

Enhancing machine learning algorithm performance through feature selection for driver behavior classification

Soukaina Bouhsissin, Nawal Sael, Faouzia Benabbou, Abdelfettah Soutana

Laboratory of Information Technology and Modeling, Faculty of Sciences Ben M'Sik, Hassan II University of Casablanca, Casablanca, Morocco

Article Info

Article history:

Received Feb 16, 2024

Revised Mar 6, 2024

Accepted Mar 20, 2024

Keywords:

Driver behavior

Feature selection

Machine learning

Road safety

UAH-DriveSet

ABSTRACT

Machine learning (ML) techniques empower computers to learn from data and make predictions or decisions in various domains, while preprocessing methods assist in cleaning and transforming data before it can be effectively utilized by ML. Feature selection in ML is a critical process that significantly influences the performance and effectiveness of models. By carefully choosing the most relevant and informative attributes from the dataset, feature selection enhances model accuracy, reduces overfitting, and minimizes computational complexity. In this study, we leverage the UAH-DriveSet dataset to classify driver behavior, employing Filter, embedded, and wrapper methods encompassing 10 distinct feature selection techniques. Through the utilization of diverse ML algorithms, we effectively categorize driver behavior into normal, drowsy, and aggressive classes. The second objective is to employ feature selection techniques to pinpoint the most influential features impacting driver behavior. As a result, random forest emerges as the top-performing classifier, achieving an impressive accuracy of 96.4% and an F1-score of 96.36% using backward feature selection in 7.43 s, while K-nearest neighbour (K-NN) attains an accuracy of 96.29% with forward feature selection in 0.05 s. Following our comprehensive results, we deduce that the primary influential features for studying driver behavior include speed (km/h), course, yaw, impact time, road width, distance to the ahead vehicle, vehicle position, and number of detected vehicles.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Soukaina Bouhsissin

Laboratory of Information Technology and Modeling, Faculty of sciences Ben M'Sik

Hassan II University of Casablanca

Casablanca, Morocco

Email: bouhsissin.soukaina@gmail.com

1. INTRODUCTION

Machine learning (ML) and feature selection are fundamental components within the realm of data science. ML typically furnishes systems with the capability to autonomously learn and improve from experience without explicit, rule-based programming. It is often recognized as one of the leading and contemporary technologies in the era of the fourth industrial revolution [1]. It involves crafting algorithms and models capable of analyzing data, recognizing patterns, and making predictions or decisions. Conversely, feature selection involves the process of identifying the most pertinent and valuable features from a given dataset [2]. In the ML domain, the quality and relevance of the selected features for training a model significantly impacts its performance and accuracy. Opting for the most meaningful features enhances the efficiency and effectiveness of models, while the inclusion of irrelevant or redundant features may cause

overfitting and heightened computational complexity. Hence, feature selection techniques have become imperative in discerning the most suitable and informative features for a specific task.

On the other hand, driver behavior is a multidimensional concept, influenced by many factors, making its accurate description and analysis challenging [3]. These factors include social, cultural, psychological, and environmental factors [4], along with individual driver attributes [5]. In the realm of driver behavior classification, feature selection holds particular significance as it aids in pinpointing the most informative features essential for accurately categorizing distinct driving behaviors [6], [7]. Moreover, is the widespread use of feature selection techniques evident in the literature when it comes to the classification of driver behavior? additionally, are machine learning algorithms consistently delivering compelling results in the context of classifying driver behavior?

Moukafih *et al.* [8], classified behavior into four distinct classes: representing aggressive driving on the highway, depicting aggressive driving on secondary roads, indicating normal driving on the highway, and signifying normal driving on secondary roads. The classification features were selected using the forward-selection method, including speed, acceleration in X, Y, and Z, pitch, roll, yaw, car angle relative to lane curvature, car position relative to lane center, road width, distance to ahead vehicle in the current lane, and time of impact to the ahead vehicle. The authors implemented random forest (RF), adaboost, and ResNet models, achieving respective F1-scores of 94.11%, 92.75%, and 88.29%, showcasing their effectiveness in accurately categorizing driver behavior across these defined classes. Saleh *et al.* [9], categorized driving behavior into three specific classes: normal, aggressive, or drowsy driving, utilizing the UAH-DriveSet dataset. The classification features encompassed acceleration along the x, y, and z axes, pitch angle, roll angle, yaw angle, vehicle speed, distance to the ahead vehicle, and the count of detected vehicles. Their implementation included ML classifiers such as the multi-layer perceptron (MLP) and decision tree (DT) models, achieving respective F1-scores of 48% and 80%. Vyas *et al.* [10] introduced an intelligent recommendation system designed to forecast the influence of prior driving attributes on stress levels, driving behavior, and energy efficiency across different driving and environmental conditions. They incorporated various features such as acceleration in different axes (x, y, and z), pitch angle, roll angle, yaw angle, vehicle speed, the number of detected vehicles, distance to the ahead vehicle, car angle relative to the lane curvature, and car position relative to the lane center. For data preprocessing, the authors employed K-nearest neighbors (KNN) for stress level identification and labeled the data using K-means++. To predict driver stress, they utilized different regression models including linear regression, support vector regression (SVR), and decision tree regressor. The achieved R² values for stress prediction were 0.85, 0.83, and 0.75, respectively. In the context of driver behavior and distraction detection, Ghandour *et al.* [6] focuses on leveraging machine learning classification methods applied to real-world data portraying diverse driving behaviors including aggressive, drowsy, and normal states. Through systematic analysis and randomization of the dataset, the study evaluates the efficacy of four distinct models: logistic regression (LR), RF, gradient boosting, and neural networks (NN). Results reveal varying levels of accuracy, with LR achieving 54%, gradient boosting at 67%, RF scoring 63%, and NN performing at 29%. These models, though applied to the same dataset, showcase different capabilities in effectively classifying and identifying diverse driver behaviors and distraction scenarios. Yi *et al.* [11] delved into the feature distribution analysis of data collected from individual drivers and the collective dataset, employing advanced data visualization techniques and statistical analyses. When substantial disparities were detected, they developed a model aimed at predicting a driver's driving state. Silva and Henrique [12], explored a unique method for identifying maneuvers using vehicle telematics data, focusing on uncovering patterns within time series data. They employed the extended motif discovery algorithm, conducting two distinct experiments: one to recognize accelerations and brakes from longitudinal acceleration time series and another to identify turns from lateral acceleration time series. Silva and Henrique [13], introduced TripMD, a system designed to extract significant driving patterns from sensor recordings such as speed and acceleration. Initially applied to journeys undertaken by a single driver, this system showcased its ability to extract an extensive array of driving patterns. Moreover, these patterns successfully differentiated between various driver behaviors. Furthermore, they demonstrated the system's effectiveness in identifying the driving behavior of an unknown driver among a group of drivers whose behaviors were known by using the patterns derived from TripMD.

From the related works analysis, we concluded that feature selection methods have not been fully utilized to enhance the prediction of driver behavior, and the performance achieved so far by machine learning algorithms in this context still needs improvement. The primary objective of this paper is to highlight the potential of feature selection techniques to enhance the accuracy and effectiveness of ML models for classifying driver behavior. Additionally, the research aims to identify features that directly influence the comprehension of driver behavior. Our methodology comprises an exploratory analysis of driver behavior data sourced from the UAH-DriveSet dataset. To accomplish this, we deploy three distinctive feature selection methodologies: filter methods, wrapper methods, and embedded methods, employing a total

of 10 techniques across these methods. Subsequently, we utilize diverse ML algorithms to categorize driver behavior into normal, drowsy, and aggressive classifications.

The paper's structure is organized as follows: in section 2, we comprehensively outline our research methodology, encompassing details about the dataset, preprocessing steps, the feature selection methods applied, and the ML algorithms utilized. Section 3 presents the results obtained from the application of feature selection and various machine learning algorithms, accompanied by a discussion of these results. Finally, the conclusion and perspective are presented in section 4.

2. RESEARCH METHOD

The objective of our paper is to employ machine learning algorithms for classifying driver behavior while utilizing feature selection methods. This approach aims to identify the most influencing features on driver behavior classification and enhance the overall performance of our ML models. Our methodology workflow is depicted in Figure 1.

The initial phase involves consolidating driving data gathered from various sensors. Subsequently, we perform data preprocessing to enhance its quality. Following this, we systematically apply ten distinct feature selection techniques. Each application is succeeded by a balancing step to ensure a representative dataset. The next stage involves splitting the data into training and testing sets. Finally, we apply eight different ML algorithms to classify driver behavior.

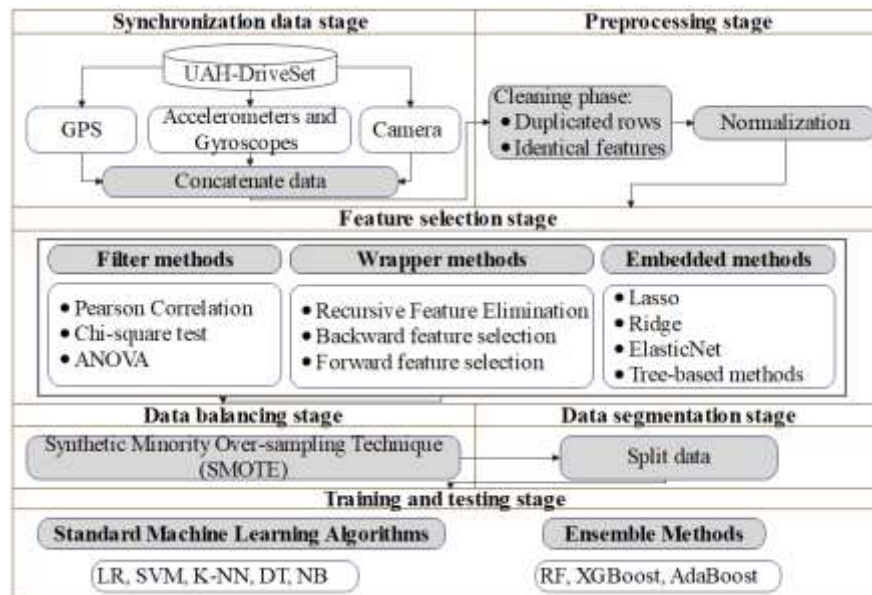


Figure 1. Research methodology

2.1. Dataset description

The UAH-DriveSet [14] dataset encompasses extensive data collected from six drivers operating various vehicles, simulating three distinct driving behaviors: normal, drowsy, and aggressive. These behaviors were observed on both highways and secondary roads. Consequently, the dataset contains over 500 minutes of naturalistic driving data, comprising raw and processed sensor data in addition to video recordings of the trips. The dataset, amounting to 3.4 GB in size, comprises video recordings along with sensory data (both raw and processed), collected through the DriveSafe application. The data collection process involved utilizing smartphone sensors like GPS signals, accelerometers, gyroscopes, and cameras, as described in Figure 2.

2.2. Data preprocessing

Several preprocessing techniques were used in the data preparation [15], including:

- Removal of missing and abnormal data: incomplete drive data and entries with missing or abnormal values were deleted.

- Elimination of duplicated rows: rows containing identical timestamps were removed to ensure unique data entries.
- Removal of redundant data: features showing identical information were eliminated to streamline the dataset.
- Transformation of categorical data: categorical data were encoded into numerical format to enable analysis.
- Data normalization: the dataset was normalized using linear scaling techniques to standardize the values across different features [16]. The following equation was employed to transform the data into a format where feature values fall within the range of 0 to 1.

$$X_i = \frac{(Xi - \min(X_i))}{(\max(X_i) - \min(X_i))} \tag{1}$$

GPS	Accelerometers	Gyroscopes	Camera
<ul style="list-style-type: none"> • Timestamp (seconds), Speed (km/h), Altitude, Latitude, Longitude, Vertical accuracy, Course, Horizontal accuracy, Difcourse: course variation 	<ul style="list-style-type: none"> • Timestamp (seconds), Acceleration in X, Y, Z, Acceleration in X, Y, Z filtered by KF, Boolean of system activated (1 if >50km/h) 	<ul style="list-style-type: none"> • Roll: the rotation of a vehicle about the longitudinal axis. Pitch: the rotation of a vehicle about the transverse axis. Yaw: the rotation of a vehicle about the vertical axis 	<ul style="list-style-type: none"> • Timestamp, X: car position relative to lane center, Phi: car angle relative to lane curvature, GPS speed, W: road width, State of the lane, Number of detected vehicles in this frame, Time of impact to ahead vehicle, Distance to ahead vehicle in current lane

Figure 2. The UAH-DriveSet features

2.3. Feature selection

Feature selection indeed aims to identify the most pertinent features from a dataset with a vast feature space. Various methodologies exist for this purpose, broadly categorized into three main types: wrapper methods, embedded methods, and filter methods [17], [18]. These approaches differ in how they assess and select features based on different criteria or algorithms to determine the most relevant subset of features for a particular task or analysis.

2.3.1. Filter methods

These evaluate features based on statistical measures. Such as pearson correlation, chi-square test, and ANOVA. While computationally efficient and easily interpretable, this method might overlook complex feature relationships.

2.3.2. Wrapper methods

These assess a model's performance by utilizing subsets of features. They involve training models with various feature subsets and selecting the subset that produces the best performance. While this approach offers increased accuracy, it can be computationally demanding, particularly for larger datasets. Techniques employed in this paper within Wrapper methods include recursive feature elimination with RF, backward selection, and forward feature selection.

2.3.3. Embedded methods

These perform feature selection during the model training phase. They automatically determine which features to use during training, reducing the risk of overfitting by dynamically selecting and updating features. Techniques explored within embedded methods consist of: Lasso, Ridge, Elastic Net, and Tree-based methods.

2.4. Balancing data

Balancing classes can offer benefits in machine learning. Yet, generating synthetic data might lead to overfitting, particularly when not executed correctly or in scenarios where the original dataset is limited in size. SMOTE is a prevalent method employed in machine learning to mitigate class imbalance by oversampling the minority class [19].

2.5. Split data

Following the data collection, preparation and feature selection, the subsequent step involves dividing the dataset into training and test sets. The training set serves as the basis for training the machine learning model, while the test set is crucial for evaluating the model's performance. To ensure a representative assessment, the data is randomly split, and in our case, a 70/30 partitioning is applied. This means that 70% of the data is allocated for training purposes, and the remaining 30% is reserved for testing the model's efficacy.

2.6. Machine learning algorithms

In the literature, a wide range of machine learning techniques has been investigated and employed for the classification of driver behavior. The classification of driving behavior relies heavily on the utilization of these diverse ML algorithms, as highlighted in the literature [3]. The algorithms utilized in this paper are outlined below.

2.6.1. Standard machine learning algorithms

ML methods involve applying algorithms that allow computer systems to understand patterns and make judgments or predictions based on data.

- LR is commonly used for binary classification tasks, LR estimates the probability that a given input belongs to a particular class [20].
- SVM proves efficient for tasks involving both classification and regression [21]. SVM identifies the optimal decision boundary (hyperplane) to separate distinct classes or predict values.
- KNN is an algorithm that assigns labels to data points by considering the majority class among their KNN within the feature space [22].
- DT are a supervised learning technique utilized for both classification and regression. They entail constructing a tree-like structure to make decisions based on input features [23].
- Naive bayes (NB) uses Bayes' theorem to classify instances based on the probability of attributes belonging to different classes [24].

2.6.2. Ensemble methods

Ensemble methods are machine learning algorithms that aggregate predictions from numerous different models to provide a more robust and accurate final prediction.

- RF is an ensemble technique that constructs multiple decision trees during training and merges their predictions to improve accuracy and reduce overfitting [25].
- Extreme gradient boosting (XGBoost) is an optimized implementation of gradient boosting that is highly efficient and widely used in various ML competitions [26].
- Adaptive boosting (AdaBoost) is a boosting technique that combines multiple weak learners to create a strong classifier. Each subsequent model gives more weight to misclassified data points from the previous model [26].

2.7. Hyperparameters of feature selection techniques and algorithms

The hyperparameters of feature selection techniques and algorithms encompass crucial settings and configurations that significantly influence their performance. Table 1 presents the hyperparameters associated with feature selection techniques, while Table 2 outlines the hyperparameters specific to ML algorithms. These tables serve as comprehensive references, detailing the key settings and configurations essential for optimizing the performance of each respective technique or algorithm.

Table 1. Hyperparameters of feature selection techniques

Methods	Techniques	Hyperparameters
Filter methods	Pearson correlation	Threshold=0.4
	Chi-square test	Score_func=chi2, k=8
	ANOVA	Score_func=f_classif, k=9
Wrapper methods	Recursive feature elimination	Estimator=base_model, step=1, cv=5 base_model=randomforestregressor(n_estimators=100, max_depth=10)
	Backward feature selection	Estimator=base_model, direction='forward' base_model=kneighborsclassifier(n_neighbors=3)
	Forward feature selection	Estimator=base_model, direction='backward' base_model=kneighborsclassifier(n_neighbors=3)
Embedded methods	Lasso	Alpha=1e-05
	Ridge	Alpha=4.01
	ElasticNet	Alpha=0.1, l1_ratio=0.1
	Tree-based methods	Randomforestclassifier, n_estimators=100, random_state=42

Table 2. Hyperparameters of algorithms

Type	Algorithms	Hyperparameters
Standard machine learning algorithms	LR	Max_iter=1,000, random_state=42, C=1.0
	SVM	Kernel='rbf', C=1.0, gamma='scale', random_state=42
	KNN	N_neighbors=3, weights='uniform', algorithm='auto
	NB	Priors=None, var_smoothing=1e-09
	DT	Min_samples_split=2, min_samples_leaf=1, max_depth=None, criterion='gini'
Ensemble methods	RF	N_estimators=100, criterion='gini', random_state=42, min_samples_split=2, min_samples_leaf=1
	XGBoost	Random_state=42, max_depth=3, learning_rate=0.1, n_estimators=100
	AdaBoost	N_estimators=50, learning_rate=1.0

2.8. Software and hardware

To implement this study, we used python programming language. The experiments were conducted on a Victus by HP laptop equipped with an AMD Ryzen 5 5600H CPU with Radeon Graphics, operating at 3301 MHz, comprising 6 Cores and 12 Threads (8 CPUs). The laptop features 16 GB of RAM and is powered by an NVIDIA GeForce RTX 3050 laptop GPU.

3. RESULTS AND DISCUSSION

3.1. Feature selection results

These findings provide a comprehensive overview of the key features consistently identified by the various methods, contributing to a more nuanced understanding of the significant attributes influencing driver behavior. Based on the results obtained from the feature selection methods, the following set of features were selected across all methods: speed, course, course variation, acceleration in X, Y, and Z filtered by KF, X (meters), phi, road width, roll, pitch, yaw, state algorithm, distance to ahead vehicle, impact time, number of vehicles, number of lanes in the current road, and road type. In Table 3, we have outlined the features selected by each technique in the filter method, wrapper method, and embedded method.

Table 3. Result of feature selection methods

Filter methods	Wrapper methods	Embedded methods
Pearson correlation	Recursive feature elimination	Lasso
Speed, course, course variation, acceleration in X, Y, Z filtered, roll, yaw, X, phi, road width, state of the lane, distance to ahead vehicle	Speed, course, roll, yaw, X, road width, distance to ahead vehicle, impact time, number of vehicles, road type	Speed, course, course variation, acceleration in X, Y, Z filtered, roll, yaw, X, phi, distance to ahead vehicle, impact time, number of lanes, road type
Chi-square test	Backward feature selection	Ridge
Speed, course, pitch, yaw, X, impact time, number of vehicles, number of lanes, road type	Speed, course, course variation, pitch, yaw, road width, impact time, number of lanes, road type	Speed, course variation, acceleration in Y filtered, phi, distance to ahead vehicle, impact time, number of lanes
ANOVA	Forward feature selection	ElasticNet
Speed, course, acceleration in Y filtered, yaw, X: (meters), phi, road width, number of vehicles, road type	Speed, course, course variation, yaw, road width, distance to ahead vehicle, impact time, number of lanes, road type	Speed, course, yaw, number of vehicles Tree-based methods Speed, course, roll, yaw, road width, distance to ahead vehicle, impact time

Based on the results of feature selection, we can conclude that several key features significantly influence driver behavior as shown in Figures 3 and 4. The most prominent features are as follows:

- Speed (km/h): speed appears to have a substantial impact on driver’s behavior. Higher speeds might indicate a more aggressive or risk-prone driving style.
- Course: the course or direction of the vehicle is a vital aspect influencing driver’s behavior, potentially reflecting patterns in navigation and decision-making.
- Yaw: which indicates the rotation around the vertical axis of the vehicle, seems to be closely associated with various driving behaviors, possibly indicating changes in direction or steering patterns.
- Impact time: the time of impact is a crucial indicator, potentially suggesting drivers' responsiveness and ability to manage their vehicle in critical situations.
- Road width: is another influential factor, potentially affecting driving choices, especially in relation to maneuvering and spatial awareness.
- Distance to ahead vehicle: its significance in assessing the space between vehicles and potential driver decision-making.

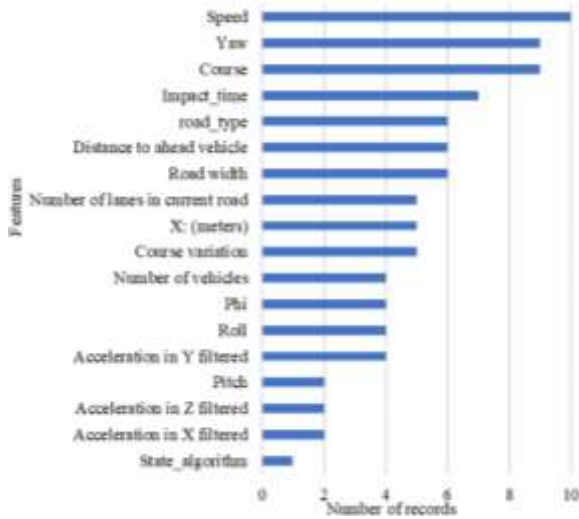


Figure 3. Number of records of each feature

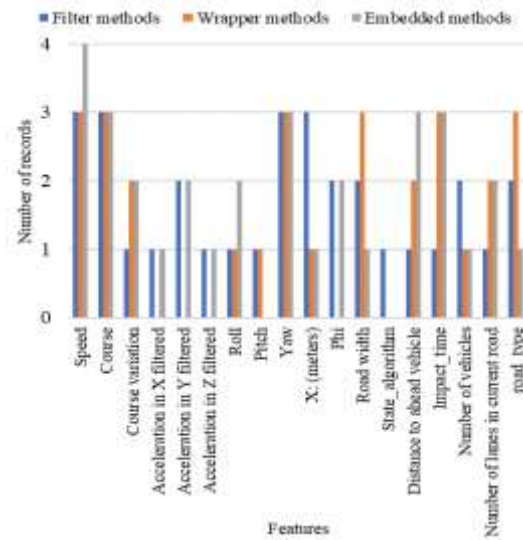


Figure 4. Number of records of each feature in each feature selection method

Moreover, the literature frequently employs certain features in the analysis of driver behavior. Bouhsissin *et al.* [3], these commonly utilized features include speed, acceleration, accelerator pedal, rotation angle, lateral and longitudinal acceleration, as well as time of impact. The frequency of these features suggests a consensus in the literature regarding their significance and relevance for understanding diverse aspects of driver actions.

The convergence of two results indicates the presence of shared features that hold significant importance across both types of studies. Vehicle speed emerges as a prominent indicator, based on its significant presence in the dataset in literature and in our experimentation and it also suggests its relevance in analyzing driving patterns and behavior. Acceleration is indeed a crucial parameter in understanding driver behavior, and both literature and experimentation support this notion as it provides insights into how drivers control their vehicles. The rotation angle of the vehicle emerges as another pivotal factor, its relevance in understanding the rotational movements of the vehicle, which can be crucial for assessing driver behavior during turns or maneuvers. Time of impact is a critical determinant to understanding the potential risk scenarios, and driver responsiveness during critical situations on the road. The distance to the ahead vehicle holds profound importance, offering insights into vehicle spacing and the decision-making processes of drivers as they maintain safe distances on the road. Vehicle position, as reflected in the literature and experimentation, emerges as a noteworthy contributor to our understanding of driver behavior. It provides crucial information about the spatial orientation and positioning of the vehicle, making it a key factor in the analysis of driver behavior across diverse road scenarios. Finally, detected vehicles are instrumental in identifying the presence of other vehicles within proximity. This feature is of utmost importance in assessing driver awareness and responsiveness, especially in multi-vehicle settings, ultimately enhancing our comprehension of driver behavior and its implications for road safety.

3.2. Machine learning results

We will assess our machine learning models by examining metrics like accuracy, precision, recall, and F1-score, in addition to analyzing the mean execution time.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \times 100 \tag{2}$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{3}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{4}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

The results of ML algorithms without feature selection are presented in Table 4. Tables 5 to 7 display the results of ML algorithms with feature selection using filter methods, wrapper methods, and embedded methods respectively.

Table 4. Machine learning algorithms results without feature selection

Type	Algorithms	Accuracy	F1-score	Precision	Recall	Time
Standard machine learning algorithms	LR	59.66	59.73	59.49	61.41	2.19
	SVM	77.45	77.62	77.33	78.60	44.89
	KNN	91.07	91.04	90.96	91.01	0.01
	NB	61.99	52.49	56.04	52.84	0.02
	DT	89.86	89.90	89.80	89.94	0.72
Ensemble methods	RF	94.37	94.33	94.25	94.34	9.04
	XGBoost	95.10	95.14	95.06	95.19	2.14
	AdaBoost	70.31	70.73	70.30	71.55	2.41

Table 5. Machine learning algorithms results with filter methods

Technique	Algorithm	Accuracy	F1-score	Precision	Recall	Time	
Standard machine learning algorithms							
Pearson correlation	LR	52.87	53.19	53.03	54.97	3.05	
	SVM	70.31	69.81	69.66	71.35	55.40	
	KNN	89.55	89.53	89.56	89.51	0.06	
	NB	64.31	56.46	58.83	55.86	0.01	
	DT	88.03	88.10	88.05	88.18	0.47	
Chi-square test	LR	57.05	57.09	56.93	58.59	0.50	
	SVM	75.56	74.67	74.52	76.63	43.64	
	KNN	92.47	92.51	92.43	92.55	0.05	
	NB	57.25	55.08	55.80	54.70	0.02	
ANOVA	DT	90.43	90.40	90.30	90.38	0.23	
	LR	56.07	56.04	55.98	57.59	0.92	
	SVM	75.79	75.63	75.60	77.07	35.41	
	KNN	92.78	92.82	92.83	92.86	0.05	
	NB	61.62	57.80	59.08	57.07	0.03	
	DT	89.90	89.89	89.83	89.88	0.28	
	Ensemble methods						
	pearson correlation	RF	93.15	93.10	92.99	93.09	9.39
XGBoost		93.77	93.81	93.70	93.85	4.20	
AdaBoost		70.60	70.82	70.21	71.79	2.12	
Chi-square test	RF	94.73	94.69	94.58	94.67	7.62	
	XGBoost	93.47	93.50	93.33	93.53	1.36	
	AdaBoost	70.32	70.50	70.19	70.88	1.38	
ANOVA	RF	94.11	94.16	94.11	94.24	8.03	
	XGBoost	93.81	93.87	93.82	93.95	2.13	
	AdaBoost	64.39	64.74	64.50	66.00	2.27	

Table 4 presents insights showcasing the varied performance of different algorithms of ML without feature selection. XGBoost outperforms other models in terms of accuracy, F1-score, precision, and recall, achieving 95.10%, 95.14%, 95.06%, and 95.19%, respectively, while maintaining a moderate processing time of 2.14 seconds. KNN also stands out for its high accuracy at 91.07% and low processing time. RF displays strong performance in accuracy, F1-score, precision, and recall, with 94.37%, 94.33%, 94.25%, and 94.34%, respectively, but requires more time (9.04 seconds) compared to KNN. Conversely, NB and LR exhibit relatively lower performance metrics compared to other models.

The results of the filter methods are presented in Table 5. Notably, among the standard ML algorithms, KNN consistently demonstrates a high level of accuracy across different feature selection techniques. The KNN algorithm achieves an accuracy rate of 92.78% with ANOVA. This indicates its robustness in maintaining high accuracy regardless of the feature selection method used. In terms of ensemble methods, RF and XGBoost consistently exhibit strong performance across different feature selection techniques, maintaining high accuracy rates. RF achieves accuracy rates 94.73% with Chi-square test, while XGBoost achieves accuracy rate 93.81% with ANOVA. These findings highlight the robustness of KNN, DT, RF, and XGBoost across different feature selection techniques, showcasing their stability and efficiency in maintaining high accuracy in classification tasks. Additionally, the feature selection technique using Chi-square test consistently showcases slightly higher accuracy values across most algorithms compared to pearson correlation and ANOVA.

Table 6. Machine learning algorithms results with wrapper methods

Technique	Algorithm	Accuracy	F1-score	Precision	Recall	Time
Standard machine learning algorithms						
Recursive feature elimination	LR	57.51	57.45	57.26	59.46	2.04
	SVM	71.08	71.18	70.97	72.22	35.31
	KNN	89.18	89.23	89.14	89.28	0.09
	NB	57.41	57.34	57.26	57.65	0.02
	DT	83.79	83.77	83.51	83.76	0.24
Backward feature selection	LR	52.90	52.97	52.81	54.58	2.45
	SVM	74.99	73.31	73.42	75.64	61.19
	KNN	96.04	96.05	96.03	96.05	0.15
	NB	56.24	49.95	52.34	50.13	0.03
	DT	91.85	91.79	91.77	91.75	0.33
Forward feature selection	LR	55.95	55.86	55.58	58.02	1.12
	SVM	77.20	75.86	75.61	77.48	41.37
	KNN	96.29	96.27	96.28	96.25	0.05
	NB	51.33	49.81	50.50	49.66	0.02
	DT	92.09	92.05	92.10	92.02	0.32
Ensemble methods						
Recursive feature elimination	RF	89.25	89.25	89.11	89.30	8.13
	XGBoost	86.56	86.66	86.44	86.82	1.65
	AdaBoost	66.32	66.53	66.14	67.46	1.47
Backward feature selection	RF	96.40	96.36	96.33	96.32	7.44
	XGBoost	94.99	94.97	94.93	94.98	5.50
	AdaBoost	64.12	64.59	64.03	65.65	2.26
Forward feature selection	RF	95.77	95.67	95.70	95.61	6.05
	XGBoost	94.59	94.60	94.53	94.64	1.42
	AdaBoost	64.95	65.35	64.76	66.12	1.57

Table 7. Machine learning algorithms results with embedded methods

Technique	Algorithm	Accuracy	F1-score	Precision	Recall	Time
Standard machine learning algorithms						
Lasso	LR	59.97	60.05	59.84	61.51	2.09
	SVM	75.65	74.47	74.06	75.96	46.42
	KNN	90.04	90.03	89.92	90.02	0.07
	NB	62.98	56.14	58.45	55.78	0.02
	DT	88.30	88.32	88.17	88.33	0.42
Ridge	LR	53.04	52.91	52.75	54.57	1.01
	SVM	60.65	61.14	60.65	62.18	59.63
	KNN	63.38	63.72	63.53	64.47	0.04
	NB	55.48	45.08	50.50	46.86	0.02
	DT	61.35	61.57	61.21	61.84	0.30
ElasticNet	LR	52.66	52.04	52.16	54.69	0.19
	SVM	68.95	67.04	67.35	69.92	38.81
	KNN	90.82	90.84	90.91	90.86	0.03
	NB	52.00	52.11	51.94	53.31	0.01
	DT	89.16	89.18	89.18	89.21	0.18
Tree-based methods	LR	53.72	53.20	53.48	56.57	0.72
	SVM	74.89	73.57	73.28	75.16	39.79
	KNN	95.36	95.31	95.31	95.26	0.04
	NB	53.16	53.41	52.74	54.10	0.02
	DT	91.38	91.37	91.32	91.37	0.24
Ensemble methods						
Lasso	RF	92.31	92.28	92.17	92.33	8.84
	XGBoost	92.96	92.99	92.83	93.04	6.34
	AdaBoost	69.43	69.64	69.21	70.47	2.98
Ridge	RF	69.87	69.90	69.43	69.93	5.89
	XGBoost	69.13	68.99	68.48	68.86	4.17
	AdaBoost	61.45	61.67	60.98	62.54	1.50
ElasticNet	RF	92.34	92.36	92.38	92.42	5.00
	XGBoost	87.91	88.09	87.98	88.54	0.96
	AdaBoost	62.94	61.43	61.95	64.91	1.00
Tree-based methods	RF	95.53	95.44	95.43	95.36	7.49
	XGBoost	94.03	94.06	93.94	94.14	1.48
	AdaBoost	67.08	67.25	66.59	68.63	1.49

Based on the results presented in Table 6, which illustrates the outcomes obtained from the wrapper methods. The KNN stood out as the most robust standard ML algorithm, exhibiting high accuracy in all feature selection methods with 96.29%. Ensemble methods like RF and XGBoost displayed more consistent

performance across feature selection techniques compared to standard ML algorithms with 96.40% and 94.99%. NB and LR generally displayed lower accuracy and performance compared to KNN and ensemble methods across all feature selection techniques. Forward and backward feature selection methods showed varying impacts on the algorithms' performance across different models. These findings suggest that different feature selection methods can significantly impact the performance of ML models, with KNN and RF exhibiting more robust performance across multiple techniques.

Table 7 displays the outcomes of ML algorithms employing embedded methods. Notably, among the standard ML algorithms, KNN and DT demonstrated superior accuracy and performance in multiple feature selection techniques. They achieve exceptional accuracy rates of 95.36% and 91.38%, respectively, specifically with Tree-based methods. Ensemble methods, especially RF and XGBoost, consistently performed better across various feature selection methods compared to standard ML algorithms. RF and XGBoost notably achieved impressive accuracy rates of 95.53% and 94.03%, respectively, especially with Tree-based methods. These findings emphasize the strength of ensemble methods, particularly RF and XGBoost, in delivering high accuracy, showcasing their potential as robust models for classification tasks when coupled with appropriate feature selection techniques, such as Tree-based method.

The Figure 5 encompasses a comprehensive assessment of various algorithms' accuracy and execution time across different feature selection techniques. The highest accuracy of 96.4% was achieved by RF with backward feature selection, then forward feature selection with KNN reached an accuracy of 96.29%, also KNN achieved an accuracy of 96.04% with backward feature selection. Conversely, certain algorithms exhibited lower accuracies across various feature selection techniques, such as LR, and NB, all achieving accuracies below 65%. In terms of time, the KNN algorithm demonstrates faster processing than the RF algorithm. KNN achieves an accuracy of 96.04% with the backward feature selection technique in 0.15 seconds, while RF achieves an accuracy of 96.4% in 7.44 seconds.

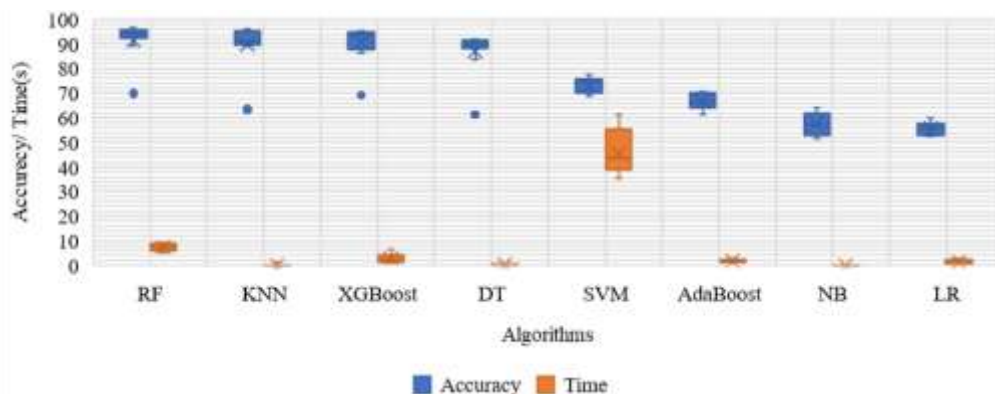


Figure 5. Comparison of accuracy and execution time results of all machine learning algorithms

3.3. Discussion

In summary, this study demonstrates how feature selection techniques enhance ML models' performance for driver behavior classification. Additionally, we introduce a ML process aimed at the development of these models, facilitating an assessment of their performance. Furthermore, the study aims to identify pivotal features that contribute to or shape driver behavior. The results highlight the varied performance of algorithms when combined with various feature selection techniques, showcasing the potential of specific pairings to substantially enhance accuracy in the classification of driver behavior. In the other hand, the study also endeavors to discern pivotal features contributing to or shaping driver behavior, employing different feature selection methods, and considering various aspects of driving behavior such as speed, acceleration, lane detection, and car position.

In comparison with previous studies, our method demonstrates superior results. For instance, when compared to studies [8] and [9], where the authors achieved F1-scores of 94.11% and 80% respectively using Random Forest, our study outperforms with an F1-score of 96.36%. Additionally, our accuracy stands at 96.4%, surpassing the 63% accuracy reported in paper [6]. These findings underscore the efficacy of our approach in achieving higher classification performance in the context of driver behavior. Additionally, our conclusion highlights that the most influential features for driver behavior encompass speed (km/h), course, yaw, impact time, road width, distance to the ahead vehicle, vehicle position, and number of detected vehicles.

4. CONCLUSION

In this paper, we conducted a classification of driver behaviors utilizing various feature selection techniques and ML algorithms. We explored Filter methods including pearson correlation, Chi-square test, and ANOVA, wrapper methods involving recursive feature elimination, backward feature selection, and forward feature selection, and embedded methods using lasso, ridge, elasticnet, and tree-based methods. In general, a comprehensive analysis of features influencing driver behavior reveals several critical factors. Vehicle speed, rotation angle, time of impact, distance to ahead vehicle, vehicle position, and number of detected vehicles are prominent variables consistently associated with driving patterns. Each technique of feature selection was applied with ML algorithms LR, SVM, K-NN, DT, NB, and ensemble methods RF, XGBoost, and AdaBoost. The RF algorithm emerged as the top-performing classifier, achieving an impressive accuracy of 96.4% and 96.36% F1-score using backward feature selection in 7.43 seconds. Additionally, KNN displayed notable performance as the fastest classifier, with results close to those of RF. In future work, we aim to explore different deep learning algorithms for driver behavior classification and compare their results with the findings presented in this article.





REFERENCES

- [1] I. H. Sarker, "Machine learning: algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, p. 160, May 2021, doi: 10.1007/s42979-021-00592-x.
- [2] S. Wang, J. Tang, and H. Liu, "Feature selection," in *Encyclopedia of Machine Learning and Data Mining*, Boston, MA: Springer US, pp. 1–9, 2016, doi: 10.1007/978-1-4899-7502-7_101-1.
- [3] S. Bouhsissin, N. Sael, and F. Benabbou, "Driver behavior classification: a systematic literature review," *IEEE Access*, vol. 11, pp. 14128–14153, 2023, doi: 10.1109/ACCESS.2023.3243865.
- [4] S. Bouhsissin, N. Sael, and F. Benabbou, "Prediction of risks in intelligent transport systems," in *Lecture Notes in Networks and Systems*, vol. 489 LNNS, pp. 303–316, 2022, doi: 10.1007/978-3-031-07969-6_23.
- [5] S. Bouhsissin, N. Sael, and F. Benabbou, "Evaluating data sources and datasets in intelligent transport systems through a weighted scoring model," *International Journal of Transport Development and Integration*, vol. 7, no. 4, pp. 353–365, Dec. 2023, doi: 10.18280/ijtdi.070409.
- [6] R. Ghandour, A. J. Potams, I. Boulkaibet, B. Neji, and Z. Al Barakeh, "Driver behavior classification system analysis using machine learning methods," *Applied Sciences (Switzerland)*, vol. 11, no. 22, p. 10562, Nov. 2021, doi: 10.3390/app112210562.
- [7] S. Bouhsissin, N. Sael, and F. Benabbou, "Classification and modeling of driver behavior during yellow intervals at intersections," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-4/W, pp. 33–40, Dec. 2022, doi: 10.5194/isprs-archives-XLVIII-4-W3-2022-33-2022.
- [8] Y. Moukafih, H. Hafidi, and M. Ghogho, "Aggressive driving detection using deep learning-based time series classification," in *IEEE International Symposium on INnovations in Intelligent SysTems and Applications, INISTA 2019 - Proceedings*, pp. 1–5, Jul. 2019, doi: 10.1109/INISTA.2019.8778416.
- [9] K. Saleh, M. Hossny, and S. Nahavandi, "Driving behavior classification based on sensor data fusion using LSTM recurrent neural networks," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, Oct. 2018, doi: 10.1109/itsc.2017.8317835.
- [10] J. Vyas, D. Das, and S. K. Das, "Vehicular edge computing-based driver recommendation system using federated learning," in *Proceedings - 2020 IEEE 17th International Conference on Mobile Ad Hoc and Smart Systems, MASS 2020*, pp. 675–683, Dec. 2020, doi: 10.1109/MASS50613.2020.00087.
- [11] D. Yi, J. Su, C. Liu, M. Quddus, and W. H. Chen, "A machine learning based personalized system for driving state recognition," *Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 241–261, Aug. 2019, doi: 10.1016/j.trc.2019.05.042.
- [12] M. I. Silva and R. Henriques, "Finding manoeuvre motifs in vehicle telematics," *Accident Analysis and Prevention*, vol. 138, p. 105467, Apr. 2020, doi: 10.1016/j.aap.2020.105467.
- [13] M. I. Silva and R. Henriques, "TripMD: driving patterns investigation via motif analysis," *Expert Systems with Applications*, vol. 184, p. 115527, Dec. 2021, doi: 10.1016/j.eswa.2021.115527.
- [14] E. Romera, L. M. Bergasa, and R. Arroyo, "Need data for driver behaviour analysis? Presenting the public UAH-DriveSet," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 387–392, Nov. 2016, doi: 10.1109/ITSC.2016.7795584.
- [15] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data," *Frontiers in Energy Research*, vol. 9, 2021, doi: 10.3389/fenrg.2021.652801.
- [16] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft Computing*, vol. 97, p. 105524, Dec. 2020, doi: 10.1016/j.asoc.2019.105524.
- [17] R. E. Nogales and M. E. Benalcázar, "Analysis and evaluation of feature selection and feature extraction methods," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 153, Sep. 2023, doi: 10.1007/s44196-023-00319-1.
- [18] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919–926, 2016, doi: 10.1016/j.procs.2016.07.111.
- [19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.
- [20] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*. Wiley, 2013, doi: 10.1002/9781118548387.
- [21] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning: Methods and Applications to Brain Disorders*, Elsevier, pp. 101–121, 2019, doi: 10.1016/B978-0-12-815739-8.00006-7.
- [22] J. Jeffers, J. Reinders, and A. Sodani, "Chapter 24 - machine learning," *Intel Xeon Phi Processor High Performance Programming (Second Edition)*, pp. 527–548, 2016, [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128091944000247>.





- [23] D. Dhall, R. Kaur, and M. Juneja, "Machine learning: a review of the algorithms and its applications," in *Lecture Notes in Electrical Engineering*, vol. 597, 2020, pp. 47–63. doi: 10.1007/978-3-030-29407-6_5.
- [24] I. Wickramasinghe and H. Kalutarage, "Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation," *Soft Computing*, vol. 25, no. 3, pp. 2277–2293, Feb. 2021, doi: 10.1007/s00500-020-05297-6.
- [25] D. Petkovic, R. Altman, M. Wong, and A. Vigil, "Improving the explainability of random forest classifier – user centered approach," *Pacific Symposium on Biocomputing*, vol. 0, no. 212669, pp. 204–215, 2018, doi: 10.1142/9789813235533_0019.
- [26] H. Belyadi and A. Haghghat, "Supervised learning," in *Machine Learning Guide for Oil and Gas Using Python*, Elsevier, pp. 169–295, 2021, doi: 10.1016/b978-0-12-821929-4.00004-4.

BIOGRAPHIES OF AUTHORS







Soukaina Bouhsissin     received the B.Sc. degree in Mathematical Sciences and Computer Science from Hassan II University of CASABLANCA Morocco in 2018 and the M.Sc. degree in data science and big data from Hassan II University, Morocco, in 2020. She is currently pursuing the Ph.D. in Computer Science. Her research interests include driver behavior, intelligent transport system, machine learning, and deep learning. She can be contacted at email: bouhsissin.soukaina@gmail.com.







Nawal Sael     teacher-researcher since 2012, Authorized Professor since 2014 and Professor of Higher Education in the Department of Mathematics and Computer Science at the Ben M'Sick Faculty of Sciences in Casablanca, Morocco since 2020 and her engineer degree in software engineering from ENSIAS in 2002. Here research interests include data mining, machine learning, deep learning, and Internet of things. She can be contacted at email: saelnawal@hotmail.com.



Faouzia Benabbou     teacher-researcher since 1994, Authorized Professor since 2008 and Professor of Higher Education in the Department of Mathematics and Computer Science at the Ben M'Sick Faculty of Sciences in Casablanca since 2015. She is a member of the Information Technology and Modeling Laboratory. His research areas include cloud Computing, data mining, machine learning, and natural language processing. She can be contacted at email: faouzia.benabbou@univh2c.ma.



Abdelfettah Soutana     is a graduate of the master's degree in software quality from Hassan II University, Casablanca, Morocco, in 2015. He is currently working on his Ph.D. at the Laboratory of Information Processing and Modeling at the Ben M'Sik Faculty of Science. His research focuses on machine learning, deep learning, and internet of things. He can be contacted at email: soutana.abdelfettah@gmail.com.