Sigma J Eng Nat Sci, Vol. 44, No. 2, pp. 919–953, April, 2026

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2026.2018

Research Article

Driving style recognition with machine learning for intelligent control of vehicles

Ahmet BEŞKARDEŞ[1], Yakup HAMEŞ[1,*]

[1]*Department of Electrical and Electronics Engineering, İskenderun Tecnical University, İskenderun, Hatay, 31200, Türkiye*

ARTICLE INFO

Article history
Received: 09 September 2024
Revised: 14 October 2024
Accepted: 30 January 2025

Keywords:
Classification Algorithms; Data-Driven Approach; Driving Style Recognition; Machine Learning Models

ABSTRACT

In recent years, extensive research and development efforts have been increasingly carried out in both academia and industry to optimize the safety, efficiency, and environmental impact of vehicles traveling on roads. This study aims to contribute to these efforts by focusing on determining the driver's driving style. The approach of our study is crucial for enhancing driving safety and optimizing fuel efficiency. Data science and machine learning techniques were employed to identify the driver's driving style based on data collected during specific driving activities. Comprehensive driving data from various vehicle types were gathered as speed-time series using a global positioning system-based mobile application. The extracted features underwent necessary preprocessing and transformation to ensure their suitability for machine learning models. The prepared dataset was applied to k-nearest neighbors, support vector machine, decision tree, artificial neural network, logistic regression, naive Bayes, random forest, and light gradient boosting machine models. Three distinct driving styles were predicted: aggressive, normal, and calm. The accuracy rates achieved were 98.6% on residential district roads, 95.5% on urban roads, and 100% on motorways, with decision tree-based algorithms proving to be the most effective.

Cite this article as: Beşkardeş A, Hameş Y. Driving style recognition with machine learning for intelligent control of vehicles. Sigma J Eng Nat Sci 2026;44(2):919–953.

INTRODUCTION

With the significant advancements in the digitalization of road vehicles, accurately determining and adapting to the driving style has become crucial for optimizing safety, efficiency, and environmental impact. Correctly estimating the driving style and adapting the vehicle to this style is essential for enhancing vehicle efficiency, traffic safety, and driver comfort. The driving style of individuals significantly influences the efficient utilization of vehicles. Driving style is influenced by various factors, including individual traits like gender, age, attitude, and personality; environmental conditions such as the type of road, weather, journey distance, and traffic; and vehicle characteristics like type, model, and power [1]. To design vehicle control systems that accommodate individual driving styles, accurate prediction of these styles is necessary. Driving aggressively leads to higher energy consumption, whereas driving calmly helps to lower energy consumption [2]. If

***Corresponding author.**
*E-mail address: yakup.hames@iste.edu.tr

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

a vehicle's intelligent control system can adapt to the driver's driving style, it can optimize energy management by prioritizing performance when desired and fuel economy when calmness is preferred. Accurate assessment of driving style enhances driver safety [3,4], comfort, and enjoyment [5], fuel economy [6], and reduces emissions. Therefore, the ability to recognize driving styles has been integrated into new-generation advanced driver assistance systems (ADAS), significantly enhancing fuel consumption and driver safety [7].

The degree of acceleration and deceleration, turning speed, and frequency of lane changes when driving are typically considered when describing a driving style. However, it should not be forgotten that the same driver will exhibit different driving styles in different vehicles and environmental conditions. Three different driving styles can be defined as aggressive, calm, and moderate. Aggressive drivers try to minimize travel time, drive impatiently and excessively, do not maintain a safe following distance, make sudden and unexpected maneuvers, and change lanes frequently [8]. This situation causes dangerous situations in traffic as well as high fuel consumption [9]. Calm drivers avoid risky reactions and activities, do not accelerate or decelerate excessively, and drive more carefully [10]. Moderate drivers, on the other hand, are defined as an intermediate group that shows both aggressive and calm characteristics [7]. Table 1 shows the factors affecting driving styles according to vehicle, road, environment, and human factors.

Driving style prediction algorithms are developed based on the features extracted from the collected signals and the classification methods that will use these features. Martinez et al. reported that driving style recognition applications are developed using rules, models, and learning algorithms [7].

In the studies conducted through the rules, an output is obtained from the rules written according to the value of events such as acceleration, deceleration, left-right turn, or lane change, and the driving style is categorized accordingly [11]. Rule-based algorithms are simple to use and easy to understand, but they limit the number of parameters that can be studied. Therefore, the robustness and accuracy of the results are also very limited [7]. Working with more parameters is only possible with more advanced control systems such as fuzzy logic. The fuzzy system, which can be modeled with more parameters, can give results with higher accuracy while preserving the simplicity, robustness and easy understanding of the rule-based system. Dorr et al. conducted an online study by adding the road type and the distance between the vehicles into the input variables to determine the driving style [2]. Gilman et al., on the other hand, developed an application based on 17 factors with a fuzzy system in their study where they prioritized fuel consumption [12]. Fernandez and Ito developed a fuzzy system to predict traffic accidents and optimize route management. The inputs of the system were vehicle speed, acceleration, deceleration, and the driver's age, and the output were five different driver profiles very passive, passive, normal, aggressive, and dangerous [13]. Rule-based applications have the disadvantage of requiring expert knowledge as well as being straightforward and practical. In order to eliminate this negativity, the proposed model should be supported by looking at the previous data.

In model-based algorithms, driving style definitions are made based on certain formulas by making the necessary parameter settings using data-based methods [7]. With these models, it may be possible to achieve faster results in a simulation environment or in real-world applications.

In learning algorithms, a driving definition is made based on data. In this method, more robust and highly accurate results can be produced with the learning algorithms chosen in accordance with the problem after comprehensive data analysis. Machine learning algorithms are unsupervised, supervised, and hybrid. In unsupervised learning, there are no output tags that must be assigned in response to previous inputs. Therefore, there is no need to understand the process of the problem. This process is done through statistical analysis in these algorithms, which produces more clustering solutions. A dataset containing output tags is used to train the algorithm in supervised learning. When the model trained with historical data is given new data that it has not seen before, it is expected to predict the output. In supervised learning algorithms used for regression and classification, methods such as k-nearest neighbor (k-NN), decision trees, random forests, SVM,

Table 1. Influencing factors on driving style

Influencing factors on driving style		
Vehicle and road factors	**Environmental factors**	**Human factors**
Vehicle type and model	Traffic	Gender
Vehicle design	Season	Age
Vehicle power	Geographic region	Driving experience
Road type	Time of the day	Condition (alcohol, fatigue, stress)
Sight distance		Driving distance
Lighting conditions		Other drivers

neural networks, Naive Bayes, and logistic regression are most commonly used.

Studies with learning algorithms in defining driving style are much more than the other two algorithms. Van Ly et al. evaluated the characteristics of acceleration events, braking, and turning events from the data collected from inertial sensors and classified driver behavior with two different classification algorithms [14]. In this study, instead of being satisfied with two classifiers, increasing the variety of methods can lead researchers to better results. Zhang et al. extracted features describing driving habits from the data obtained from phone and car sensors and performed driver classification with support vector machines [15]. Quintero and colleagues modeled driver behavior with an artificial neural network by evaluating position, speed, acceleration, and steering angle data obtained from a global positioning system (GPS) data recording system [16]. Although a high success rate was achieved with neural networks in this study, if other algorithms were applied to the same data, comparing them in terms of processing time, interpretability, and practicality would be possible. Shahverdy and others used data on speed, acceleration, gravity, throttle, and engine speed (RPM). Using a deep learning model with these data, they predicted five different driver types with very high accuracy: normal, aggressive, distracted, sleepy, and drunk [17]. However, in this study, only three drivers and a single vehicle were tested, and the drivers drove according to the instructions they received before driving. Therefore, this study needs to be expanded with more realistic and larger dataset. Vaitkus et al. extracted statistical features such as mean, median, mode, variance, standard deviation, range, minimum, maximum, skewness, and kurtosis from triaxial accelerometer data for driving on a repetitive route. From the values of these features, they estimated two different driving styles with k-NN, either aggressive or normal [18]. However, as they stated, the high accuracy rate obtained in the tests performed in the same season, in the same traffic conditions, and on the same route can be misleading. Performing these tests with different and more driving data will increase the model's reliability. In addition, conducting tests with more diverse classification algorithms and increasing the number of categories of the output class (such as aggressive, normal, and calm) will further increase the value of the study. Kedar-Dongarkar and Das collected the acceleration, braking, and throttle values of the vehicle, and estimated the driver of the vehicle as one of the aggressive, moderate, and conservative profiles using a simple classifier method from these data [19]. After applying the sensor data to two different neural networks, Brombacher et al. detected defensive and sporty driving maneuvers and calculated a driving style classification score in five categories [20]. Saleh et al. extracted three different driving behaviors as normal, aggressive, and sleepy with the time series classification model from nine different sensor data obtained from the smartphone [21]. Wang et al. collected 400 driving data from 20 drivers in a driving simulator and applied the vehicle speed and throttle opening properties from this dataset to a semi-supervised support vector machine. With this method, he estimated three different driving styles, normal, vague, and aggressive [22]. Meteier et al. used physiological signals and machine learning techniques to predict sleep deprivation, driving environment, and sleepiness in experiments with 63 different drivers in a driving simulator. Although they could not predict sleepiness well, they achieved very good results in sleep deprivation [23]. Seraj used a rule-based classification method to analyze the driving data obtained from different roads such as highways, main arteries and ramps to analyze the longitudinal and lateral controls of drivers. His classification algorithm-based study revealed three different classes of drivers: calm, rational, and aggressive [24]. This study is distinguished by its use of real-world data across various road types, implementation of both short-term and long-term classification methods, and integration Advanced Driver Assistance Systems (ADAS). Khan et al. created a dataset containing 15 features from CAN-Bus data collected via OBD-II. They were able to distinguish two drivers almost perfectly with their classification algorithm, but their accuracy dropped significantly for ten drivers [25]. Benterki et al. identified two driving styles, calm and aggressive, using the Next Generation Simulation dataset with spectral clustering and K-means algorithms. This study is an example of unsupervised learning methods in driving style diagnosis [26].

Similar methods have been applied for the prediction of driving skills, aggression status, or road type. Chandrasiri et al obtained driving data from a driving simulator and tagged the profiles of these rides. From these data, they created a model within the scope of the driving environment, the behavior of the drivers, and the reactions of the vehicles to the environment. With this model they developed based on SVM and k-NN, they estimated a driver's driving skills [27]. Yu and others identified 16 key features from driving data collected over 6 months from real driving environments. By applying this data to neural networks and support vector machines, they predicted and detected abnormal conditions such as side slip, rapid U-turn, large radius turn, and sudden braking [28]. Bernardi and others extracted a comprehensive set of features from the data they took from the monitoring system they built into the vehicles. Applying these features to the time series classification model, they identified different driving styles, drivers' familiarity with their vehicles, and road types [29]. Moukafih et al used the UAH-DriveSet dataset, which consists of data from speed and acceleration sensors and a vehicle front view camera. They applied this data to a time series classification called the Long Short-Term Memory Fully Convolutional Network (LTSM-FCN). They estimated driver aggression status with this classification model [30]. Sun et al. reported that studies conducted under real traffic conditions would yield better results [31], while Eckert et al. reported that driving cycles derived from various road

and traffic conditions would further improve these results [32].

The studies described above fill important gaps in the prediction of driving style and its contribution to driving. However, for a reliable driving prediction, features such as the number of drives, driving time, driving purpose, distance traveled and the fact that these were done under real traffic conditions are extremely important. Therefore, in this study, we developed an approach based on real data that we produced with meticulous care and dedication, considering realistic uncertainties. Recognizing the crucial role of real-world factors like driving experience, purpose, and environmental conditions, we meticulously collected a comprehensive dataset encompassing diverse scenarios. We conducted both a large number and a wide variety of driving experiments to create this dataset. We used different vehicles such as cars, minibuses, and buses. We used vehicles on three different road types: residential, urban, and highway. We drove in all four seasons of the year and at different times of the day. These drives were in different geographical regions such as sea level, and mountainous areas, and were carried out for different purposes such as commuting to work or traveling for vacation. Our professional drivers who made the drives were of different ages, gender, and driving skill categories. While conducting our experiments, we paid particular attention to the following issues:

i. All drivers were in good mental and physical condition. Drivers did not drive while drowsy or intoxicated.
ii. We did not give drivers any instructions regarding vehicle operation. We left the driving choices up to the drivers' judgment: calm, normal, or aggressive. After the driving, we labeled the style of driving according to the personal evaluations of the drivers.
iii. We did not allow distracting activities such as talking to the person next to them or making a phone call while driving.
iv. Residential and urban driving activities were carried out in full compliance with traffic rules.

The technical contributions of this document are summarized below:

i. Driving data of 610 hours and 38 thousand km was collected with nine professional drivers.
ii. The speed-time information of these drives is recorded simply and practically. We extracted 17 meaningful features solely from speed-time data, demonstrating the potential for robust driving style characterization using readily available information.
iii. Driving data was evaluated using an extensive data processing and formatting methodology.
iv. With k-NN, support vector machines (SVM), decision trees, neural networks (NN), logistic regression, Naive-Bayes, random forest, and light gradient boosting machine (LightGBM) classifiers, high-accuracy models for discrete path types have been developed.
v. These models were compared with each other in many ways and the reasons for preference were explained.
vi. We have provided a clear and accessible framework to address classification problems in the context of driving style prediction. All the steps of how to solve any classification problem can be easily understood if the flow in this article is followed.

In light of the above contributions, we show that we can predict driving style with very high accuracy in a real traffic environment for various driving events such as drivers, driving purposes, driving times, and road types.

DRIVING CYCLES AND DATA MINING

When classifying drivers or driving, as Bouhsissin [33] stated, it is necessary to ask the following research questions and plan with an appropriate methodology:

-What is the purpose of the driving process, and what is its variety, duration, and range? What kind of driving cycle will be used?

-How will the driving style problem be addressed?

-Which features should be selected?

-Which techniques will be used for data preprocessing? Which necessary transformation operations will be performed on the data?

-What types of models will be used and how will these models be optimized?

We have organized our study under subheadings according to the above-mentioned items.

Driving Cycles

By performing a meticulous study to define the driving style, we stored the driving cycles of 9 professional drivers across various roads, using different vehicles, in diverse geographical regions and times, throughout the day, and under varying traffic conditions. To do this, we used a successful and widely used mobile application for speed tracking. This application was run before the vehicle started moving and remained running while the vehicle was moving. The smartphone application determined the location of the vehicle at certain time intervals with the help of the GPS receiver and calculated the distance between two consecutive locations, the time it took to cover this distance and the speed of the vehicle. Even though the vehicles were different during the tests, the mobile application was the same. In addition, by comparing the rides made with different vehicles on the same route, we observed that the same distance was covered in total, and the speed and location changed similarly every second. In this way, we ensured the data quality and security. We saved the speed and time information of the driving cycles that we exported from the application's database to a text file in a relational database. Table 2 shows a summary of these driving cycles.

Table 2. Statistical information on driving cycles

Driving cycles summary	
Region	**Many different regions of Turkey**
Total drivers	9
Total evening drives	789
Total noon drives	524
Total morning drives	619
Total drives in the residential district	977
Total drives in the urban roads	750
Total drives in the motorway	204
Total drives in calm traffic	1881
Total drives in heavy traffic	51
Total count of records	1931
Total duration of all driving	611.4 hour
Total driving distance	37808.3 km.
Collection time for all driving cycles	19 months

Professional drivers who performed these drives, summarized in Table 2, reported that this drive was calm, normal, or aggressive after each drive was finished. However, they received no guidance or instructions on driving before the drive.

Data Mining

In systems developed for predicting driving styles, there may be some difficulties in classifying data into certain categories or classes. There may be missing, noisy, or inconsistent data in the collected data. There may be smooth transitions between the defined classes. The number and distribution of samples may be unbalanced. If the features are not selected correctly, overlaps may occur between classes. We performed detailed data preprocessing to overcome the uncertainties that reduce the classification accuracy and model reliability and cause wrong decisions. We paid attention to the uncertainty and overlap issues when selecting or eliminating features. We tried different classification methods, including ensemble models that combine the results of multiple classifiers.

We completed this study in five stages as shown in Figure 1. First, we analyzed the issue we discussed and identified the necessary information for a solution. To achieve this, we collected driving data in the direction we planned. Then, we extracted meaningful attributes from this data that would provide insights about driving style. We performed basic data preprocessing on this data set. We transformed this data so that we could apply it to classification algorithms and moved on to the application of driving style recognition with machine learning methods.

Problem Understanding

In the first stage, we collected two different types of data during the test drives. The first of these was information including the time of the drive, traffic information, destination, and who the driver was. We collected them in a table called 'drive_info' in our relational database. Secondly, we obtained the driving cycle data, which consists of the speed-time values of this drive. We stored these in another table called 'velocity'.

Feature Extraction

As we mentioned in the introduction section, there are important features that distinguish, for example, aggressive driving style from other driving styles. Aggressive drivers usually drive at higher speeds, make frequent and sudden speed changes, have high acceleration and deceleration intensity, and experience many speed fluctuations during driving. In calm drivers, on the contrary, these values are much smoother and more regular. For this purpose, we determined 13 features, such as maximum, minimum, and average speed values, speed value change, positive and negative acceleration rates, and stop-start rate. Explanations of these features are given in Table 3. We calculated the values of these features using an application we created in the .NET environment using the C# programming language.

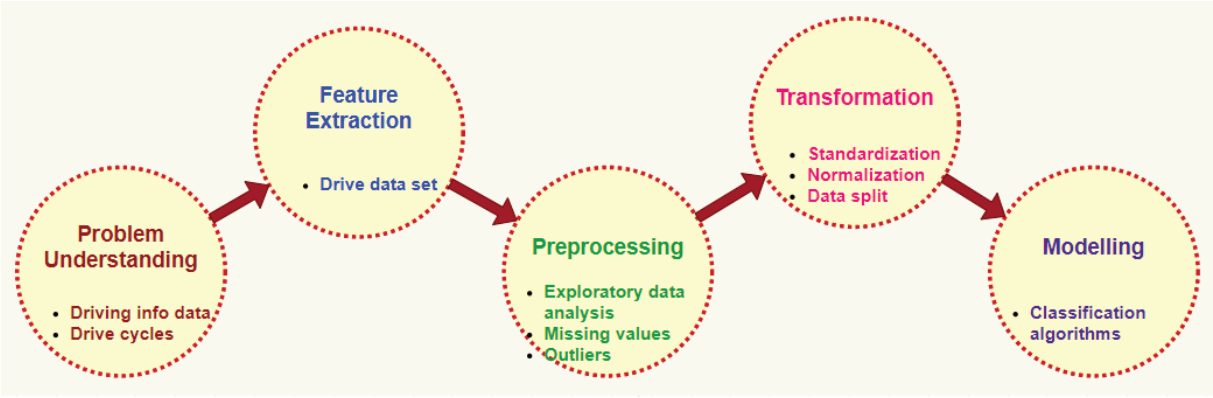

Figure 1. Stages to predict driving style.

Table 3. Attributes identified for classifying driving

Feature	Data type	Description
Max speed [km/h]	Numeric	The maximum speed of the driving
Mean speed [km/h]	Numeric	Average speed of the driving
Standard deviation [km/h]	Numeric	The standard deviation of the speed
Skewness	Numeric	Explains the distribution of velocity values. A right-skewed distribution suggests that outliers are due to high velocities, whereas a left-skewed distribution suggests that outliers are due to low velocities.
Kurtosis	Numeric	Kurtosis shows whether the tails of the normal distribution curve of velocity values are heavy or light.
Positive slop avg [m/s2]	Numeric	Mean value of positive accelerations in driving
Negative slop avg [m/s2]	Numeric	Mean value of negative accelerations in driving
Max point ratio [%]	Numeric	Ratio of peak speed points to overall time
Min point ratio [%]	Numeric	Ratio of lowest speed points to overall time
Hard acc ratio [%]	Numeric	Rate of rapid acceleration
Hard dec ratio [%]	Numeric	Rate of rapid deceleration
Overspeed [%]	Numeric	Rate of exceeding specified speeds for different road types (eg 120 km/h for motorway)
Stop and go ratio [%]	Numeric	Ratio of stop-start time to total time
Mean band ratio [%]	Numeric	Proportion of time spent traveling at average speed to total time
Driving style	Categorical	Driving style assigned at the end of the driving

There is no need to explain how some of the attributes (max, mean) in Table 3 are calculated, as the calculations of the others are given in Equations 1-12. Figure 2 illustrates how some features are calculated through visual figures.

$$Sample\ Standard\ Deviation = s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \quad (1)$$

$$Skewness = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^3}{(n-1).s^3} \quad (2)$$

$$Kurtosis = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{(n-1).s^4} \quad (3)$$

In Equation 1-3, s is the sample standard deviation, $\bar{x}$ is the sample mean, n is the total number of sample elements.

$$Positive\ Slop\ Avg = \frac{\sum_{i=0}^{m} f(i+1) - f(i)}{m}, \quad f(i+1) > f(i) \quad (4)$$

$$Negative\ Slop\ Avg = \frac{\sum_{i=0}^{n} f(i+1) - f(i)}{n}, \quad f(i+1) < f(i) \quad (5)$$

In Equations 4 and 5, *f(i)* is the velocity of the driving at the time *i*, *m* is the positive acceleration number, and *n* is the negative acceleration number.

$$Max\ Point\ Ratio = 100 \times \frac{Number\ of\ \max point}{Total\ time} \quad (6)$$

$$Min\ Point\ Ratio = 100 \times \frac{Number\ of\ \min point}{Total\ time} \quad (7)$$

In Figure 2, the maximum points are shown in red circles and the minimum points are shown in green circles.

$$Hard\ Acc\ Ratio = 100 \times \frac{Number\ of positive\ slops}{Total\ time}, \quad positive\ slop > 0.5\ m/s^2 \quad (8)$$

$$Hard\ Dec\ Ratio = 100 \times \frac{Number\ of\ negative\ slops}{Total\ time}, \quad negative\ slop < -0.5\ m/s^2 \quad (9)$$

The ±0.5 m/s² value used as a parameter while calculating the hard acceleration and deceleration rates was determined as a result of the tests performed while driving.

$$Overspeed = 100 \times \frac{Number\ of\ speeds}{Total\ time}, \quad speed > 120\ km/h \quad (10)$$

In this feature, which is valid for highway driving, the ratio of the time traveled at speeds above 120 km/h to the total time is calculated.

$$Stop\ and\ Go\ Ratio = 100 \times \frac{Number\ of\ \text{zero}\ point}{Total\ time} \quad (11)$$

This parameter, which is defined mostly to determine the traffic situation, is calculated from the ratio of the zero points shown with orange dots in Figure 2 to the total time.

$$Mean\ Band\ Ratio = 100 \times \frac{Time\ spent\ on\ the\ mean\ band}{Total\ time} \quad (12)$$

While calculating the average speed band, speeds that are 20% above and 20% below the mean speed were considered. The value of 20 was determined in consultation with the experts on the subject. Figure 2 illustrates the mean speed band.

By calculating the attribute values shown in Table 3, we generated distinct data sets for the three road types. Tables 4-6 show the summary of the datasets we created for the residential district, urban road, and motorway, respectively.

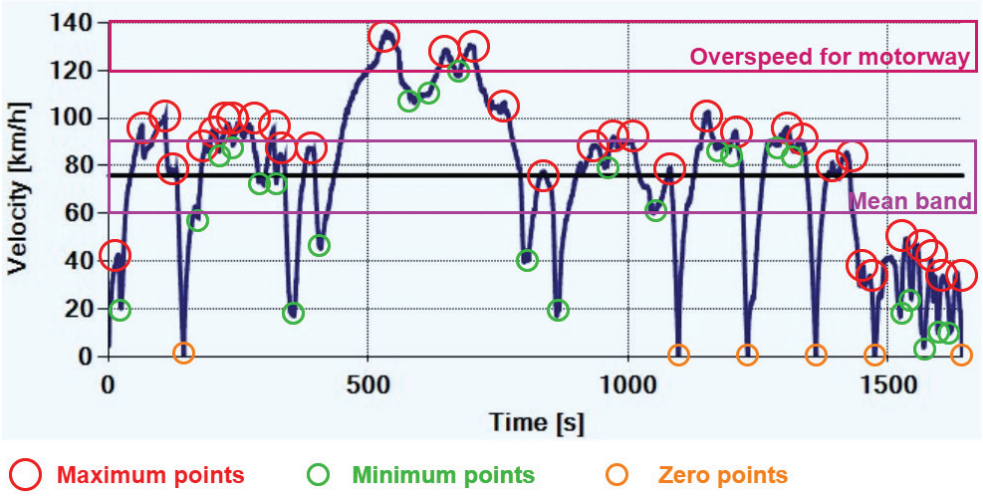

Figure 2. The maximum-minimum points and stop-start points used in the calculation of some features are shown in a sample velocity-time graph.

Table 4. Residential district dataset summary

	Max speed	Mean speed	Standard deviation	Skewness	Kurtosis	Positive slop avg	Negative slop avg	Max point ratio	Min point ratio	Hard acc ratio	Hard dec ratio	Overspeed	Stop and go ratio	Mean band ratio
Count	977	977	977	977	977	977	977	977	977	977	977	977	977	977
Mean	59.95	34.54	13.54	-0.48	2.62	0.41	-0.41	7.66	7.58	2.08	2.01	0	0.84	35.87
Std dev	8.18	4.28	2.01	0.30	0.44	0.08	0.09	1.57	1.59	0.79	0.78	0	1.24	8.90
Min	34.20	11.40	6.38	-1.43	1.45	0.22	-0.78	3.60	3.24	0.37	0	0	0	8.20
25%	53.60	31.70	12.21	-0.68	2.33	0.36	-0.45	6.48	6.37	1.53	1.50	0	0.45	29.80
50%	60.00	34.70	13.44	-0.51	2.56	0.40	-0.39	7.44	7.38	2.02	1.94	0	0.72	35.40
75%	65.20	37.40	14.70	-0.33	2.85	0.45	-0.35	8.69	8.64	2.50	2.46	0	1.05	41.40
Max	87.80	50.20	28.24	1.45	5.20	0.86	-0.21	12.81	13.30	5.91	5.01	0	35.06	69.40

Table 5. Urban road dataset summary

	Max speed	Mean speed	Standard deviation	Skewness	Kurtosis	Positive slop avg	Negative slop avg	Max point ratio	Min point ratio	Hard acc ratio	Hard dec ratio	Overspeed	Stop and go ratio	Mean band ratio
Count	750	750	750	750	750	750	750	750	750	750	750	750	750	750
Mean	86.68	43.69	22.72	-0.04	2.18	0.47	-0.48	6.21	6.14	2.17	2.26	0	1.09	22.76
Std dev	19.19	12.28	6.19	0.48	0.69	0.11	0.10	1.48	1.49	1.01	0.91	0	1.05	10.49
Min	38.20	14.70	7.94	-1.82	1.32	0.20	-1.18	2.86	2.48	0.09	0	0	0	3.30
25%	72.93	32.82	17.99	-0.34	1.73	0.39	-0.54	5.14	5.09	1.49	1.63	0	0.62	15.03
50%	90.85	44.70	24.04	-0.05	1.96	0.45	-0.48	6.02	5.96	1.98	2.17	0	0.86	20.6
75%	100.9	53.48	27.43	0.25	2.41	0.53	-0.42	7.11	6.97	2.65	2.75	0	1.31	28.1
Max	137.1	74.70	41.53	1.96	7.34	1.06	-0.23	12.77	12.77	6.97	6.36	9	21.17	80.6

Table 6. Motorway dataset summary

	Max speed	Mean speed	Standard deviation	Skewness	Kurtosis	Positive slop avg	Negative slop avg	Max point ratio	Min point ratio	Hard acc ratio	Hard dec ratio	Overspeed	Stop and go ratio	Mean band ratio
Count	204	204	204	204	204	204	204	204	204	204	204	204	204	204
Mean	132.45	76.85	30.84	-0.67	3.34	0.37	-0.41	4.72	4.71	1.23	1.34	10	0.48	37.73
Std Dev	24.73	16.53	6.80	0.75	2.43	0.08	0.09	1.80	1.80	0.69	0.65	13	0.32	20.86
Min	91.30	48.40	12.18	-3.27	1.56	0.11	-0.66	0.91	0.91	0	0	0	0	5.00
25%	112.70	63.75	26.43	-1.20	1.94	0.33	-0.46	3.55	3.53	0.76	0.85	0	0.25	22.50
50%	133.70	75.25	30.24	-0.39	2.30	0.38	-0.40	4.43	4.44	1.19	1.42	4	0.43	30.35
75%	151.28	89.78	36.06	-0.10	3.67	0.43	-0.35	5.96	5.96	1.68	1.77	15	0.66	46.47
Max	200.6	116.4	47.51	0.58	18.62	0.58	-0.18	9.42	9.38	3.31	3.12	57	2.39	94.50

Preprocessing

Before applying machine learning techniques to a data, it is necessary to ensure that the data is reliable. In this process, features that show high correlation or do not contribute to the performance of the model are removed from the data set. In addition, a data set may have missing, inconsistent, outlier or noisy data. Data preprocessing is performed to complete missing data, eliminate inconsistencies, discard outlier data, delete incorrect data if any, and clean noisy data [34].

As described in the previous section, after extracting the features, we performed exploratory data analysis (EDA) on our dataset. In this context, we first removed the overspeed feature, which is always 0 in the residential district data set. Then, by performing correlation analysis on all three data sets, we identified the features that were correlated with each other. Figures 3-5 show the correlation analysis results of the three data sets.

In correlation analysis, the pairwise relationships among all features are evaluated. The correlation coefficient between two variables ranges from -1 to 1. A value closer to -1 or 1 signifies a stronger relationship. Positive values indicate a direct relationship, whereas negative values indicate an inverse relationship. In the correlation analysis, we eliminated all but one of the features with a higher or lower correlation value than ±0.8, which we determined to be the

Residential District Dataset	Max Speed	Mean Speed	Standard Deviation	Skewness	Kurtosis	Positive Slop Avg	Negative Slop Avg	Max Point Ratio	Min Point Ratio	Hard Acc Ratio	Hard Dec Ratio	Stop and Go Ratio	Mean Band Ratio
Max Speed	1	0.70	0.73	0.32	0.05	0.20	-0.36	-0.45	-0.45	0.07	0.14	-0.20	0.19
Mean Speed	0.70	1	0.44	-0.32	0.25	-0.08	-0.07	-0.37	-0.38	-0.19	-0.13	-0.36	0.43
Standard Deviation	0.73	0.44	1	0.31	-0.44	0.32	-0.43	-0.47	-0.46	0.16	0.22	0.00	-0.38
Skewness	0.32	-0.32	0.31	1	-0.37	0.36	-0.39	-0.12	-0.12	0.32	0.36	0.17	-0.32
Kurtosis	0.05	0.25	-0.44	-0.37	1	-0.28	0.25	0.11	0.10	-0.25	-0.25	-0.14	0.84
Positive Slop Avg	0.20	-0.08	0.32	0.36	-0.28	1	-0.72	-0.01	-0.01	0.80	0.65	0.16	-0.31
Negative Slop Avg	-0.36	-0.07	-0.43	-0.39	0.25	-0.72	1	0.22	0.23	-0.58	-0.80	-0.07	0.24
Max Point Ratio	-0.45	-0.37	-0.47	-0.12	0.11	-0.01	0.22	1	1	0.34	0.19	0.08	0.02
Min Point Ratio	-0.45	-0.38	-0.46	-0.12	0.10	-0.01	0.23	1	1	0.34	0.19	0.09	0.02
Hard Acc Ratio	0.07	-0.19	0.16	0.32	-0.25	0.80	-0.58	0.34	0.34	1	0.71	0.14	-0.29
Hard Dec Ratio	0.14	-0.13	0.22	0.36	-0.25	0.65	-0.80	0.19	0.19	0.71	1	0.10	-0.28
Stop and Go Ratio	-0.20	-0.36	0.00	0.17	-0.14	0.16	-0.07	0.08	0.09	0.14	0.10	1	-0.29
Mean Band Ratio	0.19	0.43	-0.38	-0.32	0.84	-0.31	0.24	0.02	0.02	-0.29	-0.28	-0.29	1

Figure 3. Correlation analysis results in the residential district data set.

Urban Road Dataset	Max Speed	Mean Speed	Standard Deviation	Skewness	Kurtosis	Positive Slop Avg	Negative Slop Avg	Max Point Ratio	Min Point Ratio	Hard Acc Ratio	Hard Dec Ratio	Overspeed	Stop and Go Ratio	Mean Band Ratio
Max Speed	1	0.83	0.90	-0.14	-0.28	-0.02	-0.14	-0.42	-0.41	-0.15	-0.14	0.21	-0.15	-0.35
Mean Speed	0.83	1	0.75	-0.61	-0.28	-0.22	0.04	-0.57	-0.56	-0.40	-0.36	0.20	-0.33	-0.12
Standard Deviation	0.90	0.75	1	-0.11	-0.55	0.06	-0.20	-0.41	-0.40	-0.08	-0.09	0.18	-0.05	-0.61
Skewness	-0.14	-0.61	-0.11	1	0.11	0.30	-0.19	0.34	0.34	0.40	0.34	-0.03	0.28	-0.23
Kurtosis	-0.28	-0.28	-0.55	0.11	1	-0.09	0.16	0.22	0.22	0.00	0.01	-0.03	0.04	0.74
Positive Slop Avg	-0.02	-0.22	0.06	0.30	-0.09	1	-0.70	0.25	0.24	0.78	0.65	0.06	0.21	-0.22
Negative Slop Avg	-0.14	0.04	-0.20	-0.19	0.16	-0.70	1	-0.10	-0.08	-0.60	-0.74	-0.06	-0.15	0.22
Max Point Ratio	-0.42	-0.57	-0.41	0.34	0.22	0.25	-0.10	1	1	0.63	0.60	-0.06	0.13	0.08
Min Point Ratio	-0.41	-0.56	-0.40	0.34	0.22	0.24	-0.08	1	1	0.63	0.59	-0.06	0.13	0.07
Hard Acc Ratio	-0.15	-0.40	-0.08	0.40	0.00	0.78	-0.60	0.63	0.63	1	0.81	0.02	0.22	-0.18
Hard Dec Ratio	-0.14	-0.36	-0.09	0.34	0.01	0.65	-0.74	0.60	0.59	0.81	1	0	0.20	-0.12
Overspeed	0.21	0.20	0.18	-0.03	-0.03	0.06	-0.06	-0.06	-0.06	0.02	0	1	-0.04	0.01
Stop and Go Ratio	-0.15	-0.33	-0.05	0.28	0.04	0.21	-0.15	0.13	0.13	0.22	0.20	-0.04	1	-0.16
Mean Band Ratio	-0.35	-0.12	-0.61	-0.23	0.74	-0.22	0.22	0.08	0.07	-0.18	-0.12	0.01	-0.16	1

Figure 4. Correlation analysis results in the urban road data set.

Motorway Dataset	Max Speed	Mean Speed	Standard Deviation	Skewness	Kurtosis	Positive Slop Avg	Negative Slop Avg	Max Point Ratio	Min Point Ratio	Hard Acc Ratio	Hard Dec Ratio	Overspeed	Stop and Go Ratio	Mean Band Ratio
Max Speed	1	0.64	0.62	0.00	0.09	0.12	-0.34	-0.46	-0.46	-0.26	-0.16	0.72	-0.29	0.14
Mean Speed	0.64	1	0.04	-0.71	0.62	-0.44	0.25	-0.65	-0.65	-0.64	-0.66	0.69	-0.66	0.69
Standard Deviation	0.62	0.04	1	0.54	-0.58	0.64	-0.68	0.03	0.03	0.32	0.38	0.50	0.26	-0.59
Skewness	0.00	-0.71	0.54	1	-0.85	0.69	-0.63	0.45	0.45	0.62	0.72	-0.15	0.55	-0.85
Kurtosis	0.09	0.62	-0.58	-0.85	1	-0.70	0.57	-0.49	-0.49	-0.60	-0.66	0.10	-0.53	0.90
Positive Slop Avg	0.12	-0.44	0.64	0.69	-0.70	1	-0.80	0.49	0.48	0.81	0.81	0.01	0.63	-0.72
Negative Slop Avg	-0.34	0.25	-0.68	-0.63	0.57	-0.80	1	-0.14	-0.14	-0.51	-0.68	-0.12	-0.42	0.57
Max Point Ratio	-0.46	-0.65	0.03	0.45	-0.49	0.49	-0.14	1	1	0.83	0.76	-0.34	0.64	-0.54
Min Point Ratio	-0.46	-0.65	0.03	0.45	-0.49	0.48	-0.14	1	1	0.83	0.76	-0.34	0.64	-0.54
Hard Acc Ratio	-0.26	-0.64	0.32	0.62	-0.60	0.81	-0.51	0.83	0.83	1	0.91	-0.24	0.76	-0.67
Hard Dec Ratio	-0.16	-0.66	0.38	0.72	-0.66	0.81	-0.68	0.76	0.76	0.91	1	0	0.71	-0.71
Overspeed	0.72	0.69	0.50	-0.15	0.10	0.01	-0.12	-0.34	-0.34	-0.24	0	1	-0.28	0.14
Stop and Go Ratio	-0.29	-0.66	0.26	0.55	-0.53	0.63	-0.42	0.64	0.64	0.76	0.71	-0.28	1	-0.63
Mean Band Ratio	0.14	0.69	-0.59	-0.85	0.90	-0.72	0.57	-0.54	-0.54	-0.67	-0.71	0.14	-0.63	1

Figure 5. Correlation analysis results in the motorway data set.

Residential District Dataset	Urban Road Dataset	Motorway Dataset
Max Speed	Max Speed	Max Speed
Mean Speed	Skewness	Mean Speed
Standard Deviation	Kurtosis	Standard Deviation
Skewness	Positive Slop Avg	Positive Slop Avg
Min Point Ratio	Negative Slop Avg	Negative Slop Avg
Hard Acc Ratio	Min Point Ratio	Max Point Ratio
Hard Dec Ratio	Hard Dec Ratio	Overspeed
Stop and Go Ratio	Overspeed	Stop and Go Ratio
Mean Band Ratio	Stop and Go Ratio	Mean Band Ratio
	Mean Band Ratio	

Figure 6. List of features for three datasets.

threshold value. In this case, the features of our three datasets are shown in Figure 6.

One of the operations we performed during the data preprocessing phase was the control of missing data. However, no such imperfect data was found in our three data sets. Then, we checked for outliers. We used the local outlier factor (LOF) method because we aimed to reduce skewness. In the outlier analysis we performed by looking at the 20 neighbors around each point, we removed the data points that were above the threshold value we determined. By removing outliers, we ensured that our models did not suffer from over-learning. The results of the outlier analysis for the three data sets are shown in Figures 7-9.

In Figures 7-9, black dots show data points, red circles show the calculated outlier value for each point, and blue dots show samples with outlier scores exceeding the specified threshold value. At the end of this process, we removed 10 data points from the residential district data set and 7 data points from the urban road data set. No outliers were found in the motorway data set. After removing outliers, our sample counts were 967 for the residential district, 743 for the urban road, and 204 for the motorway.

Transformation

Transformation techniques are used to analyze raw data in a machine learning project and make it suitable for modeling. Methods such as scaling, standardization, normalization, and dimensionality reduction are among the common transformation techniques. After the transformation process, the performance of the model is expected to increase, especially linear models produce better results for scaled data. The computational cost is reduced with dimensionality reduction techniques. The effect of feature values with different units of measurement on the model is balanced. Transformation techniques also contribute visually and provide better interpretation of the obtained results.

In this section, we apply the standardization process after separating the data set as training and test data. Splitting the dataset will allow comparison of the accuracy values in the training and test data while finding the best parameters in the classification algorithms. In all three datasets, we split the dataset into 70% training

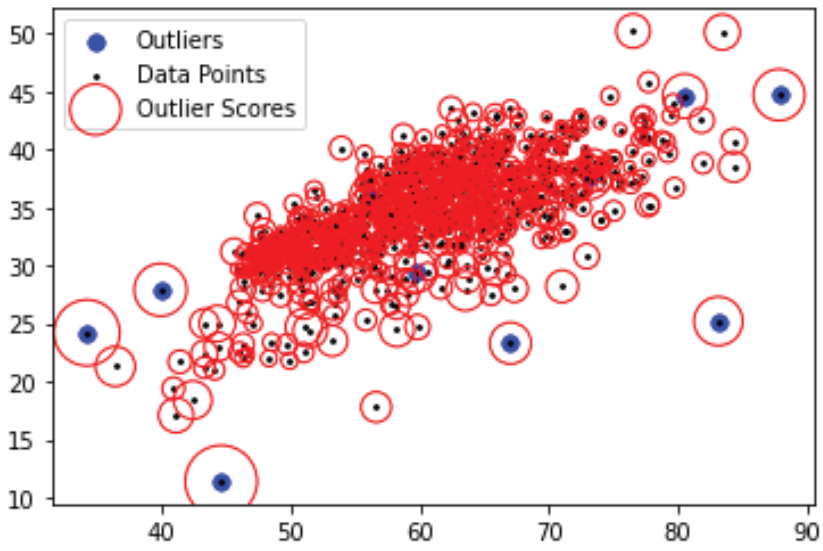

Figure 7. Outlier analysis for residential district.

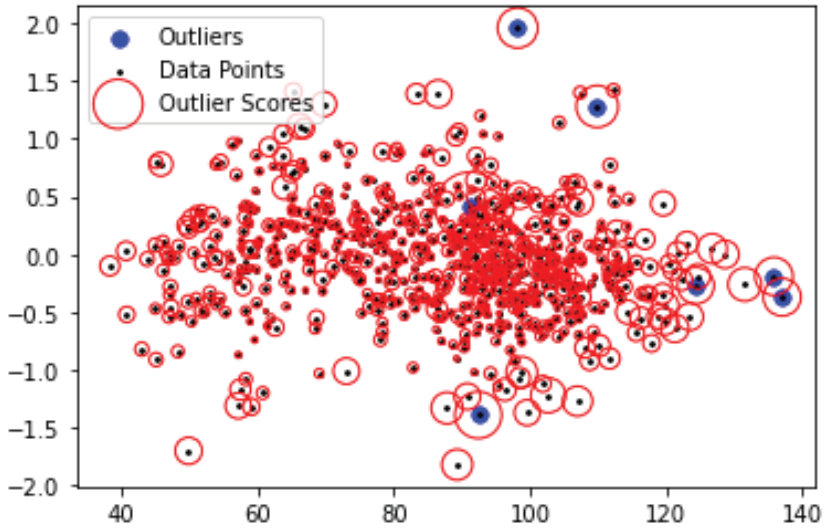

Figure 8. Outlier analysis for urban road.

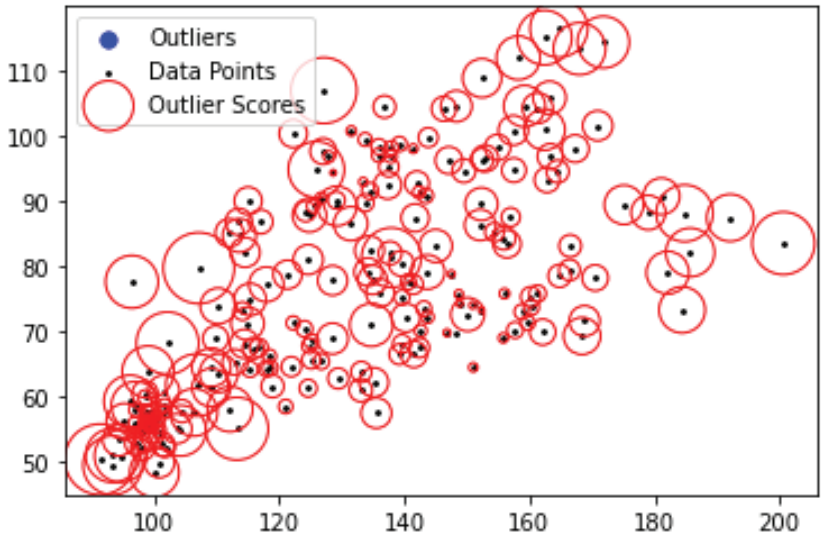

Figure 9. Outlier analysis for motorway.

data and 30% test data. Then we performed the standardization process to eliminate the scale difference between the features. With the standardization process shown in Equation 13, we ensure that the mean of the data of a feature is 0 and the standard deviation is 1. This process makes a significant contribution to the success of models that use distance-based calculations in particular.

$$z = \frac{x - \mu}{\sigma} \tag{13}$$

Here x is the variable value, μ is the mean value and σ is the standard deviation.

Modelling

After the pre-processing phase of the data is completed, the next step is to select an appropriate algorithm for the problem and run the machine learning process. Figure 10 shows the separation of the data into training and test data sets, training the model with training data, determining the hyperparameters that will give the best results in this model, and testing the model with these parameters with the test data.

We used the GridSearchCV [35] algorithm to find the best parameters of a model. While using this algorithm, as shown in Figure 11, we divided all the data into 10 pieces and used a different part as a test and the others as training

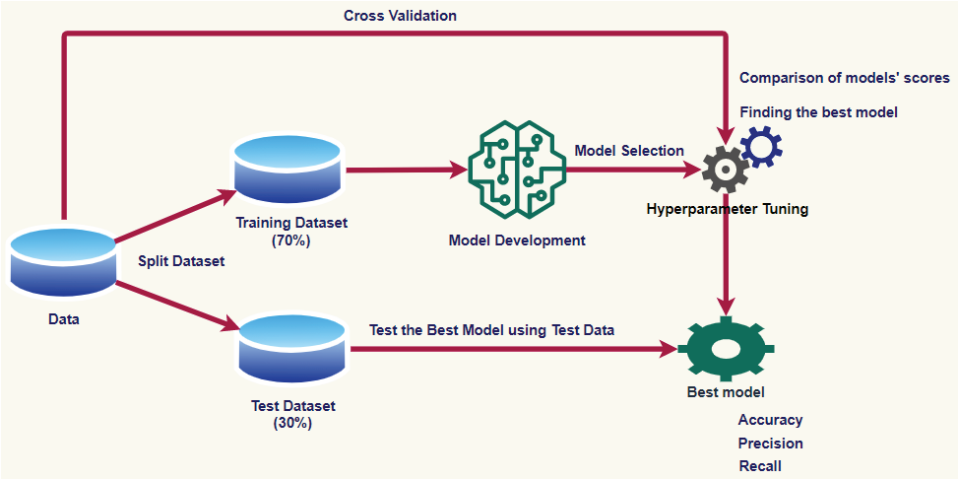

Figure 10. Finding the best model and testing it with test data.

data in each iteration. We determined the parameters that gave the highest accuracy value by trying the parameter to be adjusted for each classification model in turn.

On the one hand, the best parameters are found with GridSearchCV; on the other hand, it is important to ensure that the accuracy rates achieved in the training and test data sets are close to each other so that the model with these parameters does not overfit or underfit. As shown in Figure 12, if the model performs very well in the training dataset but not in the test dataset, there is an overfitting problem. That is, the variance of the model is high, it has memorized the training data and cannot generalize itself for new data. If the results are bad in both datasets, the model was over simplistic, produced high biases, and was not trained enough to generalize. When deciding on the best parameters of the model, it is necessary to find the best trade-off between variance and bias, as shown in Figure 13.

The model complexity shown in Figure 13 expresses the learning ability of a model and its fit to the data. Simple models have a high bias value because they do not learn well and cannot adapt to the data. Complex models, on the other hand, fit the data much better and react to even the

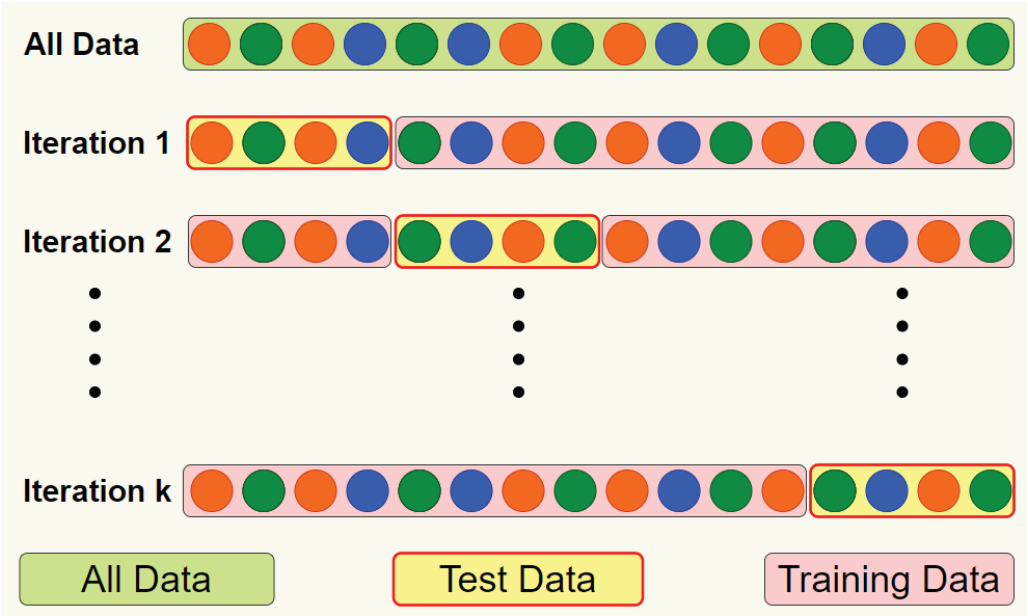

Figure 11. In the cross-validation method, all data is divided into k folds, each time a different fold is separated as test data, while the other folds are used as training data. The success of the model is calculated by taking the average of the result obtained in all iterations.

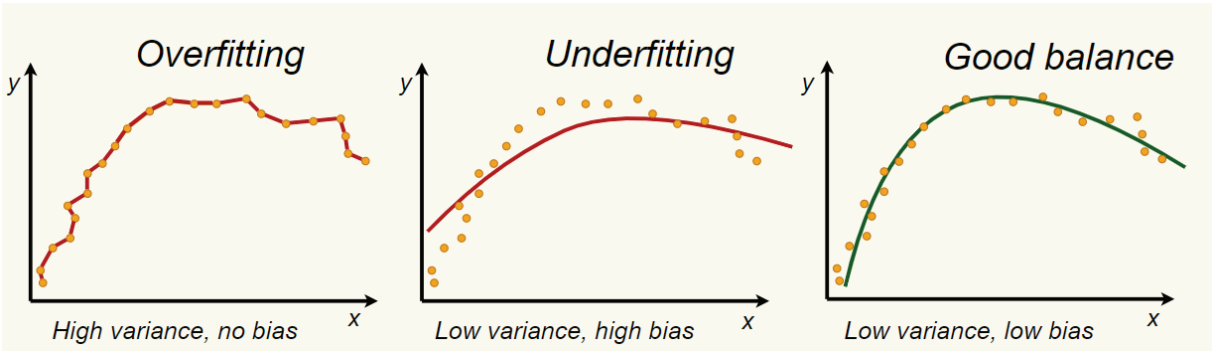

Figure 12. There is a problem of generalization in case of overfitting and inability to learn in case of underfitting. When the variance and biases are reduced, the ideal learning called good balance is reached.

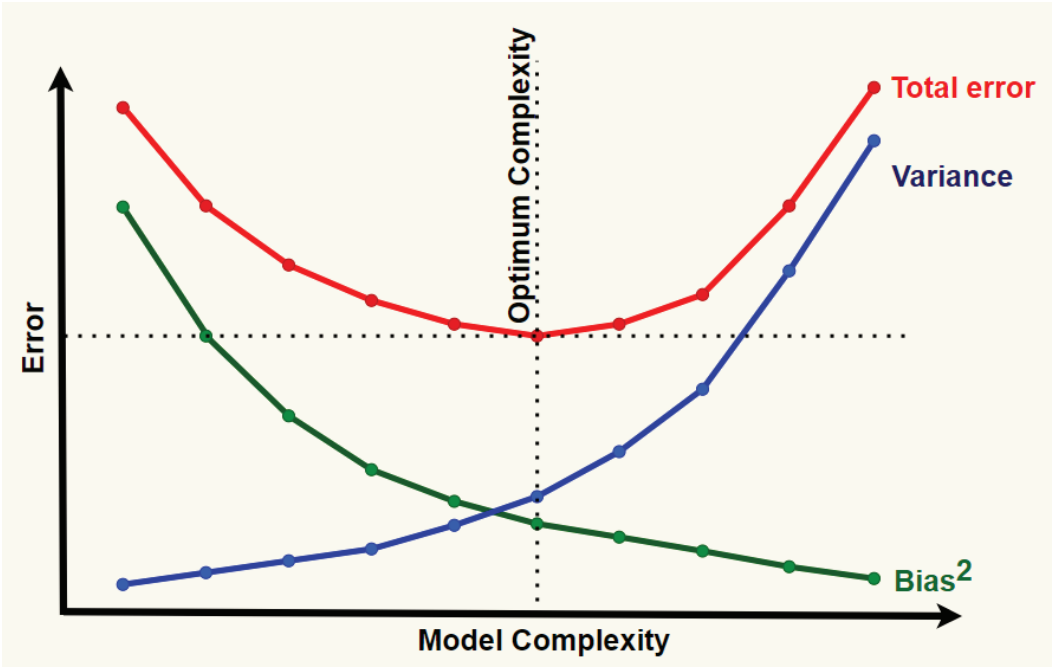

Figure 13. Optimal balance is achieved by making a trade-off between variance and bias when finding the optimal parameters of a model.

smallest changes in the data, which leads to poor performance on the test data. For optimum complexity, it is necessary to establish this balance and trade-off well.

Classification Algorithms

In situations where a clear link between outputs and inputs is not evident, machine learning algorithms employ a range of statistical, probabilistic, and optimization techniques to provide solutions by learning from past experiences, to draw meaningful conclusions from unstructured complex data stacks, and to detect patterns in large data sets [36]. These algorithms, categorized as supervised, unsupervised, and semi-supervised, have been successfully applied in areas such as intrusion detection [37], e-mail filtering [38], customer purchasing behavior detection [39], and manufacturing process [40]. In supervised learning, the model is trained to establish a relationship between the input set and the output in a labeled data set, and the most appropriate outputs are produced for the new data. In this process, input and output data are matched so that the model can make accurate predictions. Supervised learning is used especially in tasks such as classification and regression. Classification algorithms are used to separate data into specific categories, and these algorithms are used in various applications, such as determining whether an e-mail is spam or diagnosing a disease. In this study, which determines the driving style of a driver, supervised machine learning algorithms were used for the classification task.

When selecting machine learning models, it is necessary to pay attention to the type of data in the data set, the size of the data set, the complexity of the model to be designed, the generalization capabilities of the models, the resource consumption of the models, that is, the computational costs, and the interpretability of the outputs they produce. The fact that our data set consists of only numerical values, the size of our data set is not very large, and the computers on which the models work have large resources did not cause any restrictions in terms of computational costs. For this reason, it was possible to conduct experiments with many models of different complexities. The drive style was predicted with k-NN, support vector machines, decision trees, artificial neural networks, logistic regression, naive Bayes, random forest and LightGBM algorithms.

K-NN Classifier

The k Nearest Neighbor algorithm is a machine learning method that is straightforward, easy to grasp, and simple to implement [41]. Nearest neighbor methods find previously defined training examples at the closest distance to the new point and estimate the class of the new point accordingly. The k parameter specifies the number of nearest neighbors to be considered when classifying test data [42]. The distance calculation is determined by methods such as Manhattan, Euclidean, Minkowski as shown in Equation 14.

$$d(x,y) = \left(\sum_{i=1}^{n}(|x_i - y_i|)^q\right)^{\frac{1}{q}} \tag{14}$$

In this equation, referred to as the Minkowski method, setting (q = 1) results in the Manhattan distance, while setting (q = 2) uses the Euclidean distance for calculations.

Neighbor-based methods are considered non-generalizing machine learning methods because they can simply evaluate all the training data. Despite its simplicity, k-NN has been successful in many classification problems. Since it does not require parameters, it is often effective in classification scenarios where the decision boundary is highly irregular. It stores training data samples but does not create a model with that data. In classification using this method, the class of each point is determined by simple majority voting of its nearest neighbors, as shown in Figure 14.

Since k-NN, being one of the lazy algorithms, does not create any model, the training phase is costless and does not take time. However, the testing phase requires more time and memory. If the data set contains a large amount of data, scanning and storing all of them may strain the computer's memory usage. The success rate decreases in data sets with very high dimensionality and uneven distribution. To get better results with k-NN, outliers in the data set should be cleaned, and scaling should be done.

Support Vector Machines Classifier

Support vector machines (SVM) are a technique devised by Vapnik, grounded in statistical learning theory [43]. SVM is capable of classifying both linear and non-linear datasets. In this approach, the input space is transformed into a multidimensional inner product space known as the feature space, where the optimal planes are identified to enhance the classifier's generalization capability. Optimization theory and related statistical learning principles are utilized to identify the optimal planes. The marginal distance for a class refers to the gap between the decision hyperplane and the closest instance belonging to that class. To carry out the classification, we must identify the hyperplane that maximally separates the two classes, as illustrated in Figure 15. SVM offers several advantages: it can handle a large

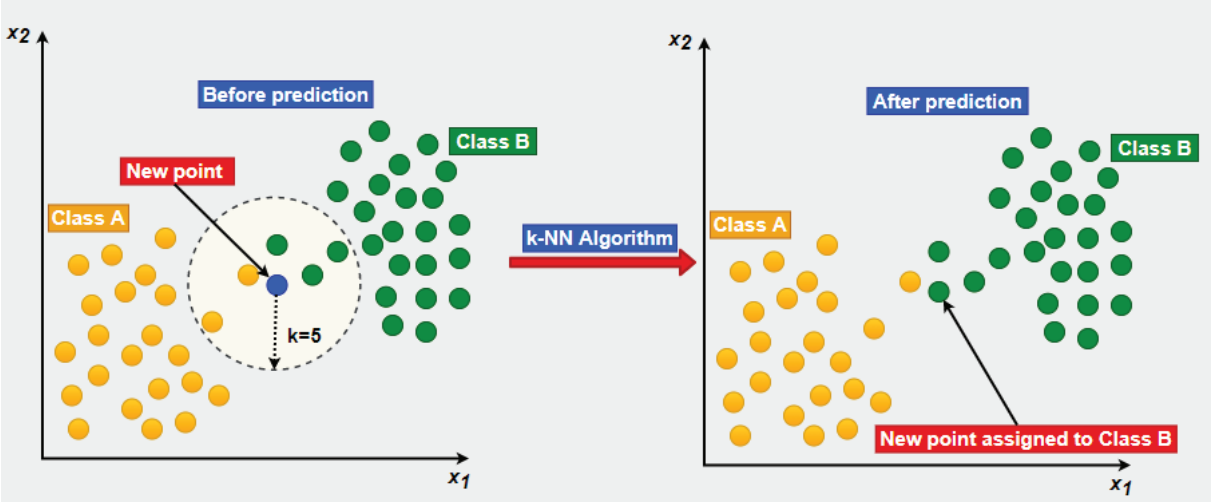

Figure 14. Assigning a new sample shown in blue to the green class with the k-NN algorithm with the nearest neighbor parameter k=5.

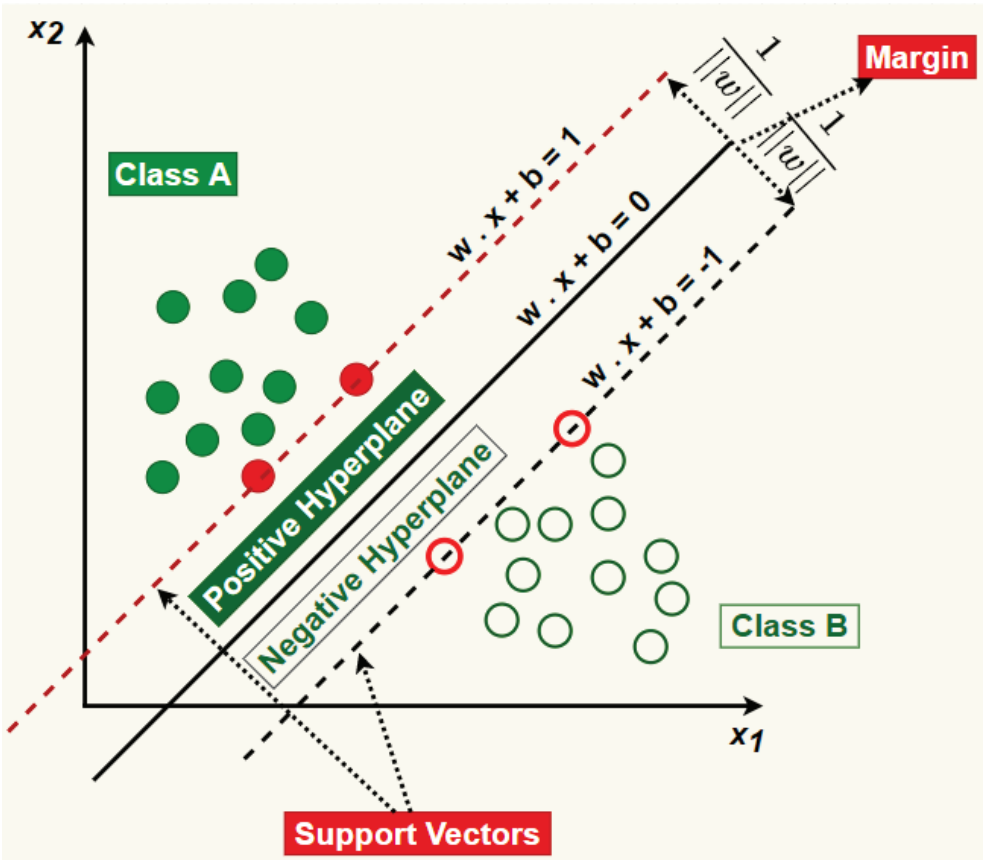

Figure 15. A hyperplane is determined by the SVM classifier that provides the furthest distance between the two classes.

number of independent variables, requires minimal input for learning, applies to both linearly separable and non-separable data, and delivers high accuracy results.

Let $y_i \in \{-1,+1\}$ be the class label corresponding to N data vectors, each of which consists of a data pair in n-dimensional real space. In this instance, x_i belongs to the first class if y_i = -1 and the second class if y_i = +1. Equations 15,16 provide a hyperplane of f(x) that separates linearly separable data, where f(x) > 0 if x_i belongs to the positive class and f(x)<0 if x_i belongs to the negative class. This hyperplane can be characterized as:

$$f(x) = w.x + b = \sum_{j=1}^{n} w_i.x_i + b \tag{15}$$

$$y_i f(x_i) = y_i(w.x + b) \geq 0 \quad i = 1,2,\dots N \tag{16}$$

Here, b is the threshold, and w is an n-dimensional vector. Furthermore, Equation 17 can be expressed as follows if w and x are the points that are closest to the extreme plane at a distance of 1/w:

$$y_i(w.x + b) \geq 1, \qquad i = 1,2,\dots N \tag{17}$$

The ideal parser extreme plane is the plane farthest from the extreme plane to the closest point. Maximizing this parser plane gives the SVM its capacity for generalization.

Decision Tree Classifier

Decision trees are one of the oldest and leading machine learning algorithms. The structure of this technique is based on the hierarchical decomposition of data [44]. The decision tree as a classifier was introduced by D. Morgan [45] and developed by JR Quinlan [46]. A decision tree models a decision logic over previously realized data. All internal nodes in the decision tree represent input variables or tests on the relevant attribute. Depending on the test result, the classification algorithm continues the test and branching process until it reaches a leaf node [46]. The extreme nodes, called leaves, indicate the decision of the model. The decision tree is a white-box model, offering a complete explanation of why a sample belongs to a particular class rather than another. Additionally, this method allows us to identify which feature is most significant for the output. This algorithm creates a decision-making process based on information entropy. Entropy shows the likelihood and uncertainty of an unexpected event. If all samples are identical, the entropy is 0. If the values are evenly distributed,

the entropy is 1. E(S) represents the information entropy of the dataset S. It is calculated as in Equation 18;

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{18}$$

Here, p represents the probability of success. The entropy for all input attributes is computed as shown in Equation 19;

$$E(T,x) = \sum_{c \in X} P(c)E(c) \tag{19}$$

Where P(c) is the probability of the potential data point in x, E(c) is its value, T is the output attribute, and x is the input attribute. Figure 16 shows the classification of the driving style with the decision tree according to the attributes extracted from the drive cycle in a residential area.

Neural Network Classifier

To carry out a particular activity or function, machine learning algorithms known as artificial neural networks (ANNs) are inspired by the human brain. Warren McCulloch and Walter Pitts laid the foundations of this algorithm by describing a simple mathematical model for a neuron that receives inputs, processes these inputs, and

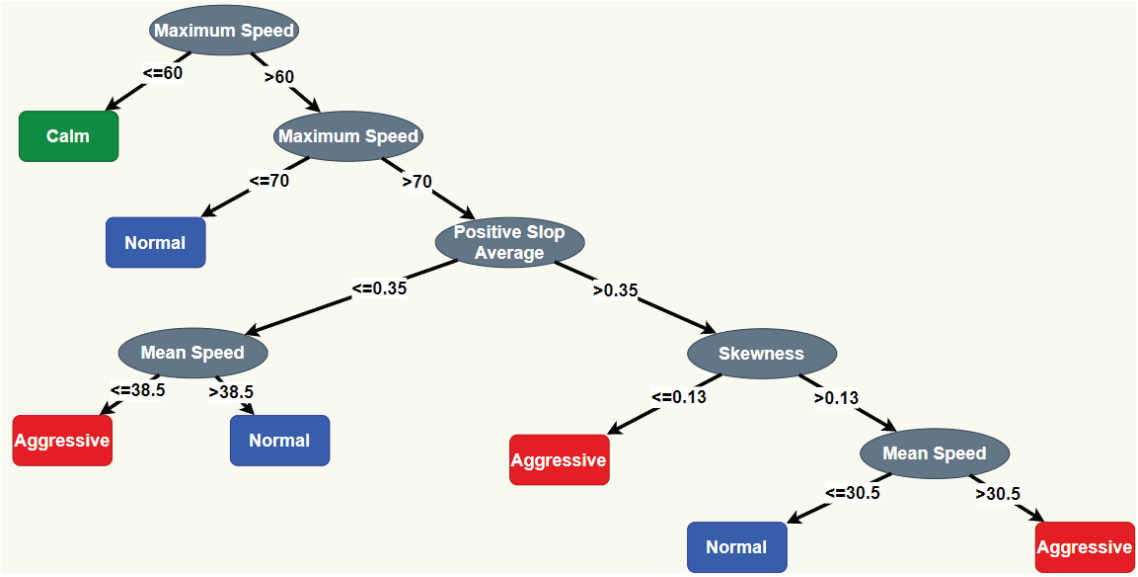

Figure 16. Classification of driving style with decision tree based on features extracted from drive cycle.

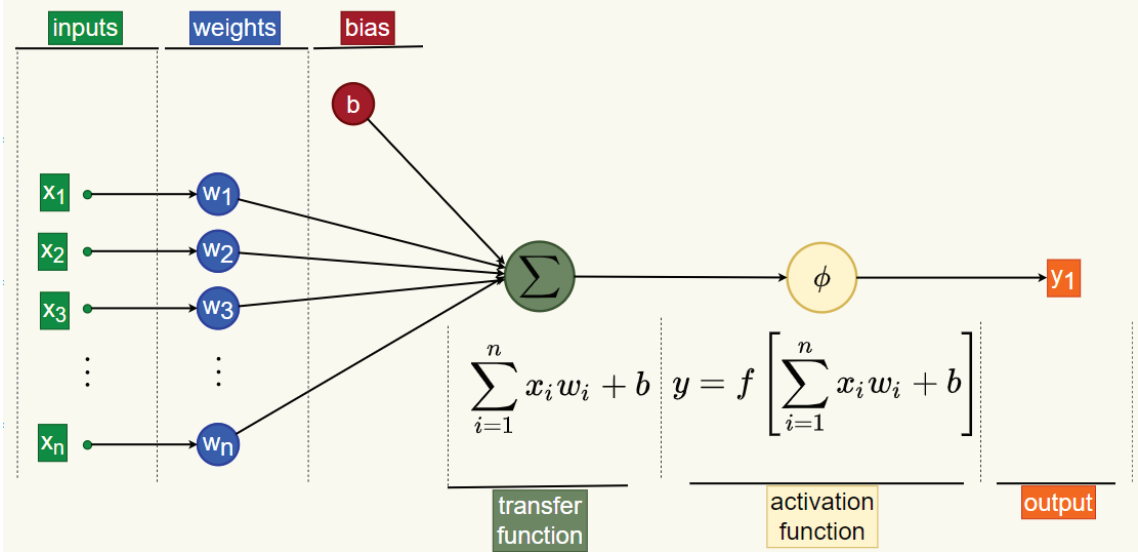

Figure 17. Calculations in a neuron.

returns an output [47]. A collection of input/output units with a weight assigned to each neuron is called a neural network. In order to forecast the right class label of the inputs, the ANN continuously modifies the weights during the learning phase.

In an artificial neural network, the inputs are multiplied by a weighting factor and summed and transferred to a transfer function. Figure 17 shows the input and output parameters, weight values, transfer, and activation functions of an artificial neuron.

The artificial neural model's mathematical expression shown in Figure 17 is shown in Equation 20.

$$y = f\left[\sum_{i=1}^{n} x_i w_i + b\right] \tag{20}$$

Here, x_i represents the inputs, w_i the weights, b the bias, and y the output. f indicates the activation function. The difference between the expected output and the obtained output is called the error. The weighting factors should be changed when the error is greater than the specified threshold. With this process, called backpropagation, the neurons are trained once again for new calculations and the errors are reduced [48]. The first phase of backpropagation is the feed-forward phase, in which the input information at the node is forwarded to calculate the output information. The second stage involves adjusting the connection strengths in response to variations between the output units' estimated and actual values. The main goal here is to reduce the discrepancy between the calculated and actual outputs through repeated processes [49]. This process is called artificial neural network learning.

There are two types of neural networks, feedforward and feedback. A feedforward neural network is not iterative. Neurons in one layer are only connected to neurons in the next layer and do not form a loop, signals only travel towards the output layer. In a feedforward neural network, the neurons are organized in layers and the outputs of the neurons in one layer are given as inputs to the next layer [50]. The mathematical expression of the structure shown in Figure 18 is given in Equation 21. Feedback neural networks, also known as recurrent neural networks, contain loops. Signals move in both directions, forming loops in the network. Feedback loops can cause the behavior of the network to change over time, depending on its input.

$$y_{i,k} = \varphi_{i,k}\left(\sum_{j=1}^{n_{k-1}} w_{j,k-1}^{i,k}\, y_{j,k-1} + b_{i,k}\right) \tag{21}$$

In Figure 18, n is the number of inputs, m is the number of outputs, k is the number of layers, and $h(k)$ is the number of neurons in the k-numbered layer. In Equation 21, $y_{i,k}$ is the output of neuron i in layer k. $n_{(k-1)}$ is the number of neurons in the (k-1) layer. $w_{j,k-1}^{i,k}$ is the connection weight between neuron i in layer k and neuron number j in layer (k-1). $b_{i,k}$ is the residual value for neuron i in layer k. $\varphi_{i,k}$ is the activation process applied to neuron i in layer k. Linear, threshold, sigmoid or tangent hyperbolic functions are used as activation functions.

Logistic Regression Classifier

Despite its name, logistic regression is used more for classification problems than regression in machine learning applications. In the literature, logistic regression is also

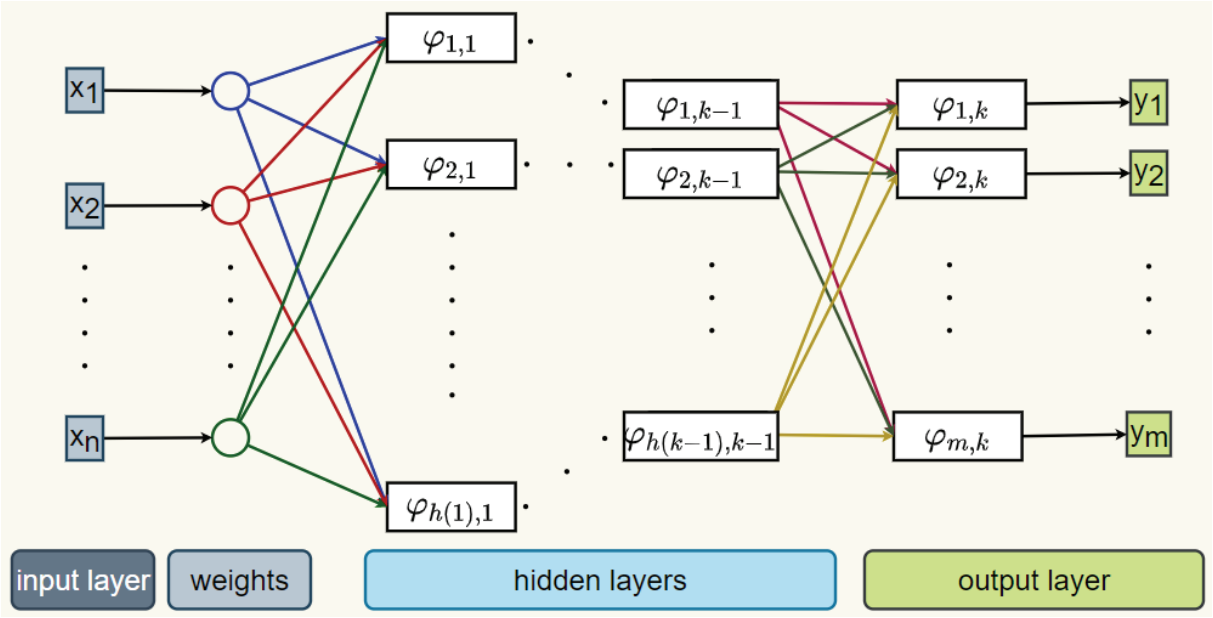

Figure 18. Structure of feedforward neural network.

known as maximum entropy classification or log-linear classifier. In logistic regression, the probability of an event can be statistically modeled using the values of categorical or numerical independent variables [51]. Logistic regression is a statistical model in mathematics that finds the relationship between x and y using a logistic or logit function. The logit function maps y as the sigmoid function of x as in Equation 22 and produces a value between 0 and 1 as shown in Figure 19. In linear regression, we use the equation shown in Equation 23, where each unit change in X influences $p(X)$ by β_1, resulting in a continuous variable. Since logistic regression aims to determine the probability of the dependent variable falling into a specific category, we will modify the left side of the equation in Equation 23 to yield a value between 0 and 1, as shown in Equations 24-26.

$$f(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{1+e^{x}} \tag{22}$$

$$p(X) = \beta_0 + \beta_1 X \tag{23}$$

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \tag{24}$$

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X} \tag{25}$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \tag{26}$$

As illustrated in Equation 26, the relationship between X and p(X) is no longer linear. The value of P(X) will be between 0 and 1.

Naive Bayes Classifier

Naive Bayes classifier is a simple, fast, accurate, and reliable supervised learning algorithm. The naive Bayes classifier is based on Bayes' theorem with the assumption of independence, which assumes that a particular feature in a class is independent of other features. Even though these features are dependent on each other, they are considered independent, thus simplifying the calculation. This is where the name Naive comes from. In this algorithm, statistical and probabilistic inferences are made between previous inputs and outputs [52]. The probability of a previous example is used to approximate each particular class.

According to Bayes' rule, the probability of both events A and B occurring is the product of the probability of A and the conditional probability of B given A, as demonstrated in Equation 27.

$$P(A \cap B) = P(A) \times P(B|A) \tag{27}$$

Here, P(A) represents the probability of event A occurring on its own, and P(B|A) denotes the probability of event

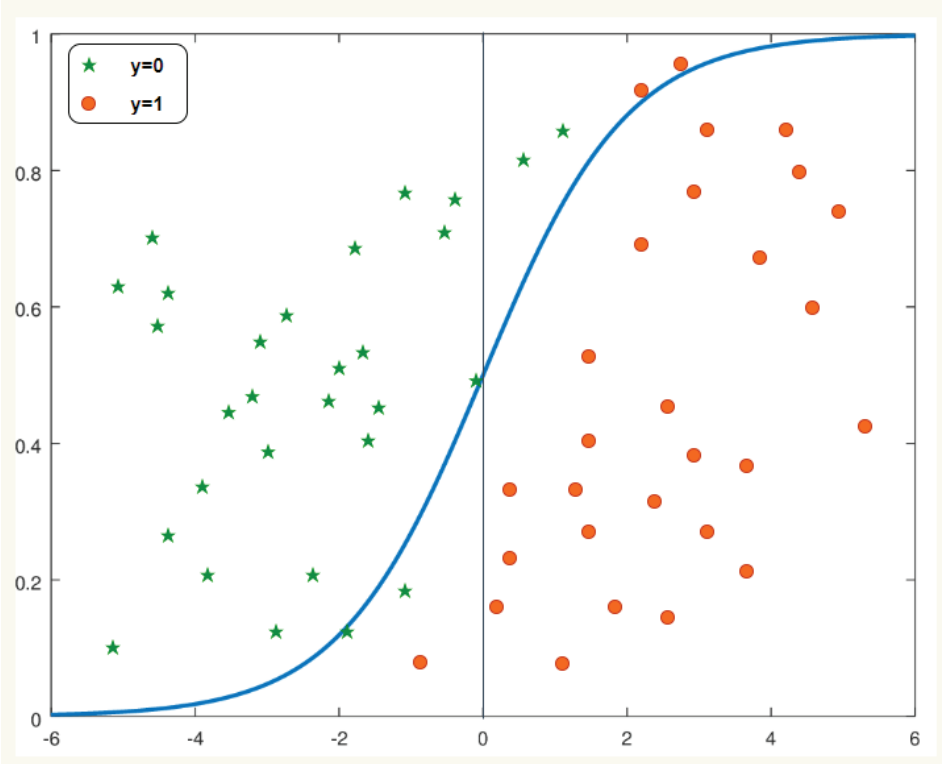

Figure 19. In logistic regression, the y values that will determine the output class take values between 0 and 1.

B occurring given that A has already occurred. This equation can also be written as shown in Equation 28:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{28}$$

Conditional probability distributions show the cause-and-effect linkages among a Bayesian Network model's inputs. Equation 29 represents the fully integrated probability distribution of a Bayesian network from the arguments $X_1, X_2, \ldots, X_n$.

$$\begin{aligned} P(X_1, X_2, \ldots X_n) &= P(X_1|X_2, X_3, \ldots X_n)P(X_2|X_3, \ldots X_n) \\ &\quad \ldots P(X_{n-1}|X_n)P(X_n) \\ &= \prod_{i=1}^{n} P(X_i|X_{i+1}, \ldots, X_n) \end{aligned} \tag{29}$$

The Naive Bayes algorithm, the theory of which is explained in more detail in [52], produces both fast and successful results in matters such as document classification, medical diagnosis, monitoring of system performance, and spam filtering, despite its simplified assumptions [53].

Random Forest Classifier

Random forests are a supervised machine-learning method that uses many decision trees for classification and regression problems. In random forest applications, subsets of the data set are applied to various decision trees. The outputs from these trees are aggregated to generate the final output. As shown in Figure 20, random forests take the predictions of all trees, instead of relying on a single decision tree, and produce the final output using a technique called majority voting. Assessing the outcomes of decision trees executed with various and random feature selections collectively enhances accuracy and mitigates the overfitting problem [54].

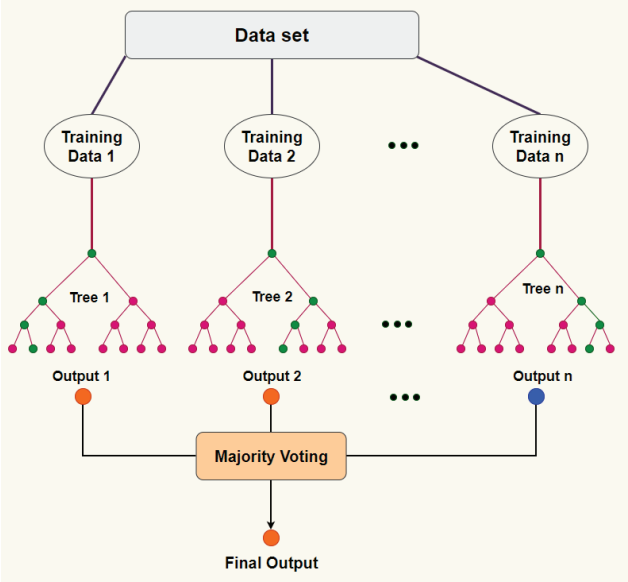

Figure 20. Random forest produces more accurate and effective outputs by re-evaluating the predictions from many decision trees.

The random forest method produces flexible and highly accurate predictions for both classification and regression problems. It works well with both categorical and numerical values. Since it uses a rule-based algorithm, it is not necessary to normalize or standardize the data set. However, it is important to pay attention to the number of decision trees used; as the number of trees increases, the training time will increase, and finding the final result will require more computational power. Random forest is not good at interpretability and determining the variables affecting the

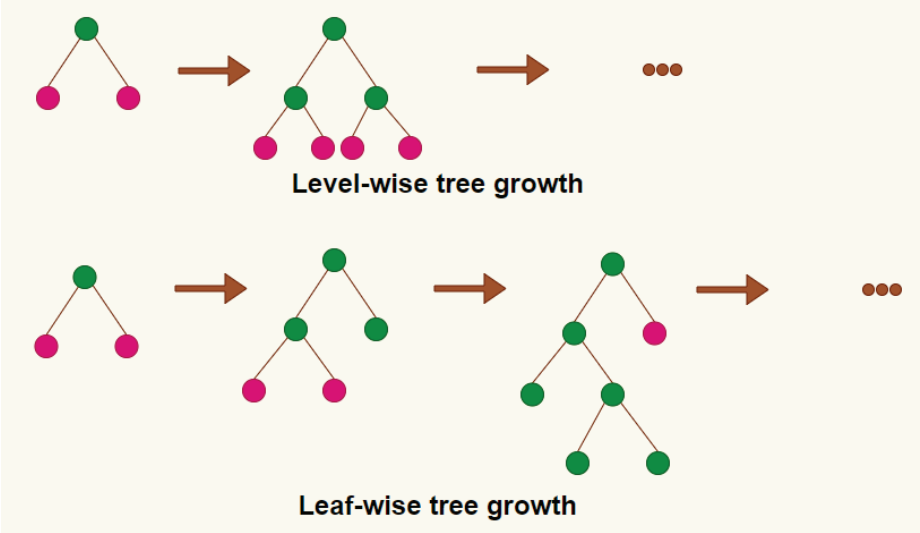

Figure 21. In the leaf-wise growth method, the leaf with the highest loss is split.

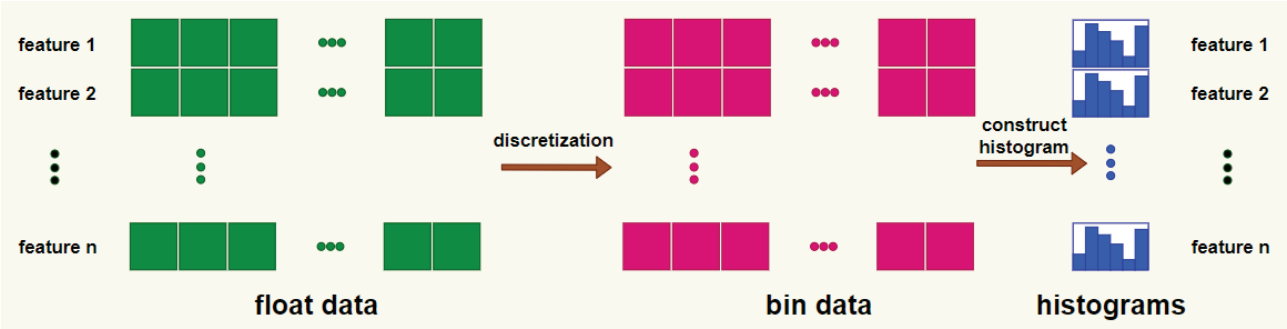

Figure 22. Histogram algorithm of LightGBM.

result; at this point, it differs from decision trees. Since a large number of decision tree outputs are converted to the final output by voting or averaging, this method loses its feature of being a white-box model.

Light GBM Classifier

LightGBM (Light Gradient Boosting Machine) is a machine learning method developed by Microsoft for classification, regression and time series problems. It has a similar structure to Gradient Boosting Decision Trees (GBDT) and XGBoost algorithms, which are efficient implementations of decision trees. The biggest difference between LightGBM and methods such as GBDT, XGBoost, and CART is that it uses a leaf-based splitting strategy instead of a level-based one [55]. As seen in Figure 21, the level-wise growth method keeps the trees balanced, while the leaf-wise growth method results in a more unbalanced growth pattern. Leaf-wise minimizes the total loss by splitting the leaf with the most loss. Since the tree grows depending on the number of leaves, it can cause over-learning when working with small data sets. To prevent this, it is necessary to adjust the tree depth parameter well.

Another feature of LightGBM is the histogram algorithm. As shown in Figure 22, each attribute dataset is divided into k integer intervals, and a histogram is created. The gradient value and sample count of each divided data sample are stored in the histogram. In this way, the data is made ready for use in a simpler way, less memory is consumed, and the training speed and efficiency of the model are higher [56,57].

Comparison of Classification Algorithms

When comparing machine learning models, in addition to metrics such as accuracy, precision, recall, f-score, ROC-AUC score, it is also necessary to consider features such as training and prediction time, resource usage, generalization capability, i.e. not overfitting, understandability, easy optimization, and the interpretability of the outputs. In this section, the advantages and limitations of the eight classification algorithms listed above are presented together in Table 7.

Accuracy Measurement

The performance of a classification algorithm is measured by finding some success metric from the results that the classification model calculates for the test data. In studies using classification algorithms, it is not sufficient to consider only the accuracy rate. Especially in unbalanced datasets, the accuracy rate alone cannot give accurate information about the classification performance. For this, a confusion matrix is created, as shown in Table 8, which compares the actual target values with those predicted by the model to evaluate the classification performance [58]. According to this table, a 'True Positive' (TP) situation occurs if a situation that actually exists positively in the forecasting process is predicted positively. If the existing condition is negative and the model's prediction is negative, a 'True Negative' (TN) condition occurs. If the existing condition is negative but the model predicts the result as positive, a 'False Positive' (FP) condition with a Type1 error occurs. A 'False Negative' (FN) condition with a Type 2 error occurs if the existing condition is positive and the model predicts the result as negative. Accuracy, precision, sensitivity (recall), true positive rate, false positive rate, and F1 score values are calculated from these values written in this matrix. In equations 30-34, the mathematical expressions of the success criteria mentioned above are given. The metric named support shows the number of samples found in each class of the test dataset.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{30}$$

$$Precision = \frac{TP}{TP + FP} \tag{31}$$

$$Recall = True\ Positive\ Rate = \frac{TP}{TP + FN} \tag{32}$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN} \tag{33}$$

Table 7. Advantages and limitations of the classification algorithms described in this article

Classification algorithm	Advantages	Limitations
k-NN	-Understandable and easy to apply. -No training time, fast. -Does not contain many parameters. -Easy to optimize. -Can process samples with missing data.	-Very high dimensional data will reduce success. - Vulnerable to inconsistent data. -Scaling required. -Classes in the dataset must be balanced. -Classification performance may decrease as attributes are given equal importance.
Support vector machines	-Can work with a large number of variables. -High accuracy and performance. -Can learn with little input. -Applicable to non-linear data. -It is also effective in cases where the number of dimensions is more than the number of samples.	-The computational cost increases as the dataset gets larger or more complex. -Performance degrades in noisy or incorrect data. -It is difficult to understand the input-output relationship in the model. -In cases where the number of features is more than the number of samples, care should be taken to avoid overfitting and regularization terms when choosing kernel functions.
Decision tree	-The classification tree is easy to understand and interpret. Trees can be visualized. -Can handle both numeric and categorical data. -It is possible to validate a model using statistical tests. -Learning and prediction time is low.	-The model can generate overly complex trees that do not generalize data well with over-learning. This requires mechanisms such as pruning, setting the minimum number of specimens required per leaf, or setting the maximum depth of the tree. -The outputs of decision trees are piecewise fixed approaches. Therefore, they are not good at extrapolation. -The decision tree model creates biased trees if some classes are dominant. Therefore, it is recommended to balance the dataset before working with the model. - It is very sensitive to minor distortions in data.
Neural networks	-Can detect non-linear relationships between dependent and independent variables. -It can be applied to both classification and regression problems. -Gives successful results for datasets containing many entries.	-It is a black box model. Because the interaction between input and output is unpredictable, it requires more time and computational power. -Can only work with numeric data. -For complex problems, the computational cost of the training process increases.
Logistic regression	-Easy to understand and apply. -Performs very well on the linearly separable dataset. -Does not require any settings. -The computational cost is low.	-The accuracy of the model decreases if the input variables have complex relationships. -In cases where the number of features is more than the number of samples, an overfitting problem occurs.
Naive bayes	-It is fast and useful especially in large datasets. -Suitable for multiple classification problems. -Suitable for numerical and categorical data.	- If the classes are close to each other, success will decrease. -Normal distribution is taken as a basis when converting numeric data into categorical data.
Random forest	-Since it combines the predictions of multiple decision trees, it reduces the variation that may occur in a single tree and provides higher accuracy. -Because it integrates the predictions of several trees, it can withstand noisy, missing, or outlier data.	-Using many trees causes excessive memory usage and increased training time. -Since random forest combines the predictions of multiple trees, it becomes difficult to understand the underlying reason for the decision made by the model.
LightGBM	-Histogram-based algorithm that separates continuous feature values into separate bins increases speed and reduces memory usage. -Provides higher accuracy with leaf-based binning approach.	-The leaf-based splitting strategy may lead to overfitting, especially on small data sets, when very complex trees are produced.

Table 8. Confusion matrix

	Predicted	
Actual	True Positives (TP)	False Negatives (FN)
	False Positives (FP)	True Negatives (TN)

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (34)$$

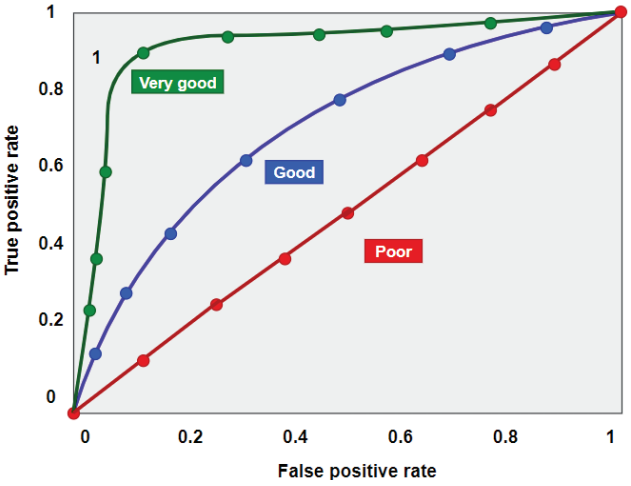

Figure 23. ROC curve showing classifier performance.

One of the criteria that indicates the performance of the classifier is the receiver operating characteristic (ROC). A ROC curve is formed by plotting the true positive rate versus the false positive rate at various threshold settings, as shown in Figure 23 [59]. The area under the ROC curve (AUC) is commonly used to evaluate the performance of a classifier. The larger the AUC value, the better the classifier predicts. Figure 23 represents three ROC curves within an abstract dataset. Since the area under the red ROC curve is half of the rectangle, its AUC value is 0.5. The blue and green curves cover a larger area, and their AUC values are higher. It can be said that the green curve, whose AUC value is very close to 1 in the figure, has a very high predictive accuracy.

As shown in Figure 23, a poor model making random predictions will have equal TPR and FPR values. The ideal scenario is to maximize TPR while minimizing FPR. In this case, true positives will be correctly classified and false positives will not be incorrectly classified.

RESULTS AND DISCUSSION

In this section, first, detailed test results of eight classification algorithms will be presented. Second, the success of these eight algorithms will be compared in terms of accuracy and processing time. Third, all the results we obtained in this study will be compared with other driving style prediction studies in the literature.

Classification Model Results

This section includes performance evaluation to compare classification algorithms to recognize driving style. This study was carried out using Scikit-learn library [35] in Anaconda Spyder environment on a 64-bit computer with an 11th Gen Intel® Core™ i7-1185G7 processor at 3.00 GHz. Three different driving style diagnoses were made with eight different classifiers using three different data sets consisting of features extracted from real driving data.

Before analyzing results, it is essential to choose the best-suited architecture for effective machine learning algorithm application. In most cases, this optimal architecture isn't immediately apparent and requires experimentation with various hyperparameter settings. Hyperparameters are adjustable parameters that control a machine learning model's behavior. Some key hyperparameters include:

-Nearest neighbor distance criterion
-Decision tree maximum depth
-Decision tree minimum leaf node samples
-Random forest number of trees
-Neural network layer and neuron counts

To identify the optimal hyperparameters, a common approach involves defining potential value ranges for each parameter, systematically trying different combinations, and evaluating their performance using appropriate metrics (e.g., accuracy, precision, recall). This process helps to balance training and testing accuracy, avoiding overfitting. In other words, when choosing hyperparameters, it is not enough to focus on achieving the highest accuracy on the training data; the hyperparameters that provide the highest accuracy should be selected such that the accuracies on both the training and test data are as close to each other as possible.

In this section, the results obtained with eight classification algorithms will be shown. The parameters of these algorithms were adjusted by choosing the most appropriate complexity value to avoid overfitting and underfitting situations. The selected parameters for each algorithm are explained in their respective sections.

K-NN Results

The results shown in Figure 24 were obtained with the k-NN model, which we preferred because it is straightforward to comprehend, optimize, and apply. Neighbors, weights, and metric parameters were adjusted to optimize the model. Selecting the 'distance' parameter as the weight parameter means that neighbors closer to the query point will have a greater influence than those farther away. Choosing Euclidean as the metric parameter means that the distances are determined by taking the square root of the sum of the squares of the distances in each dimension.

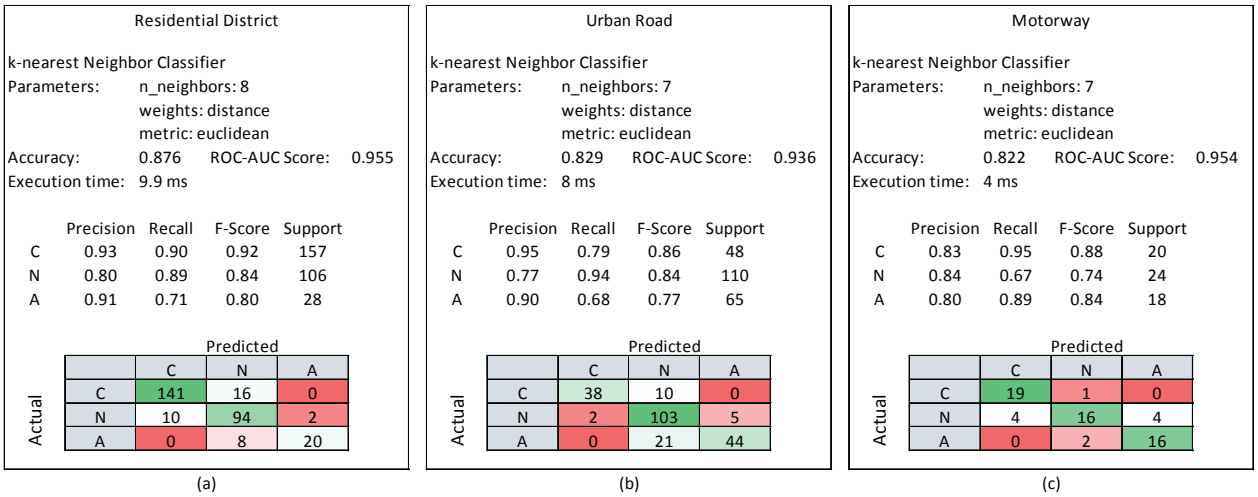

Figure 24. Results for a) residential district, b) urban road, c) motorway datasets with the k-NN algorithm.

Cleaning outliers, addressing the multidimensionality of the data set, and scaling affected the model results positively, and an average accuracy of 84.2% was reached for the three data sets. The confusion matrix shows that it cannot separate the aggressive class from the normal class. We evaluate that higher accuracy can be achieved if the classes in the data set are more balanced.

SVM Results

The results obtained with the SVM, which we prefer because it is not affected by the number of samples and dimensions, and has high accuracy and performance, are shown in Figure 25. We adjusted the regularization parameter, gamma, and kernel parameters to optimize the model. The regularization parameter, which takes only positive values, tells the model how much we want to avoid misclassifying each training sample. By determining the most appropriate value, we aim to maximize the margin between both classes and minimize the amount of misclassification. The gamma parameter is required when using the RBF kernel, which defines how far the effect of a single training sample reaches. The smaller the gamma value, the more linear the model behaves. When gamma is selected high, the model curvature will be high. As the gamma decreases, the regions separating the different classes become more generalized, causing overfitting as it gets larger. The kernel parameter specifies the kernel type to be used in the algorithm. The kernel function enables the linear separation of classes by transforming a low-dimensional input space into a higher-dimensional space. Since the data set is relatively small, the computational cost is low. An average of over 90% accuracy was achieved for the three data sets. The highest scores were obtained for the residential district dataset with 98.6%.

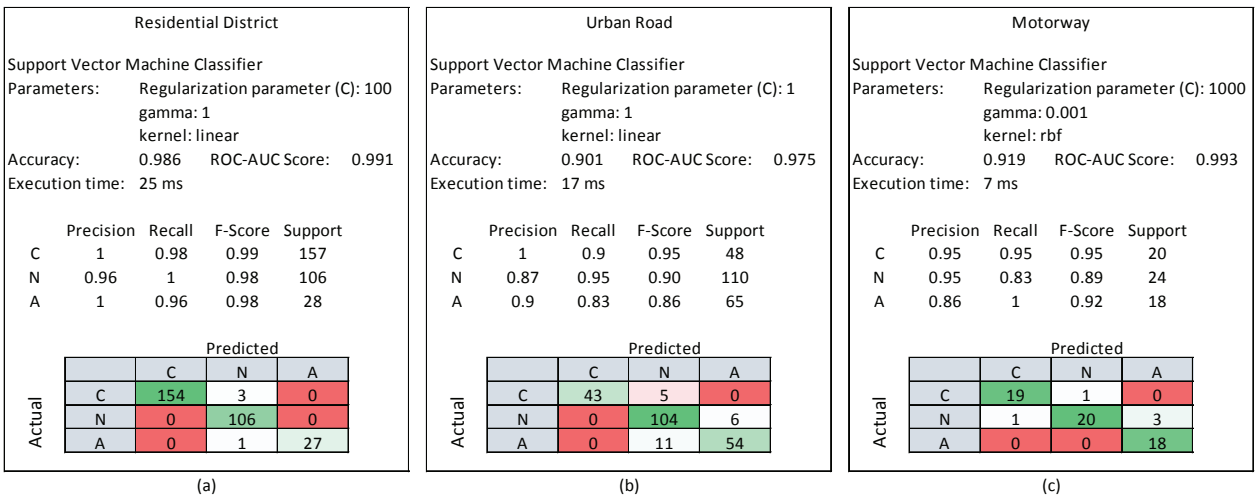

Figure 25. Results for a) residential district, b) urban road, c) motorway datasets with the SVM classifier.

Decision Tree Results

Figure 26 shows the results obtained with the decision tree, which is the most interpretable and visualizable among the classification algorithms. We used the Gini index for two datasets and information entropy for one dataset to decide which feature to split during model optimization. The "best" strategy for choosing the split at each node yielded the best result. For the most accurate classification, we tried different maximum depth and minimum sample number parameters. The response time of the decision tree is reasonable, and the average accuracy rate for all datasets is 96.86%. According to these results, the decision tree is one of the most suitable algorithms for classifying driving style.

MLP Results

The artificial neural network produces solutions to nonlinear problems with activation functions. The best learning is achieved with the tangent hyperbolic function, which serves as an activation function providing faster learning and a wider range for classification. Figure 27 shows the different regularization terms (alpha), the hidden layer and the number of neurons in these layers, the learning style, and the number of iterations. Although the success rate is high, this is the model with the highest processing time. In this study, the processing time is relatively low since the number of samples in the data set is not very large. However, when the data size increases, this should also be considered.

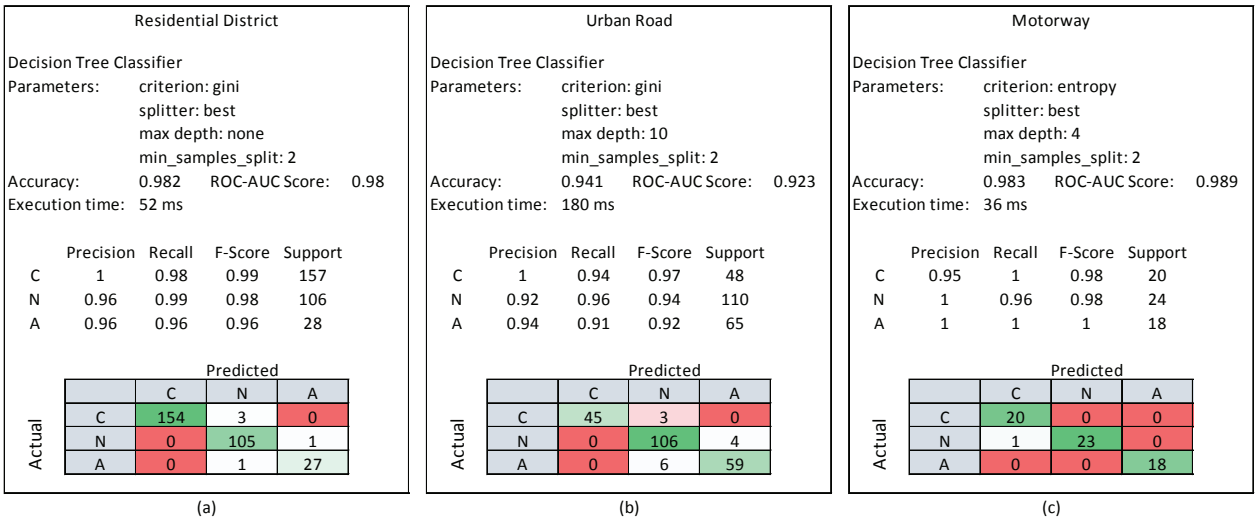

Figure 26. Results for a) residential district, b) urban road, c) motorway datasets with the decision tree algorithm.

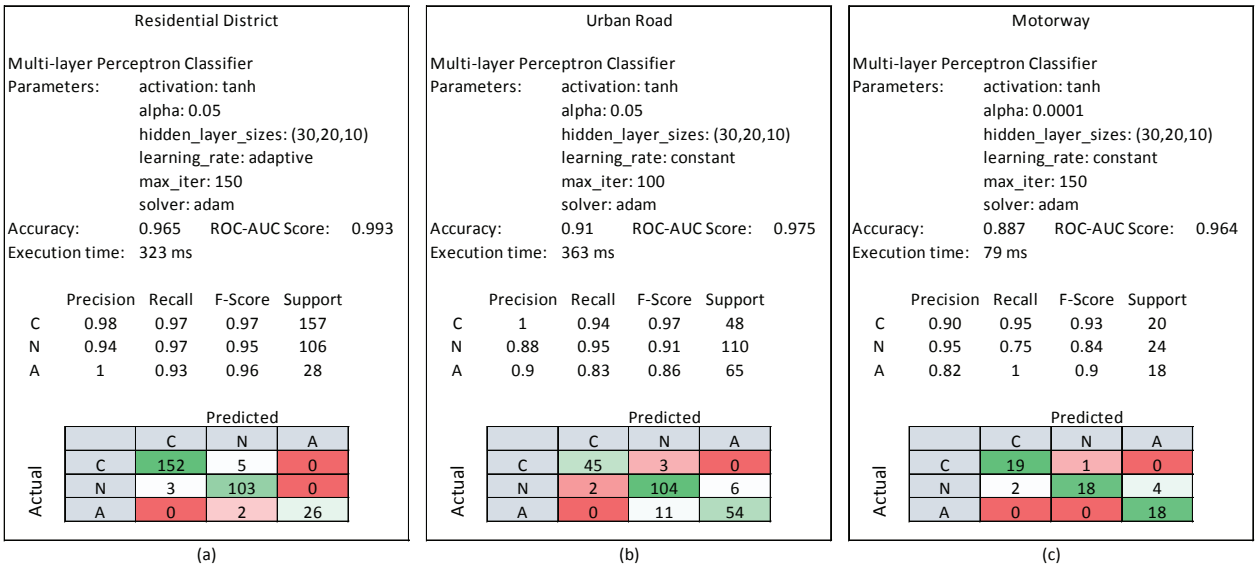

Figure 27. Results for a) residential district, b) urban road, c) motorway datasets with the MLP model.

Logistic Regression Results

We reached the optimum solution in logistic regression with newton-cg, which is preferred in multi-class problems. The highest success rate was obtained with the default penalty factor. The regularization factor (C) was set to 100 for two road types and 1 for one road type. When the results obtained in Figure 28 are examined, it is seen that logistic regression provides a favorable outcome in terms of transaction cost and accuracy.

Naive Bayes Results

The Naive Bayes classifier does not contain many tuning parameters. The only hyperparameter, the variance smoothing parameter, adds a user-defined value to the distribution's variance. This essentially broadens or smooths the curve so that the model can explain more samples that are further away from the distribution mean. Here, too, we calculated the best parameter (var_smoothing) values using the grid method and showed the results in Figure 29.

Random Forest Results

Random forest evaluates predictions made from many trees together instead of a single decision tree. It yields a final outcome from all the predictions produced according to the majority vote mechanism. The most appropriate selection of the number of trees and their features increases the accuracy and prevents the overfitting problem. Here, the hyperparameters of how many trees will be used (n_estimators), how many features will be considered in each tree (max_features), the depth of each tree, i.e., how many times it will be split (max_depth), and how many times the nodes in the tree should be split (max_leaf_nodes) are of great importance. The parameters in Figure 30 provide the

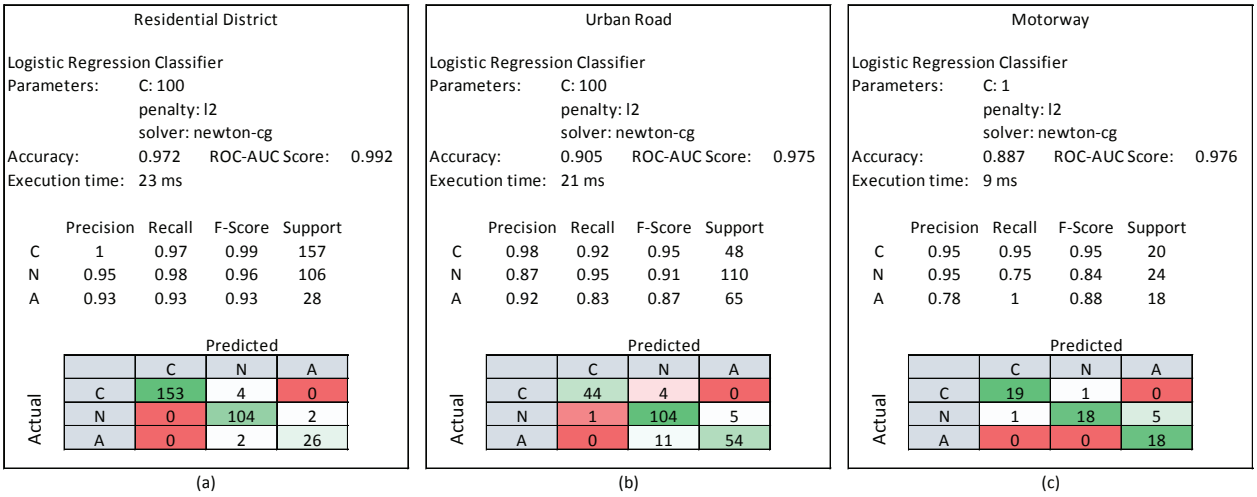

Figure 28. Results for a) residential district, b) urban road, c) motorway datasets with the logistic regression classifier.

Figure 29. Results for a) residential district, b) urban road, c) motorway datasets with the Naive Bayes algorithm.

best strategies in the random forest algorithm, achieving an accuracy of over 95% in all types of paths.

Light GBM Results

LightGBM, which is designed to optimize model efficiency and memory usage, utilizes an optimal decision tree structure. Key components include leaf-based tree growth and histogram-based algorithms. Effective hyperparameter tuning is crucial for LightGBM's performance. We determined the hyperparameter settings to best classify the driving data, including the learning rate, the number of trees (n_estimators), the type and number of loss functions to be optimized (objective, num_class), and the maximum number of leaves in each tree (num_leaves), as shown in Figure 31. While determining these parameters, we ensured that the model did not fall into over-learning error. The results in Figure 31 show that we achieve very high accuracies, similar to other methods.

Comparison of Classification Algorithms

In this study, we comprehensively analyzed the performance of different machine learning algorithms in

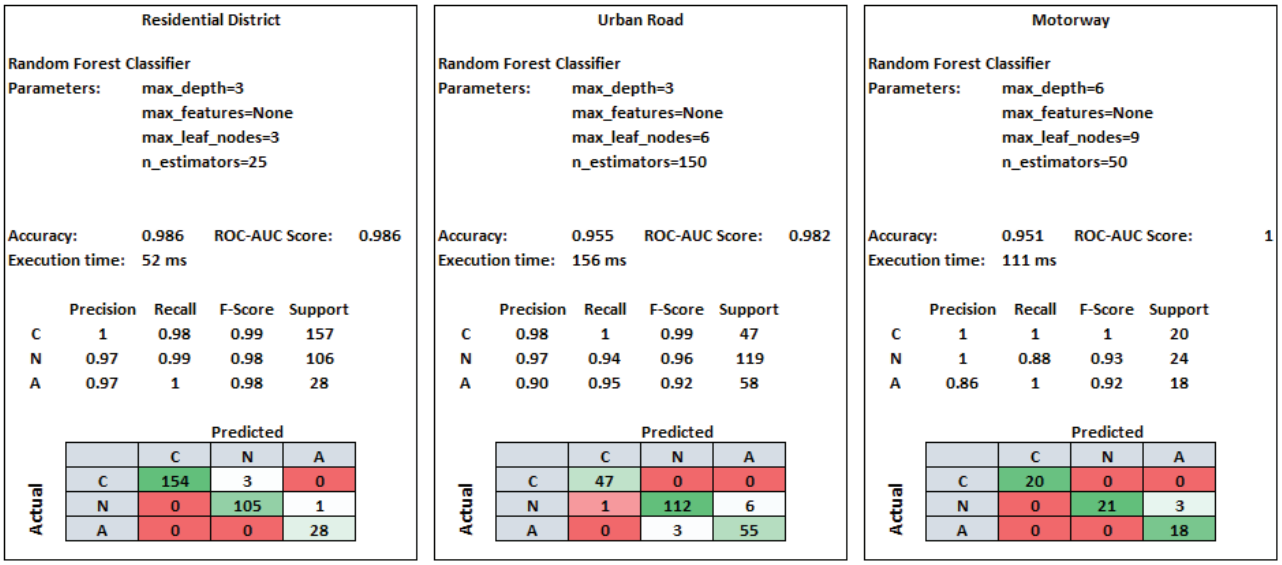

Figure 30. Results for a) residential district, b) urban road, c) motorway datasets with the random forest algorithm.

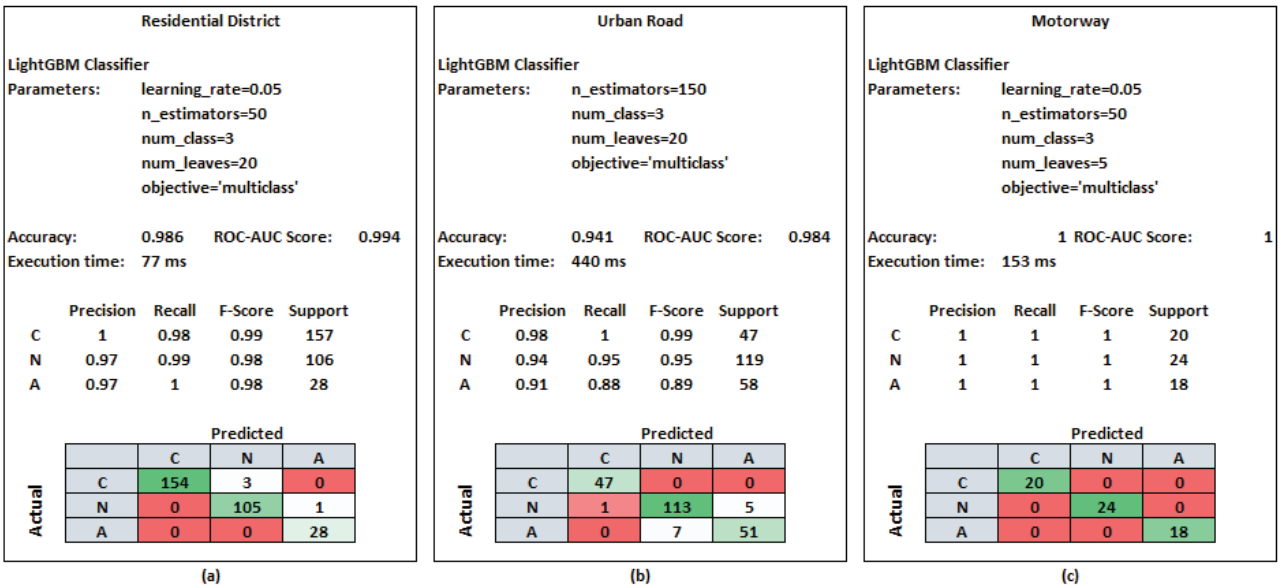

Figure 31. Results for a) residential district, b) urban road, c) motorway datasets with the LightGBM algorithm.

classifying driving data. Our findings reveal that each algorithm has its own advantages and disadvantages. For instance, k-NN, logistic regression, and naive Bayes exhibit very low processing times, although their accuracy rates may be slightly lower. On the other hand, algorithms such as SVM and artificial neural networks provide robust solutions for nonlinear problems. Decision trees offer significant benefits in terms of interpretability and visualization. Despite the disadvantages in processing time, LightGBM and random forest algorithms produce outputs based on the majority decision method and achieve higher accuracy values.

The above sections demonstrate that highly accurate solutions were obtained with all classification models. When comparing the performances of classification algorithms, we first looked at the accuracy rate. We supported this success criterion with the ROC_AUC score calculated from the confusion matrix metrics. We also added the response times to compare the resource usage of the models. Table 9 shows the results obtained from all classifiers. The performance comparison of the classification algorithms for each data set is graphically shown in Figure 32, Figure 33, and Figure 34, respectively.

When Figures 32-34 are examined, it will be seen that almost all models perform driving classification with high accuracy. Since the data of the overspeed feature can be separated more sharply in the motorway data set, greater accuracies were obtained in this data set. Similarly, in the residential district dataset, calm drivers were more easily separated from normal and aggressive drivers because they were driving at a slow speed, which positively affected the accuracies. In urban use, the traffic factor on the roads limited the speed of aggressive drivers, making it somewhat difficult to distinguish driving styles and having a negative effect on accuracy rates. Decision tree-based algorithms stand out in all data sets. We attribute this to decision

Table 9. Performances of classification algorithms for three different datasets

	Residential District			Urban Road			Motorway		
	Accuracy	ROC_AUC Score	Execution Time	Accuracy	ROC_AUC Score	Execution Time	Accuracy	ROC_AUC Score	Execution Time
k-NN	87.6 %	0.955	9.9 ms	82.9 %	0.936	207 ms	82.2 %	0.954	4 ms
Support vector machines	98.6 %	0.991	25 ms	90.1 %	0.975	17 ms	91.9 %	0.993	7 ms
Decision tree	98.2 %	0.980	52 ms	94.1 %	0.923	180 ms	98.3 %	0.989	36 ms
Multi layer perceptron	96.5 %	0.993	323 ms	91.0 %	0.975	363 ms	88.7 %	0.964	79 ms
Logistic regression	97.2 %	0.992	23 ms	90.5 %	0.975	21 ms	88.7 %	0.976	9 ms
Naive bayes	86.5 %	0.969	3 ms	78.9 %	0.931	3 ms	91.9 %	0.985	3 ms
Random forest	98.6 %	0.986	52 ms	95.5 %	0.982	156 ms	95.1 %	1	111 ms
LightGBM	98.6 %	0.994	77 ms	94.1 %	0.984	440 ms	100 %	1	153 ms

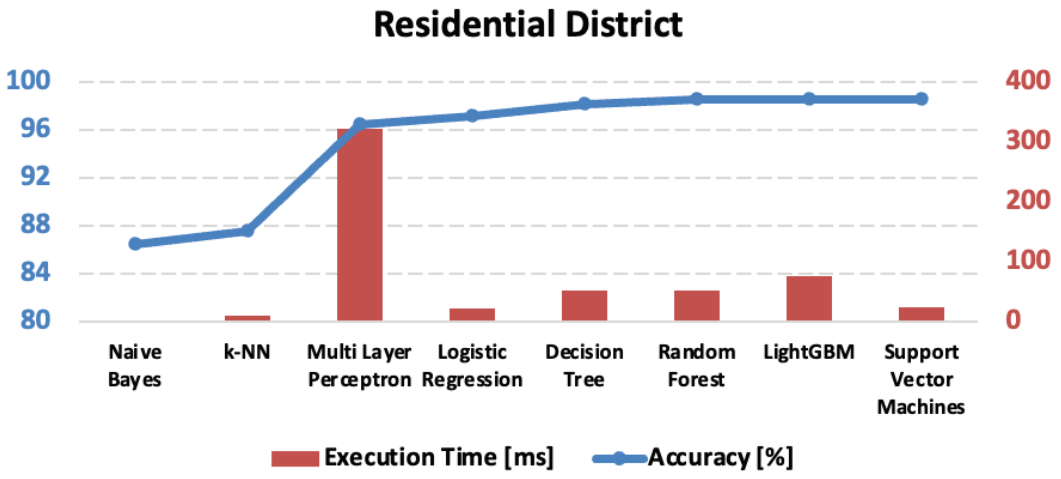

Figure 32. Comparison of classification algorithms for residential area dataset according to processing time and general accuracy criteria.

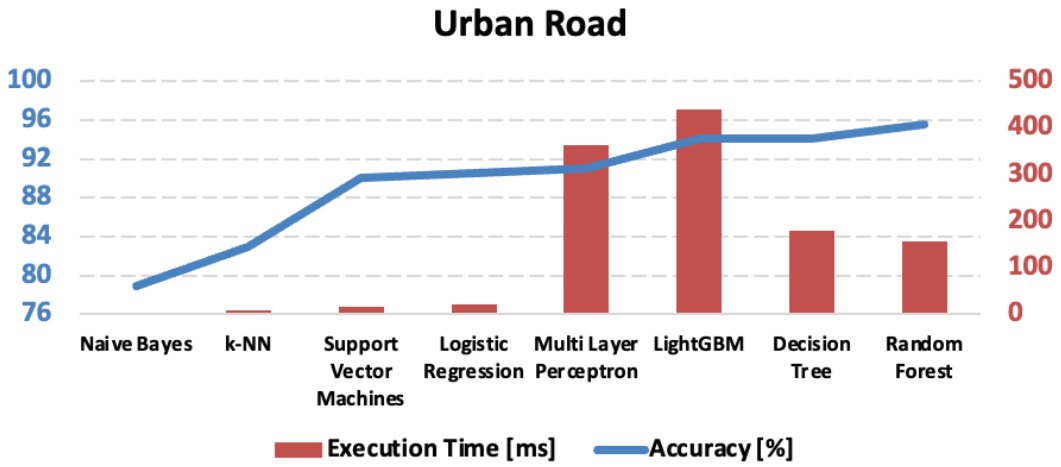

Figure 33. Comparison of classification algorithms for urban road dataset according to processing time and general accuracy criteria.

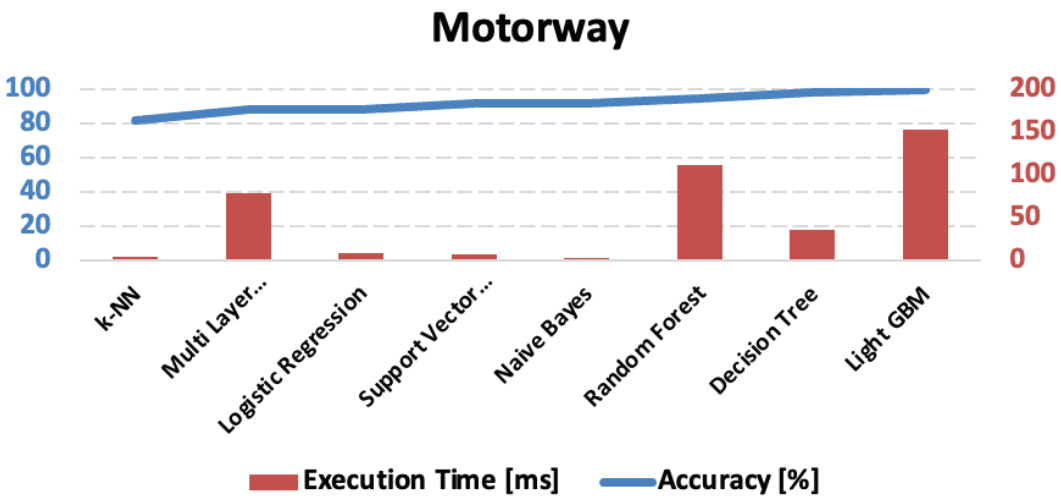

Figure 34. Comparison of classification algorithms for motorway dataset according to processing time and general accuracy criteria.

trees eliminating unimportant features and making faster and more accurate classifications according to the features they consider important. Due to the nature of our problem, decision tree-based models used prominent features such as maximum and average values of speed, sudden accelerations, and excessive speed ratio more effectively and made more accurate predictions. Our study also revealed the advantage of ensemble learning methods in coping with the overfitting problem. Although the processing times are reasonable for all models, it is necessary to consider this in cases where the data set is very large.

Comparison with Other Studies

We compare the methods proposed in this section with the studies mentioned in the literature summary of the introduction section, based on the criteria of data, algorithm, output information, and success rates. Information from 21 different studies is presented together in Table 10.

A careful examination of Table 10 reveals that driver or driving classification can be done successfully. New techniques have increased classification success for such problems. In particular, ensemble learning techniques are leading in this regard. Therefore, in our study, in addition to many traditionally used classification algorithms, we also used models based on ensemble learning such as random forest and LightGBM. As can be clearly seen in Table 10, our proposed methodology is highly effective in the following areas when compared to previous studies:

i. The models we developed were trained and tested with real road data, not simulation data.

Table 10. Comparison of studies on driving or driver behavior classification

Purpose of the study	Driving Data	Method	Features/Inputs	Outputs	Achievement	Authors
Classifying the driver's driving style	Speed profile of 11 standard drive cycles developed by Sierra Research	Rule-based classifier	Jerk profile	Calm, normal, aggressive	An accurate classification is shown depending on the fuel consumption.	Murphey et al.
Online driving style recognition	23 km track data simulated with IPG-Driver	Fuzzy logic	Longitudinal/ lateral acceleration, deceleration, speed, time gap	Sporty, normal, comfortable	They made driving classification with an average accuracy of 68%.	Dörr et al.
Classify drivers by their behavior	Data collected from a 3D virtual driving simulator	Fuzzy logic	Age, acceleration, braking, speed	Very passive, passive, normal, aggressive, dangerous	They tried to predict different undesirable situations such as traffic accidents, road congestion.	Fernandez and Ito
Diagnosing a driver profile to reduce the number of dangerous maneuvers	Driving data from two drivers on similar and different routes	k-means clustering, SVM	Acceleration, braking, and turning events	Differentiating between drivers	They were able to distinguish between 60-65% of similar drivers.	Van Ly et al.
Driver classification	1. Information on the driving of 3 different vehicles and 14 drivers on a predefined route 2. Driving data collected over 4 days from three couples sharing the same car	Support vector machine	Gyroscope, torque, GPS, accelerometer, pedal position, throttle positions, engine RPM	Information on which driver is driving which vehicle	Average classification accuracies were 75.83%, 85.83% and 86.67% using phone sensors only, car sensors only, and all sensors, respectively.	Zhang et al.
Driver classification	Data of 304 routes where 11 people drive with different vehicles on the same road	Neural networks	Velocity, acceleration, angle, GPS signals	1. Mishandling steer and pedals, speeding and getting out of the lane and road 2. Aggressive and moderate	The created model classified 97% of the drivers as good and 3% as bad drivers. The model was able to identify the driver with an accuracy of 97% to 98%.	Quintero et al.
Detection of driver behaviors to reduce the risk of traffic accidents	Data collected from the driving of three drivers in the same vehicle	Deep convolutional neural networks	Acceleration, gravity, throttle, speed, and engine RPM	Normal, aggressive, distracted, drowsy, and drunk driving	They reported that the optimal test results were 99.76% accurate.	Shahverdy et al.

Table 10. Comparison of studies on driving or driver behavior classification (continued)

Purpose of the study	Driving Data	Method	Features/Inputs	Outputs	Achievement	Authors
Driving classification	110 route driving information provided by the urban public transport system	k-NN	3-axis accelerometer signal statistical features	Normal and aggressive	In the experiments conducted in the same season, in traffic conditions and on the same route, 100% accuracy was achieved.	Vaitkus et al.
Driver classification for optimization of energy usage	Data of ten different drivers' rides on a predetermined route	Probability density function	Features extracted from the vehicle's powertrain signals	Aggressive, moderate and conservative	Based on the power demands placed on the vehicle powertrains, a 77% accuracy rate was recorded.	Kedar-Dongarkar and Das
Driving classification	58 test trips from different drivers	Neural networks	Attributes such as mean, max, min, variance extracted from longitudinal and lateral events	Very sporty, sporty, normal, defensive, very defensive	In the most relevant subset of 42 trips, a classification accuracy of 81% was achieved without misclassification.	Brombacher et al.
Driving behavior classification	UAH-DriveSet	Stacked long-short term memory recurrent neural networks	Accelerations along the x, y, z axes, Roll, pitch and yaw angles, Vehicle speed, Distance to vehicle ahead, Number of vehicles detected	Normal, aggressive and drowsiness	According to the confusion matrix data in classification skill, they reached an F1 score of 0.91.	Saleh et al.
Driving style classification	Data from a driving simulator with 20 licensed drivers	Semi supervised support vector machine	Vehicle speed, throttle opening	Aggressive, vague, normal	It has been reported that 86.6% was reached as the best classification rate with semi-supervised svm.	Wang et al.
Classification of driver fatigue	Data from a driving simulator with 63 adult participants	1. ANOVA 2. Random Forest 3. Neural Network 4. k-NN	Electrocardiogram, electrodermal activity signals, sleepiness and emotional state statement	Sleep deprivation, driving environment, sleepiness	Sleep deprivation, driving environment, and sleepiness predicted with an accuracy of 99%, 85%, and 73% respectively.	Meteier et al.
The classification of short and long-term driving behavior	13,792 real driving data in Ann Arbor, Michigan	1.Logistic regression 2. Support vector machine 3. Decision tree 4. Discriminant analysis	GPS data, brake status, headlight status, vehicle speed, acceleration, steering position, throttle position, yaw rate	Jerk, leading headway, yaw rate	It separated the classes with a confidence level of up to 99%.	Seraj

Table 10. Comparison of studies on driving or driver behavior classification (continued)

Purpose of the study	Driving Data	Method	Features/Inputs	Outputs	Achievement	Authors
Driver identification	KIA motors corporation dataset	k-NN	The 51 features extracted from vehicle, road and environmental parameters were reduced to 15.	Drivers	Accuracy for two drivers was 99.9%, and accuracy for 10 drivers was 76.36%.	Khan et al.
Classification of driving styles	Next Generation Simulation dataset	1. k-means 2. Spectral clustering	Features such as speed, acceleration, and yaw angle	Calm and aggressive drives	The K-means method gave slightly better results than spectral clustering.	Benterki et al.
Driving skill classification in curve driving scenes	Data from a driving simulator with 16 adult participants	1. k-NN 2. Support vector machine	Lateral and longitudinal controls	High-skilled driving, low/average-skilled driving	With SVM, classification accuracy was 95.7% in the full curve driving scene and 89% in the case of segmented curves.	Chandrasiri et al.
Abnormal driving behaviors detection	6-month driving data collected from real driving environments	1. Support vector machine 2. Neural network	16 key features were extracted from the driving behavior models.	Weaving, swerving, sideslipping, fast u-turn, turning with a wide radius, sudden braking	Average accuracy was 95.36 percent with the SVM classifier model and 96.88 percent with the NN classifier model.	Yu et al.
Driver and path detection	A real dataset of 292 observations	Neural networks	16 features related to road, speed and fuel consumption have been extracted.	City street, highway, dirt road	The best values obtained were 97% for determining the gender of the driver, 94% for determining the road type, and 97% for determining the driver familiarity.	Bernardi et al.
Aggressive driving detection	UAH-DriveSet	Long short term memory fully convolutional neural network	Speed, acceleration, roll, pitch, yaw, position, angle, road with, distance to ahead vehicle in current lane, time of impact to ahead vehicle	Normal driving, aggressive driving	They achieved an F score of 95.88% for a window length of 5 minutes.	Moukafih et al.
Object detection for road safety	4966 images	1. Faster R-CNN 2. RetinaNet, 3. YOLOv5	Images	Eye closure, yawning, smoking, mobile phone usage, and seatbelt compliance	As the best result, YOLOv5 recorded 125 FPS, 42 MB compact model size and 93.6% success.	Zia et al.

Table 10. Comparison of studies on driving or driver behavior classification (continued)

Purpose of the study	Driving Data	Method	Features/Inputs	Outputs	Achievement	Authors
Distracted driving detection system	Images and video data	CNN	Images and video frames of drivers	Using a mobile phone, eating, and looking away from the road	They detected driver distraction with 95% success.	Anitha et al.
Driving style classification	**38 thousand km driving data collected in 610 hours with nine professional drivers**	**1. k-NN 2. SVM 3. Decision tree 4. NN 5. Log. Regression 6. Naive Bayes 7. Random Forest 8. LightGBM**	**Max/mean/st deviation of speed, skewness, kurtosis, max/min point ratio, acc/dec ratio, pos/neg slop avg, overspeed, stop an go ratio, mean band ratio**	**Aggressive, normal, calm driving.**	**The best results for three different roads: 1. 91.9% (motorway) 2. 87.6% (residential district) 3. 96.5% (residential district) 4. 97.2% (residential district) 5. 98.3% (motorway) 6. 98.6% (residential district) 7. 98.6% (residential district) 8. 100 % (motorway)**	**Beskardes and Hames (this paper)**

ii. Our study has the longest range and the greatest total driving time with 9 professional drivers on different roads and with different vehicle types.
iii. It was studied with eight different classification algorithms. The success obtained with each of these algorithms is as high as the success obtained in other studies and is compatible with the literature.

CONCLUSION

Efforts to increase driving comfort and safety in road vehicles are continuously advancing. Driving style prediction systems, which are part of these efforts, can make important contributions to safety, such as detecting risky behaviors and preventing potential accidents caused by these behaviors. In vehicles integrated with driving style prediction systems, inefficient driving habits such as sudden acceleration or deceleration, and frequent stop-and-go can be detected and eliminated. More efficient driving can be achieved, resulting in lower fuel consumption and fewer harmful gas emissions. Driver fatigue can be detected in these vehicles and the driver can be warned. In this study, we estimated the driving style of drivers to contribute to the optimal utilization of these technologies. We used our meticulously compiled dataset, which reflects real driving characteristics as much as possible, rather than ready-made datasets. We worked on a driving style discrimination technique that can be used for purposes such as determining a driver's driving style from the data of their driving and offering more suitable vehicle options or adapting a vehicle to the driver's driving style. For this, we collected 38 thousand km of driving data with professional drivers. We extracted the necessary features from this data and calculated their value. After the necessary formatting and transformation processes with data mining techniques, we determined the driving style with eight classification algorithms. Using k-nearest neighbor, support vector machines, decision trees, neural networks, logistic regression, Naive-Bayes, random forest, and light gradient-boosting machine classifiers, we obtained consistent and highly accurate

results. We compared these methods extensively with each other and with similar studies in the literature. We have shown that the proposed methods are compatible with the literature, are suitable for determining driving style, and can be used for safe and enjoyable driving.

In this study, although the participants were of different ages, education levels, and genders, the number of participants was limited. Increasing the number of participants can advance the research further. Although the driving tests were conducted in many different regions, the majority were in the Mediterranean region. Evaluating more driving data from different regions could provide new insights into the differences in roughness, climate, and traffic conditions. The data evaluated to estimate the driving style was taken from a single source, namely an application on a smart device. Including data collected from devices such as in-car cameras in addition to this data source could show that information received from different sources supports each other and increases the prediction success rate.

In further studies, more detailed analyses could be performed by applying the data set not only to classification algorithms but also to time series-based algorithms. In the current situation, the estimation of the driving style is done after the drive is over. As a further stage, we can design a system that dynamically detects the changing driving style of the driver during a drive and adapts the vehicle to this current situation. For real-world applications, if the developed models are run on a tool that can be integrated into a car, a vehicle-human interaction can be established. This way, some warning, calming, or relaxing notifications can be sent to the driver from the vehicle's multimedia system. Moreover, adaptive systems can be designed to enable the vehicle to respond better to the driver's driving style to increase driving efficiency and comfort.

ACKNOWLEDGEMENTS

This work is part of Ahmet Beşkardeş's doctoral thesis.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence was not used in the preparation of the article.

REFERENCES

[1] Rolim C, Baptista P, Duarte G, Farias T, Pereira J. Real-Time Feedback Impacts on Eco-Driving Behavior and Influential Variables in Fuel Consumption in a Lisbon Urban Bus Operator. IEEE Trans Intell Transp Syst 2017;18:3061–3071. [CrossRef]

[2] Dörr D, Grabengiesser D, Gauterin F. Online driving style recognition using fuzzy logic. 2014 17th IEEE Int Conf Intell Transp Syst ITSC 2014 2014:1021–1026. [CrossRef]

[3] Zia H, Hassan I ul, Khurram M, Harris N, Shah F, Imran N. Advancing Road Safety: A Comprehensive Evaluation of Object Detection Models for Commercial Driver Monitoring Systems. Futur Transp 2025;5. [CrossRef]

[4] Anitha M, Namachivayam S, Christus ATA, Sb V. Real-Time Distracted Driver Detection and Alert System for Enhanced Road Safety 2024;2:196–213.

[5] Sim G, Ahn S, Park I, Youn J, Yoo S, Min K. Automatic longitudinal regenerative control of evs based on a driver characteristics-oriented deceleration model. World Electr Veh J 2019;10. [CrossRef]

[6] D'Agostino C, Saidi A, Scouarnec G, Chen L. Learning-Based Driving Events Recognition and Its Application to Digital Roads. IEEE Trans Intell Transp Syst 2015;16:2155–2166. [CrossRef]

[7] Marina Martinez C, Heucke M, Wang F-Y, Gao B, Cao D. Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey. IEEE Trans Intell Transp Syst 2018;19:666–676. [CrossRef]

[8] Meiring GAM, Myburgh HC. A Review of intelligent driving style analysis systems and related artificial intelligence algorithms. Sensors 2015;15:30653–30682. [CrossRef]

[9] Xu L, Hu J, Jiang H, Meng W. Establishing style-oriented driver models by imitating human driving behaviors. IEEE Trans Intell Transp Syst 2015;16:2522–2530. [CrossRef]

[10] Imamura T, Yamashita H, Zhang Z, Othman MDR bin, Miyake T. A study of classification for driver conditions using driving behaviors. 2008 IEEE Int Conf Syst Man Cybern 2008. p. 1506–1511. [CrossRef]

[11] Murphey YL, Milton R, Kiliaris L. Driver's style classification using jerk analysis. 2009 IEEE Work Comput Intell Veh Veh Syst 2009, p. 23–28. [CrossRef]
[12] Gilman E, Keskinarkaus A, Tamminen S, Pirttikangas S, Röning J, Riekki J. Personalised assistance for fuel-efficient driving. Transp Res Part C Emerg Technol 2015;58:681–705. [CrossRef]
[13] Fernandez S, Ito T. Driver Classification for Intelligent Transportation Systems using Fuzzy Logic 2016:1212–1216. [CrossRef]
[14] Van Ly M, Martin S, Trivedi MM. Driver classification and driving style recognition using inertial sensors. 2013 IEEE Intell. Veh. Symp., 2013, p. 1040–1045. [CrossRef]
[15] Zhang C, Patel M, Buthpitiya S, Lyons K, Harrison B, Abowd GD. Driver Classification Based on Driving Behaviors. Proc. 21st Int Conf Intell User Interfaces. New York, NY, USA: Association for Computing Machinery; 2016, p. 80–84. [CrossRef]
[16] Quintero M CG, López JO, Cuervo Pinilla AC. Driver behavior classification model based on an intelligent driving diagnosis system. IEEE Conf Intell Transp Syst Proceedings, ITSC 2012:894–899. [CrossRef]
[17] Shahverdy M, Fathy M, Berangi R, Sabokrou M. Driver behavior detection and classification using deep convolutional neural networks. Expert Syst Appl 2020;149:113240. [CrossRef]
[18] Vaitkus V, Lengvenis P, Žylius G. Driving style classification using long-term accelerometer information. 2014 19th Int Conf Methods Model Autom Robot MMAR 2014 2014:641–644. [CrossRef]
[19] Kedar-Dongarkar G, Das M. Driver classification for optimization of energy usage in a vehicle. Proced Comput Sci 2012;8:388–393. [CrossRef]
[20] Brombacher P, Masino J, Frey M, Gauterin F. Driving event detection and driving style classification using artificial neural networks. 2017 IEEE Int Conf Ind Technol 2017. p. 997–1002. [CrossRef]
[21] Saleh K, Hossny M, Nahavandi S. Driving behavior classification based on sensor data fusion using LSTM recurrent neural networks. 2017 IEEE 20th Int Conf Intell Transp Syst 2017. p. 1–6. [CrossRef]
[22] Wang W, Xi J, Chong A, Li L. Driving Style Classification Using a Semisupervised Support Vector Machine. IEEE Trans Human-Machine Syst 2017;47:650–660. [CrossRef]
[23] Meteier Q, Favre R, Viola S, Capallera M, Angelini L, Mugellini E, et al. Classification of driver fatigue in conditionally automated driving using physiological signals and machine learning. Transp Res Interdiscip Perspect 2024;26:101148. [CrossRef]
[24] Seraj M. The Classification of Short and Long-term Driving Behavior for an Advanced Driver Assistance System by Analyzing Bidirectional Driving Features. 2023.
[25] Khan M, Ali M, Haque F, Habib M. A machine learning approach for driver identification. Indones J Electr Eng Comput Sci 2023;30:276–288. [CrossRef]
[26] Benterki A, Maaoui C, Boukhnifer M, Judalet V. Driver Style Recognition Based on Vehicle Dynamic Data. 2024 10th Int Conf Control Decis Inf Technol 2024. p. 2566–25671. [CrossRef]
[27] Chandrasiri NP, Nawa K, Ishii A. Driving skill classification in curve driving scenes using machine learning. J Mod Transp 2016;24:196–206. [CrossRef]
[28] Yu J, Chen Z, Zhu Y, Chen Y, Kong L, Li M. Fine-Grained Abnormal Driving Behaviors Detection and Identification with Smartphones. IEEE Trans Mob Comput 2017;16:2198–2212. [CrossRef]
[29] Bernardi ML, Cimitile M, Martinelli F, Mercaldo F. Driver and Path Detection through Time-Series Classification. J Adv Transp 2018;2018:1758731. [CrossRef]
[30] Moukafih Y, Hafidi H, Ghogho M. Aggressive Driving Detection Using Deep Learning-based Time Series Classification. 2019 IEEE Int Symp Innov Intell Syst Appl 2019, p. 1–5. [CrossRef]
[31] Sun R, Chen Y, Dubey A, Pugliese P. Hybrid electric buses fuel consumption prediction based on real-world driving data. Transp Res Part D Transp Environ 2021;91:102637. [CrossRef]
[32] Eckert JJ, da Silva SF, Lourenço MA de M, Corrêa FC, Silva LCA, Dedini FG. Energy management and gear shifting control for a hybridized vehicle to minimize gas emissions, energy consumption and battery aging. Energy Convers Manag 2021;240:114222. [CrossRef]
[33] Bouhsissin S, Sael N, Benabbou F. Driver Behavior Classification: A Systematic Literature Review. IEEE Access 2023;11:14128–14153. [CrossRef]
[34] Alaa, T, Abdul M, Ali S, Alubady R, Adile M, Mohd K, et al. Data Management and Decision-Making Process Using Machine Learning Approach for Enterprises. Ournal Intell Syst Internet Things 2023:75–88. [CrossRef]
[35] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res 2011;12:2825–2830.
[36] Mitchell TM, Mitchell TM. Machine learning. vol. 1. New York; McGraw-Hill; 1997.
[37] Sinclair C, Pierce L, Matzner S. An application of machine learning to network intrusion detection. Proc 15th Annu Comput Secur Appl Conf 1999. p. 371–377. [CrossRef]
[38] Sahami M, Dumais S, Heckerman D, Horvitz E. A Bayesian approach to filtering junk e-mail. Learn Text Categ Pap from 1998 Work, vol. 62, 1998. p. 98–105.
[39] Kim E, Kim W, Lee Y. Combination of multiple classifiers for the customer's purchase behavior prediction. Decis Support Syst 2003;34:167–175. [CrossRef]

[40] Kumar V, Sairam SDSS, Kumar S, Singh A, Nayak D, Sah R, et al. Prediction of Iron Ore Sinter Properties Using Statistical Technique. Trans Indian Inst Met 2017;70:1661–1670. [CrossRef]

[41] Erin K, Kutlu MÇ, Boru B. Comparison of gesture classification methods with contact and non-contact sensors for human-computer interaction. Sigma J Eng Nat Sci 2022;40:219–226. [CrossRef]

[42] Dogan D, Bogosyan S. Performance Analysis of SVM, ANN and KNN Methods for Acoustic Road-Type Classification. 2019 IEEE Int Symp Innov Intell Syst Appl, 2019, p. 1–6. [CrossRef]

[43] Vapnik VN. The Nature of Statistical Learning Theory. New York: Springer; 2000. [CrossRef]

[44] Morgan JN, Sonquist JA. Problems in the Analysis of Survey Data, and a Proposal. J Am Stat Assoc 1963;58:415–434. [CrossRef]

[45] Magerman DM. Statistical Decision-Tree Models for Parsing. 33rd Annual Meeting of the Association for Computational Linguistics. Cambridge, Massachusetts, USA: Association for Computational Linguistics; 1995. p. 276–283. [CrossRef]

[46] Quinlan JR. Induction of decision trees. Mach Learn 1986;1:81–106. [CrossRef]

[47] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 1943;5:115–133. [CrossRef]

[48] Tunçkaya Y, Köklükaya E. Comparative performance evaluation of blast furnace flame temperature prediction using artificial intelligence and statistical methods. Turk J Electr Eng Comput Sci 2016;24:1163–1175. [CrossRef]

[49] Gajic D, Savic-Gajic I, Savic I, Georgieva O, Di Gennaro S. Modelling of electrical energy consumption in an electric arc furnace using artificial neural networks. Energy 2016;108:132–139. [CrossRef]

[50] Ojha VK, Abraham A, Snášel V. Metaheuristic design of feedforward neural networks: A review of two decades of research. Eng Appl Artif Intell 2017;60:97–116. [CrossRef]

[51] Bhattacharjee P, Dey V, Mandal UK. Risk assessment by failure mode and effects analysis (FMEA) using an interval number based logistic regression model. Saf Sci 2020;132:104967. [CrossRef]

[52] Zhang H. The Optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004). Florida: AAAI Press; 2004.

[53] Badik ŞT, Akar M. Machine learning classification models for the patients who have heart failure. Sigma J Eng Nat Sci 2024;42:235–244. [CrossRef]

[54] Cihan P. The machine learning approach for predicting the number of intensive care, intubated patients and death: The COVID-19 pandemic in Turkey. Sigma J Eng Nat Sci 2022;40:85–94. [CrossRef]

[55] Tian L, Feng L, Yang L, Guo Y. Stock price prediction based on LSTM and LightGBM hybrid model. J Supercomput 2022;78:11768–11793. [CrossRef]

[56] Ju Y, Sun G, Chen Q, Zhang M, Zhu H, Rehman MU. A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting. IEEE Access 2019;7:28309–28318. [CrossRef]

[57] Zhang Y, Yu W, Li Z, Raza S, Cao H. Detecting Ethereum Ponzi Schemes Based on Improved LightGBM Algorithm. IEEE Trans Comput Soc Syst 2022;9:624–637. [CrossRef]

[58] Roshni Thanka M, Bijolin Edwin E, Ebenezer V, Martin Sagayam K, Jayakeshav Reddy B, Günerhan H, et al. A hybrid approach for melanoma classification using ensemble machine learning techniques with deep transfer learning. Comput Methods Programs Biomed Update 2023;3:100103. [CrossRef]

[59] Uddin S, Khan A, Hossain ME, Moni MA. Comparing different supervised machine learning algorithms for disease prediction. BMC Med Inform Decis Mak 2019;19:281. [CrossRef]