Smartphone Inertial Measurement Unit Data Features for Analyzing Driver Driving Behavior

Khadija Kanwal, Furqan Rustam, Rajeskhar Chaganti, Anca Delia Jurcut* and Imran Ashraf*

***Abstract*—Driving behavior is an important aspect of maintaining and sustaining safe transport on the roads. It also directly affects fuel consumption, traffic flow, public health, and air pollution along with psychology and personal mental health. For advanced driving assistance systems (ADAS) and autonomous vehicles, predicting driver behavior helps to facilitate interaction between ADAS and the human driver. Consequently, driver behavior prediction has emerged as an important research topic and has been investigated largely during the past few years. Often, the investigations are based on simulators and controlled environments. Driving behavior can be inferred using control actions, visual monitoring, and inertial measurement unit (IMU) data. This study leverages the IMU data recorded using a smartphone placed inside the vehicle. The dataset contains the accelerometer and gyroscope data recorded from the real traffic environment. Extensive experiments are performed regarding the use of a different set of features, the combination of original and derived features, and binary vs multi-class classification problems; a total of six scenarios are considered. Results reveal that 'timestamp' is the most important feature and using it with accelerometer and gyroscope features can lead to a 100% accuracy for driver behavior prediction. Without using the 'timestamp' feature, the number of wrong predictions for 'slow' and 'normal' classes is high due to the feature space overlap. Although derived features can help elevate the performance of the models, the models show inferior performance to that of using the 'timestamp' feature. Deep learning models tend to show poor performance than machine learning models where random forest and extreme gradient boosting machines show a 100% accuracy for multi-class classification.**

***Index Terms*—Driver behavior; autonomous vehicles; feature engineering; machine learning; inertial measurement unit**

I. Introduction

DRIVING behavior plays a vital role to maintain and sustain safe transport [1]. It directly affects fuel consumption, traffic flow, public health, and air pollution along with psychology and personal mental health. The 'driving behavior' or 'driving style' concepts have been defined by many researchers differently [2]. [3] defines driving behavior as, *driving style concerns individual driving habits that are, the way a driver chooses to drive*. The existing research work highlights the advantages to adopt safe and more environment-friendly driver behaviors for traffic circumstances, stress relief, emissions, and much more. Moreover, driving behavior has significant importance since particular driving behaviors are in considerable relation to traffic congestion, carbon emission, etc. [4]. Driving style is varying in the way drivers select to decelerate and accelerate the distance as drivers kept from the leading vehicle whatever they drive more than the speed limit [5].

"This research was supported by University College Dublin, Ireland."

Khadija Kanwal is with the Institute of Computer Science and Information Technology, The Women University Multan, Pakistan. (e-mail: khadijakanwal.6022@wum.edu.pk)

Furqan Rustam is with the School of Computer Science, University College Dublin, D04 V1W8 Dublin, Ireland. (furqan.rustam1@gmail.com).

Rajeskhar Chaganti is with the Toyota Research Institute, Los Altos, California, USA. (e-mail: raj.chaganti2@gmail.com)

Anca Delia Jurcut is with the School of Computer Science, University College Dublin, Ireland. (e-mail: anca.jurcut@ucd.ie)

Imran Ashraf is with the department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Korea (e-mail: imranashraf@ynu.ac.kr).

Asymmetric driving behavior means that the drivers are much more attentive in deceleration than in Accelerometer which is closely related to an eminent traffic hysteresis phenomenon [6]. Recently, three characteristics of asymmetric driving behavior are defined as

1) Hysteresis: the drivers are applied to keep a larger headway while Accelerometer than deceleration provided a similar speed [7],
2) Discrete driving: the accelerating and decelerating in car-following (CF) are not sequential [8],
3) Intensity difference: the positive and negative relative speeds are different though driving at the same condition for response intensity of the drivers such as the same speed, the similar magnitudes for relative speeds, and the same gap among the following and leading vehicles.

In addition, the driving behaviors in deceleration and acceleration vary when analyzing the next-generation simulation (NGSIM) data. In [9], experimental results of the presented research have reported that the reaction time in acceleration is different from that in deceleration. However, various driving profiles have been acknowledged in existing research regarding road traffic safety. In addition, aggressive driving has been comprehensively studied in many research articles that have focused to identify braking events and harsh acceleration [10].

Road transportation is commonly used to travel from one location to another location. The emergence of technologies such as the Internet of things (IoT), computer vision, wireless communication, and artificial intelligence enables smart transportation with advanced capabilities for safe traveling.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000

The wide adoption of advanced technology-enabled vehicles for transportation is still in progress. On the other hand, road accidents are not stopping soon. The World health organization (WHO) reported that more than one million people are killed and around fifty million people are injured by road accidents every year [11]. The road accidents trend is predicted to be increasing over the next few years and expected that road accident-based deaths becomes the fifth leading cause of death by 2030 [12]. The majority of road accidents are caused by human driving behavior. Although autonomous driving and advanced safety monitoring capabilities are incorporated into the vehicles, there is no guarantee that the driver is safe unless the driving behavior is normal. Driving behavior may have a direct impact on public health, traffic flow, air pollution, and environmental condition. So, there is a need to analyze driving behavioral patterns and understand individual driving habits so that safe driving recommendations can be provided to the users.

This study leverages the data from the inertial measurement unit of a smartphone that is placed in a car and predicts driving behavior into several categories. In this regard, this study makes the following key contributions

- Importance of feature selection from the inertial measurement unit data is investigated for the driver's driving behavior. The influence of using different original features derived features and the impact of binary vs multi-class classification is investigated in this study. For driver behavior prediction, six cases are considered including binary classification, accelerometer features alone, gyroscope features alone, accelerometer and gyroscope features combined, all features combined, and accelerometer and gyroscope features plus derived features without using the 'timestamp' feature.
- Extensive analysis of prediction performance regarding driver behavior is carried out using the data recorded in a real traffic environment. The dataset is recorded using a smartphone placed in the vehicle in a fixed position and readings from the accelerometer and gyroscope are recorded.
- Experiments involve using five well-known machine learning models and two deep learning models. Such models include random forest (RF), extreme gradient boosting machine (XGBoost), support vector classifier (SVC), extra tree classifier (ETC), logistic regression (LR), long short-term memory (LSTM), and convolutional neural network (CNN). Performance is analyzed with several parameters like accuracy, precision, etc., in addition to, standard deviation and the number of correct and wrong predictions.

The rest of the paper is organized as follows: Section II discusses the state-of-the-art machine learning-based approaches to address the driving behavior prediction problem. Section III presents the proposed methodology to accurately predict driving behavior. Also, a description of the datasets is provided in the same section. Section IV includes the results from an extensive set of experiments for driving behavior, as well as a discussion of the results. Section V concludes the article.

II. Literature Review

Recent developments in driver assistance and autonomous vehicles led to a great deal of research and development. Consequently, a large body of literature can be found on different aspects related to driving. For example, the role of trajectory data and its critical applications for microscopic modeling has been discussed in detail in [13].

In the last few years, experimentation has been performed on openly-accessible trajectory datasets and reports have been published related to several traffic flow phenomena. In addition, comprehensive empirical analysis has been reported including traffic oscillations [14], traffic hysteresis [15], and heterogeneity [16]. In addition, various models have been presented for a better approximation of car lane-changing behavior [17] and the following behavior.

Conventionally, the trajectory data is collected using an image processing technique that is based on recorded videos from either fixed drones or cameras. Currently, driving datasets are getting attention due to the demand for autonomous vehicle (AV) technology. The main purpose is to comprehend the challenge of computer vision systems in a self-driving context. In addition, the vehicle-based techniques detect the vehicle operating parameters including changes in the steering, speed of the vehicle, acceleration, lane tracking, braking, and many more. On the other hand, driver-based techniques are based on devices that directly monitor the condition of the driver. Also, the driver-based techniques are the physical movements parameters like blink ratios and eye closure ratios, and facial expression tracking with video imaging methods. The most famous trajectory dataset is possibly the NGSIM database [18] which has a total duration of 150 minutes from fixed cameras at four different sites. Also, another famous dataset is the highD dataset which contains videos from camera-equipped drones and has a total duration of 16.5 hours at six locations on the highways of Germany [19]. The driving situations presented in highD and NGSIM are quite limited. The NGSIM dataset includes signalized intersection and highway driving scenarios. However, traffic lights used to control signals and interactions are slight and rare.

Currently, many new datasets concerning the vehicles at high-level automation are made available [20]. For example, Argo [21], KITTI [22], BDD100K [23], Lyft Level 5 AV [24], Waymo open [25] and nuScenes [22] contains the data for autonomous vehicles and similar driver assistance systems. These are related to Lyft Level 5 AV, AV, nuScenes, and Waymo open datasets and combine trajectories for AV and the human-driven from real-world traffic. Moreover, these datasets are mentioned as AV-oriented empirical datasets. Therefore, these datasets are mainly helpful for driving behavior research. In addition, these AV-oriented empirical databases are sophisticated which helps to understand complicated driving behaviors to understand and use by traffic flow researchers. Firstly, these datasets are combined using an array of sensors; For example, LiDAR, a novel sensor to record traffic flow, is used. Secondly, the dataset contains several sensors and is more sophisticated than the conventional dataset. It collected not only comprehensive information for the movement of autonomous vehicles but also

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000

a vast amount of information for all objects in the vicinity of the vehicle. Lastly, the format and structure of these datasets are not user-friendly. Moreover, BDD100K contains ten tasks namely lane detection, image tagging, drivable area segmentation, semantic segmentation, road object detection, instance segmentation, multi-object segmentation tracking, multi-object detection tracking, imitation learning, and domain adaptation.

The authors [26] performed a survey on driving behavior improvement using ML and DL models. The study revealed that the combination of sensors and intelligent methods improves the performance of driving behavior classification. The study [27] designed a driving behavior detection method for identifying rash drivers. The contributions in the paper mainly include the architectural aspects of a system to build the driving behavior identification, including the monitoring system. But, the authors did not evaluate the driving behavior using the ML and DL models.

The authors in [28] presented a two-level hierarchy classification of driver activity while driving. Five input features speed, longitudinal acceleration, lateral acceleration, pedal position, and yaw rate, are considered for testing the driving behavior classification. The driver's secondary task while driving is detected in the first level. Then, the different types of secondary tasks are categorized in the second level. The ML-based Decision tree achieved the best results with an accuracy of 99.8% to classify the driver's secondary tasks. The study [29] proposed a lightbgm model to detect abnormal driving behavior. The accelerometer and gyroscope sensor data are input features to predict driving behavior. The authors reported that lightbgm achieved 82% accuracy on the test dataset. The classification accuracy still needs to improve for better driving behavior detection.

The study [30] proposed a two-dimensional CNN technique to analyze the driving behavior. The sensor data such as acceleration, gravity, revolutions per minute, speed, and throttle are used as a feature to construct an input image. The output is classified into five types such as normal, aggressive, distracted, drowsy, and drunk driving using 2D CNN. The authors reported that the proposed method obtained good results in predicting driving behavior.

The study [31] explored multi-class gait classification with machine learning approaches including KNN, extreme learning machines (ELM), SVM, and multi-layer perceptron (MLP), and evaluated the performance for multi-class gait classification. The presented approach achieved the best results. The ELM is introduced to analyze the neuromuscular mechanics that is associated with the brain of patients suffering from multiple strokes and sclerosis. In addition, an artificial neural network (ANN) is applied to classify the human gait and its performance is compared with the ELM. A deep learning ensemble technique is used for human lower activities recognition to capture the learning process of bi-pedal robot locomotion in [32]. The long short-term memory (LSTM) and convolutional neural network (CNN) models are used to classify these activities. In [33], a multi-branch CNN-BiLSTM network is applied for automatic feature extraction from raw sensor data with minimum data preprocessing.

Predominantly, existing literature on driver behavior prediction is based on machine learning algorithms, however, the use of no-machine learning architectures is also observed. For example, [34] uses the hidden Markov model (HMM) and coupled HMM (CHMM) for driver behavior prediction. Combined with car and traffic data, promising results are obtained regarding different driver actions. It is believed that machine learning models are black boxes, and it is not clear how predictions are made from such trained models. Consequently, several studies prefer non-machine learning models. The study [35] leverages rule-based models for driver behavior prediction. These models maintain long-term coherence and are easy to interpret.

The study [36] utilizes an auto-regressive input-output hidden Markov model (AIO-HMM) for driver behavior prediction. The focus is especially placed on driver behavior at intersections and driver gaze and traffic light recognition are used for that purpose. Similarly, [37] determines aggressive driver behavior by using multivariate-temporal features and driver's intention using HMM.

The above-discussed research studies have several shortcomings. First, the datasets containing smartphone sensor data are not very well studied for analyzing driver-driving behavior. Second, although several studies utilize these datasets for driver behavior prediction, the impact of feature combination and machine learning techniques is not well covered in the literature. Third, the context of the dependency between the features and the prediction output is not explored very well. Last but most important, the driving behavior prediction accuracy can be improved for existing works. Keeping in view these research gaps, this study proposes a highly accurate, machine learning-based driving behavior solution with extensive performance analysis for driver behavior.

III. Proposed Methodology

In this section, we discuss the proposed methodology for driver behavior prediction. We used several machine learning models to predict driver behavior as 'slow', 'normal', or 'aggressive'. Figure 1 shows the flow of the proposed methodology. This study leverages deep learning models for driver-driving behavior prediction. The selection of deep learning models is based on the results reported in the existing literature. For example, LSTM and CNN models are commonly used on similar kinds of datasets as in [38] for human behavior prediction, and in [39] for human activity detection. Similarly, [40] used variants of CNN and LSTM for driver behavior detection.

First, we acquire the dataset from the Kaggle repository. The dataset consists of several samples related to three target classes 'slow', 'aggressive', and 'normal'. After acquiring the dataset, we find that dataset features are not correlated to target classes which does not help the machine learning models to achieve a significant accuracy. Feature engineering steps are included in our proposed methodology to improve the performance of machine learning models. In feature engineering, we generate new (derived) features using old features to train learning techniques. Data is split into training and testing subsets for training several machine learning models. We split

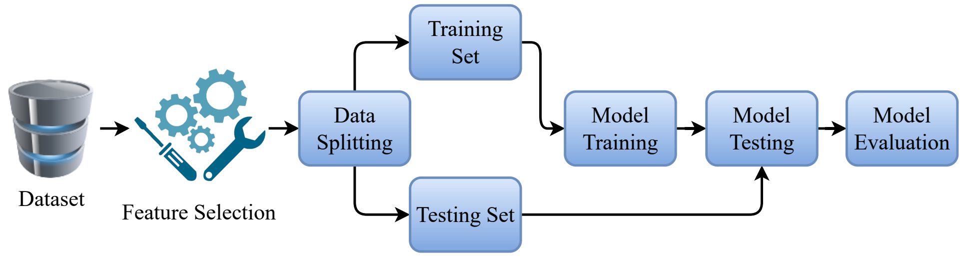


Fig. 1: Flow of the adopted methodology.

TABLE I: Dataset feature description.

Feature	Measure	Description	Count
Input			
Acceleration (X axis)	(m/s^2)	Linear acceleration towards X-axis	3084
Acceleration (Y axis)	(m/s^2)	Linear acceleration towards Y-axis	3084
Acceleration (Z axis)	(m/s^2)	Linear acceleration towards Z-axis	
Rotation (X axis)	degrees per second (°/s)	Rate of rotation towards X-axis	3084
Rotation (Y axis)	degrees per second (°/s)	Rate of rotation towards Y-axis	3084
Rotation (Z axis)	degrees per second (°/s)	Rate of rotation towards Z-axis	3084
Timestamp	seconds	timestamp when the event captured	3084
Output			
Class (SLOW)	-	Low risk driving	1273
Class (NORMAL)	-	Normal driving behavior	997
Class (AGGRESSIVE)	-	Unusual speeding, lane changes and turns	814

TABLE II: Samples from the dataset.

AccX	AccY	AccZ	GyroX	GyroY	GyroZ	Class	Timestamp
0	0	0	0.059407	-0.17471	0.101938	NORMAL	3581629
0.5503	-0.59792	-0.43771	0.03375	0.090408	0.006032	AGGRESSIVE	3582669
0.703766	-0.455	0.915689	-0.06582	0.089186	0.068341	SLOW	3583292

the dataset with an 80:20 ratio where 80% of the dataset is used for the training of models and 20% of the dataset is used for testing of models. In the end, testing and validation are performed. We evaluate all models in terms of accuracy, precision, recall, and F1 score.

A. Dataset Description

The mobile sensors generated driving behavior dataset is obtained from Kaggle [41]. The 'Sensorrecords' mobile application was used to capture the sensor data observations. This dataset is used by many recent studies [42]–[44]. The three dimensions of accelerometer and gyroscope sensor observations are mainly considered dataset features. The combination of accelerometer and gyroscope sensors helps to effectively track movement behavior. The accelerometer captures the linear acceleration along the axis, whereas the gyroscope captures the rate of rotation along the axis. The timestamp is also included as a feature in the dataset. The driving behavior dataset was collected using mobile sensing technology with accelerometer and gyroscope sensors enabled in the mobile when the user is driving the vehicle. TABLE I displays the input and output feature set along with measurement metrics and the dataset count. The driving behavior output is designated as 'normal', 'slow', and 'aggressive' driving. Normal driving behavior denotes that the driver maintains a constant speed and is aware of the surroundings. Slow driving may include low-risk driving behavior and essentially driving with fear or over conscious. The aggressive driving category includes unusual driving behavior with sudden breaks and accelerating the vehicles, unexpected lane-changing behavior, and unfocused driving due to eating, texting, etc. The dataset consists of 3084 samples with a different number of samples for driving behavior classes as slow with 1273 samples while normal and aggressive classes with 997 and 814 samples, respectively. The sample of the dataset is shown in TABLE II.

The original dataset consists of three accelerometer features, three gyroscope features, and a timestamp. These features are not much correlated to the target classes so to improve the accuracy of models we generate more features that are more correlated to the target classes. Figure 2 shows the sample values for accelerometer and gyroscope data for each of the three classes. We find that several values from the 'normal' and 'slow' target classes are similar which can create complexity for learning models to distinguish these targets based on sample values.

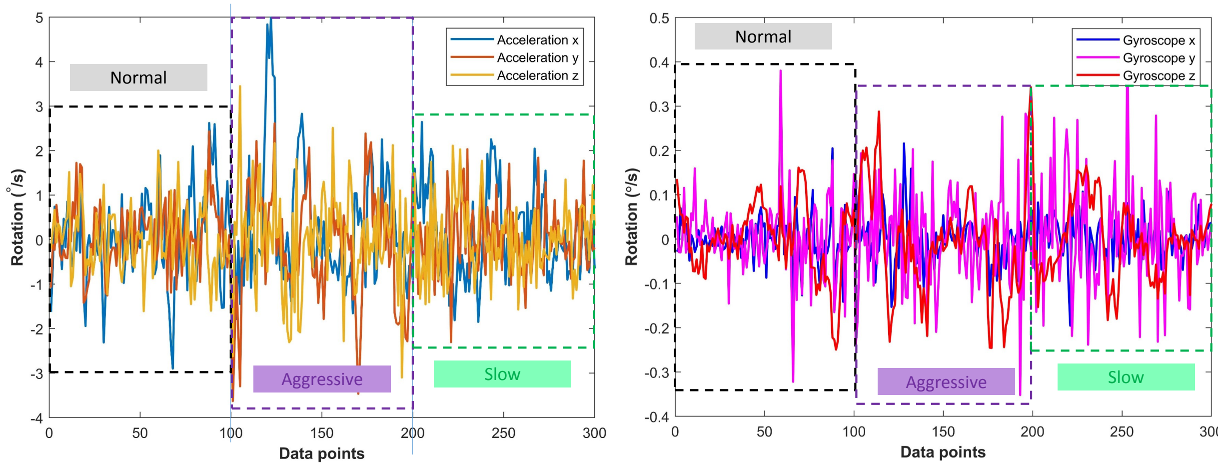
Along with the x, y, and z axes values for both the accelerometer and gyroscope, the dataset also contains a timestamp attribute. The histogram distribution of all these attributes is presented in Figure 3.

To analyze the feature correlation of these features, RF is used and the results are shown in Figure 4. It can be observed that features have different levels of correlation.

B. Feature Selection

We have seven features in the used dataset for driver behavior prediction. All features are not important for machine

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000



(a) (b)

Fig. 2: Sample data from the data for three classes, (a) Accelerometer data, and (b) Gyroscope data.

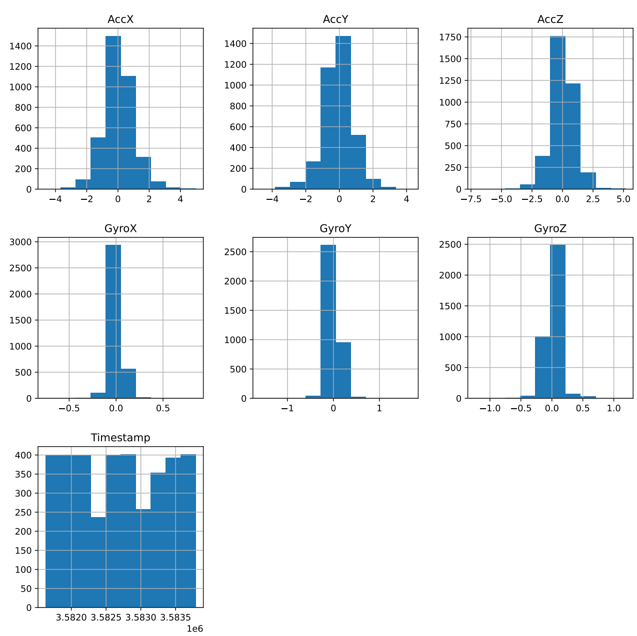


Fig. 3: Histogram distribution of all attributes of the data.

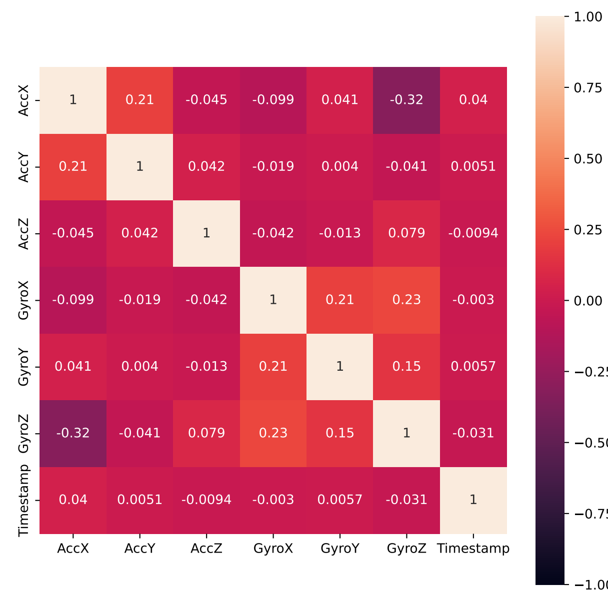


Fig. 4: Feature correlation using RF model.

learning models. So we make several scenarios/cases with feature selection.

- **Case 1:** Experiment with gyroscope features. In this case, we used only the gyroscope x, y, and z axes feature for model training.
- **Case 2:** Experiment with accelerometer features. This case considers only the accelerometer x, y, and z axes feature for model training.
- **Case 3:** Experiment without the 'timestamp' feature and binary target classes. In this case, we used both the gyroscope and accelerometer features and excluded the 'timestamp' feature. We also converted three target classes into two target classes for performance analysis and the feasibility of using the features to obtain higher accuracy. We find that several values for normal and slow classes are similar which creates complexity for the models. So we combined both target classes as one (SLOW + NORMAL= SLOW). In this way, we convert the multi-class problem into a binary class problem (SLOW, AGGRESSIVE).
- **Case 4:** Experiment without timestamp feature and three target classes. In this case, we used both the gyroscope and accelerometer features and excluded the timestamp feature. We used three target classes in this case (SLOW, NORMAL, AGGRESSIVE).
- **Case 5:** Experiment with the timestamp feature plus three target classes. In this case, we used all features (three features from the gyroscope, three features from the accelerometer, and the timestamp feature) for model training with three target classes (SLOW, NORMAL, AGGRESSIVE).
- **Case 6:** Experiment with new (derived) features and

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000

without the 'timestamp' feature for three classes. In this case, we used three gyroscope features, three accelerometer features, and four newly generated features including mean, median, ProbRF, and ProbXGBoost. The mean feature is obtained by taking means of the gyroscope and accelerometer features. Similarly, we take the median of the gyroscope and accelerometer features. Two additional features of ProbRF and ProbXGBoost are generated using the tree-based ensemble models including RF and XGBoost. We train three models on the whole dataset and then pass the whole dataset to make prediction probabilities. These prediction probabilities are used as features.

- ProbRF: To drive the new features we use machine learning models. The derived features are closer to the target which guides the learning models toward more accurate predictions. We trained random forest (RF) on original features and find the prediction probability for the target classes against each sample. That prediction probability we included in the feature set. We can define it mathematically as:

$$trained_{rf} = RF_{training}(D) \tag{1}$$

$$ProbRF = trained_{rf} \sum_{i=1}^{M} (D_i) \tag{2}$$

  where M is the size of the dataset and D is the dataset.
- ProbXGBoost: We used XGBoost also used to drive the features and similarly to RF we also pass original features to the model and find the prediction probability for the target classes against each sample. We can define ProbXGBoost mathematically as:

$$trained_{xgboost} = XGBoost_{training}(D) \tag{3}$$

$$ProbXGBoost = trained_{xgboost} \sum_{i=1}^{M} (D_i) \tag{4}$$

TABLE III shows the details regarding the use of different features for experiments. Each case considers different features including features from the accelerometer, gyroscope, timestamp, and derived features.

C. Machine Learning Models

We used several machine learning algorithms for driver behavior prediction. We used RF, XGBoost, SVC, ETC, and LR with their best hyperparameters settings. We find the best hyperparameters by tuning each model between a specific range.

1) Random Forest: RF is applied for both regression and classification problems. It is an ensemble model that uses the decision tree concept for classification. The bagging technique is applied to train a large number of decision trees with several samples of bootstrap [45]. In addition, a random forest is used to reduce the over-fitting problems with a bootstrap technique for sampling. Sampling for the training dataset using replacement is applied to attain a bootstrap sample where the training dataset and sample size are similar [46]. All classifiers that use the decision trees for the process of prediction apply the same methods to construct the decision trees. For this, attribute selection of root nodes at every level is challenging during tree construction in random forest [47]. In ensemble classification, different classifiers are trained and all classifier results are integrated through the voting process. Many contributors have described multifarious ensemble approaches; boosting and bagging are very famous ensemble techniques [48]. Several classifiers are trained on bootstrapped samples that lead to a drop-in for classification in the bagging method. As shown in TABLE IV, we choose m_ estimtr = 300 to obtain the best accuracy when using the voting method for combining the individual predictions. The maximum depth, mx_dpth is set to be 300 to reduce the probability and complexity of overfitting. The random forest class prediction is represented as

$$\hat{C}^{B}_{rf}(x) = majorityvote\{\hat{C}_b(x)\}^{B}_{1} \tag{5}$$

where B represents the number of decision trees.

2) Extra tree classifier: ETC uses the process of randomization as a base concept to construct trees [49]. For every node, the split conditions are decided randomly at every node for an extra tree, and the prime performing rule is selected to associate with that node which is based on a score calculation. This is helpful when reducing the complexity significantly of the induction process and increasing the training speed. To do so, the correlation among the decision tree is reduced. The process of node splitting is easy and the computational load for the algorithm is dropped as the Extra tree classifier is not included in locally optimal cut-points. The bagging process is not used as the whole available learning set is provided to every decision tree [50]. As described in TABLE IV, the three parameters rndm_state, mx_dpth, and m_estimtr are chosen to be 27, 300, and 300 respectively.

3) Logistic Regression: Logistic regression (LR) is a pure statistical technique that is applied for data analysis and contains one or more variables for outcome prediction. LR is applied to evaluate the class member's probability because it is the best classifier when it comes to a definite target variable. To estimate the probabilities, a logistic function (LF) is used to evaluate the behavior among dependent variables and independent variables [45]. The 'slvr' parameter is set as 'newton-cg' due to solving the multi-class classification problem. In addition, the 'multi-class parameter is set as 'multi-nomial' because of multi-class classification. The 'D' is set to 1. 'D' value is inversely proportional to regularization strength and helps to reduce the overfitting probability eventually [51]. The probability of predicting the class k, given the input sample X_i

$$Pr(Y_i = k) = \frac{e^{\beta_k . X_i}}{\Sigma_{0 \le c \le k} e^{\beta_c . X_i}} \tag{6}$$

4) Support Vector Classifier: SVC is a linear support vector classifier and used for regression, classification, and has

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000

TABLE III: Description of data attributes.

Case	AccX	AccY	AccZ	GyroX	GyroY	GyroZ	Timestamp	Mean	Median	RF_Prob	XGBoost_Prob
Case 1: Experiment with gyroscope features	x	x	x	✓	✓	✓	x	x	x	x	x
Case 2: Experiment with accelerometer features	✓	✓	✓	x	x	x	x	x	x	x	x
Case 3 : Experiment without 'timestamp' feature + binary target classes	✓	✓	✓	✓	✓	✓	x	x	x	x	x
Case 4 : Experiment without 'timestamp' feature + three target classes	✓	✓	✓	✓	✓	✓	x	x	x	x	x
Case 5 : Experiment with 'timestamp' feature + three target classes	✓	✓	✓	✓	✓	✓	✓	x	x	x	x
Case 6: Experiment with new features and without the 'timestamp' feature + three classes	✓	✓	✓	✓	✓	✓	x	✓	✓	✓	✓

many applications. SVC divides the sample data into different classes with a hyperplane or set of hyper-planes in m-dimensional space, where m is used for the number of features [52], [53]. SVC performs classification to find the "best fit" hyperplane that is differentiated among classes. To deal with the nonlinear issues, this research uses a 'linear' kernel for the support vector machine which is frequently used when the dataset has many features. The linear kernel training is faster due to the requirement of D regularization parameter optimization. In TABLE IV, D regularization parameter value is set to three, and rndm_state value is 500. The hyperplane function is denoted as

$$H(x) = \begin{cases} +1, & \text{if } w.x + b \geq 1 \\ -1, & \text{if } w.x + b \leq 1 \end{cases} \tag{7}$$

The objective function needs to be minimized such that $y_i(w.xi+)b \geq 1$ satisfy all the time.

5) Extreme Gradient Boosting: XGBoost model works in a way similar to the gradient boosting model. However, an additional feature is needed for assigning weights to every sample like in the Adaboost model [54], [55]. The eXtreme Gradient Boosting is a tree-based classifier and it has received much attention recently. XGBoost fits several distinct decision trees parallel which ensures the sequence. For this, XGBoost provides a speed boost. The eXtreme Gradient Boosting has standardized methods to control over-fittings like L1 and L2 and these methods are not available in Adaboost and GBoost models. Here, Alpha and Lambda are the L1 and L2 regularization terms, respectively. In addition, an extra key feature of Gradient Boosting is scalability. It helps to better perform on distributed systems and process large-scale datasets. Moreover, it uses a Log-Loss function, which is very helpful for loss minimization and increasing accuracy. The Log-Loss function estimates the probability of false categorizations. The loss function is defined as

$$Logloss = \frac{1}{M} \Sigma_{j=1}^{M} x_j.log(q(x_j)) + (1 - x_j).log(1 - q(x_j)) \tag{8}$$

In TABLE IV, values of four parameters are set for eXtreme Gradient Boosting. The m_estimtr = 300 implies eXtreme Gradient Boosting that is used 300 decision trees for the base-learner which takes part in the process of prediction. The parameter mx_dpth = 300 restricts the growth of the trees to a maximum of 300. The lerning_ratio = 0.2 is used to control the overfitting [55]. The rndm_state = 27 restricts the random seed specified to every Tree estimator at every boosting repetition. Additionally, it controls random permutations for features at every split.

TABLE IV: Hyperparameters used for machine learning models.

Model	Parameters
RF	m_ estimtr = 300, mx_dpth = 300, rndm_state = 27
XGBoost	m_ estimtr = 300, lerning_ratio = 0.2, mx_dpth = 300, rndm_state = 27
SVC	kernel = sigmoid, D = 3.0, rndm_state = 27
ETC	m_ estimtr = 300, mx_dpth = 300, rndm_state = 27
LR	slvr=' newton-cg', multi- class='multinomial', D =1.0

IV. Results and Discussion

In this section, a detailed description of the experimental results obtained using machine learning techniques and analysis is presented. The experiments were run on a standalone Linux machine with a system configuration of 8 GB RAM and 8-core processors. A notebook web application runs locally on the machine to perform the experiments. The software packages scikit-learn[1] were installed and the python programming language was used to write the code.

The performance metrics accuracy, precision, recall, and F1 score are used to compare the experimental results. Accuracy is defined as the sum of the true positives (TP) and true negatives (TN) divided by the sum of the TP, TN, false positive (FP), and false negative (FN). The precision is measured as the TP divided by the sum of the TP and FP. The recall is defined as the TP divided by the TP and FN. The F1 score is the harmonic mean of precision and recall.

[1]https://scikit-learn.org/stable/

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (12)$$

A. Driving Behavior Prediction using Gyroscope Features (Case: 1)

In this subsection, the driving behavior performance results when the acceleration feature is ignored are discussed.TABLE V displays the performance metrics for all five models when the acceleration feature is excluded.

TABLE V: Driving behavior prediction results using only gyroscope features.

Model	Accuracy	Target	Precision	Recall	F1 Score
RF	0.35	Aggressive	0.35	0.30	0.32
		Normal	0.31	0.29	0.30
		Slow	0.38	0.45	0.42
		Macro Avg	0.38	0.38	0.38
LR	0.38	Aggressive	0.43	0. 03	0.05
		Normal	0.00	0.00	0.00
		Slow	0.38	0.99	0.66
		Macro Avg	0.27	0.34	0.20
ETC	0.34	Aggressive	0.31	0.29	0.30
		Normal	0.32	0.31	0.31
		Slow	0.37	0.41	0.39
		Macro Avg	0.33	0.33	0.33
SVC	0.38	Aggressive	0.40	0.03	0.05
		Normal	0.00	0.00	0.00
		Slow	0.38	0.99	0.54
		Macro Avg	0.36	0.38	0.31
XGBoost	0.35	Aggressive	0.33	0.35	0.34
		Normal	0.34	0.30	0.32
		Slow	0.38	0.41	0.39
		Macro Avg	0.35	0.35	0.35

The prediction accuracy of the selected ML models varied between 35% to 38%. It is evident that driving behavior prediction using only the gyroscope data performed poorly. The SVC and LR models performed slightly better than decision tree-based models. In particular, the 'slow' target class prediction shows promising results with 99% recall for both LR and SVC models. These results also show that the 'normal' class target is misclassified as a 'slow' class target in both SVC and LR models. The distinction between normal and slow targets is challenging using mathematical models to separate the classes. The precision and recall metrics follow a similar trend as accuracy in decision tree-based models. Overall, the performance metrics indicate that acceleration features are valuable for driving behavior classification and should be included for multi-class evaluation as the performance of models using the gyroscope data alone is not satisfactory.

TABLE VI displays the accuracy of consistency analysis by measuring the standard deviation. The accuracy-based standard deviation for all the models is varying between 0.02 to 0.04. These results indicate that we can rely on the obtained accuracy values and not see significant variations even while repeating the experiments. Here, CP is the number of correct predictions and WP is the number of wrong predictions.

TABLE VI: K-fold results and confusion matrix values for driving behavior prediction using gyroscope features.

Model	Accuracy	SD	CP	WP
RF	0.35	+/-0.03	257	472
LR	0.38	+/-0.02	275	454
ETC	0.34	+/-0.02	247	482
SVC	0.38	+/-0.02	274	455
XGBoost	0.35	+/-0.04	258	471

TABLE VI shows the driving behavior target class correct and wrong prediction sample count for each model. The LR model correctly classifies more test samples with 275 correct classifications which are higher compared to other models. On the other hand, the ETC model least correctly predicts the test samples with 247 correct predictions. We can construe that the decision tree-based models least performed to correctly classify the driving behavior compared to the LR and SVC models.

B. Driving Behavior Prediction using Acceleration Features (Case: 2)

In order to evaluate the performance of the driving behavior under different feature combinations, we start with a feature set of 4 by excluding the gyroscope attributes and using only the accelerometer features. TABLE VII describes the performance of the driving behavior with the accelerometer features case. The five machine learning models RF, LR, ETC, SVC, and XGBoost are considered for our evaluation. TABLE VII clearly shows that none of the machine learning techniques performed well when the gyroscope features are excluded from the trained dataset. All five models achieved similar prediction accuracy on the test datasets. The SVC and LR obtained 40% accuracy, whereas RF, ETC, and XGBoost achieved approximately 39% accuracy. A similar trend appears in precision and recall metrics for all five models except the recall for the 'slow' class case when LR and SVC are used. The LR and SVC report 84% recall for the 'slow' target case. The 'normal' target case is greatly impacted by recall performance when the 'slow' target case is predicted using LR and SVC. The macro average obtained for both precision and recall metrics is almost the same for all five machine-learning models. Overall, these results show that gyro features are essential for predicting driving behavior and should not be ignored.

The accuracy of the machine learning models is verified using the standard deviation measurement. TABLE VIII depicts that the accuracy standard deviation in all five models is minimal and near zero. So, the accuracy results are consistent when the standard deviation is considered regarding the machine learning models.

The test dataset samples' correct and wrong predictions for all the five machine learning models are shown in TABLE VIII. The SVC model correctly predicts the highest number of test samples with 291 correct predicts, which is higher than

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000

TABLE VII: Driving behavior prediction results using only accelerometer features.

Model	Accuracy	Target	Precision	Recall	F1 Score
RF	0.39	Aggressive	0.39	0.37	0.38
		Normal	0.34	0.28	0.31
		Slow	0.41	0.49	0.44
		Macro Avg	0.38	0.38	0.38
LR	0.40	Aggressive	0.48	0. 27	0.34
		Normal	0.21	0.02	0.04
		Slow	0.39	0.84	0.53
		Macro Avg	0.36	0.38	0.31
ETC	0.39	Aggressive	0.38	0.39	0.39
		Normal	0.38	0.35	0.36
		Slow	0.39	0.42	0.40
		Macro Avg	0.38	0.38	0.38
SVC	0.40	Aggressive	0.48	0.27	0.34
		Normal	0.21	0.02	0.04
		Slow	0.39	0.84	0.53
		Macro Avg	0.36	0.38	0.31
XGBoost	0.38	Aggressive	0.35	0.38	0.36
		Normal	0.38	0.33	0.35
		Slow	0.41	0.43	0.42
		Macro Avg	0.38	0.38	0.38

TABLE VIII: K-fold results and confusion matrix values for driving behavior using accelerometer features.

Model	Accuracy	SD	CP	WP
RF	0.39	+/-0.02	281	448
LR	0.40	+/-0.02	278	451
ETC	0.39	+/-0.02	281	448
SVC	0.40	+/-0.02	291	438
XGBoost	0.38	+/-0.03	278	451

all other models. On the other hand, the XGBoost and LR models show the least correctly predicted test-driving behavior samples, each with 278 correct predictions.

C. Driving Behavior Prediction Without Timestamp Features and With Two Targets Classes (Case : 3)

In the above test case, we have seen that it is difficult to discriminate between the 'slow' and 'normal' target classes which reduces the prediction accuracy of models. So, we evaluate the performance of the models by combining the 'normal' and 'slow' target classes as one target class and excluding the timestamp feature from the input dataset. So, the number of input features is 6, and the output classes are 2 in this scenario.

TABLE IX presents the performance metrics of the models when two output classes are considered, and the timestamp is excluded from the input dataset. The results indicate that all five models achieve 100% accuracy, prediction, recall, and F1 score. So, for binary classification, the decision tree-based models, SVC and LR are able to classify the target classes even if the timestamp is not present in the input datasets. When the gyroscope and accelerometer features are used to train the models, the models classify the 'aggressive' and 'normal' driving with 100% accuracy. However, the 'normal' and 'slow' driving behavior classifications require additional features to capture the driving behavior.

TABLE X supports the fact that accuracy is 100% for this dataset when the output classes are categorized into two classes and no standard deviation is observed for this case.

TABLE IX: Driving behavior prediction results without timestamp feature and for binary classification problem.

Model	Accuracy	Target	Precision	Recall	F1 Score
RF	1.00	Aggressive	1.00	1.00	1.00
		Slow	1.00	1.00	1.00
		Macro Avg	1.00	1.00	1.00
LR	1.00	1.00	1.00	1.00	1.00
		Slow	1.00	1.00	1.00
		Macro Avg	1.00	1.00	1.00
ETC	1.00	Aggressive	1.00	1.00	1.00
		Slow	1.00	1.00	1.00
		Macro Avg	1.00	1.00	1.00
SVC	1.00	Aggressive	1.00	1.00	1.00
		Slow	1.00	1.00	1.00
		Macro Avg	1.00	1.00	1.00
XGBoost	1.00	Aggressive	1.00	1.00	1.00
		Slow	1.00	1.00	1.00
		Macro Avg	1.00	1.00	1.00

TABLE X: k-fold results and confusion matrix values for driving behavior prediction without timestamp feature and using binary classification problem.

Model	Accuracy	SD	CP	WP
RF	1.00	+/-0.00	446	0
LR	1.00	+/-0.00	446	0
ETC	1.00	+/-0.00	446	0
SVC	1.00	+/-0.00	446	0
XGBoost	1.00	+/-0.00	446	0

TABLE X shows the number of samples that are correctly classified for all five models when the classification categories are two. All the 446 testing samples are correctly classified as either 'normal' or 'aggressive' driving.

D. Driving Behavior Prediction Without Timestamp Feature and Three Target Classes (Case: 4)

In general, the timestamp feature may add little value to accurately predict the detection or classification using machine learning models. We excluded the timestamp from the input features to test the case and trained the models with 6 features, 3 features each from the accelerometer and gyroscope. TABLE XI shows the performance metric values for the selected five models when the timestamp feature is excluded from the input feature. The prediction accuracy for all five models is slightly better than in the previous two cases. However, the overall performance follows a similar trend as the last two cases. Except for the 'slow' target classification using LR and SVC, the performance is nominal. The 'aggressive' target classification precision for RF, LR, ETC, and SVC has been slightly improved as well compared to the gyroscope and Accelerometer feature alone. Overall, based on the performance metrics obtained when one of the features is excluded from the input feature set suffers a performance loss. This can be the fact that target classification is multiclass and the input features are not enough to distinguish the multi classes, in particular, the 'normal' versus 'slow' target classes.

TABLE XII indicates that the accuracy is consistent for all the models, even if machine learning training and testing experiments are repeated with a slight standard deviation between 0.03 to 0.04. So, we can confirm that the timestamp

TABLE XI: Driving behavior performance results without timestamp feature and for three classes.

Model	Accuracy	Target	Precision	Recall	F1 Score
RF	0.43	Aggressive	0.55	0.39	0.46
		Normal	0.36	0.30	0.31
		Slow	0.42	0.59	0.49
		Macro Avg	0.44	0.43	0.42
LR	0.38	Aggressive	0.51	0.20	0.29
		Normal	0.25	0.06	0.09
		Slow	0.38	0.85	0.52
		Macro Avg	0.38	0.37	0.30
ETC	0.42	Aggressive	0.51	0.39	0.44
		Normal	0.35	0.31	0.33
		Slow	0.41	0.55	0.47
		Macro Avg	0.42	0.41	0.41
SVC	0.38	Aggressive	0.49	0.21	0.29
		Normal	0.27	0.06	0.10
		Slow	0.37	0.84	0.52
		Macro Avg	0.38	0.37	0.30
XGBoost	0.38	Aggressive	0.40	0.36	0.38
		Normal	0.35	0.32	0.34
		Slow	0.40	0.46	0.43
		Macro Avg	0.38	0.38	0.38

exclusion also has a consistent performance loss impact on the classification results.

TABLE XII: K-fold results and confusion matrix values for driving behavior prediction without timestamp feature and for three classes.

Model	Accuracy	SD	CP	WP
RF	0.43	+/-0.04	314	415
LR	0.38	+/-0.03	279	450
ETC	0.42	+/-0.04	304	425
SVC	0.38	+/-0.03	278	451
XGBoost	0.38	+/-0.03	280	449

Interestingly, the RF performed slightly better than other models when the timestamp feature is excluded from the input feature dataset. TABLE XII shows the correct and wrong predicted test classification sample count for the models. RF can correctly classify 314 samples, whereas the SVC correctly classified the least number of data samples, i.e. 278.

E. Driving Behavior With Timestamp Feature and Three Target classes (Case: 5)

Although we obtained 100% accuracy for driving behavior using binary classification, the best accuracy still needs to be achieved for multiclass classification. So, we use all the dataset input features and keep the target classes as 3 (normal, slow, and aggressive) for performance evaluation. TABLE XIII depicts the performance metrics of the models when all the input features are included to train and test the models. We can see that decision tree-based models obtain 100% accuracy for multiclass target classification. The precision and recall are also 100% for decision tree models such as RF, ETC, and XGBoost. However, the LR and SVC models did not perform well for multiclass classification and only obtained 37% accuracy. These models cannot distinguish between the 'normal' and 'aggressive' target classes. Overall, the RF, ETC, and XGBoost techniques suit well for driving behavior target classification, in which the classes are not easily separated with mathematical computations.

TABLE XIII: Driving behavior performance results with all features and for three classes.

Model	Accuracy	Target	Precision	Recall	F1 Score
RF	1.00	Aggressive	1.00	1.00	1.00
		Normal	1.00	1.00	1.00
		Slow	1.00	1.00	1.00
		Macro Avg	1.00	1.00	1.00
LR	0.37	Aggressive	0.00	0.00	0.00
		Normal	0.00	0.00	0.00
		Slow	0.37	1.00	0.54
		Macro Avg	0.12	0.33	0.18
ETC	0.67	Aggressive	1.00	1.00	1.00
		Normal	0.49	0.43	0.46
		Slow	0.54	0.60	0.57
		Macro Avg	0.68	0.68	0.67
SVC	0.37	Aggressive	0.00	0.00	0.00
		Normal	0.00	0.00	0.00
		Slow	0.37	1.00	0.54
		Macro Avg	0.12	0.33	0.18
XGBoost	1.00	Aggressive	1.00	1.00	1.00
		Normal	1.00	1.00	1.00
		Slow	1.00	1.00	1.00
		Macro Avg	1.00	1.00	1.00

TABLE XIV reveals that the decision tree-based model's accuracy is consistent when performing the experiments multiple times. On the other hand, the LR and SVC obtained low accuracy when repeating the experiments.

TABLE XIV: K-fold results and confusion matrix values for driving behavior prediction using all features

Model	Accuracy	SD	CP	WP
RF	1.00	+/-0.00	729	0
LR	0.37	+/-0.00	271	458
ETC	1.00	+/-0.01	728	01
SVC	0.37	+/-0.02	271	458
XGBoost	1.00	+/-0.00	729	0

As we can see in TABLE XIV, the RF, ETC, and XGBoost are able to correctly classify all the driving behavior samples of 729 into three classes. On the other hand, LR and SVC perform poorly and each has 458 wrong predictions.

F. Driving Behavior Prediction Without Timestamp Feature and New (Derived) Features (Case: 6)

As the number of features in the dataset is less, additional features are included to test the machine learning models' performances. The mean of the Accelerometer in the x, y, and z-axis is included as another feature. Similarly, another feature is the mean of gyroscope values in the x, y, and z-axis. Overall, nine features are used to train the models. TABLE XV presents the performance evaluation metric values when testing the dataset with models. The prediction accuracy is improved in all five models, and the accuracy range is between 65% to 67%. The 'aggressive' target class obtained 100% precision and recall for all five models. It shows that distinguishing the 'slow' and 'normal' classes hampers the overall accuracy of driving behavior multiclass classification. The 'normal' target class characteristics should be captured in training to accurately classify all the classes in the driving behavior dataset. As the previous cases show, the SVC and LR model achieved 'slow' target classification with a recall of 86%. Overall, the mean of the sensor values is essential and can obtain significant performances in the multiclass category.

TABLE XV: Driving behavior performance results using derived features and without timestamp feature.

Model	Accuracy	Target	Precision	Recall	F1 Score
RF	0.67	Aggressive	1.00	1.00	1.00
		Normal	0.48	0.40	0.44
		Slow	0.53	0.61	0.57
		Macro Avg	0.67	0.67	0.67
LR	0.66	Aggressive	1.00	1.00	1.00
		Normal	0.44	0.14	0.21
		Slow	0.52	0.84	0.64
		Macro Avg	0.65	0.66	0.62
ETC	0.67	Aggressive	1.00	1.00	1.00
		Normal	0.49	0.43	0.46
		Slow	0.54	0.60	0.57
		Macro Avg	0.68	0.68	0.67
SVC	0.65	Aggressive	1.00	1.00	1.00
		Normal	0.36	0.08	0.14
		Slow	0.51	0.86	0.64
		Macro Avg	0.62	0.65	0.59
XGBoost	0.65	Aggressive	1.00	1.00	1.00
		Normal	0.46	0.43	0.44
		Slow	0.51	0.53	0.54
		Macro Avg	0.66	0.66	0.66

TABLE XVI shows the consistency of model accuracy values by measuring the standard deviation. The results indicate that accuracy for all the models lies between 0.62 to 0.69.

TABLE XVI: K-fold results and confusion matrix values for driving behavior prediction using derived features and without timestamp feature.

Model	Accuracy	SD	CP	WP
RF	0.67	+/-0.02	485	244
LR	0.66	+/-0.02	484	245
ETC	0.67	+/-0.02	489	240
SVC	0.65	+/-0.02	477	252
XGBoost	0.65	+/-0.03	474	255

The ETC model can correctly predict more driving behavior samples (485) than other models. XGBoost shows the least performance model with a correct classification of 474 samples. The results in TABLE XVI show that most of those correct predictions belong to the aggressive target class.

Figure 5 shows the comparison between all cases. According to the results, models show superb performance when the 'timestamp' feature is included in the dataset for training and testing the models. Although, not as successful as the 'timestamp' case, when using the additional features the performance of the models is better than using the original features without 'timestamp'. Similarly, when the problem is transformed into a binary problem (slow vs aggressive), models show superior performance.

G. Computation Complexity of Machine Learning Models

TABLE XVII shows the time taken to predict the classes in all five models. XGBoost took a minimum amount of time of 0.3sec to accurately classify the driving behavior, whereas RF took 0.7056 sec to correctly classify all driving behavior samples. Overall, based on the extensive study of the feature selection and output class selection and the corresponding model performances, we propose the XGBoost techniques achieve the best performance with minimum computation time for driving behavior sample multiclass classification. This variation is because of the number of target classes and a number of features for the experiments. Case 6 consists of original features as well as new features. So the increase in feature set size also increases the computational time.

TABLE XVII: Driving behavior computational time in each case.

Model	Execution time (sec)					
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
RF	0.7550	0.8365	0.3079	1.3159	0.7056	0.8892
LR	0.0399	0.0400	0.0461	0.0385	0.0614	0.0776
ETC	0.6408	0.6436	0.2016	0.7661	0.3788	0.5020
SVC	0.0437	0.2225	0.0150	0.3237	0.4411	0.1141
XGBoost	1.9132	2.9634	0.0438	2.2100	0.3006	1.1936

H. Results Using Additional Datasets

To prove the significance of the proposed approach, we utilized two additional datasets for case 6 where additional features are generated. Additionally used datasets include the 'driving behavior dataset' (DBD) [56] and the 'Carla driver behaviour dataset' (CDBD) [57]. DBD consists of 60 features combining accelerometer and gyroscopes features and four classes. The dataset collection includes the use of Ford Fiesta 1.25, Ford Fiesta 1.4, Hyundai i20, and three different drivers with the ages of 27, 28, and 37. The collection involves an MPU6050 sensor and Raspberry Pi 3 Model B. While the CDBD dataset consists of 6 features; three from the gyroscope and three from the accelerometer. Seven drivers contribute to this dataset and for each instance, the dataset is categorized on the driver names mehdi, selin, onder, apo, berk, hurcan and gonca. For experiments, the best-performing models RF and GBM are used and results are shown in TABLE XVIII. RF and GBM both show better results for the DBD dataset as they achieve a 1.00 accuracy score while for the CDBD they could not perform well as only RF can achieve a 0.72 accuracy score. The model's poor performance on the CDBD is because of the poor relationship between the target classes and the feature set.

Figure 6 shows the confusion matrices for RF and GBM for both datasets. For the CDBD dataset, confusion matrix values 1 to 7 indicate the apo, berk, gonca, hurcan, mehdi, wonder, and selin classes, respectively. For the DBD dataset, RF gives 0 wrong predictions and GBM gives only 1 wrong prediction. RF gives 14285 correct predictions out of 19872 predictions and GBM gives 12848 correct predictions out of 19872 predictions.

I. Performance of Deep Learning Models

This study also performs experiments using the deep learning approach. We deployed two state-of-the-art models LSTM and CNN for driver behavior predictions. This study uses two models LSTM and CNN for driver behavior prediction

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000

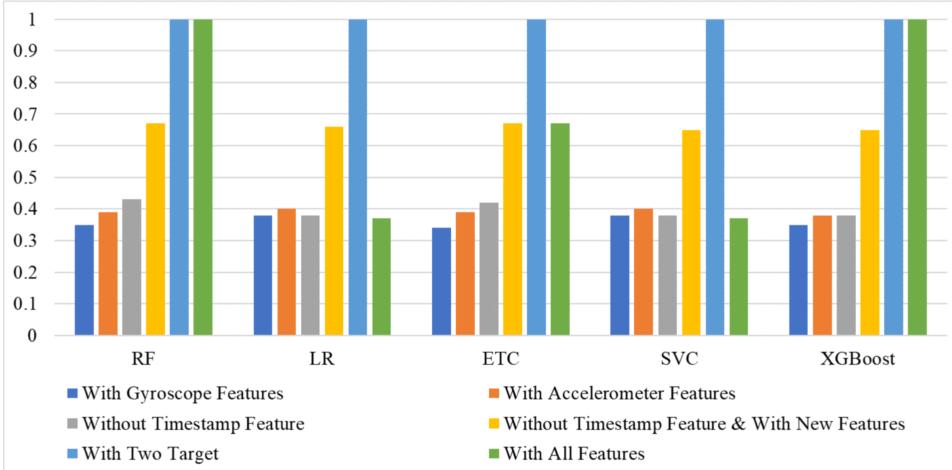


Fig. 5: Performance of all machine learning models for all cases.

TABLE XVIII: Case 6 scenario results on other datasets.

DBD							
RF				GBM			
Class	Precision	Recall	F1 Score	Class	Precision	Recall	F1 Score
1	1.00	1.00	1.00	1	1.00	1.00	1.00
2	1.00	1.00	1.00	2	1.00	1.00	1.00
3	1.00	1.00	1.00	3	1.00	1.00	1.00
4	1.00	1.00	1.00	4	1.00	1.00	1.00
macro avg	1.00	1.00	1.00	macro avg	1.00	0.99	1.00
weighted avg	1.00	1.00	1.00	weighted avg	1.00	1.00	1.00
Accuracy	1.00			Accuracy	1.00		
CDBD							
RF				GBM			
Class	Precision	Recall	F1 Score	Class	Precision	Recall	F1 Score
apo	0.99	1.00	1.00	apo	0.99	0.99	0.99
berk	0.68	0.60	0.64	berk	0.58	0.54	0.56
gonca	0.70	0.67	0.68	gonca	0.62	0.58	0.60
hurcan	0.65	0.61	0.63	hurcan	0.57	0.54	0.56
mehdi	0.65	0.62	0.63	mehdi	0.57	0.54	0.55
onder	0.65	0.64	0.65	onder	0.56	0.58	0.57
selin	0.69	0.83	0.76	selin	0.62	0.70	0.66
macro avg	0.72	0.71	0.71	macro avg	0.65	0.64	0.64
weighted avg	0.72	0.72	0.72	weighted avg	0.65	0.65	0.65
Accuracy	0.72			Accuracy	0.65		

as these are commonly used models for similar kinds of datasets. For example, the study [38] used CNN for human behavior prediction, and [39] used LSTM and CNN for human activity detection. The authors utilized variants of CNN and LSTM in [40] for driver behavior detection. The wide use of these models motivated us to choose these models in the current study for driver behavior prediction. We used these models with their state-of-the-art parameters settings as shown in Table XIX. The embedding layer is an input layer that defines the vocabulary size, output dimension, and length of the feature set. We used these models with 100 epochs and categorical_cross-entropy loss function because of multi-class data. We also used the 'Adam' optimizer to compile these models.

TABLE XX shows the results of deep learning models which indicates that both models perform well when we used the 'timestamp' feature. LSTM outperforms with a 0.84 accuracy score for case 5 when we used timestamp. Overall the performance of deep learning models is not significant in terms of accuracy as compared to machine learning models because of the small feature set. Deep learning models required a large feature set for a good fit.

TABLE XIX: Architecture of deep learning models.

Model	**Hyperparameters**	
LSTM	Embedding(500,50,input_length=..) Dropout(0.5) LSTM(64) Dense(16) Dense({3,2}, activation='softmax')	{loss= 'categorical_crossentropy, binary_crossentropy} optimizer='adam' ,epochs=100, batch_size=8}
CNN	Embedding(500,50, input_length=..) Conv1D(64,2,activation='relu') MaxPooling1D(pool_size=2) Activation('relu') Dropout(rate=0.5) Flatten() Dense({3,2},activation='softmax')	

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000

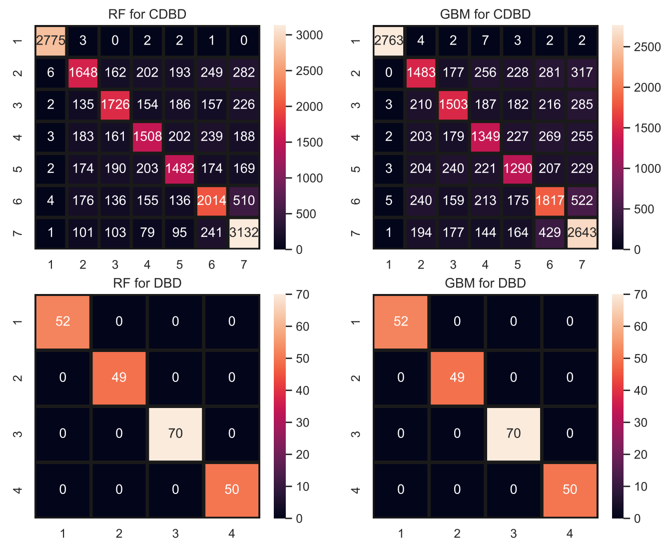


Fig. 6: Confusion matrices for RF and GBM on CDBD and DBD datasets.

TABLE XX: Results for deep learning models for all scenarios.

Case	LSTM	CNN
Case 1: Experiment with gyroscope features	0.43	0.42
Case 2: Experiment with accelerometer features	0.42	0.42
Case 3 : Experiment without 'timestamp' feature + binary target classes	0.79	0.75
Case 4 : Experiment without 'timestamp' feature + three target classes	0.45	0.42
Case 5 : Experiment with 'timestamp' feature + three target classes	0.84	0.77
Case 6 : Experiment with derived features, without 'timestamp' feature + three classes	0.59	0.58

J. Comparison With Other Approaches

To show the significance of the proposed approach, we performed a comparative analysis with other studies as well. We deployed approaches from other studies on the dataset used in the current study to perform a fair comparison. We selected recent studies which have done work on similar types of datasets. The study [58] worked on human activity detection using a multi-layer perceptron (MLP) model. The authors utilized gyroscope and accelerometer features for human activity detection. Similarly, studies [59] and [60] used smartphone accelerometers and gyroscope features using SVM and KNN models, respectively. The study [61] worked on sign classification using a machine learning approach. The authors deployed SVM using the accelerometer and gyroscope features dataset. Similarly, smartphone IMU data is used by [62] with an ensemble model for the same purpose. We deployed all these studies on our used dataset with all cases to carry out a performance comparison. TABLE XXI shows the comparison with other approaches which indicates that the proposed approach outperforms existing state-of-the-art approaches.

K. Discussions

This study performs experiments using a total of six different cases where the influence of using a different set of features is extensively investigated for driver behavior prediction. Similarly, the impact of multiclass and binary classification is also analyzed. It is found that 'timestamp' is the most important feature regarding the performance of machine learning models. Adding this feature to the training dataset dramatically increased the classification accuracy for multiclass classification. Although using additional (derived) features can show better performance even without using the 'timestamp' feature than using individual features from accelerometer and gyroscope features alone, this performance is inferior to that of using the 'timestamp' feature. Primarily, the 'slow' and 'normal' classes seem to have similar feature space, as shown in Figure 7f which leads to a higher number of wrong predictions for these classes when the 'timestamp' feature is not used. Using accelerometer features or gyroscope features alone is not sufficient to produce high performance, as shown in Figures 7a and 7b. However, when the 'timestamp' features are combined with either accelerometer or gyroscope features, the performance of the models is enhanced, as shown in Figures 7c and 7d. The deliverable things of this research are a software-based approach for driver behavior prediction which is more accurate and efficient. Linked with a data source, this approach provides driver behavior prediction.

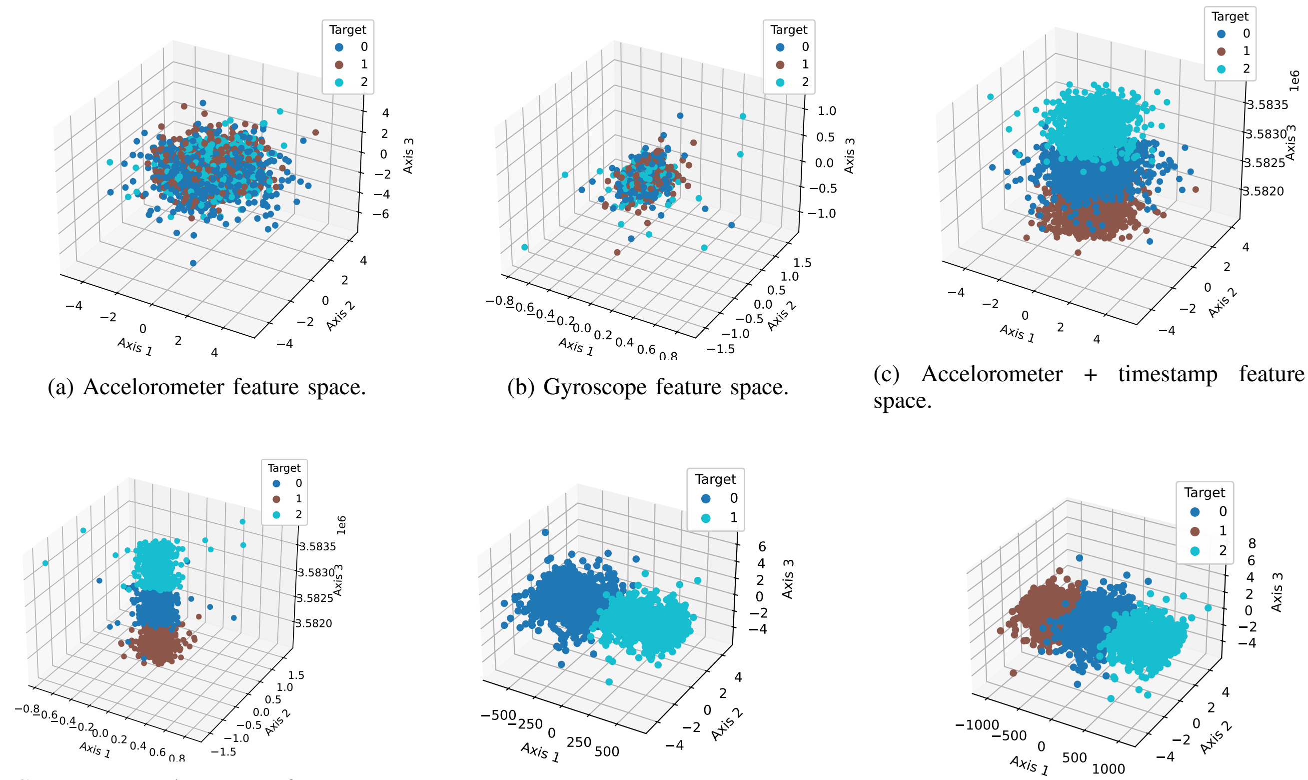
TABLE XXI: Comparison with other approaches.

Ref.	Year	Model	Cases					
			1	2	3	4	5	6
[58]	2020	MLP	0.31	0.34	0.94	0.33	0.97	0.58
[59]	2021	SVM	0.37	0.40	0.96	0.35	0.36	0.65
[60]	2022	KNN	0.37	0.38	0.93	0.34	0.36	0.67
[61]	2022	SVM	0.37	0.40	0.95	0.35	0.37	0.65
[62]	2022	EM	0.30	0.33	0.91	0.28	0.35	0.54
Our	2022	RF	0.35	0.39	1.00	0.43	1.00	0.67

V. Conclusion

Driver behavior prediction is an important part of designing the interaction between advanced driver assistance systems (ADAS) and human drivers for future transportation systems. Consequently, driver behavior prediction has emerged as an important research topic and has been investigated largely during the past few years. Often, the investigations are based on simulators and controlled environments. This study investigates the use of a different set of features, feature combinations, use of original plus derived features for driver behavior prediction using the dataset recorded in a real traffic environment. The data recorded using the smartphone accelerometer and gyroscope is used for experiments using several machine learning and deep learning models. This study designs six cases to investigate the impact of feature selection and binary vs multi-class classification problems. Results indicate that using accelerometer or gyroscope data alone is not sufficient to obtain high performance. Combining the features, though increases the performance, yet, the accuracy is still low. Primarily, the 'slow' and 'normal' class feature spaces tend to overlap which reduces the performance of the models. Adding derived features would further improve the performance, however, the best performance of 100% accuracy is achieved by RF and XGBoost models when accelerometer

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000



(a) Accelerometer feature space.

(b) Gyroscope feature space.

(c) Accelerometer + timestamp feature space.

(d) Gyroscope + timestamp feature space.

(e) All features with binary target classes (SLOW, AGGRESSIVE).

(f) All features with three target classes.

Fig. 7: Feature space using different features.

and gyroscope features are combined with the 'timestamp' features. Deep learning models tend to show lower accuracy than machine learning models. This study uses a small dataset which can be seen as a limitation. The small size of the dataset may not be enough for the training of models, especially deep learning models. The second limitation is the small feature set because deep learning models require a large feature set to get a good fit. In future work, we intend to collect our own dataset and perform experiments. Besides that use of non-machine learning architectures including probabilistic methods or statistical directed acyclic graphs would be a good dimension to explore.

References

[1] E. Mantouka, E. Barmpounakis, E. Vlahogianni, and J. Golias, "Smartphone sensing for understanding driving behavior: Current practice and challenges," *International journal of transportation science and technology*, vol. 10, no. 3, pp. 266–282, 2021.

[2] Z. E. Abou Elassad, H. Mousannif, H. Al Moatassime, and A. Karkouch, "The application of machine learning techniques for driving behavior analysis: A conceptual framework and a systematic literature review," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103312, 2020.

[3] H. Gwyther and C. Holland, "The effect of age, gender and attitudes on self-regulation in driving," *Accident Analysis & Prevention*, vol. 45, pp. 19–28, 2012.

[4] M. Abou-Zeid, I. Kaysi, and H. Al-Naghi, "Measuring aggressive driving behavior using a driving simulator: An exploratory study," in *3rd International Conference on Road Safety and Simulation*. Citeseer, 2011, pp. 1–19.

[5] S. L. Jamson, D. L. Hibberd, and A. H. Jamson, "Drivers' ability to learn eco-driving skills; effects on fuel efficient and safe driving behaviour," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 657–668, 2015.

[6] G. Chrysikopoulos, "Development of a driver behavior framework for manual and automated control considering driver cognition," Ph.D. dissertation, University of Kansas, 2019.

[7] Z. Zhong, J. Lee, and L. Zhao, "Traffic flow characteristics and lane use strategies for connected and automated vehicles in mixed traffic conditions," *Journal of Advanced Transportation*, vol. 2021, 2021.

[8] E. Tenenboim, A. Lucas-Alba, Ó. M. Melchor, T. Toledo, and S. Bekhor, "Car following with an inertia-oriented driving technique: A driving simulator experiment," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 89, pp. 72–83, 2022.

[9] Y. Peng, L. Cheng, Y. Jiang, and S. Zhu, "Examining bayesian network modeling in identification of dangerous driving behavior," *PLoS one*, vol. 16, no. 8, p. e0252484, 2021.

[10] A. Kontaxi, A. Ziakopoulos, and G. Yannis, "Trip characteristics impact on the frequency of harsh events recorded via smartphone sensors," *IATSS research*, vol. 45, no. 4, pp. 574–583, 2021.

[11] W. H. Organization, *Global action plan on physical activity 2018-2030: more active people for a healthier world*. World Health Organization, 2019.

[12] M. Sangare, S. Gupta, S. Bouzefrane, S. Banerjee, and P. Muhlethaler, "Exploring the forecasting approach for road accidents: Analytical measures with hybrid machine learning," *Expert Systems with Applications*, vol. 167, p. 113855, 2021.

[13] H. Yu, R. Jiang, Z. He, Z. Zheng, L. Li, R. Liu, and X. Chen, "Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives," *Transportation research part C: emerging technologies*, vol. 127, p. 103101, 2021.

[14] M. Li, Z. Cao, and Z. Li, "A reinforcement learning-based vehicle platoon control strategy for reducing energy consumption in traffic oscillations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5309–5322, 2021.

[15] G. Albano, K. Mattas, and B. Ciuffo, "The relationship between hystere-

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2023.3256000

sis and string stability in traffic flow. a simulation-based investigation," Tech. Rep., 2021.

[16] J. Yarlagadda and D. S. Pawar, "Heterogeneity in the driver behavior: An exploratory study using real-time driving data," *Journal of Advanced Transportation*, vol. 2022, 2022.

[17] Y. Ali, Z. Zheng, M. M. Haque, M. Yildirimoglu, and S. Washington, "Clacd: A complete lane-changing decision modeling framework for the connected and traditional environments," *Transportation Research Part C: Emerging Technologies*, vol. 128, p. 103162, 2021.

[18] N. Raju, S. Arkatkar, S. Easa, and G. Joshi, "Developing extended trajectory database for heterogeneous traffic like ngsim database," *Transportation Letters*, vol. 14, no. 5, pp. 555–564, 2022.

[19] R. Izquierdo, A. Quintanar, I. Parra, D. Fernández-Llorca, and M. Sotelo, "Experimental validation of lane-change intention prediction methodologies based on cnn and lstm," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3657–3662.

[20] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2118–2125.

[21] D. Roemmich and J. Gilson, "The 2004–2008 mean and annual cycle of temperature, salinity, and steric height in the global ocean from the argo program," *Progress in oceanography*, vol. 82, no. 2, pp. 81–100, 2009.

[22] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[23] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[24] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," 2020. [Online]. Available: https://arxiv.org/abs/2006.14480

[25] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[26] J. Ferreira, E. Carvalho, B. V. Ferreira, C. de Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLoS one*, vol. 12, no. 4, p. e0174959, 2017.

[27] A. Kashevnik, A. Fedotov, and I. Lashkov, "Dangerous situation prediction and driving statistics accumulation using smartphone," in *2018 International Conference on Intelligent Systems (IS)*. IEEE, 2018, pp. 521–527.

[28] O. A. Osman, M. Hajij, S. Karbalaieali, and S. Ishak, "A hierarchical machine learning classification approach for secondary task identification from observed driving behavior data," *Accident Analysis & Prevention*, vol. 123, pp. 274–281, 2019.

[29] R. R. Rachmadi, A. Sudarsono, and T. B. Santoso, "Application of lightgbm method for abnormal driving behavior classification on motorcycle driver based on smartphone sensor," *Jurnal Komputer Terapan*, vol. 7, no. 2, p. 472581.

[30] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, p. 113240, 2020.

[31] P. Patil, K. S. Kumar, N. Gaud, and V. B. Semwal, "Clinical human gait classification: extreme learning machine approach," in *2019 1st international conference on advances in science, engineering and robotics technology (ICASERT)*. IEEE, 2019, pp. 1–6.

[32] R. Jain, V. B. Semwal, and P. Kaushik, "Deep ensemble learning approach for lower extremity activities recognition using wearable sensors," *Expert Systems*, vol. 39, no. 6, p. e12743, 2022.

[33] S. K. Challa, A. Kumar, and V. B. Semwal, "A multibranch cnn-bilstm model for human activity recognition using wearable sensor data," *The Visual Computer*, pp. 1–15, 2021.

[34] N. Oliver and A. P. Pentland, "Driver behavior recognition and prediction in a smartcar," in *Enhanced and Synthetic Vision 2000*, vol. 4023. SPIE, 2000, pp. 280–290.

[35] S. Arbabi, D. Tavernini, S. Fallah, and R. Bowden, "Learning an interpretable model for driver behavior prediction with inductive biases," *arXiv preprint arXiv:2208.00516*, 2022.

[36] M. J. Rahman, S. S. Beauchemin, and M. A. Bauer, "Predicting driver behaviour at intersections based on driver gaze and traffic light recognition," *IET Intelligent Transport Systems*, vol. 14, no. 14, pp. 2083–2091, 2020.

[37] W. Xu, J. Wang, T. Fu, H. Gong, and A. Sobhani, "Aggressive driving behavior prediction considering driver's intention based on multivariate-temporal feature data," *Accident Analysis & Prevention*, vol. 164, p. 106477, 2022.

[38] M. Sajjad, S. Zahir, A. Ullah, Z. Akhtar, and K. Muhammad, "Human behavior understanding in big multimedia data using cnn based facial expression recognition," *Mobile networks and applications*, vol. 25, no. 4, pp. 1611–1621, 2020.

[39] C. Shiranthika, N. Premakumara, H.-L. Chiu, H. Samani, C. Shyalika, and C.-Y. Yang, "Human activity recognition using cnn & lstm," in *2020 5th International Conference on Information Technology Research (ICITR)*. IEEE, 2020, pp. 1–6.

[40] H. Zhang, Z. Nan, T. Yang, Y. Liu, and N. Zheng, "A driving behavior recognition model with bi-lstm and multi-scale cnn," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 284–289.

[41] "Driving behavior," https://www.kaggle.com/datasets/outofskills/driving-behavior,2022, note = Accessed: 2022-07-15.

[42] J. Hu, X. Zhang, and S. Maybank, "Abnormal driving detection with normalized driving behavior data: a deep learning approach," *IEEE transactions on vehicular technology*, vol. 69, no. 7, pp. 6943–6951, 2020.

[43] I. Cojocaru, P.-S. Popescu, and C. Mihaescu, "Driver behaviour analysis based on deep learning algorithms," 202.

[44] M. N. Azadani and A. Boukerche, "Performance evaluation of driving behavior identification models through can-bus data," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.

[45] F. Rustam, A. Mehmood, M. Ahmad, S. Ullah, D. M. Khan, and G. S. Choi, "Classification of shopify app user reviews using novel multi text features," *IEEE Access*, vol. 8, pp. 30 234–30 244, 2020.

[46] J. Fitzgerald, R. M. A. Azad, and C. Ryan, "A bootstrapping approach to reduce over-fitting in genetic programming," in *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, 2013, pp. 1113–1120.

[47] A. Lahouar and J. B. H. Slama, "Day-ahead load forecast using random forest and expert input selection," *Energy Conversion and Management*, vol. 103, pp. 1040–1051, 2015.

[48] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2011.

[49] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1617–1632, 2016.

[50] C. Désir, C. Petitjean, L. Heutte, M. Salaun, and L. Thiberville, "Classification of endomicroscopic images of the lung based on random subwindows and extra-trees," *IEEE transactions on biomedical engineering*, vol. 59, no. 9, pp. 2677–2683, 2012.

[51] G. Grégoire, "4-complements and applications," *Statistics for astrophysics2022, pages=145-180, year=2022, publisher=EDP Sciences*.

[52] N. Zainuddin and A. Selamat, "Sentiment analysis using support vector machine," in *2014 international conference on computer, communications, and control technology (I4CT)*. IEEE, 2014, pp. 333–337.

[53] W. Zheng and Q. Ye, "Sentiment classification of chinese traveler reviews by support vector machine algorithm," in *2009 Third International Symposium on Intelligent Information Technology Application*, vol. 3. IEEE, 2009, pp. 335–338.

[54] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[55] L. Zhang and C. Zhan, "Machine learning in rock facies classification: An application of xgboost," in *International Geophysical Conference, Qingdao, China, 17-20 April 2017*. Society of Exploration Geophysicists and Chinese Petroleum Society, 2017, pp. 1371–1374.

[56] "Driving behavior dataset," https://www.kaggle.com/datasets/shashwatwork/driving-behavior-dataset, accessed: 2022-12-20.

[57] "Carla driver behaviour dataset," https://www.kaggle.com/datasets/dasmehdixtr/carla-driver-behaviour-dataset, accessed: 2022-12-20.

[58] F. Rustam, A. A. Reshi, I. Ashraf, A. Mehmood, S. Ullah, D. M. Khan, and G. S. Choi, "Sensor-based human activity recognition using deep stacked multilayered perceptron model," *IEEE Access*, vol. 8, pp. 218 898–218 910, 2020.

[59] H. U. R. Siddiqui, A. A. Saleem, R. Brown, B. Bademci, E. Lee, F. Rustam, and S. Dudley, "Non-invasive driver drowsiness detection system," *Sensors*, vol. 21, no. 14, p. 4833, 2021.

[60] P. Palimkar, V. Bajaj, A. K. Mal, R. N. Shaw, and A. Ghosh, "Unique action identifier by using magnetometer, accelerometer and gyroscope: Knn approach," in *Advanced Computing and Intelligent Technologies*, M. Bianchini, V. Piuri, S. Das, and R. N. Shaw, Eds. Singapore: Springer Singapore, 2022, pp. 607–631.

[61] M. S. Amin and S. T. H. Rizvi, "Sign gesture classification and recognition using machine learning," *Cybernetics and Systems*, vol. 0, no. 0, pp. 1–15, 2022.

[62] T.-H. Tan, J.-Y. Wu, S.-H. Liu, and M. Gochoo, "Human activity recognition using an ensemble learning algorithm with smartphone sensor data," *Electronics*, vol. 11, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/3/322

Khadija Kanwal received her PhD degree from the University of Science and Technology of China (USTC), Hefei, China in 2022 and her MS degree in Computer Science from Bahauddin Zakariya University, Multan, Pakistan in 2010. Her research interests include Pattern Recognition, Image Enhancement, Image Processing and Machine Learning, Deep Learning, Big Data and Computer Vision.

Furqan Rustam received his MCS degree in the Department of Computer Science, Islamia University of Bahawalpur, Pakistan (Oct-2015 to Oct-2017). Since Nov-2018, he got himself enrolled in Master of Computer Science, Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology (KFUEIT), Rahim Yar Khan, 64200, Pakistan. He is also serving as Research Assistant at Fareed Computing & Research Center, KFUEIT, Pakistan. His recent research interests are related to Data Mining, Machine Learning, and Artificial Intelligence, mainly working on Creative Computing, and Supervised Machine Learning.

Rajasekhar Chaganti is a Cyber Security Engineer at Toyota Research Institute, Los Altos, California, USA. He worked for several technology companies as a cyber security specialist with broad knowledge of SIEM, Vulnerability Management, Endpoint Security, Cloud Security, Network Security, and Application Security. He received his Undergraduate Degree (B-Tech in Electronics and Communication Engineering) with Distinction in 2010, received his Postgraduate Degree (M-Tech in Electronics and Communication Engineering) with Distinction in 2012, and master's degree (MS in Computer Science) in 2018 from the University of Texas at San Antonio, Texas USA. He holds the industry certifications AWS Solution Architect, Certified Ethical Hacker, and CCNA Cyberops. His research interests are ML/AI for security applications, SDN Security, Threat detection in IoT, Cloud, and Blockchain. He also had experience doing patent research projects related to patent novelty, invalidation, and landscape studies in the electronics and computer science domain focusing on cybersecurity, machine learning, IoT security, and cloud computing. He served as a technical program committee member for conferences EAI AFRICOM, FedCSIS, RDAAPS, ICRIMB, CICEN and I3SC. He has been a reviewer for IEEE Access, Information and computer security, Springer cybersecurity Journal, Wireless communications and mobile computing, Journal of cybersecurity and mobility, Cybersecurity skills journal, and more. He is also actively working for the nonprofit organization cybertrustamerica to promote cybersecurity.

ANCA D. JURCUT received the Bachelor of Mathematics and Computer Science degree from the West University of Timisoara, Romania, in 2007, and the Ph.D. degree from the University of Limerick, Ireland, in 2013. From 2008 to 2013, she was a Research Assistant with the Data Communication Security Laboratory, University of Limerick. From 2013 to 2015, she was working as a Postdoctoral Researcher with the Department of Electronic and Computer Engineering, University of Limerick. She was a Software Engineer with IBM, Ireland. Since 2015, she has been an Assistant Professor with the School of Computer Science, University College Dublin (UCD), Ireland. She is currently an Assistant Professor with the School of Computer Science, UCD. Her research interests include network and data security, security for the Internet of Things (IoT), security protocols, formal verification techniques, and applications of blockchain technologies in cybersecurity.

Imran Ashraf received his Ph.D. in Information and Communication Engineering from Yeungnam University, South Korea in 2018, and the M.S. degree in computer science from the Blekinge Institute of Technology, Karlskrona, Sweden, in 2010. He has worked as a postdoctoral fellow at Yeungnam University, as well. He is currently working as an Assistant Professor at the Information and Communication Engineering Department, Yeungnam University, Gyeongsan, South Korea. His research areas include indoor positioning and localization in 5G and beyond, indoor location-based services in wireless communication, smart sensors for smart cars, and data analytics.