, Marrakesh, Morocco

***Abstract*—Driver behavior has recently emerged as a challenging topic in Traffic risk studies. Despite the advances in this topic, the challenges still remain. In fact, the current contribution deals with predicting at Real-time driver behavior based on machine learning techniques handling data sensing collected from smartphone sensors (accelerometer, gyroscope, GPS) and from OBD II. To ensure prediction at real time, we used a real-time architecture utilizing Atlas MongoDB service to synchronize data communication. Furthermore, we opt Random Forest model that demonstrates the highest performance compared to other models. This model has the advantage of predicting and preventing by warning a driver if his or her driving style is aggressive, moderate or slow. The proposed system aims to give more information about incidents to gain a better understanding of their causes.**

***Keywords*—Driver behavior prediction; OBD II; smartphone sensors; intelligent transport system; traffic safety**

I. Introduction

Traffic accidents have been on the rise over the past decade and have turned into a serious problem that affects the safety of people and their property. The World Health Organization report states [1] that pedestrians, cyclists, and motorcyclists are responsible for more than 50% of all road traffic deaths and, every year, road traffic crashes result in approximately 13.5 lakhs of casualties [2]. Due to its specific vehicular composition, Marrakech, a popular city in Morocco, poses a big challenge for traffic flow and safety studies. The traffic in the city is made up of roughly 40% motorcycles and 60% cars, which has a significant impact on traffic dynamics and safety patterns. The mix includes various factors, such as vehicle volumes, road conditions, and traffic control measures. The traffic system in Marrakech has a unique set of challenges because of the high number of motorcycles. In fact, Marrakech's traffic is unique and presents big challenges that have a significant impact on traffic flow and safety [3].

In recent years, there has been a proliferation of studies about traffic safety. Prediction of traffic accidents can help to prevent crashes, avoiding damage, providing drivers with warning alerts to potential risks or identifying potential causes and precautionary measures to prevent crashes from occurring [4]. The purpose of traffic risk studies is to identify crucial factors for future planning scenarios and to examine potential factors that may have adverse effects. According to research, most accidents occur due to driving behavior.

The author in study [5] identified multiple aspects of driver behavior from the same viewpoint and separated two aspects: (1) the research focused on individual psychological determinants as the basis for risky traffic behavior and its predictor, and (2) investigating the specific aspects of traffic climate that relate to social psychological determinants. In addition, they underlined the importance of focusing on studying driver behavior models: (a) The content aspect considers the limits of driver behavior and their distinctions from aggressive or dangerous driving, (b) the structural aspect involves identifying risky road behavior by examining the relationships between various behavioral manifestations of patterns in a wide social context, and (c) the dynamic aspect that identifies the behavior change in future [2].

Through various studies and research in psychology, it has come to light that rewarding desirable behavior can lead to a faster and longer-lasting change in human behavior than punishing undesirable behavior. Research has indicated that rewarding road safety behavior can be helpful, as evidenced using seatbelts and keeping driving speeds within permissible limits. However, a proper reward system is necessary for optimal effect. The authors in the study [6] proposed a system that constantly monitors drivers and uses real-time data to give them an appropriate score, provide feedback accordingly, and help punish errant drivers. They utilized different measured human factors as inputs to their smart algorithm to give an appropriate score to the driver based on their efficiency. Due to that, the major causes of road accidents and global warming are attributed to human factors. In this paper, we proposed an approach which considers human factors that can be analyzed with a smartphone sensor, a car and an OBD-II adapter. The measured human factors are given as inputs to a smart algorithm that delivers an appropriate score to the driver based on his efficiency.

The remainder of the paper is organized as follows: Section II provides an overview of the relevant literature. In Section III, we proposed our methodology in which we presented the data acquisition and data analysis. Section IV compared the different machine learning models and approved the choice of appropriate ML model. Section V covers the experimental and simulation results. Finally, Section VI summarize the paper and discusses future works.

II. Related Work

To evaluate road safety measures based on accident frequency and severity, several researchers have studied the safety effects of different road safety measures [7]. As a result, relevant techniques and methodologies have been developed to detect road accidents and identify critical factors that can cause crashes.

Nericell is a system proposed by study [8] that leverages smartphones to identify various vehicle conditions, such as braking, road bumps, honking, and stop-and-go traffic. It utilizes the smartphone's accelerometer, microphone, GSM communications, and GPS for data collection. The information gathered from multiple smartphones is aggregated on a

centralized server. The detection of such events does not involve the use of machine learning (ML) algorithms. The authors used the pattern matching and orientation calibration to detect the various vehicle situations. Similarly, the mobile sensor-platform for intelligent recognition of aggressive driving (MIROAD) is a driver behavior monitoring system developed by Johnson and Trivedi [9]. This system relies entirely on a smartphone's internal accelerometer, gyroscope, magnetometer, and GPS. They were the first to introduce a more advanced pattern recognition approach. MIROAD-equipped smartphones can detect and classify various aggressive and non-aggressive driving maneuvers using the variations of Dynamic Time Warping (DTW). The DTW is an algorithm to find similar patterns in temporal series. They don't introduce the machine learning models to classify the aggressive and non-aggressive behavior. Likewise, the system proposed by study [10] evaluates a person's driving as either safe or risky. It identifies risky events such as sudden maneuvers, turns, lane departures, braking, and acceleration using only the accelerometer, gyroscope, and magnetometer of a smartphone. It uses an endpoint detection algorithm to identify events and a DTW algorithm to compare input data with template events. Additionally, they incorporate a feature does not present in Johnson and Trivedi's system [9] which is the labeling of driver behavior. A Bayesian classifier labels the driver's behavior as safe or risky based on a calculated probability. Alternatively, the authors in study [11] developed and built a vehicle integrated system to inform drivers about the quality of their driving. They provided a safety index to each driver which indicating their ability to drive safely over prolonged periods. To reduce accidents, the safety index generated can be utilized to reward or retire the drivers. They based on the acceleration, speed and road traffic conditions parameters to indicate the Safety Index. For that, they utilized smartphones and high-end mobile devices having all the sensors to access data to develop an integrated application to track driver's performance. The machine learning models are not involved in their contribution. The authors in study [12] proposed a machine learning model based on set of rules to classify the driving maneuvers from time series data. They categorized the driver behavior as aggressive acceleration, non-aggressive, aggressive right turn, aggressive left turn, aggressive right lane change, and aggressive breaking aggressive left lane change. In fact, they utilized the accelerometer to collect the traffic data during longitudinal, lateral movement and the gyroscope for angular movement.

Recently, authors in study [13] proposed a classification model for drivers' behaviors. The dataset was built from recording OBD II (On Board Diagnostic II) parameters. The recorded data were analyzed and calculated to derived additional driving metrics such as Avg_fuel, Idle_engine, High_speed_breaking, Revv_engine, Dev_str, Vs_dev, Idle_instance, Avg_gear, Hb_instances, Avg_speed, Rev_instances. In this research, they employed the machine learning techniques, like SVM, AdaBoost, and Random Forest to classify driver behavior into ten classes: poorest, poor, bad, belong average, average, good, above average, extremely good, excellent, very good.

In our approach, we classify driving behavior into one of the following classes: aggressive, moderate or slow behavior to overcome the shortcomings of previous research that has focused only on aggressive behavior in various driving events. However, they have not addressed the other cases related to normal or slow behaviors. Additionally, we emphasize that slow driving can have a significant negative impact on urban traffic. A driver moving significantly slower than the flow of traffic may cause frustration for other drivers, leading to sudden lane changes or tailgating. Moreover, to accurately identify moderate and slow driving behaviors, it is essential to incorporate additional data features provided by OBD II adapter such as average speed, acceleration, throttle position, engine revolutions, and the vehicle's geographic location. These data features are crucial for a more comprehensive analysis of driving patterns.

Some significant points have been identified shown in Table I outlined here:

a) All papers use sensor data as input for the event detection algorithms, except [13] which used OBD II to create their dataset.

b) Some of them doesn't utilize the machine learning for classification of driver behaviors.

c) The majority of the research results in analyzing data using either mobile sensors such as accelerometer, gyroscope, magnetometer, GPS or OBD II.

TABLE I. COMPARISON BETWEEN EXISTING DRIVING BEHAVIOR STUDIES

Existing works	Types of Datasets	Machine learning models	Driver behavior Classes	Driver feedback application
[8]	smartphone's accelerometer, microphone, GSM communications, and GPS	Not considered	Not considered	Not developed
[9]	accelerometer, gyroscope, magnetometer, GPS and video	Not considered	non-aggressive and aggressive.	Not developed
[10]	accelerometer, gyroscope and the magnetometer	Bayesian classifier	safe or risky	Not developed
[11]	smartphones, high-end mobile devices and videos	Not considered	Not considered	Not developed
[12]	Accelerometer, gyroscope and the magnetometer	unsupervised learning technique using the sequential covering algorithm and classifying driving maneuvers from time-series data	aggressive acceleration, non-aggressive, aggressive right turn, aggressive left turn, aggressive right lane change, and aggressive breaking aggressive left lane change	Not developed
[13]	OBD II adapter and derived additional driving metrics	SVM, AdaBoost, and Random Forest	poorest, poor, bad, belong average, average, good, above average, extremely good, excellent, very good	Not developed

We deducted that there is no interest in combining smartphone sensors and OBD II features. By combining sensors and OBD II data systems offers more accuracy to classify driving behavior (aggressive, moderate, or slow) and capture the nuances of driver actions. This integrated approach enhances the ability to detect risky driving patterns and improves safety systems by offering a real-time, comprehensive analysis of driving habits.

The purpose of this paper is to present a system having advantage to predict and prevent by warning a driver if his or her driving style is aggressive, moderate or slow and to give information about incidents to gain a better understanding of their causes. Therefore, the proposed system determines the drivers with aggressive labels in localized areas to provide insight into where accidents are more likely to happen.

III. Methodology

Unfortunately, up to now the research involved the driving behavior factor for safety road are not widely studied in the literature. In fact, we proposed a road driving awareness system that focuses on driver behavior as a key factor in reducing road accidents and congestion problems in Morocco, particularly in Marrakesh city. Furthermore, road accidents in Morocco lead to over 4,000 deaths annually and approximately 140 million Dirhams (equivalent to about $14 million) of property damage [14]. During our analysis of the traffic road in Marrakech city, we observed that the behavior of drivers is influenced by various parameters, including infrastructure, vehicle performance, fatigue, and overspeed and we pointed out that motorcycles are the major factor that influences road accidents and congestion. Due to that, we propose a system that can predict various road problems by gathering the driver behaviour in real time. We began our system's development by collecting sensors and OBD II data, analyzing the effects of various features, and using machine learning algorithms to classify the provided data to visualize driver behavior in real-time, which led us to propose an awareness system.

A. Data Acquisition

By integrating data from smartphone sensors and OBD II adapter, driving behavior can be classified more precisely (aggressive, moderate, or slow), allowing for a deeper understanding of driver actions. This combined approach improves the detection of risky driving patterns and enhances safety systems by providing a comprehensive, and real-time analysis of driving habits. Each sensor provides unique data that, when combined, creates a more detailed and accurate picture of a driver's actions. These sensors are:

1) Accelerometer: detects rapid changes in speed, such as hard braking or sudden acceleration, key indicators of aggressive driving behavior (Fig. 1).

2) Gyroscope: measures rotational movement, helping to identify sharp turns or swerving, which can also indicate aggressive or unsafe driving (Fig. 1).

3) Magnetometer: used to determine the vehicle's orientation relative to the earth's magnetic field, aiding in directional awareness.

4) GPS: provides real-time location data, enabling the analysis of driving speed and route choice. It also helps assess whether a driver is adhering to speed limits and can indicate potential traffic conditions.

5) OBD-II: captures vehicle-specific data like throttle position, engine RPM, and fuel consumption, which provide insights into how efficiently the vehicle is being driven.

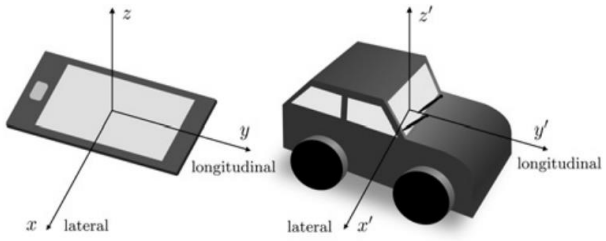

Fig. 1. Accelerometer and gyroscope sensors with vehicle coordinate system.

In our approach, we interest to label the driver behavior in three classes aggressive, moderate and slow behavior. For aggressive behavior, we collected the data set from accelerometer and gyroscope sensors to detect sudden maneuvers and sudden brakes. For moderate behavior, we need some additional features from OBD II like average speed, engine revolutions, throttle to detect average driving maneuvers. In similar way, we classify driver behavior as slow when the driver is maintaining a lower-than-average speed. To detect irregularities in the car, we must connect the OBD functions to a DLC (data link connector). The DLC is a collection of codes that make it easier to detect the work of sensors. The primary function of the OBD system is to monitor the functioning of essential engine components, such as those responsible for regulating emissions and detecting engine defects. Both hardware and firmware are crucial to extracting data from the OBD-II system. The ELM327 Bluetooth-based multi-protocol device located at the core of the system is compatible with the OBD-II protocols as shown in Fig. 2.

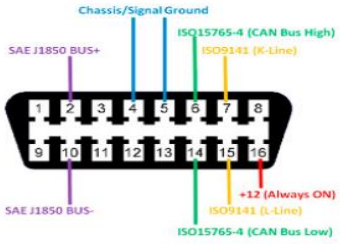

Fig. 2. OBD II system.

B. Data Analysis

To analyze the data, live vehicle data was necessary. It was collected from the Clio 4 vehicle, which was driven by same driver in Marrakesh city way. Fig. 3 illustrates the route utilized for gathering the dataset in a single direction, represented in black colored segment starting from the same point. We selected thoughtfully to simulate urban driving conditions. Therefore, we opted for a route that would enable us to collect the most relevant data possible. It was collected during crucial times, which were

chosen to reflect the different traffic conditions that are commonly observed throughout the day. The peak hours for rush are at 8:00 AM to 8:30 AM, midday from 12:00 PM to 12:30 PM, and evening from 16:30 PM to 17:00 PM. The OBD-II recorded data every second while we were driving and collected 22 features that are average fuel consumption, average speed, average speed (GPS), calculated boost, calculated instant fuel consumption, distance travelled, engine RPM x1000, fuel used, fuel used price, instant fuel power, vehicle acceleration, OBD module voltage, Calculated engine load value, engine coolant temperature, intake manifold absolute pressure, engine RPM, vehicle speed, intake air temperature, throttle position, fuel rail press, speed (GPS), altitude (GPS). At the same time, we collected the longitudinal and lateral movement from accelerometer as acc_x, acc_y, and angular movement from gyroscope gyr_z, dataset using a Samsung Galaxy A31 smartphone which is positioned at the center of a car's windshield. Table II presents the various feature categories along with their descriptions.

Fig. 3. Itinerary chosen for data collection in Marrakesh, Morocco.

TABLE II. THE OBD AND SENSORS FEATURES DESCRIPTION

Category	Features	Description
OBD II	Average speed	The average speed of the vehicle over a period of time, calculated by dividing the total distance traveled by the total time taken
	Average fuel consumption	The average rate of fuel usage, typically measured in liters per 100 kilometers (L/100 km), indicating how efficiently the vehicle uses fuel
	engine RPM x1000	The revolutions per minute of the engine, divided by 1000. It indicates how fast the engine's crankshaft is rotating.
	Distance travelled	The total distance covered by the vehicle during a trip, typically measured in kilometers
	Average speed (GPS)	The average speed of the vehicle as determined by the GPS system. It may differ from the calculated average speed due to factors like GPS signal accuracy
	vehicle acceleration	The rate at which the vehicle's speed increases, typically measured in meters per second squared (m/s²)
	OBD module voltage	The voltage level of the OBD module, typically corresponding to the vehicle's battery or alternator voltage
	Calculated engine load value	The calculated value representing the percentage of the engine's maximum capability being used. It provides an idea of how hard the engine is working
	Engine coolant temperature	The temperature of the engine's coolant, indicating how hot the engine is running.
	Intake manifold absolute pressure	The pressure inside the intake manifold, typically measured in kilopascals (kPa) or bar, used to calculate engine load and airflow
	Engine RPM	The number of revolutions per minute (RPM) at which the engine's crankshaft rotates, indicating engine speed
	Vehicle speed	The current speed of the vehicle, usually measured in kilometers per hour (km/h)
	Intake air temperature	The temperature of the air entering the engine, which can affect performance and fuel efficiency
	Throttle position	The position of the throttle valve, typically measured as a percentage from fully closed (0%) to fully open (100%), indicating how much air is being allowed into the engine
	Fuel rail press	The pressure of the fuel in the fuel rail, typically measured in kilopascals (kPa), indicating how much fuel pressure is being supplied to the injectors
	Speed (GPS)	The real-time speed of the vehicle as measured by GPS, typically more accurate than the speedometer reading
	Altitude (GPS)	The vehicle's altitude above sea level as measured by GPS, typically in meters
Accelerometer	acc_x	It measures the rate of change in velocity along the **x-axis**. This typically corresponds to motion in the left-right direction
	acc_y	It measures the acceleration along the **y-axis**, capturing vertical motion (up and down). It detects forces like gravity or upward/downward movement in space
	acc_z	It measures the acceleration along the **z-axis**, typically in the forward-backward direction. This axis is perpendicular to both the x and y axes, capturing motion like moving a device toward or away from a surface
Gyroscope	gyr_x	It measures the angular velocity around the **x-axis**. This represents the rate of rotation around the horizontal axis that runs from left to right
	gyr_y	It measures the angular velocity around the **y-axis**. This axis runs from front to back, so the gyroscope detects rotations in the left or right direction (side-to-side tilting)
	gyr_z	Measures the angular velocity around the **z-axis**. The z-axis runs vertically through the device (up-down direction), it detects rotational movements like spinning or turning the device around this vertical axis

Both datasets were merged by data acquisition module developed on mobile application and sent in real time for storing in NoSQL database. In this work, we assume that a driving behavior can be labeled into one of the classes described in the following subsections.

a) Aggressive behavior: As previously stated, we keep a gyroscope and accelerometer to recognize risky behavior. The accelerometer and gyroscope value are utilized during the trip to detect driving events like hard braking, sudden acceleration, or aggressive turning. To ensure non aggressive driving, it is recommended to use the recommended range of acceleration or deceleration values presented by [11][12][15]. For safe accelerating and braking the acc_x should be between -3 to 3 m/s². In contrast, the maneuvers such as left turn (LT), right turn (RT) influence the lateral and angular changes in a motion. Over the duration of safe LT and RT, acc_y between -1.5 to 1.5 m/s². Similarly, gyr_z perform between -0.4 to 0.4 m/s². In fact, we can identify aggressive driving, typically, when the threshold for aggressive acceleration rate greater than 2.5 to 3.0 m/s². This means a sudden increase in speed, indicating rapid acceleration. The threshold for aggressive braking rate (negative acceleration) greater than -2.5 to -3.0 m/s². In case of aggressive corning, the lateral acceleration can be greater than 1.5 m/s² and a yaw rate exceeding 15 to 20 degrees per second can indicate sharp turns. We summarized the different type of aggressiveness behavior in Table III.

TABLE III. CATEGORY OF AGGRESSIVENESS BEHAVIOR SEVERITY

Degree of Behavior aggressiveness	Type of Behavior aggressiveness
acc_x greater than -2.5 to -3.0 m/s	Braking with high speed
acc_x 2.5 to 3.0 m/s².	Sudden increase in speed
acc_y greater than 1.5 m/s and gyr_z exceed 15 to 20 degrees	Zig-zag with high speed

b) Moderate behavior: The proposed method considers a good average speed between 50 and 60 km/h. In addition, the optimal engine load indicates how much power the engine is producing relative to its maximum capacity. The normal driving rate its between 20% to 60% of maximum engine load.

c) Slow behavior: Driving the vehicle at a speed that is significantly less than the posted limits or traffic flow. Furthermore, being too cautious can result slow acceleration, early braking, and hesitancy at intersections which could lead to delays and hinder the flow of traffic.

Our research does not assume that driver behavior will remain consistent during the entire trip. Instead, we aim to detect varying actions taken by drivers during the same trip and classify them. In data analysis process, we used gradient boosting on decision trees (GBDT) models as XGBoost, Adaboost, Gradient boosting and decision tree to identify the three potential driving behavior classes. The GBDT algorithms gathers multiple weak learners, typically a decision tree, to create a stronger model that surpasses the base model. When boosting decision trees are added, they learn from the errors of previous individual trees. Each tree is connected in a series and tries to reduce the error of the previous tree. Boost algorithms may be slow to learn because of this sequential connection, but they are great at performing in the classification. In addition, we utilized random forest model which uses bagging techniques to create multiple decision trees using bootstrapped samples. The bagging method produces random samples with replacements from the input data and instructs decision trees based on the samples. A Random Forest classifier trains multiple decision trees on the same training set to enhance classification precision and combat overfitting [13][16].

IV. APPROPRIATE MACHINE LEARNING MODEL

One of the key steps is data preprocessing, which includes data clean-up and normalization. Then, we employed machine learning models for classification. We adopted in this paper XGBoost, AdaBoost, gradient boosting, decision trees, and random forests, to identify three potential driving behavior classes: aggressive, moderate, and slow. The classification techniques utilized here are outlined in the following subsections.

a) XGBoost model: XGBoost is increasingly used in the field of driver behavior analysis due to its effectiveness in handling complex, structured datasets and its ability to provide high predictive accuracy. It's a supervised algorithm that combines multiple base learners to enhance the performance of strong learners.

To analyze driver behavior, we train the model on sensors and OBD datasets $D_{sensors} = \{(X_i, y_i)\}$, $D_{obd} = \{(X_j, y_j)\}$ containing n samples and m decision variables. The driver behavior class y_i is predicted by the trained model. y_i is the target value corresponding to X_i . To understand the functions within the model, the learning objective function of XGBoost is defined as follows:

$$= \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (1)$$

where, l is a differentiable convex function that measures the discrepancy between the predicted classification value $\hat{y}_i$and the actual category y_i, this is known as the loss function. The Ω function represents the regularization term, which penalizes model complexity to help prevent overfitting.

b) AdaBoost model: AdaBoost algorithm, a widely used ensemble classifier. It works by sequentially training multiple base learners on different sets of training data. The base learners are generated sequentially. The first learner, L_1 is trained on a random subset of the data. The second learner, L_2 is trained using data where half are correctly classified by L_1 and the other half are misclassified by L_1 . The third learner, L_2 focuses on examples misclassified by both L_1 and L_2. The algorithm assigns a weight, w_1, to each training sample. In each iteration, these weights are updated, particularly increasing the weights for the misclassified samples, to help the learner focus on harder cases. The final classifier chosen in each iteration, h_r , is assigned an importance score, α_r , based on its mean squared error e_r, according to the formula:

$$\alpha_r = \frac{1}{2} \ln\left(\frac{1-e_r}{e_r}\right) \quad (2)$$

This process continues iteratively, and different base learners work together to reduce the classification error.

c) Gradient boosting model: Gradient Boosting Machine (GBM) is one of the most widely used supervised machine learning techniques. It was introduced by [17] and has been shown to be highly effective for various predictive tasks. While the training process of GBM, a function is used to distinguish between aggressive, moderate and slow driving behavior. The dataset contains O observations, each characterized by features. The training set is represented as X_i, where related labels y_i with $y_i \in \{0,1,2\}$ where 0 indicates an aggressive behavior, 1 indicates a moderate behavior and 2 indicates a slow behavior.

The goal is to minimize the aggregated loss function through multiclass function estimation. This is typically expressed as an extension of binary classification, where multiple functions are estimated, one for each class. Instead of estimating a single function, the model estimates a separate function for each class label. This is done using the following formula:

$$P(y = N|X_i) = \frac{\exp(f_N(X_i))}{\sum_{j=1}^{N} \exp(f_j(X_i))} \quad (3)$$

Where $f_N(X_i)$ is the function learned for class N at input X_i , $P(y = N|X_i)$ is the predicted probability that the instance belongs to class N, exp denotes the exponential function.

d) Random forest model: Random forest (RF) model utilizes a method called bootstrap aggregating (also known as bagging) to create multiple decision trees and enhance model performance. Bagging involves selecting random subsets of features to determine the best split points in the tree-growing process. For classification tasks, RF bases the final prediction on a majority vote across all trees. The RF predictor function is represented as:

$$f(x) = \frac{1}{K} \sum_{i=1}^{K} T(x) \quad (4)$$

where K is the number of trees, and T(x) represents the individual tree predictions. RF reduces the correlation between trees by using bagging, which creates diversity among trees by drawing different data subsets with replacement. This resampling method allows some data points to be reused multiple times, while others might not be used at all. The diversity enhances robustness to slight variations in input data.

When growing trees, RF selects the best split point from a randomly chosen subset of features at each node. This decreases the strength of individual trees but reduces their correlation, ultimately improving generalization.

e) Decision trees model: The decision tree algorithm is trained using labeled data where each driving behavior is tagged. The tree is constructed by recursively splitting the data based on the most informative features. At each node of the tree, a specific feature is used to split the data, resulting in child nodes representing subsets of the data with more homogeneous driving behaviors. This process continues until the data is sufficiently categorized, or a stopping criterion (such as maximum depth or minimum samples) is met. At each node, the best feature $X_i$and threshold θ are selected to maximize the information gain or minimize Gini impurity. The Gini impurity formula used to assess the purity of a node is:

$$Gini(T) = 1 - \sum_{i=1}^{N} p_i^2 \quad (5)$$

Where N is the number of classes (e.g., "aggressive", "moderate", "slow") and p_i is the proportion of data points in class i at node T.

To assess the efficiency of driver behavior classification, we compare the performances of XGBoost, AdaBoost, Gradient boosting, Decision tree, Random Forest using the following metrics:

- Macro Accuracy: accuracy is the ratio of correctly predicted instances (both positive and negative) over the total number of instances. In multi-class problems, macro accuracy is the overall accuracy across all classes, treating each class equally.

$$\text{Average Accuracy} = \frac{1}{N} \sum_{i=0}^{N} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \quad (6)$$

- Macro Precision: precision measures how many of the predicted positive instances were true positive. It's the ratio of true positives (TP) to the sum of true positives and false positives (FP). For multi-class classification, macro precision is the unweighted average precision across all classes.

$$\text{Macro Precision} = \frac{1}{N} \sum_{i=0}^{N} \frac{TP_i}{TP_i + FP_i} \quad (7)$$

- Macro recall is used to evaluate the performance of a classification model, particularly in multiclass problems. It measures the model's ability to correctly identify positive instances for each class and then averages these recalls without considering class sizes.

$$\text{Macro Recall} = \sum_{i=0}^{N} \text{Recall}_i = \frac{1}{N} \sum_{i=0}^{N} \frac{TP_i}{TP_i + FN_i} \quad (8)$$

- The macro F1-score combines the concepts of precision and recall into a single score, providing a balanced measure of a model's performance across all classes.

$$\text{Macro F1} - \text{score} = \frac{1}{N} \sum_{i=0}^{N} \frac{\text{Precision}_i \, \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad (9)$$

Where N is behavior classes number which means the three classes in our study, and TP_i, FP_i, TN_i, and FN_i are the true positives, false positives, true negatives, and false negatives. Macro-averaging approach is employed to solve the problem of imbalanced data when calculating accuracy, recall, precision, and F1-score.

We divided our dataset into 70% of training and 30% of testing. Tables IV and V presented the comparison of different machine learnings algorithms based on sensors and OBD II data, respectively, using the metrics described above. We deduct that Random Forest model has achieved an accuracy of 96% and 99% in both sensors and OBD testing sets. The Random Forest

model demonstrated his efficiency in multi classification tasks. In fact, we chose it as the appropriate machine learning model to predict the three driver behavior classes already proposed.

TABLE IV. PERFORMANCE OF THE DIFFERENT MACHINE LEARNINGS MODELS USING SENSORS DATABASE

Metrics	ML models				
	XGBoost	*AdaBoost*	*Gradient Boosting*	*Decision Tree*	*Random forest*
Average Accuracy	92%	87%	88%	90%	96%
Macro Precision	91%	72%	90%	79%	97%
Macro Recall	72%	52%	54%	78%	82%
Macro F1-score	78%	51%	54%	79%	88%

TABLE V. PERFORMANCE OF THE DIFFERENT MACHINE LEARNINGS MODELS USING OBD DATABASE

Metrics	ML models				
	XGBoost	*AdaBoost*	*Gradient Boosting*	*Decision Tree*	*Random forest*
Average Accuracy	99%	96%	97%	98%	99%
Macro Precision	95%	87%	91%	80%	98%
Macro Recall	82%	84%	70%	77%	86%
Macro F1-score	88%	85%	79%	78%	92%

V. SIMULATION

Fig. 4 illustrates an overall architecture for visualizing driver behavior from data acquisition to classification and driver behavior prediction in real-time such as aggressive, moderate or slow. This architecture is based on three principal keys: (1) Sensing required data from driver's environment and context, (2) Analysing and labeling the driver's behavior class based on the gathered data, and (3) Visualizing the obtained label on driver's mobile application. These main steps are presented in more detail in the following subsections.

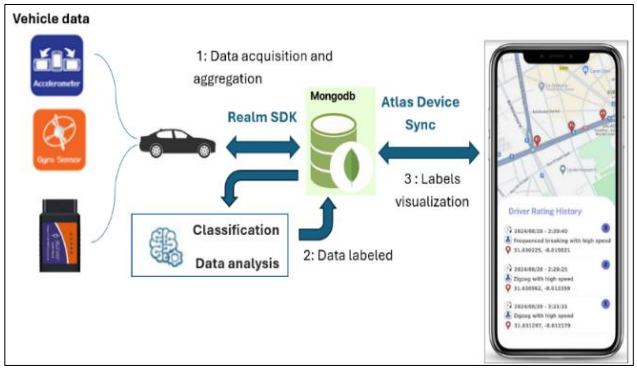

Fig. 4. Smart system for driver behavior prediction.

C. Data Acquisition and Aggregation

As mentioned earlier, the driver behavior prediction objective needs to collect various features that can help the system to categorize driver behavior into three classes: aggressive, moderate, and slow. These features are gathered both from the driver's mobile phone and from an OBDII that is connected to the driver's car and his mobile phone. The collected data related these features are concatenated in a single row json to be stored at real-time in a NoSql Mongodb database. Then the inserted data will be analysed by the chosen ML model which is Random Forest. To meet the challenge of communicating at real-time these data to Mongodb database, we used the following technologies:

- MongoDB Realm which is a lightweight database,
- Realm SDKs to synchronise in real-time these data,
- MongoDB Atlas that provides cloud-hosted managed instances of MongoDB for availability.

MongoDB Realm enables synchronisation between Realm Database and MongoDB Atlas, seamlessly stitching together the two components into an application layer for mobile application. This advantage comes from the fact that the core of Atlas MongoDB service is based on call rest API and push services.

D. Data Labeled

Once the data is stored in MongoDB database, Atlas trigger fires the module "Classification Data Analysis" implementing random forest algorithm to predict the corresponding driver behavior. The obtained label will then be updated in Mongodb database.

The random forest model read the row recently inserted in Mongodb. It analyses first the value that comes from mobile sensors to predict if the driver's behavior was aggressive or not. If the detected behaviour is aggressive, the algorithm is considered finished. Otherwise, the algorithm starts the second process by reading the OBD data to predict if the driver's behavior is moderate or slow.

Once the class is determined by the random forest model, the obtained label is then updated in MongoDB database.

The detail of the implemented algorithm is presented as follows:

Algorithm 1: Prediction Driver behavior

1. **Input**: Each Last added Row from MongoDB at real-time
2. {dataOBD, dataSensors}
3. dataOBD: Data related to OBD columns
4. dataSensors: Data related to Sensor's columns
5. **Output**: The variable "Behavior": {Agg, Slow, Moderate}
6. **Begin**
7. # **First step** – Prediction of aggressive behavior case
8. Behavior = Prediction of the behavior related to dataSensors
9. # **Second step** – Prediction of Moderate or Slow behavior cases
10. If (Behavior! = Agg)
11. Behavior = Prediction of the behavior related to dataOBD
12. # Return the result Aggressive, Moderate or Slow
13. Return Behavior
14. **End**

E. Labels Visualization

The objective of this application is not only to provide driver with valuable feedback about their driver behavior but in addition, to motivate them to be more attentive and conscious of their actions. The application was built using Flutter technology, a well-known open-source framework for designing robust and multi-platform mobile applications. Users can receive real-time updates and access to the most up-to-date information thanks to its seamless integration with APIs. Flutter technology is connected to Atlas Device Sync in order get the updated rows in MongoDB done by the classification ML model module (step 2). The map is used to display pertinent information about driver behavior in real-time using machine learning algorithms, and each color represents a different type of driver behavior. An example of driver interface is shown in the Fig. 5.

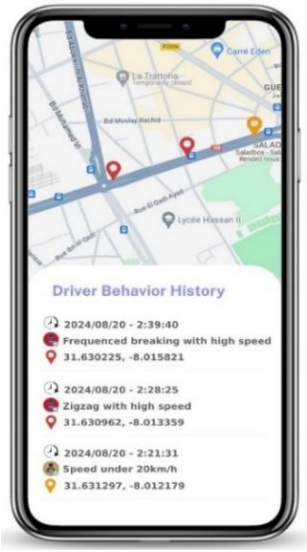

Fig. 5. Driver behavior history application.

VI. Conclusion and Future Work

Driving behavior is a crucial problem that affect the driver safety. To overcome the problem, we proposed a mobile application to aware the driver about his/her driving behavior styles. In this contribution, we developed a smart system to predict driver behavior in real-time utilizing machine learning model and the data collected from smartphone sensors and OBD II installed in a car. This system has the advantage to display on the map the relevant information about driver behavior predicted by machine learning algorithm in real time by assigning different color in which identifies a different type of driver behavior. Simulation have shown the ability of random forest model to predict three classes: aggressive, slow, moderate in both OBD and sensors data to achieve an accuracy of 92% and 88% respectively in Marrakesh city as case study. As a future work, our upcoming project involves developing a real-time recommendation system for safer routes based on driver behaviors. To establish safe routes through intersections, we propose an approach that considers the safety index of each road segment. This index will be calculated in real time, based on the safety indices of vehicles currently traveling on the respective road segment at any given moment. The safety index for each vehicle at time t will be computed in real time using the aggressiveness classification model (which distinguishes between aggressive and slow behaviors) and subsequently stored in a MongoDB repository.

References

[1] WHO/T. Pietrasik. 2020 https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[2] H. Lazar, Z. Jarir, "Road traffic accident prediction: a driving behavior approach". 8th International Conference on Optimization and Applications (ICOA), 1–4. 2022. doi: 10.1109/ICOA55659.2022.9934000

[3] A. Charef, Z. Jarir, and M. Quafafou. "The Impact of Motorcycle Positioning on Start-Up Lost Time: The Empirical Case Study of Signalized Intersections in Marrakech using VISSIM". Eng. Technol. Appl. Sci. Res., vol. 14, no. 3, pp. 14313–14318, Jun. 2024.

[4] I. J. Mrema and M. A. Dida, "A Survey of Road Accident Reporting and Driver's Behavior Awareness Systems: The Case of Tanzania", Eng. Technol. Appl. Sci. Res., vol. 10, no. 4, pp. 6009–6015, Aug. 2020.

[5] TV. Kochetova. "The Patterns of Drivers' Traffic Behavior: Evidence From Three Countries". Front Psychol. 2022 Apr 7;13:869029. doi: 10.3389/fpsyg.2022.869029. PMID: 35465507; PMCID: PMC9021888.

[6] S Khedkar, A Oswal, M Setty, S Ravi. "Driver Evaluation System Using Mobile Phone and OBD-II System". Int. J. Comput. Sci. Inf. Technol. Vol. 6 (3) , 2015.

[7] G. Yannis, A. Dragomanovits, A. Laiou, F. La Torre, L. Domenichini, T. Richter, S. Ruhl, D. Graham, and N. Karathodoro. "Road traffic accident prediction modelling: a literature review." Proceedings of the Institution of Civil Engineers - Transport 2017 170:5, 245-254 http://dx.doi.org/10.1680/jtran.16.00067

[8] P.Mohan, V.N.Padmanabhan, R.Ramjee,: "Nericell: rich monitoring of road and traffic conditions using mobile smartphones". Proc. of the Sixth ACM Conf. on Embedded Network Sensor Systems, 2008, pp. 323–336.

[9] D.A., Johnson, M.M.Trivedi. "Driving style recognition using a smartphone as a sensor platform". 14th Int. Conf. on Intelligent Transportation Systems (ITSC), 2011, pp. 1609–1615.

[10] H.Eren, S. Makinist, E.Akin, A.Yilmaz. "Estimating driving behavior by a smartphone'. Intelligent Vehicles Symp. (IV), 2012, pp. 234–239.

[11] S. Chigurupati, S. Polavarapu et al., "Integrated computing system for measuring driver safety index," Int. J. Emerg. Technol. Advanced Eng., vol. 2, no. 6, 2012.

[12] M. M.Haque, S. Sarker, M.A.A. Dewan. "Driving maneuver classification from time series data: a rule based machine learning approach". Appl Intell 52, 16900–16915. 2022. https://doi.org/10.1007/s10489-022-03328-3

[13] R.Kumar, A. Jain. "Driving behavior analysis and classification by vehicle OBD data using machine learning". The Journal of Supercomputing. 2023 Nov;79(16):18800-19.

[14] M. Benhadou, I. Chairi, and A. Lyhyaoui." Accident Prediction Model - Case Study Region of Tangier". Journal of Traffic and Logistics Engineering, 23-27. 2019.

[15] J. Ferreira, E. Carvalho, B, V. Ferreira, C. de Souza and Y. Suhara, A. Pentland, and G. Pessin.,"Driver behavior profiling: An investigation with different smartphone sensors and machine learning, PLoS one, 12(4), 2017.

[16] M.Domor, S.Yanxia.. A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. IEEE Access. PP. 1-1.2022. 10.1109/ACCESS.2022.3207287.

[17] Jerome H..Friedman."Greedy function approximation: A gradient boosting machine".Annals of Statistics 29 (2001): 1189-1232.