Article

Driver Behavior Classification System Analysis Using Machine Learning Methods

Raymond Ghandour *, Albert Jose Potams [ID], Ilyes Boulkaibet, Bilel Neji * and Zaher Al Barakeh

College of Engineering and Technology, American University of the Middle East, Kuwait;
albert.potams@aum.edu.kw (A.J.P.); ilyes.boulkaibet@aum.edu.kw (I.B.); zaher.al-barakeh@aum.edu.kw (Z.A.B.)
* Correspondence: raymond.ghandour@aum.edu.kw (R.G.); bilel.neji@aum.edu.kw (B.N.)

Abstract: Distraction while driving occurs when a driver is engaged in non-driving activities. These activities reduce the driver's attention and focus on the road, therefore increasing the risk of accidents. As a consequence, the number of accidents increases and infrastructure is damaged. Cars are now equipped with different safety precautions that ensure driver awareness and attention at all times. The first step for such systems is to define whether the driver is distracted or not. Different methods are proposed to detect such distractions, but they lack efficiency when tested in real-life situations. In this paper, four machine learning classification methods are implemented and compared to identify drivers' behavior and distraction situations based on real data corresponding to different behaviors such as aggressive, drowsy and normal. The data were randomized for a better application of the methods. We demonstrate that the gradient boosting method outperforms the other used classifiers.

Keywords: machine learning; driver behavior; classification; ANN; logistic regression; random forest

1. Introduction

The increasing number of vehicles in the world raises the issue of traffic and road safety. Enhancing road safety and reducing accident numbers is now a high priority for all governments. The lack of concentration of drivers on the road or driver distraction are the major causes of these accidents. Despite the fact that the driving task requires continuous attention to be paid to the road and the surrounding vehicles, drivers are usually distracted by other activities such as answering a phone call, looking at a billboard, chatting, eating, talking with passengers and radio calibration. In addition, distraction may be the result of a stressful situation that induces an abnormal driving behavior. Stressful situations are now very common and are due to impaired driving, traffic jam, excessive workload, personal problems, etc. Distraction situations due to stress can be reflected in driver behavior, which can be represented by an aggressive driving, a lane departure or hard, repetitive braking. Moreover, the mental and medical states of a driver impact their driving behavior. If a driver is sick or very tired, this may lead to drowsy driving.

It is estimated that in the USA, 14% of the total number of traffic crashes reported by the police in 2018 was due to a distracted driver, while the number of fatal crashes due to distracted drivers amounted to 8% [1]. It is also important to note that of a total of 2841 reported fatalities with a distracted driver in 2018 by the police in the USA, 61% involved the driver's loss of concentration, 21% involved passengers, 14% involved pedestrians and 3% involved cyclists [1]. This directly shows that distracted drivers pose a major hazard to themselves as well as people in their vicinity. In order to avoid such incidents, efforts have been made to ensure drivers' constant awareness utilizing different approaches and prediction techniques [2]. As a matter of fact, car manufacturers have already implemented different detection techniques that monitor and predict driver behavior in the hope of increasing total vehicle safety [2]. Such systems tend to be costly and encumbering due to the multitude of variables and needed sensors, as well as potential ethical and privacy-related aspects. The magnitude of processed information is introducing new constraints

regarding the system time response and computational power consumption put in place by manufacturers or regulations [3]. Hence, finding the best processing method needed to process all of this information is a major issue.

In this paper, we investigate an efficient classification method based on the machine learning approach that can be utilized for driver behavior and distraction detection. Therefore, different classification methods are applied on a set of data recorded from real driving tests and compared together in order to define the most efficient among them. The following section presents a summary of the driver distraction factors and detection and classification methods. Section 3 presents the data description and analysis. The proposed classification methods are presented in Section 4. Then, the simulation results and comparison are detailed and discussed in Section 5. Finally, a conclusion and future works are proposed to enhance road safety.

2. Driver Behavior Overview

This section presents an overview of the factors affecting drivers' behavior, driver behavior sensing methods and driving behavior classification models.

2.1. Factors Affecting Drivers' Behavior

While driving, a driver's behaviors change when the driver is engaged in other, non-driving activities. This causes risks in terms of road safety as the drivers should be fully concentrated on their driving task. A driver in a distracting situation will most probably fail to respond to hazards and/or to anticipate them. Driver distraction has been classified by several researchers into two categories: internal and external distractions. Internal distraction can happen when the driver is eating, chatting and using entertainment systems. External distraction occurs when a driver looks at the outside of their vehicle as well as roadside advertising panels. Another classification presented in the literature classifies driver distraction into different types: most commonly auditory, biomedical, cognitive and visual. Auditory distraction happens when a driver is disturbed by a sound that breaks their concentration and focus on the road. This can be due to music, listening to the radio or chatting with other passengers [4]. Biomedical or physical distraction occurs when drivers are not controlling the steering wheel because their hands are used to answer a call, eating or adjusting vehicle settings [5]. Cognitive distraction is related to the driver's mind. This is represented by a reduction of the driver's awareness and concentration on the road. It can be due to mental pressure or personal/financial issues [6–8]. Visual distraction occurs when the driver's eyes are off the road. This can happen when the driver is adjusting a GPS address or looking at a billboard on the roadside [9]. A combination of these distraction factors will affect the driver's behavior and will lead to very dangerous driving situations [10].

2.2. Driving Behavior Sensing Methods

Several sensing modalities have been used in the literature to identify driving behavior. They have been classified into three types of data: vehicle control data, visual data and physiological data [10]. In the vehicle control data type, the identification is based on vehicle dynamics data such as the supervision of the steering wheel, the pedal position or the throttle hold rate [11,12]. Several researchers have proposed steering wheel patterns to detect driver behavior. They stated that distracted drivers tend to increase the distance with the leading vehicle or tend to drive faster than normal [13]. Others proposed steering wheel metrics to detect abnormal behaviors [14], such as the standard deviation of the steering wheel angle, the steering wheel reversal rate [15], the high frequency of steering [16] and the steering entropy [17,18]. In the visual data type, the driver behavior is deduced from images and videos of the driver's facial expressions [19,20], the movements of the eye ball [21–26], head movements [27–30] and body movements [31]. In the physiological data type, a driver's abnormal behavior is detected by analyzing brain activity signals and their heart rate. These signals are used to detect the major distraction factor, which is driver

fatigue [32,33]. Physiological data can now be acquired using nano sensors embedded in human bodies [34,35].

2.3. Driving Behavior Classification Methods

Several methods and techniques have been used in the literature to detect abnormal driver behavior, such as artificial neural networks, gradient boosting machine [36], dynamic Bayesian network, logic regressions, support vector machine and various machine learning methods: decision tree, random forest, k-NN, SVM and Naive Bayes [37–40]. They have been used to detect the physiological and visual attributes to identify and to propose distraction reduction techniques. Some of these methods were used to detect the vehicle dynamics to determine abnormal driving situations, such as the double-class classifier based on Gaussian mixture model [41] and hidden Markov models to simulate driver behavior based on car following, lane changes and pedal orientation [42]. Naive Bayes (BN) has also been used to predict driver behavior based on preliminary collected data [43]. Driver behavior is detected based on vehicle velocity and steering wheel data using many sensors including a GPS unit, cameras and ranging sensors (ultrasonic and laser sensors) to monitor the turning patterns of the steering wheel and recognize lane and acceleration patterns. The data are compared to reference data reflecting a normal driving condition to detect abnormal driver behavior [44–46].

3. Data Description and Analysis

In this paper, a large dataset, called UAH-DriveSet, obtained from six different drivers and vehicles, is used for driving behavioral analysis [47]. The data were created by Romera et al. [47] to include three different driving behaviors: normal, drowsy and aggressive. Furthermore, two different types of roads, motorway roads and secondary roads, were used to record the data. As a result, a data set with more than 500 min of naturalistic driving time was recorded, in which raw and processed sensor data were included together with video recordings of the trips. This data set is represented in 3.2 GB of recorded data encompassing video recordings, sensory data (raw data) and processed data. For more simplicity, the features built for the final data were a simplification of the raw data [47] with a much richer relation to the classes of normal, drowsy and aggressive behaviors. More details about data features and comparisons are presented in the next subsections.

3.1. Breakdown of Data Set

In the modern world, the types of roads, level of traffic and other parameters describing drivers environment are highly varied. Building a model which encompasses all these conditions, to be able to read into a driver's mental state, can be quite challenging. To be able to read into subtle details that distinguish a person with heavy eyes driving through a lonely path from another cutting through traffic while providing similar features is one example of many such cases. Therefore, a model has been proposed where limitless data sets of different driver environments can be constantly added and the algorithms proposed can consistently learn and adapt for use in the real world. The original data set has been classified into two example data sets of drivers traversing through Motorways and Secondary roads with lesser traffic and restricted lane width. These pathways have each been driven through by six different drivers providing diversity in our data bank. Furthermore, each of these data sets have further been classified into two models which focus on the car, its orientation with the road and the overall traffic in the driver environment. Select features have been picked which have a high correlation with the target data set. A search for consistency along with encouraging results would be the primary focus of our models for varied driver environments and traversing roads.

3.2. Data Sets for Application of Machine Learning Models

Based on the early work provided by Romera et al. [47], the information used to create the classification models focuses on features that give a high correlation with our target

classification set. Two sets of data were referenced to run the different models [47]: "lane detection" with a set of 214,151 sample points and "traffic status" with a set of 46,542 sample points corresponding to the motorway class. Using the available data, two different feature sets were constructed, in which the first one, lane detection, focused on the position and orientation of the car of interest. The second feature set, traffic status, related to the car's environment. Both features sets are described as follows:

1. Lane detection

    (a) Timestamp (seconds);
    (b) Position of car relative to lane center (X) (meters);
    (c) Orientation of car relative to lane curvature (phi) (degrees);
    (d) Lane width (Width_road) (meters);
    (e) Driver state of mind (status) (0 = normal, 1 = aggressive, 2 = drowsy).

2. Traffic status

    (a) Timestamp (seconds);
    (b) Distance to car ahead in nearest lane (Dist_vehicle) (meters);
    (c) Time for impact with nearest car (Impact_time) (seconds);
    (d) Number of vehicles in the same frame (Num_vehicles);
    (e) GPS speed (GPS_speed) (km/h);
    (f) Driver state of mind (0 = normal, 1 = aggressive, 2 = drowsy).

The data were collected from six different drivers [47] for each of the above sets and corresponding to the motorway class to compare the results from each of the models and search for consistency in the results. Then, the data were randomized before analyzing and applying any machine learning classifiers. The initial data that were in the order of "normal", "drowsy" and "aggressive" sets were shuffled to ensure that the models provided a good representation of their performance for more practical, varied test sets.

3.3. Feature Comparison, Profiling and Analysis

To address the dimension problem, Romera et al. [47] proposed a way to track the lane marking based on rear camera images instead of using the whole image datasets. In this section, a feature comparison is performed to improve the types of features used to create the classification models.

3.3.1. Feature Comparison

First, a pair-plot is performed to carry out a correlation of the parameters. Distinct characteristics in each of the data sets [47] based on the status of the driver can be observed. The comparison between characteristics of the "lane detection" set is carried out as shown below.

(a) As can be seen in Figure 1, there is a distinct relation between the "aggressive" and "drowsy" target variables with the inclination of the car to the lane centre. It can be noticed that there is an unexpected positioning of the car in the lane center for "drowsy" cases.

(b) The car orientation relative to the lane curvature seems to be opposite for several "aggressive" and "drowsy" cases. Given the number of cases, the clear distinction between the two target variables gives explicit information to work with.

(c) Next, regarding the comparison between the status and width of the road, while it is clear that the number of cases of drivers being aggressive seems to increase with either narrow roads or—even more so—wide roads, the number of cases showing an increase among drowsy drivers is relatively significant.
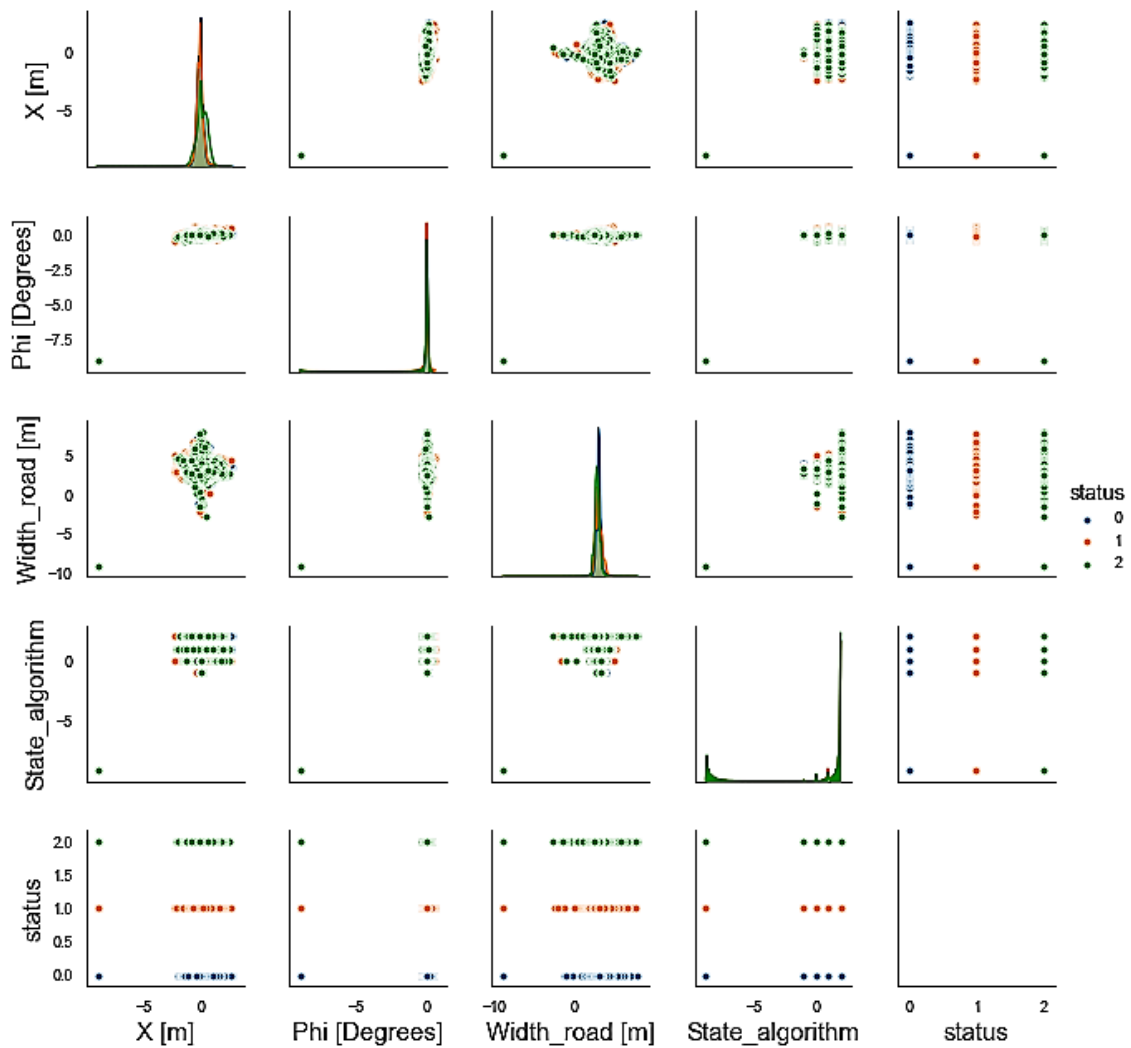
Figure 1. Feature comparison of "lane detection" dataset.

Next, a comparison between the characteristics of the "traffic detection" set is presented.

(a)   As can be seen in Figure 2, the distance from the nearest car does not appear to affect the drowsy cases, which can be explained from the even spread. However, the cases of aggressive driving are split between those with little distance, causing drivers to possibly make dangerous maneuvers, and those having a significant distance between cars, encouraging them to race.

(b)   The above characteristics can also be observed with cases where the time of impact between the car of concern and the car in front is considered. With a reduced time of impact, drivers seem to opt for the same choice of driving normally.

(c)   As the number of vehicles increases, the likelihood of drivers going at a controlled, normal pace seems to be implied from the relation between the number of vehicles and the type of driving.

(d)   The GPS speed can be most easily interpreted, with an extremely high pace more often than not suggesting aggressive driving. The roads and their corresponding speed limits may be something to be considered in the future with respect to this parameter.
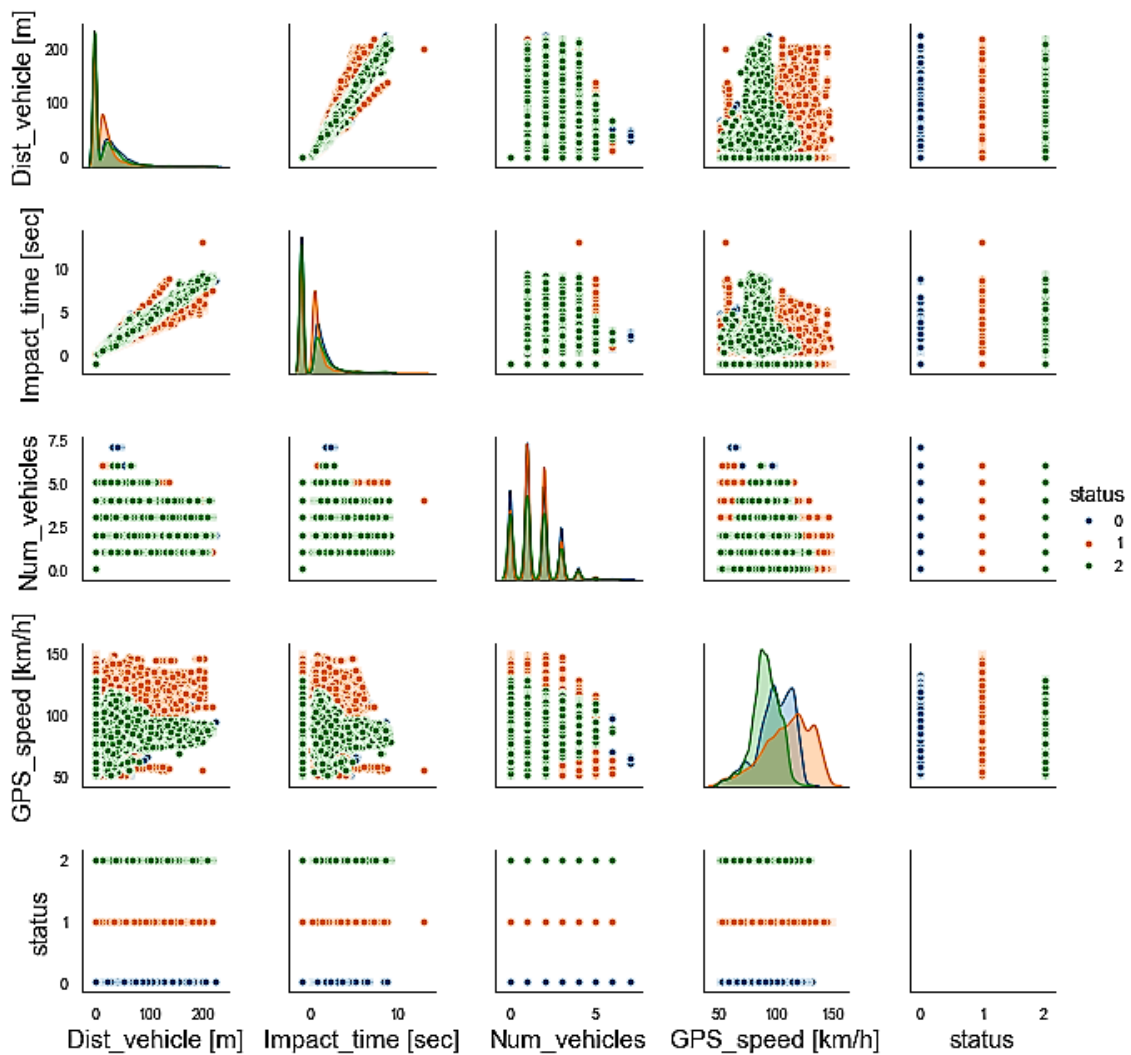
Figure 2. Feature comparison of "traffic detection" dataset.

3.3.2. Profiling

The profiling of features used for the listed models includes a complete description of the parameters and the value spread. The Pandas profiling library has been used for feature profiling and has also been adopted for the motorway roads class.

3.3.3. Analysis

A complete description and analysis of each of the features in both "lane detection" and "traffic status" was carried out. These were used to control the parameters during the machine learning algorithms based on their relevance to the target status.

4. Proposed Classification Methods

A large panel of classification methods have been used to model driver behavior in the literature. An extensive survey shows that most of these methods utilize statistical regression algorithms in order to classify different states of drivers based on a set of data representing certain physical aspects. Among these classification methods, we can recognize probabilistic, hierarchical or clustering algorithms. In this section, we discuss some of the commonly used algorithms and methods for driver behavior detection.

4.1. Logistic Regression

Since the late 1980s, the logistic regression algorithm has witnessed rising interest with increased utilization in research. It is well suited for applications whose main objective is to analyze categorical outcome variables. Logistic regression is usually used to obtain binary regression models where the output is either "0" or "1". It is a simple approach utilized when the desired outcome is binary [48].

4.2. Artificial Neural Networks

Neural networks and other hierarchical algorithms such as deep learning and SVN are widely used in situations dealing with a large set of data and variables. These methods' main advantages are the adaptability of the model architecture towards any application and the learning capability, which have led researchers to opt to use them in almost every possible scenario. Inspired by the human brain and created to solve problems in the same manner as human brains, the architecture of these methods is defined by the interconnection of different neurons over different layers of hierarchy progressing from input data to the desired outputs. Different versions of these algorithms exist or are being developed to serve different objectives; i.e., recognizing hidden patterns, correlations between raw data, classification and clustering and the possibility of creating dynamic models that are able to adapt to time-variant situations [49–51].

4.3. Gradient Boosting Classifier

Among the ensemble techniques based on decision trees method, the gradient boosting method is based on the competition-winning concept in which weak learners are iteratively boosted by optimizing a loss function. Generally, this approach tends to focus on observations that were difficult to predict in previous iterations and synthesizes an ensemble of weak learners. This approach has the ability to generate a model by optimizing a differential loss function. The gradient boosting classifier can be summarized by the following elements [52–54].

- A loss function that depends on the problem type is formulated and used to create the prediction model. Note that a logarithmic loss function will be used in the case of building a classifier model. Besides, this boosting algorithm focuses on optimizing, at each stage, the unexplained loss from prior iterations;
- As one of its basic elements, the gradient boosting classifiers use decision trees as a weak learner to make predictions;
- Finally, the concept of additive models is used, in which trees are added one at a time (add weak learners) while the existing trees in the model are not altered. Then, a gradient descent procedure is used to minimize the loss function when adding trees.

The gradient boosting algorithm can be summarized by the following steps:

(a)　Initialization:

$$f_0(x) = \gamma \sum_{i=1}^{N} L(y_i, \gamma) \tag{1}$$

(b)　For $m = 1, 2, \ldots, M$:

- For $i = 1, 2, \ldots, N$:

$$r_{im} = -\Big[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\Big]_{f=f_{m-1}} \tag{2}$$

　–　Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm}, j = 1, 2, \ldots, J_m$

　–　For $j = 1, 2, \ldots, J_m$:

$$\gamma_{jm} = argmin_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma) \tag{3}$$

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x_{jm}) \tag{4}$$

(c)   Output:

$$\hat{f}(x) = f_M(x) \tag{5}$$

4.4. Random Forest Classifier

The random forest (RF) classifiers are considered to be one of the most powerful ensemble classification techniques and are used frequently in the field of data science for solving countless problems across many industrial areas [55,56]. The random forest technique was first introduced by Breiman [55] to integrate an ensemble of randomized decision trees in order to build a prediction model. As one of the ensemble methods, the RF method relies on aggregating the results of an ensemble of decision tree estimators. Generally, decision tree classifiers suffer from high variance, in which the rules obtained by splitting the training data into two random parts to fit two decision trees are likely to be different. To deal with this technical problem, an ensemble of randomized decision trees is used to create parallel estimators; then, by averaging the decision trees, the variance component of the obtained model can be minimized, which will eventually bring the prediction close to an ideal model. Usually, the RF classifier concentrates on sampling both variables and observations of training data to synthesize independent decision trees, and then majority voting is applied to obtain a better classifier. The RF classifier can be summarized as follows [55–57].

- First, M samples are selected from the training set using the bootstrapping algorithm;
- Next, N_{Tree} samples are composed to form N_{Tree} training sets and used to train N_{Tree} parallel decision tree models, respectively;
- For each decision tree model, the best split is selected from the randomly selected M_{Tree} feature subset. Note that no pruning of the decision tree is requested in the splitting process, and the splitting process can only be stopped when all training samples of each tree belong to the same class at the corresponding node;
- Finally, the results obtained by these N_{Tree} decision trees are combined to form one random forest model for new data prediction. Note that the sensitivity, out of bag (OOB) error, accuracy and specificity are statistical factors used to evaluate the performance of the discrimination models of the random forest algorithm. More details about these statistical factors can be found in [55].

5. Experimental Results and Comparison

The algorithms listed in Section 4 were run on both "lane detection" and "traffic status" datasets. The machine learning models were run on a data set split into 70% for training and 30% for testing. A clear distinction between the models is described in this section. Due to the imbalanced data set, the following standard set of evaluation metrics was used to evaluate the performance of the classifiers:

Accuracy: This is a common evaluation metric and is used to compare the four different classification models. Generally, accuracy is used to determine how often the predictions made are true and in favor of or against the event with respect to all the instances of the event;

Precision: This parameter is used to determine the successful predictions from all predictions made in favor of the event. It provides a variable to determine the ability to successfully predict an event;

Recall: This metric is used to determine the successful predictions from all true instances of the event. It provides a parameter to determine how true the made predictions are in favor of an event with respect to all of its instances;

F1-score: This gives a weighted average of the precision and recall metrics. It is the best metric for averaging out and balancing all the evaluation metrics as a whole;

ROC curve: This provides a trade-off between the true positive rate (ratio of correct observations to total number of observations) and false positive rate (ratio of false observations to total number of observations);

Macro and micro evaluation metrics: These are evaluation metrics that are widely used when working with multi-class classification problems. Macro evaluation computes the average of each class evaluation metric, such as precision. Micro evaluation focuses on the data from each class and computes its metric with respect to the entire data.

It should be noted that the dataset used in this paper is imbalanced in terms of the output classes. Therefore, micro evaluation metrics would provide a better understanding of the performance of the models.

5.1. Lane Detection Dataset

The model was run on all the samples from the parameters of the "lane detection" dataset using each of the machine learning algorithms presented in the previous section. The results are presented through the plots and the numerical outcomes observed from the used evaluation metrics.

5.1.1. Logistic Regression Classifier

The model was run in Python with an initial focus on accuracy. The logistic regression model from the "sci-kit library" was run on our data. A range of values for the parameters "solver" and "C" (regularization parameter) was used to run the model. These values were run in a loop to find the logistic regression model that was closest to ideal and its corresponding parameter values for the dataset at hand. Once the preferred parameter values were obtained (C = 0.5, solver = "lbfgs"), all the evaluation metrics of interest, in addition to accuracy, were determined. The visual representation is shown in Figure 3.
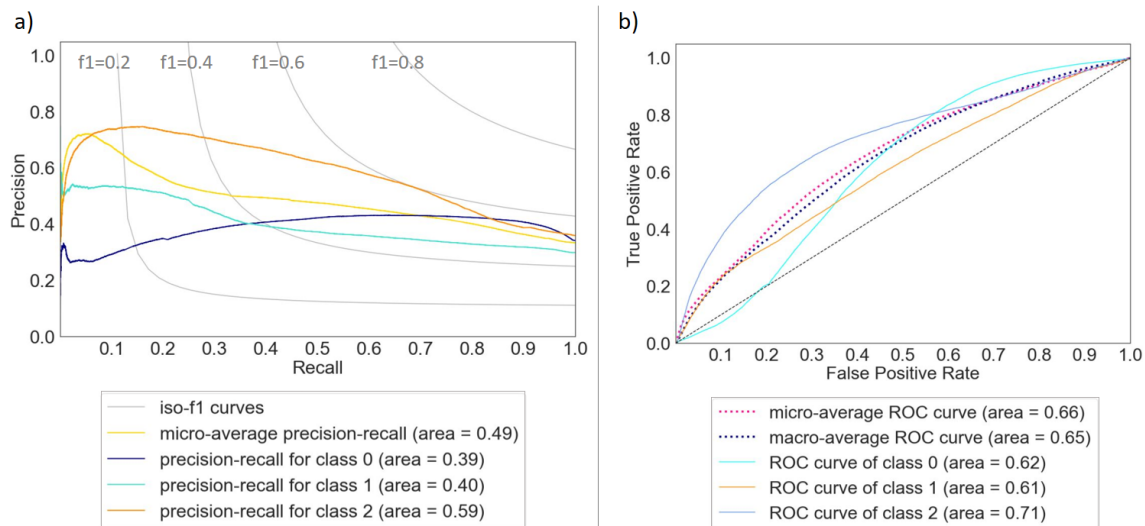

Figure 3. Logistic regression simulation results: (**a**) precision–recall curve; (**b**) ROC curve.

It can be observed from Figure 3a that there is significant lack of consistency between the precision–recall curves of each class. This seems to increase for class 0 (normal), while it drops for classes 1 and 2 (aggressive and drowsy). This is more evident in Table 1. The values also do not seem to show any pattern that can be used to reference future test values.

The ROC curves in Figure 3b do not provide encouraging results either. The curve approaches the diagonal and therefore points to a low performance.

Table 1. Multi-class evaluation metrics with logistic regression classifier for lane detection.

	Precision	Recall	F1-Score	Support
Class 0 (normal)	0.44	0.67	0.53	40,320
Class 1 (aggressive)	0.55	0.03	0.05	35,752
Class 2 (drowsy)	0.53	0.69	0.60	42,206

Furthermore, it is clear that, in the breakdown of class-wise precision, recall and F1-scores, the values of recall and F1-score for class 1 (aggressive cases) are very poor. This is a significant anomaly. One would expect a model with poor scores to have either under-fitting, in which case the class with the fewest data samples (aggressive) would have poor metrics, or to have overfitting, in which case the class with maximum samples (drowsy) would have poor metrics. This is not observed, which further highlights the poor performance of this algorithm.

5.1.2. Gradient Boosting

The classifier ran the model through the data after adjusting certain parameters. These parameters were specifically for the gradient boosting classifier used in Python. The number of estimators and the learning rate were the parameters that were run over a range of values, and the parametric values that gave the best accuracy results (number of estimators = 400 and learning rate = 0.015) for the gradient boosting classifier on the data were used to determine the remaining evaluation metrics.

The precision–recall curve and ROC curve exhibit very good representations of the results. They are presented in Figure 4 and Table 2.
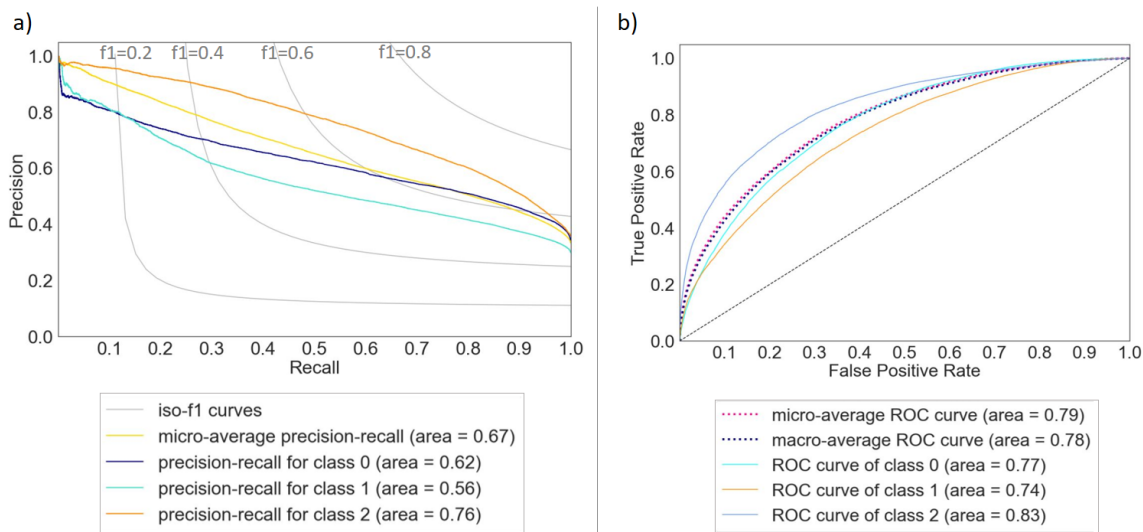

Figure 4. Gradient boosting simulation results: (**a**) precision–recall curve; (**b**) ROC curve.

Two characteristics stand out here: the classes have a consistency in their precision–recall curves and their relation is almost linear, giving a good transition from high precision to high recall. This therefore results in a good F1-score. More importantly, it is observed that the lowest precision–recall curve and correspondingly lowest F1-score are found in the aggressive class (class 1). This is understandable given the smaller set of samples in the aggressive state.

The inclination of all the ROC curves in Figure 4b to the top left corner indicates a high performance. This is supported by the area values for each ROC curve. The high scores in each class also point to how well the algorithm has adapted to the different classes and show encouraging results.

Table 2. Multi-class evaluation metrics with gradient boosting classifier for lane detection.

	Precision	Recall	F1-Score	Support
Class 0 (normal)	0.56	0.65	0.60	40,320
Class 1 (aggressive)	0.57	0.40	0.47	35,752
Class 2 (drowsy)	0.65	0.71	0.68	42,206

The comparable precision, recall and F1-scores overall are very encouraging because they point to a high likelihood of having avoided both under-fitting and over-fitting.

It can be further noticed that the scores are also consistent among each class. However, the one anomaly that remains is the relatively low scores of recall and precision for class 1 (aggressive). They are still much improved compared to the results from logistic regression. The lower scores in this class, as discussed above, most likely stem from the fewer sample points that were available for this class.

The high micro-average precision-recall score, in Figure 4b, helps us to understand that the model has performed well on such an imbalanced data in term of sample points per class.

5.1.3. Random Forests

For this classifier, the parameters used to maximize the output accuracy were criterion, class_weight, max_depth and min_samples_leaf. The model random forest classifier from the Anaconda distribution of Python was run over a range of values for each parameter in individual loops. A plot was made in each case to observe the values for each parameter that provided a high accuracy without overfitting the data. The final set of values used for the parameters of the classifier was as follows: criterion is "Entropy", class_weights are 0 to 0.5 and 1 to 0.5, max_depth is 5 and min_samples_leaf is 5. The corresponding results are presented in Figure 5 and Table 3.
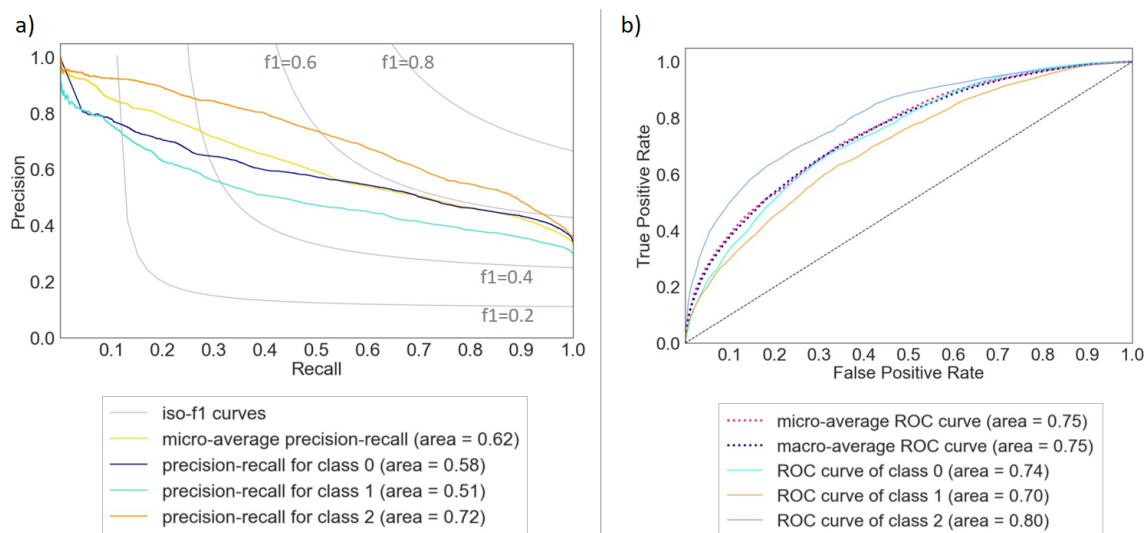

Figure 5. Random forests simulation results: (**a**) precision–recall curve; (**b**) ROC curve.

Similar to the precision–recall curve of gradient boosting, there is a continuous relationship for the metrics of all three classes, and the curves seem to approach linearity. The steep drop in precision with increase in recall is one characteristic that stands out as unfavorable. The instability should preferably be avoided. As can be observed in Figure 5a, the precision–recall curves for all classes individually as well as for the whole dataset show a poor trade-off between precision and recall. An encouraging score for one metric provides a poor score for the other metric.

Table 3. Multi-class evaluation metrics with random forest classifier for lane detection.

	Precision	Recall	F1-Score	Support
Class 0 (normal)	0.53	0.64	0.58	40,320
Class 1 (aggressive)	0.62	0.20	0.31	35,752
Class 2 (drowsy)	0.56	0.78	0.65	42,206

The random forest model also provides a consistent set of precision, recall and F1-scores, leading to a decrease in the possibility of both under-fitting and over-fitting.

The class-wise scores are also encouraging. However, it can be observed that there is a significant variation in the metrics across the different classes. There is also a significant drop in recall and F1-scores for class 1 (aggressive).

5.1.4. Neural Networks

The model was run with deep neural networks, making use of 2 hidden layers with 24 units in each layer for the first test and 48 units per layer in a second test. The "Adam" optimizer was used and the algorithm was run with batches of 40 in a range of 10–50 epochs. The "relu" function was used for the hidden layers and the "sigmoid" function was used for the output layer.

The overall classifier model's accuracy is never greater than 0.30. Multiple epochs, varied numbers of layers and multiple other parameters run independently as well as together provide an accuracy of 0.3 at best. The vast and yet highly varied data sets with different sample count result in an overall poor result.

5.1.5. Results Comparison

To compare the performance of the classifiers used, and taking into consideration the nature of the imbalanced data set, all metric values in the previous tables are combined into a single overall score for each classifier.

Generally, there are several ways to obtain a single performance score for each classifier. In our case, the overall accuracy, which is the ratio between the true predictions made by classifiers over the overall instances, is initially used to evaluate each classifier and obtain the best parameters for each. Another important score is the macro score, which represents a simple arithmetic mean value (in which equal weights are given to all classes) of the scores of each class. It is also used to evaluate the performance of all algorithms. Moreover, the Weighted score can be useful to compare the performance of the of the classifiers used, where this metric weights the score of each class using the number of its samples. Lastly, the micro score is also used to perform the comparison. The micro score considers all the samples from each class and determines the performance score rather than the average of the scores from each class as in the case of macro scores. This is particularly useful to determine the performance of a model on imbalanced datasets such as the one considered in this paper.

Table 4 shows that the gradient boosting algorithm outperforms ANN, RF and LR classifiers. This is due to its flexibility and efficiency in using clone decision trees, where all scores of this algorithm are greater than the other classifiers.

The gradient boosting algorithm has the ability to provide a certain consistency across all classes. On the other hand, the ANN classifier has the worst accuracy results, as all of its metric scores are less than 0.3 (30%). The RF classifier shows acceptable accuracy compared to the logistic regression classifier, where all metric scores are greater than 50%.

Table 4. Lane detection—results comparison.

	Logistic Regression	Gradient Boosting	Random Forests	Neural Networks
Accuracy	0.48	0.60	0.56	0.30
Micro Precision	0.48	0.60	0.56	0.30
Micro Recall	0.48	0.60	0.56	0.30
Micro F1-score	0.48	0.60	0.56	0.29
Macro Precision	0.51	0.59	0.57	0.30
Macro Recall	0.46	0.59	0.54	0.28
Macro F1-score	0.39	0.58	0.51	0.27
Weighted Precision	0.50	0.60	0.57	0.30
Weighted Recall	0.48	0.60	0.56	0.29
Weighted F1-score	0.41	0.59	0.52	0.28

5.2. Traffic Status Dataset

The model was run on all the samples from the parameters of the "traffic status" dataset using each of the machine learning algorithms discussed. The results are presented through the plots and the numerical outcomes observed from the used evaluation metrics.

5.2.1. Logistic Regression

The algorithm was run on a range of values for the parameters "solver" and "C" (regularization parameter), similarly to the "lane detection" dataset. These values were run in a loop to find the most ideal logistic regression model and its corresponding parameter values for the dataset at hand.

Once the preferred parameter values were obtained (C = 0.5, solver = "lbfgs"), all the evaluation metrics of interest in addition to accuracy were determined. The visual representation is shown in Figure 6.
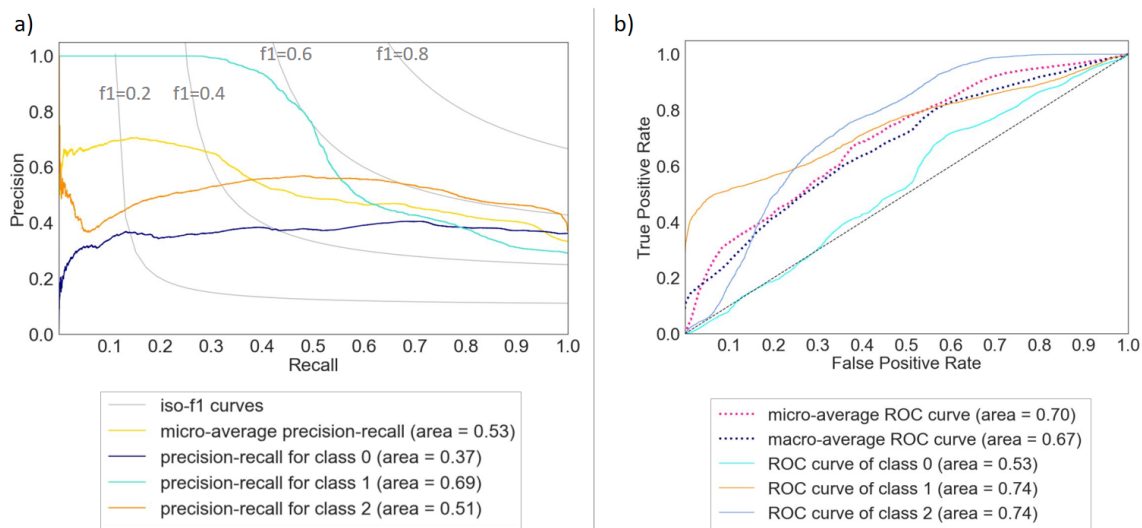

Figure 6. Logistic regression simulation results: (**a**) precision–recall curve; (**b**) ROC curve.

Similar to what is noted in the precision–recall curves of the "lane detection" dataset, the curves across all three curves are highly inconsistent and varied. However, in Figure 6a, precision–recall curves consistent with the amount of data in each class are observed. It is also observed that the aggressive class (class 1) shows a varied precision–recall curve compared to the other two classes.

The low ROC curve suggests the likelihood of the normal class data having been overfit with a much larger sample ratio to the remaining two classes.

The overall metric scores are consistent on an overall scale as well as across the different classes (Table 5). However, the scores are low. The model does not seem to

provide a highly encouraging classification system. The normal class metrics as observed in the ROC curve show poor results relative to the drowsy and aggressive classes.

Table 5. Multi-class evaluation metrics with the logistic regression classifier for traffic status.

	Precision	Recall	F1-Score	Support
Class 0 (normal)	0.49	0.40	0.44	7840
Class 1 (aggressive)	0.56	0.58	0.57	6394
Class 2 (drowsy)	0.55	0.63	0.59	7652

5.2.2. Gradient Boosting

A similar process to that followed with the "lane detection" data was carried out with the "traffic status" data to obtain nearly ideal values for the parameters "number of estimators" and "learning rate" in the gradient boosting classifier. Both numeric and graphical results are presented in Figure 7 and Table 6.
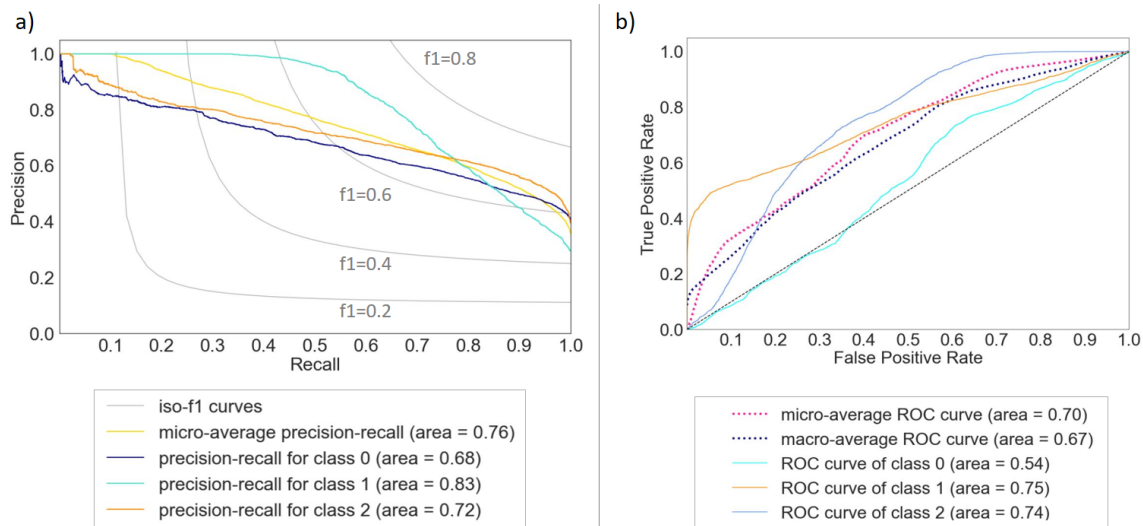

Figure 7. Gradient boosting simulation results: (**a**) precision–recall curve; (**b**) ROC curve.

Promising results from precision–recall curves can be observed, with a high average precision–recall curve for all three classes combined as well. The precision–recall curve for class 1 (aggressive) is understandable given the smaller set of sample points. We also observe that the micro-average precision–recall curve score is high, giving a good indication of how the algorithm has adapted to a multi-class model well.

Similar inferences can be made from Figure 7b. The ROC curves, except class 1 (aggressive), are high and approach the extreme top left. The comparable ROC curve scores illustrate the robustness of the algorithm across various classes.

Table 6. Multi-class evaluation metrics with gradient boosting classifier for traffic status.

	Precision	Recall	F1-Score	Support
Class 0 (normal)	0.63	0.62	0.62	7840
Class 1 (aggressive)	0.86	0.62	0.72	6394
Class 2 (drowsy)	0.62	0.77	0.69	7652

The model is highly consistent across all classes and on average with highly encouraging metric scores. There are different numbers of data samples in each class; despite this, the model seems to provide consistent and much improved metrics.

5.2.3. Random Forests

The same set of parameters was used for the random forest classifier, and the final values for each that give the best results were also the same as for the "lane detection" dataset. The results are listed and plotted in Table 7 and Figure 8.
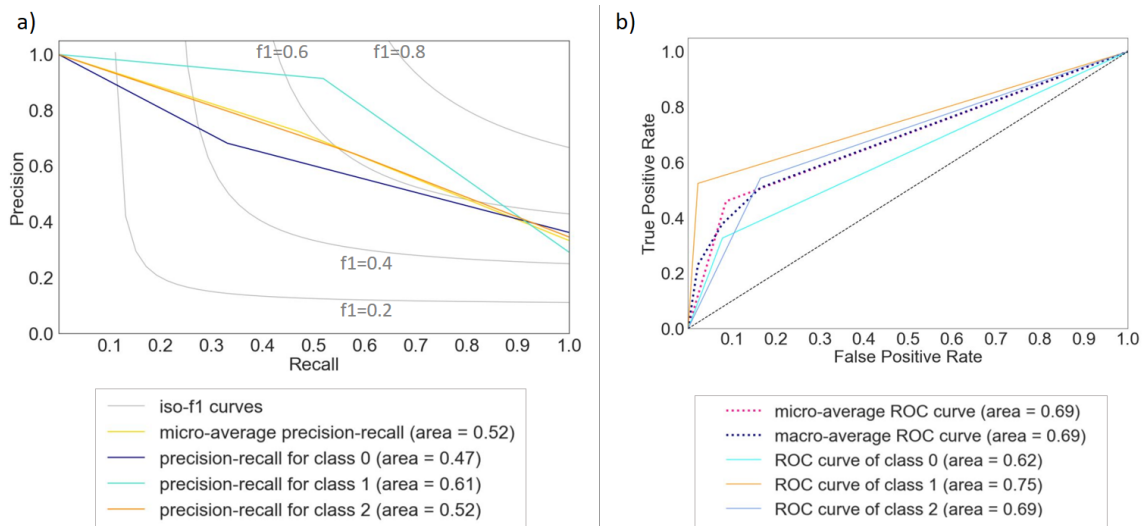

Figure 8. Random forests simulation results: (**a**) precision–recall curve; (**b**) ROC curve.

The results are very similar in characteristics to those obtained with the "lane detection" dataset. We obtained encouraging metric curves but with linear relations that were slightly too steep, as observed in Figure 8a. The class 1 curve is quite different from the others. In Figure 8b, it can be noticed that the ROC curve area is average. The curves are inclined to the extreme top left at the middle, which is preferred, but the variation is too steep and short lived. Here again, the ability of the model to be able to provide consistent results has to be questioned.

Table 7. Multi-class evaluation metrics with random forest classifier for traffic status.

	Precision	Recall	F1-Score	Support
Class 0 (normal)	0.61	0.52	0.56	7840
Class 1 (aggressive)	0.88	0.56	0.68	6394
Class 2 (drowsy)	0.55	0.79	0.65	7652

The random forest metrics above suggest good results as well, but they are quite varied in terms of the precision, recall and F1-score, giving different levels of performance. It can also be noted that the performances across the classes are also quite unevenly distributed.

5.2.4. Neural Networks

The starting network parameters were the same as those used for the "lane detection" dataset. The number of epochs and batch sizes were varied over the same range. Similar results to those observed with the "lane detection" dataset could be observed, with the overall model accuracy not exceeding 0.29. In addition, multiple epochs, a varied number of layers and multiple other parameters including the activation functions were run independently as well as together in loops, providing an accuracy at best of 0.29. The vast and yet highly varied data sets, with different sample numbers, result in poor overall results again.

5.2.5. Result Comparison

Similarly to the results obtained using the lane detection dataset, the gradient boosting algorithm outperformed all other classifiers, where its micro precision score reached 0.67 (67%) as indicated in Table 8. Moreover, the ANN classifier provided the worst accuracy results, where all of its metric scores were less than 0.3 (30%). Again, the RF classifier shows improvements compared to the lane detection dataset results and provides an acceptable accuracy compared to the LR classifier with metric scores of more than 63%. The logistic regression classifier did not perform better than GB and RF classifiers, yet it shows some accuracy improvements compared to the lane detection dataset results and provides acceptable metric scores (more than 53%). The focus on micro scores is in keeping with the need for a metric that accounts for imbalanced datasets.

Table 8. Traffic status—results comparison.

	Logistic Regression	Gradient Boosting	Random Forests	Neural Networks
Accuracy	0.54	0.67	0.63	0.29
Micro Precision	0.54	0.67	0.63	0.29
Micro Recall	0.54	0.67	0.63	0.26
Micro F1-score	0.54	0.67	0.63	0.25
Macro Precision	0.53	0.70	0.68	0.29
Macro Recall	0.54	0.67	0.63	0.27
Macro F1-score	0.53	0.68	0.63	0.25
Weighted Precision	0.53	0.69	0.67	0.29
Weighted Recall	0.54	0.67	0.63	0.28
Weighted F1-score	0.53	0.67	0.63	0.25

5.3. Overall Results—Discussion

Overall, the accuracies of GB, RF and LR classifiers increased for the "traffic status" dataset. Generally, despite the fact that some differences may exist, the two sets of data describe a set of features that are connected to each other. The "lane detection" dataset focuses more on the features of the targeted car relative to the road. On the other hand, the "traffic status" dataset describes features relating the target (car of interest) with its surrounding vehicles/objects. As a result, the second dataset provides a rich and diverse set of data to determine the state of mind of the driver. In conclusion, although the obtained results are not the same, the models built based on these datasets gave consistent results, particularly the gradient boosting algorithm.

Despite the use of imbalanced samples for each class (aggressive, normal and drowsy), the gradient boosting algorithm provides consistent and encouraging results with both datasets across all classes. This helps in further substantiating the robustness of the model with various types of data set features. A level of consistency is observed.

The authors in [47] employed all features in one single model with the entire data. This also provides a probabilistic result with the possibility for the driver to be normal, drowsy or aggressive. On the other hand, two feature sets and their corresponding models were run for robustness. The results obtained in this paper provide a clear classification for each set of features. The driver is classified as being in only one out of normal, drowsy or aggressive states. In terms of classification metrics, the accuracies of both models are comparable.

Further research can be built on the improvement of the datasets. Even though the models were run separately on the datasets, consistent results were provided by these models. It should be noted that the two datasets use features that have a high correlation with the target classes. Combining these two datasets and other, more similar datasets could help in providing more concrete classification results. Moreover, the results from various datasets could be used to obtain a weighted percentage of the likelihood of the prediction being true. A concrete value could then be allotted to the driver being in an aggressive, drowsy or normal state of mind.

Furthermore, the model results were obtained from motorways road data. Adding other diverse types of roads, such as secondary roads and their corresponding features, could be used to set up a large database. This would result in a better account of the robustness of the model on various roads, including other factors.

6. Conclusions

In this paper, classification, machine learning and artificial intelligence methods are used to classify driver behavior. Two sets of data were considered: lane detection and traffic status. These data were collected from six different driver behaviors classified into three states: normal, drowsy and aggressive. They were randomized for a better analysis and application of the machine learning models. Logistic regression, gradient boosting, random forest and neural networks results were presented for the two sets. They showed relatively good performance, accuracy and precision, considering the high complexity of the imbalanced data. The obtained results demonstrate that the gradient boosting method outperforms the other used classifiers.

A comprehensive analysis of features from the data set and their relevance to classifying the state of mind of the driver is a key contribution of this work. These features were distributed across two sets and can further be extended to other feature sets of larger and more varied data. The model allows for continuous improvement with increased data. The results also provided a clear distinction in comparison to the probabilistic approach applied in [47].

Following these achievements, and in order to improve our classification results, additional factors will be taken into account, such as the road speed limit and the mental workload. In addition, a hybrid classification system based on the combination of multiple methods will be proposed.

References

1. National Highway Traffic Safety Administration. *Traffic Safety Facts Research Note: Distracted Driving 2018*; NHTSA: Washington, DC, USA, 2020.
2. Qasim Khan, M.; Lee, S. A Comprehensive Survey of Driving Monitoring and Assistance Systems. *Sensors* **2019**, *19*, 2574, doi:10.3390/s19112574. [CrossRef]
3. National Highway Traffic Safety Administration. *Laboratory Test Procedure For FMVSS 111 Rear Visibility (Other than School Buses)*; Department of Transportation, NHTSA: Washington, DC, USA, 2018.
4. Banbury, S.P.; Macken, W.J.; Tremblay, S.; Jones, D.M. Distraction and Short-Term Memory: Phenomena and Practical Implications. *J. Hum. Factors* **2001**, *43*, 12–29. [CrossRef]
5. *Understanding the Distracted Brain: Why Driving While Using Hands-Free Phones Is Risky Behavior*; White Paper 0412 000082045; The National Safety Council: Itasca, IL, USA, 2012.
6. Yuzoff, N.M.; Ahmad, R.F.; Guillet, C.; Malik, A.S.; Saad, N.M. Selection of Measurement Method for Detection of Driver Visual Cognitive Distraction: A Review. *IEEE Access* **2017**, *5*, 22844–22854. [CrossRef]
7. Chun, M.M.; Golomb, G.D.; Turk-Browne, N.B. A taxonomy of external and internal attention. *Annu. Rev. Psychol.* **2011**, *62*, 73–101. [CrossRef]
8. Farah, H.; Zatmeh, S.; Toledo, T; Wagner, P. Impact of distracting activities and drivers' cognitive failures on driving performance. *Adv. Transp. Stud.* **2016**, *1*, 71–82.

9. Calvi, A.; Benedetto, A.; D'Amico, F. Effects of mobile telephone tasks on driving performance: A driving simulator study. *Adv. Transp. Stud.* **2012**, *26*, 29–44. [CrossRef]

10. Ghandour, R.; Neji, B.; El Rifaie, A.; Al Barakeh, Z. Driver distraction and stress detection systems: A review. *Int. J. Eng. Appl. Sci. (IJEAS)* **2020**, *7*, 39–46. [CrossRef]

11. MAcDonald, W.A.; Hoffman, E.R. Review of relationships between steering wheel reversal rate and driving task demand. *J. Hum. Factors* **2020**, *22*, 733–739. [CrossRef]

12. Zylstra, B.; Tsimhomi, O.; Green, P.A.; Mayer, K. *Driving Performance for Dialing, Radio Tuning, and Destination Entry While Driving Straight Roads*; The University of Michigan Transportation Research Institute: Ann Arbor, MI, USA, 2003.

13. Jin, L.; Niu, Q.; Hou, H.; Xian, H.; Wang, Y.; Shi, D. Driver cognitive distraction detection using driving performance measures. *Discret. Dyn. Nat. Soc.* **2012**, *2012*, 432634. [CrossRef]

14. Martinez-Garcia, M.; Gordon, T. A New Model of Human Steering Using Far-Point Error Perception and Multiplicative Control. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018; pp. 1245–1250. doi:10.1109-SMC.2018.00218. [CrossRef]

15. Ostlund, J.; Peters, B.; Thorslund, B.; Ensgtrom, J.; Markkula, G.; Keinath, A.; Markula, G. Driving performance assessment-methods and metrics. In *Technical Report*; 2005. Available online: http://www.aide-eu.org/pdf/sp2_deliv_new/aide_d2_2_5.pdf (accessed on 8 October 2021).

16. McLean, J.R.; Hoffman, E.R. Analysis of Drivers' Control Movements. *J. Hum. Factors* **1971**, *13*, 407–418. [CrossRef]

17. Kircher, K.; Ahlstrom, C. Predicting visual distraction using driving performance data. In *Annals of advances in automotive medicine, Annual Scientific Conference*; Association for the Advancement of Automotive Medicine: Chicage, IL, USA, 2010; Volume 54, pp. 333–342.

18. Boer, E.; Rakauskas, M.; Ward, N.J.; Goodrich, M.A. Steering entropy revisited. In Proceedings of the 3rd International Driving Symposium on Human Factors in Driver Assessment, Rockport, Maine, 27–30 June 2005.

19. Ji, Q.; Zhu, Z.; Lan, P. Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE Trans. Veh. Technol.* **2004**, *53*, 1052–1068. [CrossRef]

20. Park, S.; Trivedi, M. Driver activity analysis for intelligent vehicles: Issues and development framework. In Proceedings of the IEEE Proceedings Intelligent Vehicles Symposium, Las Vegas, NV, USA, 6–8 June 2005; pp. 644–649.

21. Tabrizi, P.R.; Zoroofi, R.A. Drowsiness detection based on brightness and numeral features of eye image. In Proceedings of the Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 12–14 September 2009; pp. 1310–1313.

22. Farber, E.; Foley, J.; Scott, S. Visual attention design limits for its in-vehicle systems: The society of automotive engineers standard for limiting visual distraction while driving. In *Transportation Research Board Annual General Meeting*; National Academies of Sciences, Engineering, and Medicine (NASEM), Washington, DC, USA, 2000; pp. 2–3.

23. Bergasa, L.M.; Nuevo, J.; Sotelo, M.A.; Barea, R.; Lopez, M.E. Real-time system for monitoring driver vigilance. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 63–77. [CrossRef]

24. Ji, Q.; Lan, P.; Looney, C. A probabilistic framework for modeling and real time monitoring human fatigue. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2006**, *36*, 862–875.

25. Craye, C.; Karray, F. Multi-distributions particle filter for eye tracking inside a vehicle. *Image Anal. Recognit.* **2013**, *6*, 407–416.

26. Martinez-Garcia, M.; Eve Zhang, Y.; Gordon, T. Memory Pattern Identification for Feedback Tracking Control in Human-Machine Systems. *Hum. Factors J. Hum. Factors Ergon. Soc.* **2021**, *63*, 210–226. doi:10.1177/0018720819881008. [CrossRef]

27. Murphy-Chutorian, E.; Doshi, A.; Trivedi, M.M. Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In Proceedings of the IEEE Intelligent Transportation Systems Conference, Seattle, Washington, DC, USA, 30 September–3 October 2007; pp. 709–714.

28. Sun, G.; Jin, Y.; Li, Z.; Zhang, F.; Jia, L. A vision-based head status judging algorithm for driving fatigue detection system. *Adv. Transp. Stud.* **2015**, 51–64. doi:10.4399/978885488868505

29. Pohl, J.; Birk, W.; Westervall, L. A driver-distraction-based lane-keeping assistance system. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2007**, *221*, 541–552.

30. Kircher, K.; Ahlstrom, C.; Kircher, A. Comparison of two eye-gaze based real time driver distraction detection algorithms in a small-scale field operational test. In Proceedings of the Fifth International Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, Big Sky, MT, USA, 22–25 June 2009; pp. 16–23.

31. Craye, C.; Karray, F. Driver distraction detection and recognition using RGB-D sensor. *arXiv* **2015**, arXiv:1502.00250.

32. Shiwu, L.; Linhong, W.; Zhifa, Y.; Bingkui, J.; Feiyan, Q.; ZHongkai, Y. An active driver fatigue identification technique using multiple physiological features. In Proceedings of the International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), Jilin, China, 19–22 August 2011; pp. 733–737.

33. Lal, S.K.; Craig, A. Driver fatigue: Electroencephalography and psychological assessment. *Psychophysiology* **2002**, *39*, 313–321. [CrossRef]

34. Neji, B.; Xu, J.; Titus, A.H.; Meltzer, J. Micro-fabricated DC comparison calorimeter for RF power measurement. *Sensors* **2014**, *14*, 20245–20261.. [CrossRef]

35. Neji, B.; Ferko, N.; Ghandour, R.; Karar, A.S.; Arbess, H. Micro-Fabricated RTD Based Sensor for Breathing Analysis and Monitoring. *Sensors* **2021**, *21*, 318. [CrossRef]

36. Huynh, X.P.; Park, S.M.; Kim, Y.G. Detection of driver drowsiness using 3D deep neural network and semi-supervised gradient boosting machine. In Proceedings of the Asian Conference on Computer Vision, in Lecture Notes in Computer Science, Taipei, Taiwan, 20–24 November 2016; Springer: Cham, Switzerland, 2016; Volume 10118, pp. 134–145.

37. Azman, A.; Ibrahim, S.Z.; Meng, Q.; Edirisinghe, E.A. Physiological measurement used in real time experiment to detect driver cognitive distraction. In Proceedings of the International Conference on Electronics, Information, and Communication (ICEIC), Kota Kinabalu, Malaysia, 15–18 January 2014; pp. 1–5.

38. Jiao, Y.; Peng, Y.; Lu, B.L.; Chen, X.; Chen, S.; Wang, C. Recognizing slow eye movement for driver fatigue detection with machine learning approach. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 4035–4041.

39. Jo, J.; Lee, S.J.; Kim, J.; Gi Jung, H.; Park, K.R. Vision based method for detecting driver drowsiness and distraction in driver monitoring system. *Opt. Eng.* **2011**, *50*, 1–24.

40. Alizadeh, V.; Dehzangi, O. The impact of secondary tasks on drivers during naturalistic driving: Analysis of EEG dynamics. In Proceeding of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 2493–2499.

41. Yang, J.; Chang, T.N.; Hou, E. Driver distraction detection for vehicular monitoring. In Proceedings of the 36th Annual Conference of the IEEE Industrial Electronics Society (IECON), Glendale, AZ, USA, 7–10 November 2010; pp. 108–113.

42. Miyajima, C.; Takeda, K. Driver-behavior modeling using on-road driving data: A new application for behavior signal processing. *IEEE Signal Process. Mag.* **2016**, *33*, 14–21.

43. Kumnagai, T.; Akamatsu, M. Prediction of human driving behavior using dynamic Bayesian network. *IEICE Trans. Inf. Syst.* **2006**, *89*, 857–860. [CrossRef]

44. Lanata, A.; Valenza, G.; Greco, A.; Gentili, C.; Bartolozzi, R.; Bucchi, F.; Scilingo, E.P. How the autonomic nervous system and driving style change with incremental stressing conditions during simulated driving. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1505–1517. [CrossRef]

45. Lee, D.S.; Chong, T.W.; Lee, B.G. Stress events detection of driver by wearable glove system. *IEEE Sensors J.* **2017**, *17*, 194–204. [CrossRef]

46. Mas, A.; Merienne, F.; Kemeny, A. Lateral control assistance and driver behavior in emergency situations. *Adv. Transp. Stud.* **2011**. [CrossRef]

47. Romera, E.; Bergasa, L.M.; Arroyo, R. Need data for driver behaviour analysis? Presenting the public UAH-DriveSet. In Proceedings of the International Conference on Intelligent Transportation Systems, Rio de Janeiro, Brazil, 1–4 November 2016; pp. 387–392.

48. Peng, C.Y.J.; Lee, K.L.; Ingersoll, G.M. An introduction to logistic regression analysis and reporting. *J. Educ. Res.* **2002**, *96*, 3–14. [CrossRef]

49. Hassoun, M.H. *Fundamentals of Artificial Neural Networks*; MIT Press: Cambridge, MA, USA, 1995.

50. Said, S.; Boulkaibet, I.; Sheikh, M.; Karar, A.S.; Alkork, S.; Nait-Ali, A. Machine-Learning-Based Muscle Control of a 3D-Printed Bionic Arm. *Sensors* **2020**, *20*, 3144. [CrossRef] [PubMed]

51. Mbuvha, R.; Boulkaibet, I.; Marwala, T. Automatic Relevance Determination Bayesian Neural Networks for Credit Card Default Modelling. *arXiv*, **2019**, arXiv:1906.06382.

52. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232.

53. Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurorobotics* **2013**, *7*, 21.

54. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [CrossRef]

55. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]

56. Lin, X.; Sun, L.; Li, Y.; Guo, Z.; Li, Y.; Zhong, K.; Wang, Q.; Lu, X.; Yang, Y.; Xu, G. A random forest of combined features in the classification of cut tobacco based on gas chromatography fingerprinting. *Talanta* **2010**, *82*, 1571–1575. [CrossRef] [PubMed]

57. Liang, J.; Li, M.; Du, Y.; Yan, C.; Zhang, Y.; Zhang, T.; Zheng, X.; Li, H. Data fusion of laser induced breakdown spectroscopy (LIBS) and infrared spectroscopy (IR) coupled with random forest (RF) for the classification and discrimination of compound salvia miltiorrhiza. *Chemom. Intell. Lab. Syst.* **2020**, *207*, 104179. [CrossRef]