applied sciences

MDPI

Article

Environment Classification Using Machine Learning Methods for Eco-Driving Strategies in Intelligent Vehicles

Jose del C. Julio-Rodríguez [1,2,*], Carlos A. Rojas-Ruiz [2], Alfredo Santana-Díaz [2,*], M. Rogelio Bustamante-Bello [2] and Ricardo A. Ramirez-Mendoza [2]

[1] Faculty of Mechanical Engineering, RWTH Aachen University, 52074 Aachen, Germany
[2] School of Engineering and Sciences, Tecnologico de Monterrey, Toluca 50110, Mexico; a01368743@tec.mx (C.A.R.-R.); rbustama@tec.mx (M.R.B.-B.); ricardo.ramirez@tec.mx (R.A.R.-M.)
* Correspondence: jjulio@tec.mx (J.d.C.J.-R.); asantana@tec.mx (A.S.-D.)

Abstract: This work presents the development of a classification method that can contribute to precise and increased awareness of the situational context of vehicles, for it to be used in autonomous driving applications. This work aims to obtain a method for machine-learning-based driving environment classification that does not involve computer vision but instead makes use of dynamics variables from Inertial-Measurement-Unit (IMU) sensors and instantaneous energy consumption measurements. This article includes details about the data acquisition, the electric vehicle used for the experiments, and the pre-processing methods employed. This explores the viability of a method for classifying a vehicle's driving environment. The results of such a system can potentially be used to provide precise information for path planning, energy optimization, or safety purposes. Information about the driving context could be also used to decide if the conditions are safe for autonomous driving or if human intervention is recommended or required. In this work, the feature selection process and statistical data pre-processing methods are evaluated. The pre-processed data are used to compare 13 different classification algorithms and then the best three are selected for further testing and data dimensionality reduction. Two approaches for feature selection based on feature importance and final classification scores are tested, achieving a classification mean accuracy of 93 percent with a real testing dataset that included three driving scenarios and eight different drivers. The obtained results and high classification accuracy represent a first approach for the further development of such classification systems and the potential for direct implementation into autonomous driving technology.

Keywords: electric vehicles; driving environment classification; machine learning; electromobility; energy consumption

check for updates

Citation: Julio-Rodríguez, J.d.C.; Rojas-Ruiz, C.A.; Santana-Díaz, A.; Bustamante-Bello, M.R.; Ramirez-Mendoza, R.A. Environment Classification Using Machine Learning Methods for Eco-Driving Strategies in Intelligent Vehicles. *Appl. Sci.* **2022**, *12*, 5578. https://doi.org/10.3390/app12115578

Academic Editor: Dario Richiedei

Received: 19 April 2022
Accepted: 24 May 2022
Published: 31 May 2022

1. Introduction

As transport technology evolves, new opportunities arise. Formerly, the efficiency of energy usage in street vehicles was close to being maxed out up to a point limited by the physical principles of the internal combustion engine. With the renewed interest in electric vehicles (EVs) and the mass production of them, a vast new field for improvement is to be worked on by research engineers, physicists, and designers of many areas, ranging from batteries technology, charging stations, power train mechanics, and of course vehicle energy management. This era in vehicle technology has seen the advent of two major paradigm changes, the electrification of street vehicles and the implementation of higher levels of autonomous driving, thanks to the increased availability of high computational power on board in the vehicles and cloud computing connectivity.

There is interest in increasing vehicle energy efficiency for environmental, technical, and economic reasons. Environmentally, all energy taken from the power grid contains a portion that comes from fossil fuels with the consequential undesired contamination. These proportions in the energy origin varies greatly from one country to another. The technical and economic reasons come together as car manufacturers want to improve the mileage per unit of energy of their vehicles through technological development and to offer a better option to customers. Besides, there are also governmental regulation policies in many countries aimed to directly stimulate the development of these technologies and reduce emissions.

The development of autonomous, intelligent, and connected vehicles and its integration with eco-driving technics have the potential to further increase the energy efficiency. The integration of all the aforementioned methods comprises a synergy of great interest for studying, evaluating, and implementing, taking energy usage optimization to new high possibilities.

This document explains the procedure and method employed to formulate, develop, and test an ML Classification Algorithm for an effective real-time driving environment classification. The data was collected from three hundred tests consisting of driving in an electric vehicle in a 1.5 km route inside our university campus while recording energy and vehicle dynamics data using specialized instrumentation (Dewesoft's SIRIUS Data acquisition System). The tests included eight different drivers and three different "driving styles". The collected data was then analyzed and processed to obtain composite variables that provided more information to the algorithm. Several ML technics were tested on the pre-processed dataset and evaluated based on training time, prediction time, and classification scores. The main contributions of this work include the proposal and assessment of a methodology, the identification of key features for driving scenario classification tasks, and the evaluation of results for determining if such a system can be useful for intelligent driving systems.

This paper is organized as follows: Section 2 describes the previous works, and the state of the art is presented in Section 3. Section 4 presents the methodology employed in this work and details of the experimental data acquisition including the EV used for the tests, the route, and the sensors. Section 5 presents the features used and the data pre-processing. Section 6 describes the data analysis and the initial classification with Machine learning algorithms. Section 7 presents further classification tasks while also discussing its results. Finally, the work's conclusions are drawn in Section 8 and future work is proposed in the last section.

2. Previous Works

The improvement of vehicles' energy consumption is a recurring subject of research. The motivations are extensive and so are the works that propose their own approach to achieve this objective. The development and convergence of technologies for automotive applications offer a great deal of opportunities for addressing this matter. Concepts such as automated, connected, electrified, and shared vehicles offer their own set of advantages, but by combining these concepts, the resulting synergy can greatly enhance the achievable results. Qi [1] describe these four concepts: (i) vehicle automation includes automated vehicle dynamics control such as adaptive cruise control (ACC), and automated powertrain functions such as power-split control for PHEVs; (ii) connected vehicle technology includes vehicle to vehicle V2V, vehicle to infrastructure V2I and V2X connectivity for applications such as reducing traffic congestion and optimized stop-and-go maneuvers at signalized intersections; (iii) electrification is the migration from the fossil fuel energy source in the vehicle to the use of electric powertrain architectures, including variations such as plug-in PEVs, battery electric vehicles, and Hybrid EVs (hydrogen fuel cells or extended range ICE vehicles); (iv) shared vehicle technologies provide the possibility of sharing a single car among several users, reducing the total travel distance of many vehicles, therefore reducing the net vehicle energy consumption. Simulation results by Qi, implementing the combined four concepts, reported average energy savings of 12% and 22% for ADAS and partially automated driving, respectively.

This section presents a short summary of research in electric vehicles power-efficiency improvements by means of power train control and autonomous driving.

2.1. Synergy Opportunities of New Technologies

The electrification of transport provides new opportunities for improving energy efficiency; vehicle connectivity together with the implementation of onboard intelligent system can take these improvements to higher new levels. The synergy of these systems is thoroughly explained by Asher [2].

Asher identified and defined three approaches for improving the energy efficiency of fully electric vehicles (EVs), Hybrid Electric Vehicles (HEVs), and Plug-in Hybrid Electric Vehicles (PHEVs). The first approach is the intervention of the power train system, in what he calls "Optimal Energy Management Strategy" or Optimal EMS, reducing energy losses on the physical components of the power train or reprograming and tweaking the motors controller software. The second approach is the implementation of Eco-driving strategies on the autonomous driving algorithm of the vehicle. The third approach is the combination of Optimal EMS and Eco-driving. Asher tested the three approaches through simulations on a validated PHEV model using four different driving cycles: the city-focused Urban Dynamometer Driving Schedule (UDDS), the highway-focused Highway Fuel Economy Test (HWFET), the aggressive US06 drive cycle, and the New York City Cycle (NYCC). According to the results of his work, the third approach achieved the highest energy efficiency improvements with 40% for the city drive cycle.

Connectivity between vehicles also allows the gathering of information and the evaluation of the vehicles' performance in novel ways, such as sending data in real time to laboratories that can monitor and evaluate the vehicle powertrain for research and improvement purposes [3]. V2I communication can also be used to give preferential treatment to emergency or special vehicles at intersections by calling the phase required by them [4].

2.2. Road Perception

Path, maneuver, and trajectory planning components of autonomous on-road driving (often combined as one) take vehicular dynamics, obstacles, road geometry, and traffic interactions into account [5]. Road geometry has a great impact on the energy used by a vehicle to travel it [6,7]. The slope can demand a great amount of energy during ascending but can also help the vehicle to regenerate energy during descending if a regenerative braking system is available and used correctly. But road geometry can affect vehicle energy efficiency in other situations, for example in horizontal curves. During turning maneuvers, the average driver tends to decelerate before the curve and then accelerate again after leaving it, while the optimum use of energy would imply to maintain a constant speed during the curve and planning ahead the optimum speed profile before arriving to it [8].

3. State of the Art

Automated learning allows for a great amount of information to be processed, classified, and correlated. Autonomous driving systems rely on data from several sensors being delivered at a high rate; this information is filtered, processed, and used by the autonomous driving system for real-time decision making. A high processing capacity onboard vehicles and the volume of data acquired by autonomous driving systems can be used for correlating energy consumption with driving and road data, which is valuable information for improving the energetic efficiency of the vehicle. Vehicle connectivity can greatly expand the impact of vehicles collecting data by allowing them to share this information with other vehicles and central data servers for it to be used for system improvements that can then be downloaded back to all vehicles as system enhancement updates.

The use of ML algorithms for processing vehicle data and classifying driving information has been reviewed by Elassad et al. [9]. The authors reviewed and grouped 86 works from 46 different journals and conferences according to the ML technics employed in each one (See Figure 1). The review provided an insight in the extended capabilities of ML tools, comparing the results of each work according to metrics such as accuracy, precision, and recall. The review of Elassad et al. concluded that the selection of the best ML technic in automotive data classification applications depends on the particularities of the task to perform, the conditions of the experiments conducted, and the characteristics of the available data.

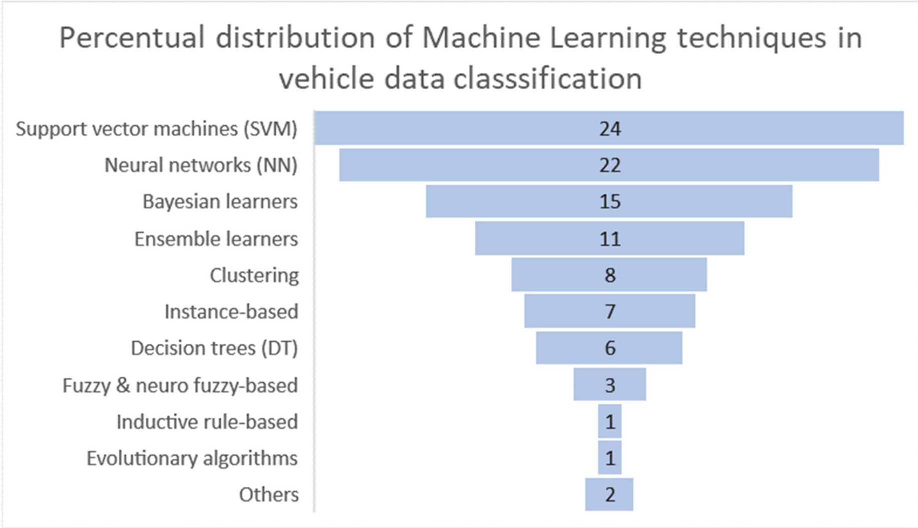

Figure 1. Distribution of ML technique used in vehicle data classification (Data by Elassad et al. [9]).

The tendency of cars innovations aims toward the realization of the holistic view of its role in the modern society, where the vehicle is not only an element to move from one point to another, but it is fully integrated in human daily experiences and its environment [10].

MPC is a powerful method for vehicle control, but its computing performance can be improved by integrating machine learning technics into its functioning [11].

ACC is a useful functionality that allows for increased comfort and as an additional safety measure during driving. As ACC corresponds to level one autonomous driving, its functionality is closely affected by human interaction in the context of the vehicle itself, beside the interactions with exterior agents such as pedestrians and drivers of other vehicles. So, the driver is a key factor that weighs into the ACC performance. Quantifying the effect of the driver is of utter importance as stated in other works regarding the research of ACC technology [12].

Driver behavior can account for up to 30% more energy consumed when comparing aggressive and moderate driving styles [13]. Additionally, the operating characteristics of the individual drivers require an ACC that best adapts to them, for a safer, more comfortable, and more efficient driving experience [12]. Eco-driving involves applying a set of strategies to reduce the vehicles total energy consumption for a given displacement. As seen in the previous section, there are many approaches for improving the energy consumption of a vehicle. Ajanović et al. [14] groups the driving behavior-related approaches for improving energy efficiency into: Eco-routing, using road slope information, traffic light assistance, platooning, and overtaking.

Real-time awareness of the driving environment is a valuable input for an online energy management optimization system in a vehicle. Planning ahead for a specific route could ideally allow a minimized energy consumption driving strategy to be prepared, but there are always changing factors along the route, such as traffic and red lights at intersections that need to be accounted for by means of probabilistic calculations or real-time adaptation [15]. Artificial Intelligence can be a suitable tool for improving the energy efficiency of vehicles in complex scenarios and ever-changing roadway and traffic conditions [16].

The "road slope" energy improving strategies normally involves knowing the road topography of the route beforehand, and performing optimization using this information, together with vehicle dynamics models [17]. Ajanović et al. also state in their work that the optimized speed trajectories are rarely used directly to provide a reference value for lowlevel controllers such as cruise control, nonetheless, that is the path in which automotive technology is advancing.

The energy consumption of an electric vehicle can be predicted with a precise model of the powertrain that includes all sub-components [18], but an individual model must be developed for each car variation. Some works on the field of energy efficiency create macro models of the whole transportation sector using artificial neural networks for long-term prediction of energy demand [19,20]. The computation of EVs energy consumption in real-time and short-range predictions have been also studied by other authors while other works study energy performance by considering particular segments of a predefined route or focus on the effect of single variables such as the vehicle weight [21,22].

Many shortcomings of traditional systems can be overcome by implementing data approaches such as deep learning methods, taking advantage of their ability to find correlations between the given inputs automatically (unsupervised learning) and share this acquired knowledge with other systems [23]. Deep learning algorithms are especially useful for image recognition and classification [24].

The implementation of deep learning in autonomous vehicle applications generally uses deep-neural-network-based controllers, together with perception modules. Sensors provide the input information to the controller algorithm and it outputs the actuation for driving the vehicle safely under predefined rules [5,25]. Other uses of deep learning for vehicle applications includes the prediction of specific driver speed profiles [26], the prediction of optimal speed profiles for a specific sector on a route, and the prediction-classification of road, traffic, and driving environments with the objective of improving the vehicle energy efficiency [27]. A summary of the reviewed state-of-the-art and the comparison of characteristics with this work is presented in Table 1.

Table 1. State-of-the-art summary.

Article	Authors	Functions	Model	Methods	Evaluation
Increasing the Fuel Economy of Connected and Autonomous Lithium-Ion Electrified Vehicles	Asher et al., 2018 [2]	Energy Management, V2V, environment Perception	Vehicle Dynamics Model, Energy Management Model	Dynamic programming and Pontryagin's Minimization Principle	SIMULATION using models to compare results of different control strategies
On the Optimal Speed Profile for Eco-Driving on Curved Roads	Ding et al., 2019 [8]	Velocity profile optimization for curved roads	Vehicle Dynamics Model, Fuel Consumption Model	Dynamic programming Optimization	Algorithm verification using co-simulation of CarSim and Matlab/Simulink
Design and Implementation of Ecological Adaptive Cruise Control for Autonomous Driving with Communication to Traffic Lights	Bae et al., 2018 [15]	V2I, EAD *, Surrounding traffic consideration	Vehicle Dynamics Model	Robust Model Predictive Control	ACC ** tested in a Hardware in the Loop setup with SPaT *** information
Vehicle Deceleration Prediction Based on Deep Neural Network at Braking Conditions	Min et al., 2020 [26]	Decelerations predictions	Deep learning	Deep neural network (RNN, LSTM, conventional neural network), K means clustering method	Vehicle velocity, relative distance between the vehicle and the traffic light, reference acceleration
Quantifying the Impact of Traffic on Electric Vehicle Efficiency	Jonas et al., 2022 [27]	Impact of traffic on Electric Vehicle efficiency	Statistical models	Regression models, ANOVA	Total energy consumption, total distance, Average consumption per mile, Mean variation in speed, Mean variation in acceleration, Mean variation in jerk
Road surface real-time detection based on Raspberry Pi and recurrent neural networks	Wang et al., 2021 [28]	Road surface detection	Recurrent Neural Network	Allan variance, Machine learning algorithms (KNN, L2 logistic regression, Decision tree, SVM cross validation), Deep learning Algorithms (LSTM, RNN)	Three axis accelerometer (x, y, z) and three axis gyroscope (x, y, z)

Table 1. *Cont.*

Article	Authors	Functions	Model	Methods	Evaluation
An IMU-based traffic and road condition monitoring system	Lei et al., 2018 [29]	Traffic and road condition monitoring system	Fast Fourier Transform	Least squares optimization for speed estimation considering sensor bias, DCM filter for attitude angle estimation	Relation between vertical acceleration and Present Serviceability Rating (PSR)
Map Matching and Lane Detection Based on Markovian Behavior, GIS, and IMU Data	Trogh et al., 2020 [30]	Map matching and lane detection	Markovian behavior	Viterbi (hidden Markov model)	Direction of road segments, maximum allowed speed per road segment, and driving behavior
Safe and Ecological Speed Profile Planning Algorithm for Autonomous Vehicles Using a Parametric Multi-objective Optimization Procedure	Orfila et al., 2019 [31]	Velocity profile optimization for predefined route	Data-driven approach	Global Optimization using Simulated Annealing	Velocity profile data comparison—Algorithm results Vs human drivers experimental data
Energy Management Strategy for a Hybrid Electric Vehicle Based on Deep Reinforcement Learning	Hu et al., 2018 [32]	Energy Management	HEV Model	Deep Reinforcement Learning	Trained control Algorithm tested in MATHLAB and ADVISOR Co-simulation
Methodology for Finding Maximum Performance and Improvement Possibility of Rule-Based Control for Parallel Type-2 Hybrid Electric Vehicles	Jeoung et al., 2019 [33]	Rule based controller, Energy Management	Parallel Type 2 HEV Model	Dynamic programming and Pontryagin's Minimization Principle	Controller algorithm evaluation in SIMULATION HEV MODEL
A Learning-Based Stochastic MPC Design for Cooperative Adaptive Cruise Control to Handle Interfering Vehicles	Kazemi et al., 2018 [34]	CACC, Surrounding traffic consideration, V2V	Data-driven approach	Artificial Neural Network	Model Predictive Controller evaluated in SIMULATED test runs from a real driving tests dataset
Cooperative Adaptive Cruise Control with Fuel Efficiency Using PMP Technique	Rasool et al., 2019 [35]	CACC, Surrounding traffic consideration	Platoon model, ICE power train Model	Pontryagin's Minimization Principle	Validation of the controller with the models in a SIMULATION
On combining Big Data and machine learning to support eco-driving behaviors	Delnevo et al., 2019 [36]	HMI, ADAS	Data-driven approach	Machine Learning	Algorithm testing in SIMULATION using real data
Real-Time Optimal Eco-Driving for Hybrid-Electric Vehicles	Zhu et al., 2019 [37]	ADAS, HEV	Data-driven approach	Dynamic programming Optimization/Artificial Neural Network	Mutual Validation of the obtained speed profiles using DPO and ANN methods
Environment classification using machine learning methods for eco-driving strategies in intelligent vehicles.	**(THIS PROPOSED WORK)**	**Driving Environment classification**	**Machine learning models**	**Machine Learning algorithms (KNN, SVM, Decision tree)**	**Linear velocity, three axis acceleration, energy consumption, jerk, roll, pitch**

* EAD: Eco Approach and Departure, ** ACC: Adaptative Cruise Control, *** SPaT: Signal Phase and Timing, V2I: Vehicle to Interface communication, EMS: Energy Management System, CACC: Cooperative Adaptative Cruise Control, ANN: Artificial Neural Networks, DP: Dynamic programing optimization HEV: Hybrid Electric Vehicle, ADAS: Advanced Driving Assistance System, SoC: State of charge.

The works reviewed in the State-of-the-Art section cover a wide range of applications of ML in automotive technology, mainly in the creation of models to predict and act upon vehicle states or, in other cases, to classify the driver according to pre-defined categories of driving styles. These works approach the environment assessment as a computer vision problem or as a historic modeling of roads based on pre-registered information. The real-time assessment and classification into categories are the key points for the new approach proposed in this document. This work proposes a procedure for classifying driving scenarios based on vehicle dynamics and energy sensors data, describing the suitable pre-processing for the raw data and the recommended input variables that should be included to get the maximum amount of information from the data, while minimizing the processing load and processing time in the classification system. The results of this work can be used to further research the implementation of subsystems that may help autonomous driving systems or a human driver (through ADAS) to select the best driving strategy in real time, for energy-optimization, safety, and/or general performance improvement purposes.

4. Methodology

The methodology for this work develops from the idea that intelligent vehicle's awareness of the driving environment can help to improve safety and energy efficiency of the vehicle, among other possible applications. After the formulation of the idea, the objective was to use an inertial measurement unit (IMU) and energy sensors for registering data that could be processed and adapted so that an automated system could use it to achieve the ability to accurately classify its instantaneous driving environment. The classification performed by the automated system can be later used by other systems for energy optimization, safety recommendations, or adjusting the driving strategy to any specific goals.

The requirement of the system to be able to learn from data, clearly suggested the need for developing a classification system based on machine learning tools. So, the next step was to review the available ML technics and to select the most suitable for testing on the actual datasets.

The data collection consisted of equipping an electric vehicle with sensors and performing real route tests on a predefined circuit. The data collected was then exported and pre-processed as explained in Section 5. The pre-processed datasets were then used to train and evaluate the ML algorithms as explained in Sections 6 and 7. Finally, the results analysis allowed us to draw the work's conclusions, its possible application, and future work. A summary of the methodology is presented in Figure 2.

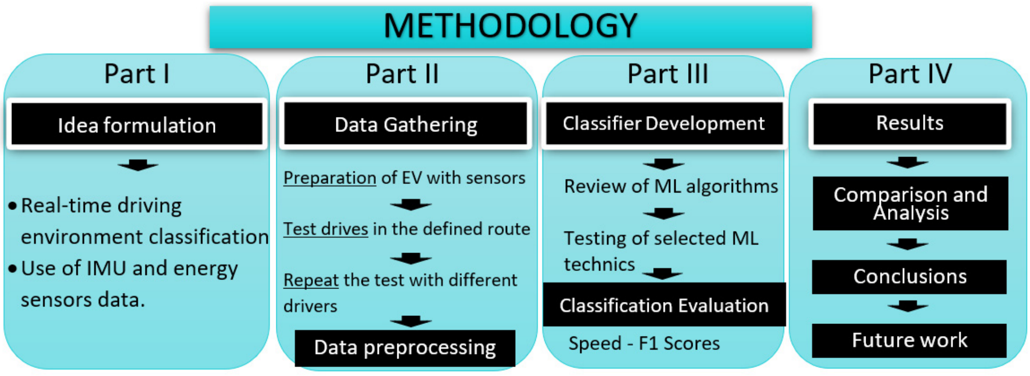

Figure 2. Design methodology of the EDS.

4.1. Work Process Overview

A general overview of this work is presented in Figure 3, including the Data-acquisition phase and the design process of the ML classifier. Furthermore, the context in which such a system could be used is presented as future work.

4.2. Experimentation and Test Route

The experimentation consisted of real driving within the Tecnológico de Monterrey University Campus in the city of Toluca, Mexico. The defined route was selected so that it included various differentiable scenarios. For this series of experiments, the selected route had three different zones that were defined as class elements for the classification task.

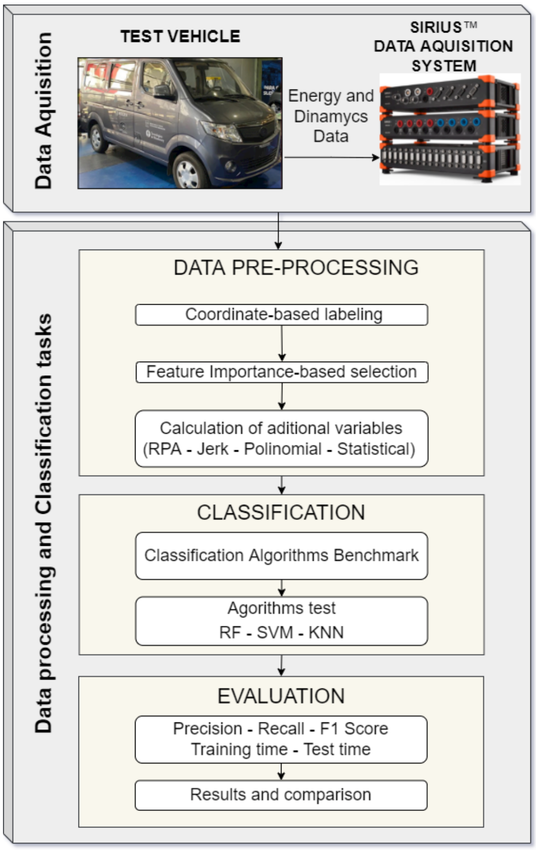

Figure 3. Experimentation and algorithm design process (provisional).

The acquisition of data was performed by using a Dewesoft's SIRIUS data acquisition system, which included the DS-IMU/GYRO modules and sensors. The test circuit was divided in three zones: "Transit", "Flatland", and "Cobblestone". These zones are shown in Figure 4 and are described as follows:

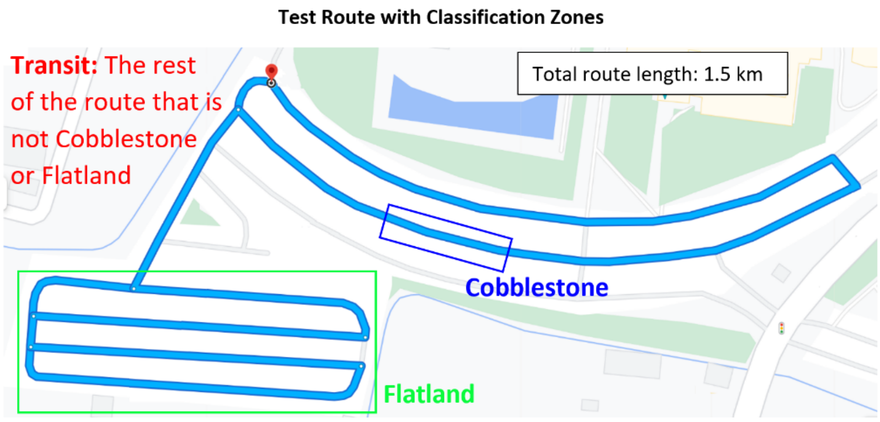

Figure 4. Test route with defined zones.

Transit: This is the default classification of the parts of the route that do not belong to the other two classes. This zone is characterized for having a top speed restriction of 10 km for pedestrian safety; also, there are several crossings and intersections, and it is expected that pedestrians or other vehicles may roam, so the driver should always be careful and especially attentive.

Flatland: This zone is a 150 m by 50 m open-space parking lot that was totally empty while the testing was being conducted. It is considered flat because there is no significative difference in the height of the area within the Flatland zone and there are also no obstacles.

After measuring the height on several points inside the area, the maximum slope registered was 0.007. The route within this zone includes a series of close radius and successive turning maneuvers.

Cobblestone: This is a small zone of about 70 m long, located inside the "Transit" zone; this means that both zones share some characteristics. The difference is the terrain roughness of this area due to it not being asphalted but rather covered with cobblestone. Figure 5 presents a photography of this zone.

Figure 5. Roughness detail of "Cobblestone" Zone.

In the transit and cobblestone zones, it is recommended that the vehicle speed is moderated, but it is possible to observe a range of speeds in the gathered data. In the Cobblestone zone, it is expected that the IMU sensors register high values of vibration in all axes, but mostly on the z axis, and a possible decrease in longitudinal velocity as an effort of the driver to reduce the effects of the road roughness on the passenger comfort.

In the Flatland zone, the closed turning maneuvers will induce lateral acceleration and yaw-rates that should be characteristic of the zone, both in magnitude and frequency. In addition, the roll angle (rotation around the vehicle's x axis) is expected to have a distinctive behavior.

4.3. Test Vehicle

The vehicle used for this work is a custom Battery Electric Vehicle (BEV) available at the Research Center for Automotive Mechatronics (CIMA) (Figure 6) at the Tecnológico de Monterrey University in Toluca, México. This test vehicle is a FAW-G60 that features an electric powertrain. The powertrain consists of a three-phase 72 volts induction motor, a 5 velocities gearbox, and a mechanical differential. Details of the motor and the battery are shown in Table 2. Furthermore, the powertrain's architecture is displayed in Figure 7.

Figure 6. Test car for data gathering.

Table 2. Specifications of the test EV's motor and battery.

Motor	
Power	15 HP
Nominal Voltage	76 V
Nominal current	115 A
Peak current	132 A
Max speed	3000 RPM
Battery	
Total voltage	96 V
Number of cells	32
Total Energy Capacity	28,800 Wh
Discharge rating 2C	0.5
Discharge current	150 A

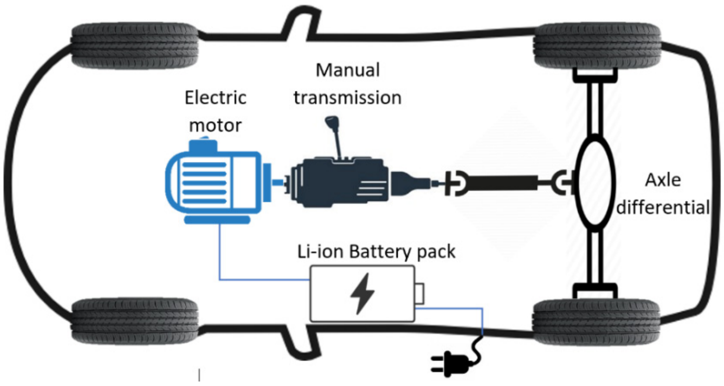

Figure 7. Test EV powertrain architecture.

The test vehicle was subjected to an evaluation protocol at CIMA facilities to obtain all the relevant vehicle characteristics, including aerodynamic coefficients, acceleration capacity, output of mechanical power, and torque. The summary of these results is presented in Table 3.

Table 3. Test vehicle physical and performance characteristics.

Vehicle brute mass	1370 kg
Wheels' dynamic radius	0.25019 m
Rolling resistance coefficient	0.014984
Aerodynamic drag coefficient	0.436
Vehicle frontal Area	2.15 m^2
Gearbox efficiency	0.95
Axle differential efficiency	0.95
Top Speed	100 km/h

After obtaining the critical characteristics of the vehicle dynamics, the next step was to perform the tests on the pre-defined route as described in the previous section and to register the driving data that includes both vehicle dynamics and energy consumption information.

5. Features and Data Preparation

5.1. Data Pre-Processing

The collected data was then analyzed and processed for it to be later used in training and testing datasets for machine-learning algorithms. The three zones to be classified were first labeled by using GNSS coordinates associated to each data sample. Then, the GNSS information was removed before training the classification algorithm so that it relied on the vehicle's dynamics and energy consumption information as the only available features. This information can be pre-processed to enhance the ML algorithm performance. This pre-processing includes the selection of the best input features and the combination of them, followed by statistical values calculation.

5.2. Variable's Information and Features Analysis

The gathered information includes vehicle dynamics data such as linear acceleration on x, y, and z axis, linear speeds on x, y, and z axis. Moreover, angular velocity and angular acceleration around the x, y, and z axis are measured.

The measurements of lateral velocity and acceleration can provide information of movements that are being performed on the y axis, such as lane changing and other characteristic maneuvers of certain environments, e.g., low-speed collision avoidance and continuous lane changing, both characteristics of high traffic urban environments. Another variable related to the change of the vehicle direction is the angular displacement around the z axis, also known as *heading*. The *heading* provides information on the vehicle's turning maneuvers in angular magnitudes. Differentiation of the *heading* variable in time produces a well-known magnitude, the yaw rate. The yaw rate provides information on the vehicle-rotation velocity around its z axis, and it is a valuable factor in vehicle stability assessment. The *heading* was not directly used as a feature in this work due to it acting as a "compass" that precisely gives away the vehicles' orientation relative to that at the starting position and can be considered data leakage.

The rotation of the vehicle around its x axis is referred to as the *rolling* angle. The rolling angle of the vehicle is affected by vehicle characteristics and by the vehicle's dynamics. The suspension subsystem of the vehicle can influence the *rolling* ranges during turning maneuvers and, in some cases, the effect of the wind in the laterals of the vehicle can induce rolling if the suspension is not rigid enough. After performing a characterization of the rolling behavior of a vehicle, it is possible to use this variable to obtain useful information of the driving environment.

Together with the *Heading* and *Rolling*, there is the *Pitch* angle, which represents the rotation of the vehicle around its lateral axis (y axis). This measurement is useful to measure the vehicle's inclination and can be used to monitor the effect of acceleration and braking maneuvers and as an indicator of the road grade.

The *slip angle* is a variable that corelates the direction of the movement of a vehicle with the direction of its longitudinal axis. When the vehicle is moving on a straight line, the slip angle is zero, and its absolute value increases as the vehicle takes a curve. The slip angle is greater for higher vehicle speeds and for a smaller instantaneous turning radius.

5.2.1. Driver Effect in the Driving Environment Classification

The classifier should be robust enough so that accurate classification can still be possible for different drivers. Although different driving styles may affect the dynamics parameters of the vehicle such as longitudinal speed, lateral jerk, and braking frequency and intensity, the system must assess these differences and correctly classify the current driving environment based on the combined behavior of the available features. To include the effect of different drivers as a noise input to the system, the tests carried out for this work included eight different drivers.

5.2.2. Electric Power Variables

Energy consumption profiles are of a great interest for assessing the vehicle's battery range and energy efficiency. Instantaneous energy data also provides valuable information of many aspects of the driving, including the driving strategy. In addition, some particularities of the current driving environment can be inferred from this data. The electrical current between the battery and the electric motor is directly related with the mechanical energy required to accelerate/decelerate the vehicle. When complemented with additional information from other sensors, it is possible to create a comprehensive model of the instantaneous situation of the vehicle and use it to optimize energy use through driving strategy adjustment.

In this work, the electric energy was measured at the vehicle's motor controller, which manages the power delivery and regeneration between the motor and the battery pack. The variables measured were current and voltage, which are directly related with the electric motor torque and speed. An instantaneous high current demand can cause momentary battery voltage drops. This information can help in characterizing both the electric power demand profiles and the battery response to them.

5.3. Data Processing and Environment Parameters

The raw data comes from samples taken by Dewesoft's real-time inertial measurement unit (IMU) and subsequently exported at a frequency of 100 (Hz) to be processed through Python. The signals that were extracted for processing were: linear speed in (m/s), linear acceleration (m/s^2), lateral acceleration (m/s^2), vertical acceleration (m/s^2), time (s), distance (m), electric power (w), roll and pitch.

An example of linear acceleration and linear velocity data over time is shown in Figure 8.

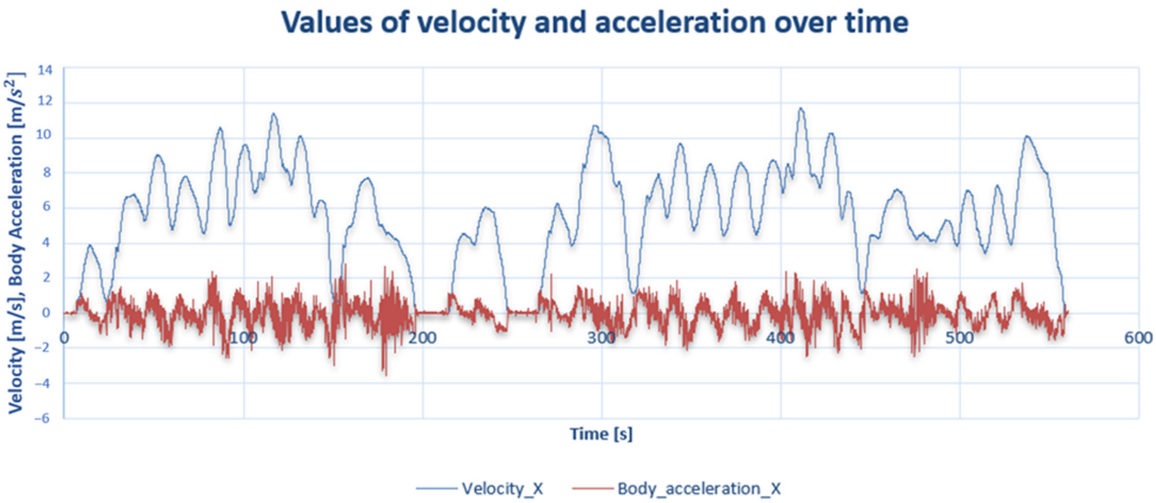

Figure 8. Values of velocity and acceleration over time.

To synthesize these signals into values that serve as test characteristics, the mean and standard deviation has been calculated. Another compounded variable calculated is the relative positive acceleration (RPA) that is defined as a fundamental measure for the characterization of routes [38]. RPA is a parameter introduced in 1997 by C. Van de Weijer [39] for characterizing vehicles in emission testing cycles of Diesel Internal Combustion Engines.

In this study, RPA is defined as the sum for a given time delta of the product of the instantaneous speed times the instantaneous positive instant acceleration divided by the traveled distance during the given time interval. This definition is mathematically defined in Equation (1).

$$\mathrm{RPA} = \frac{1}{\mathrm{X}} \int_{0}^{\mathrm{T}} \mathrm{v}(\mathrm{t}) * a^{+}(\mathrm{t}) \cdot \mathrm{dt} \quad (1)$$

where v(t) is the instantaneous velocity at time t, $a^{+}(\mathrm{t})$ is the instantaneous acceleration of positive magnitude, X is the total distance traveled, and T is the total duration. RPA can also be interpreted as a measurement of the total energy required for the instantaneous positive longitudinal accelerations performed by the powertrain while traveling a given distance.

The jerk is defined as the coefficient of variation between the standard deviation and the mean of the derivative of the linear acceleration in time [40]. The calculation method for the jerk is presented in Equation (2).

$$\gamma = \frac{SD_J}{\overline{J}} \quad (2)$$

where SD_J is the standard deviation of the jerk, and $\overline{J}$ is the mean value.

6. Data Analysis

For the analysis and construction of the initial dataset that relates all the tests and with which the machine learning algorithms were trained, 21 characteristics were considered. The complete list can be seen in Table A2 in the Appendix A.

For the classification of the road, three previously defined labels were used, which are "Transit", "Cobblestone", and "Flatland", which correspond to the distinguishable zones of a route inside the Toluca campus of the Tecnológico de Monterrey. Eight different drivers aged between 22 and 30 years participated in these tests. The total number of samples collected from the three classes was 476, which consisted of complete travels around the defined test route.

For the creation of the data table with which the machine learning algorithms were later trained, two main approaches were taken. The first method was to process the information of each complete class gathering the global characteristics of each analyzed sector, thus obtaining a dataset of 476 samples. The second approach consisted of partitioning the original data by using two-second windows and then calculating the statistical values for these subsamples, which contained 200 entries each (100 (Hz) data recording). The total of the samples processed after the partition of each of the tests was 34,137.

As a method of analyzing the information contained in the dataset, different artificial intelligence algorithms such as the k-nearest-neighbors (KNN), support vectors machine (SVM), and the decision tree were tested, the latter being mainly used as a method for feature selection and reduction of the dataset's dimensionality. As mentioned in the previous section, a study was initially carried out in which the samples without partitioning were considered; that is, statistical values of the data were recorded for each zone of the complete test route separately. After that first approach, the data set was statistically processed again, this time in 2-s intervals, which contained 200 subsamples each.

6.1. Analysis of Samples from Pure Datasets

As stated in the previous section, the dataset that contains the recorded data per travel, is a matrix of 476 rows by 25 columns. The analysis of this dataset starts by exploring the behavior of the K-nearest-neighbors algorithm (KNN) against all dimensions. Table 3 presents the evaluation metrics of the KNN algorithm; the accuracy in the test and in the training is 80% in both cases. The Precision, Recall, and F1-score are also shown; these values are a direct measurement of the algorithm's performance.

The same training and testing procedures were performed for a second-degree polynomial kernel support vector machine, the results were much better than those of the KNN algorithm. The results of this preliminary analysis are also presented in Table 4.

Table 4. KNN performance metrics.

	KNN F1-Score	SVM F1-Score	Support Samples
Flatland	0.8	0.99	45
Cobblestone	0.7	1.0	36
Transit	0.88	0.99	38
Accuracy	0.8	0.99	119
Macro avg.	0.8	0.99	119

However, seeking to improve these metrics and the general processing performance, a dimensional reduction is proposed for the dataset used with the ML algorithms. This reduction of the dimensions was carried out by exploring the importance that each one of the characteristics has when a decision tree is trained. This dimensional reduction also served as a method to avoid data leakage that leads to overfitting of the decision tree and to increase training and prediction time for different algorithms, without significant improvement in the tests results scores.

This process consisted of obtaining the contribution that each characteristic made to the decision tree at the time of being trained. Then, the following step was training and evaluating KNN and SVM algorithms with metrics against the different data tables obtained after eliminating the dimensions that led to overfitting or poor performance of the classification algorithms.

Finally, after different tests and evaluations, it was found that the best performance of the classification task occurs in front of three-dimensions that are the standard deviation of the pitch (Pitchsd), the standard deviation of the Roll (Rollsd), and the average lateral or Y-axis acceleration (Aymn).

Figure 9 presents the relationship that exists between each of these dimensions and how the data from the different samples are located within these point clouds. It can be seen that there are regions in which only the values of the same class are located and, thus, a good classification with these features can be achieved.

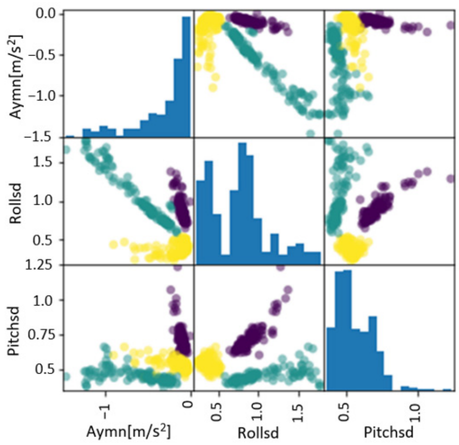

Figure 9. Relationship of the characteristics obtained after the dimensional reduction.

After reducing the number of features in the dataset, the next step was to train and evaluate the nearest neighbor algorithms and the SVM again. This time, the result for the test accuracy of the KNN algorithm was 99.2%.

Although the reduction of the characteristics improved the performance of the KNN going from an accuracy of 80% to one of 99.2%, the same did not happen with the SVM algorithm with the second-degree polynomial kernel function; it remained the same.

6.2. Data Segmentation into Subsamples

In the previous section, results of the ML algorithm's training were obtained by using the dataset with statistical values of the samples analyzed in their entirety. In this section, the data is sub-divided by creating and analyzing partitions of 2-s durations.

The future application of this work is achieving that the driving environment can be accurately classified in almost real-time; the information generated is relevant for automated driving systems only if it is available and updated during real-time decision making, along with other information that is being processed from the vehicle's state.

With this premise, the data from the original dataset was divided using discrete time-windows that contained information from 2 s of sampling and was subsequently processed with the statistical methods described in the beginning of this section to generate a new dataset. The division into small windows allowed for information of more events to be maintained; for example, if a driver maintains a constant speed for 20 s and then suddenly speeds up for another 10 s, a time-window of size 30 s will not be able to capture the two different events that took place in it (Murphey et al., 2009) [40].

The graph in Figure 10 presents this analysis of division in windows, where the x-axis is divided into segments of 2 s. It can be seen how the average speed value of the entire sample (green line) differs from each of the average speed values of the subsamples (red line), which is why it was decided for each test to be analyzed as a succession of discrete events.

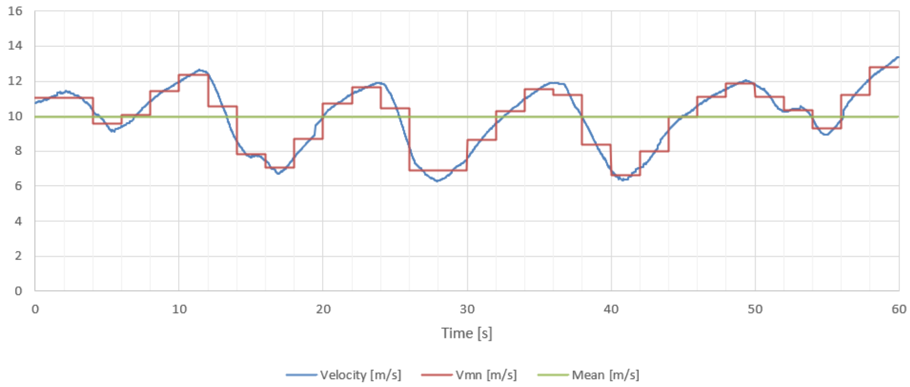

Figure 10. Speed averages in windows of 2 (s) and total average in one minute.

7. Results and Discussion

This section presents the results from the data processing and the classification algorithms performance. The results from pre-processing stages, such as feature selection, are also included in this section. The considered information includes the classification scores (accuracy, precision, recall, and F1-score) as well as training and prediction time. This metrics allow for comparisons to be made between the different pre-processing methods and Classifiers.

The first results presented correspond to the ones obtained with the original 21 features, that include the composite features RPA and Jerk (Section 5.3).

7.1. Feature Importance Results based on Mean Decrease in Impurity

"*Feature importance are computed as the mean and standard deviation of accumulation of the impurity decrease within each tree*" ("MATLAB Documentation", n.d.) [41]. This automated feature selection allows us to identify the contribution to the class separation in each level of a decision tree classifier. It is a straightforward method to identify notable contributions to the classification task among all the features.

7.2. Classifiers Benchmark

The pre-processed dataset was then split in a 75-25 proportion for training and testing, respectively. In addition, the trained classifier was later evaluated with data from a driving test that was not included in the original dataset. The results for this new dataset were consistent with the previous results, achieving an average F1-Score of 93%.

In this work, 14 classification algorithms were tested in a benchmark evaluation using the pre-processed data. As it is explained further in this document, the analysis then proceeded with only the three algorithms with the highest scores in the benchmark: Random Forest, KNN, and SVM-RBF. The other classification algorithms initially tested were: Ridge Classifier, Perceptron, Passive-Aggressive, Stochastic gradient descent, Nearest Centroid (Rocchio classifier), and Naive Bayes. For each one of those, extensive documentation can be found [42]. In this work, the default configuration parameters of these algorithms in python scikit-learn were used [43].

The benchmark allowed us to compare 14 different classifiers' performance, by using the same dataset in all of them. The results showed that Random Forest, KNN, and SVM classifiers provided consistently the best classification scores (Figure 11). In attention to the former, subsequent analyses were carried on with these three classifiers only.

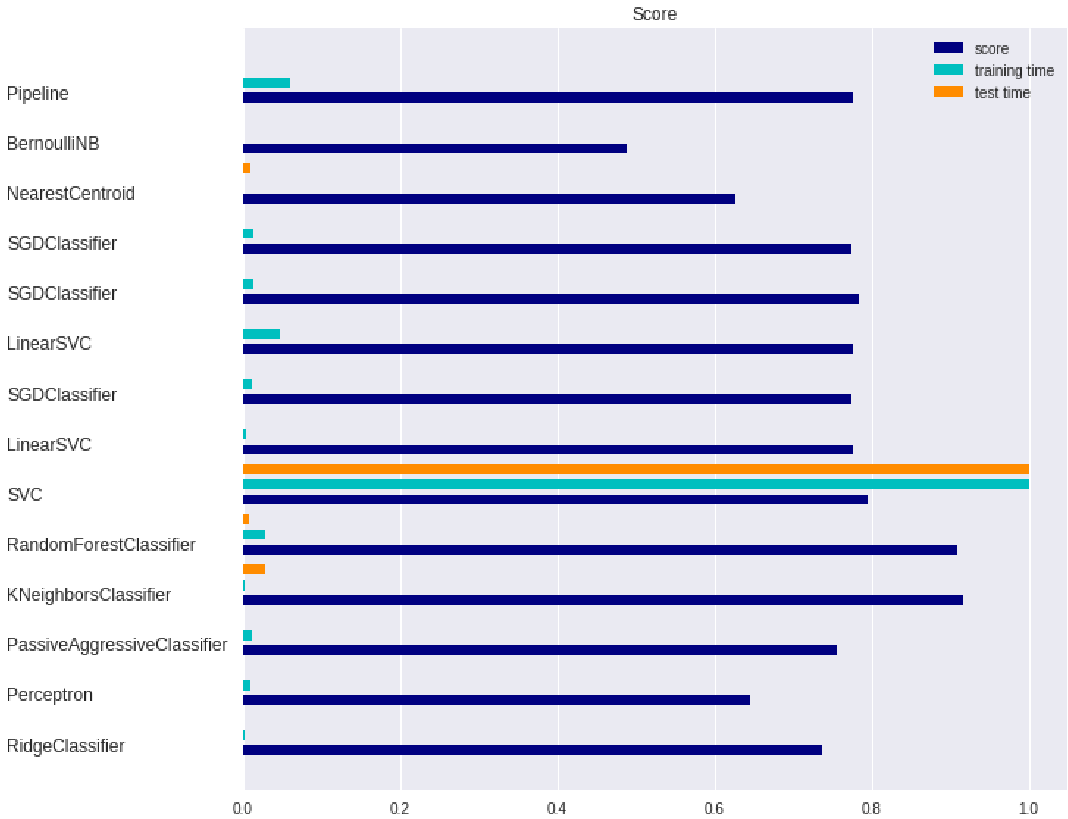

Figure 11. Benchmark of 14 different classification algorithms for the same dataset.

7.3. Processing and Analysis of Samples in 2-s Windows

By dividing each of the tests into 2-s subsamples and obtaining their statistical values, a new dataset is generated. It contains all the tests performed with 34,137 rows and 21 columns, with which the RF, KNN, and SVM algorithms are trained and evaluated.

The obtained performance of the KNN classifier with the aforementioned approach was quite poor (57.3% test accuracy), a dimensional reduction was performed in the same way that it was done for the set of 476 samples. Then, the behavior of the three algorithms could be checked again with this new reduced dimensionality dataset. After performing the steps described in Section 7.1, dimensions are reduced from 21 to 8, as shown in Figure 12, to reduce processing time and to evaluate the classification score again.

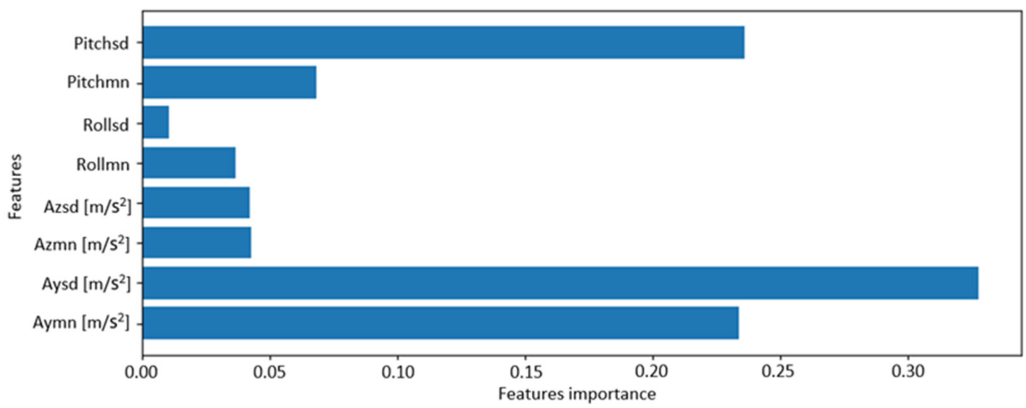

Figure 12. Importance of pre-selected features obtained with help of a decision tree classifier.

With this new data set of 34,137 rows and eight dimensions, the RF, KNN, and SVM algorithms were trained and evaluated again; the results are presented in Table 5, where it can be seen that the performance of the k nearest neighbors improved when considering only six neighbors. On the contrary, the SVM decreased its performance according to the precision, recall, and F1-score metrics. However, its accuracy in the test was not highly affected. Random Forest scores also decreased when reducing the features dimensionality.

Table 5. Evaluation metrics for KNN and SVM after dimensional reduction.

	KNN k = 19	SVM (RBF)	Random Forest
Test accuracy	93.2%	88.9%	90.7%
Precision	93%	89%	91%
Recall	94%	89%	91%
F1-score	93%	89%	91%
Support samples	8535	8535	8535

The Figure 13 table presents the resultant confusion matrices of the three classification algorithms and an example of classification area graphs created by them for a pair of features.

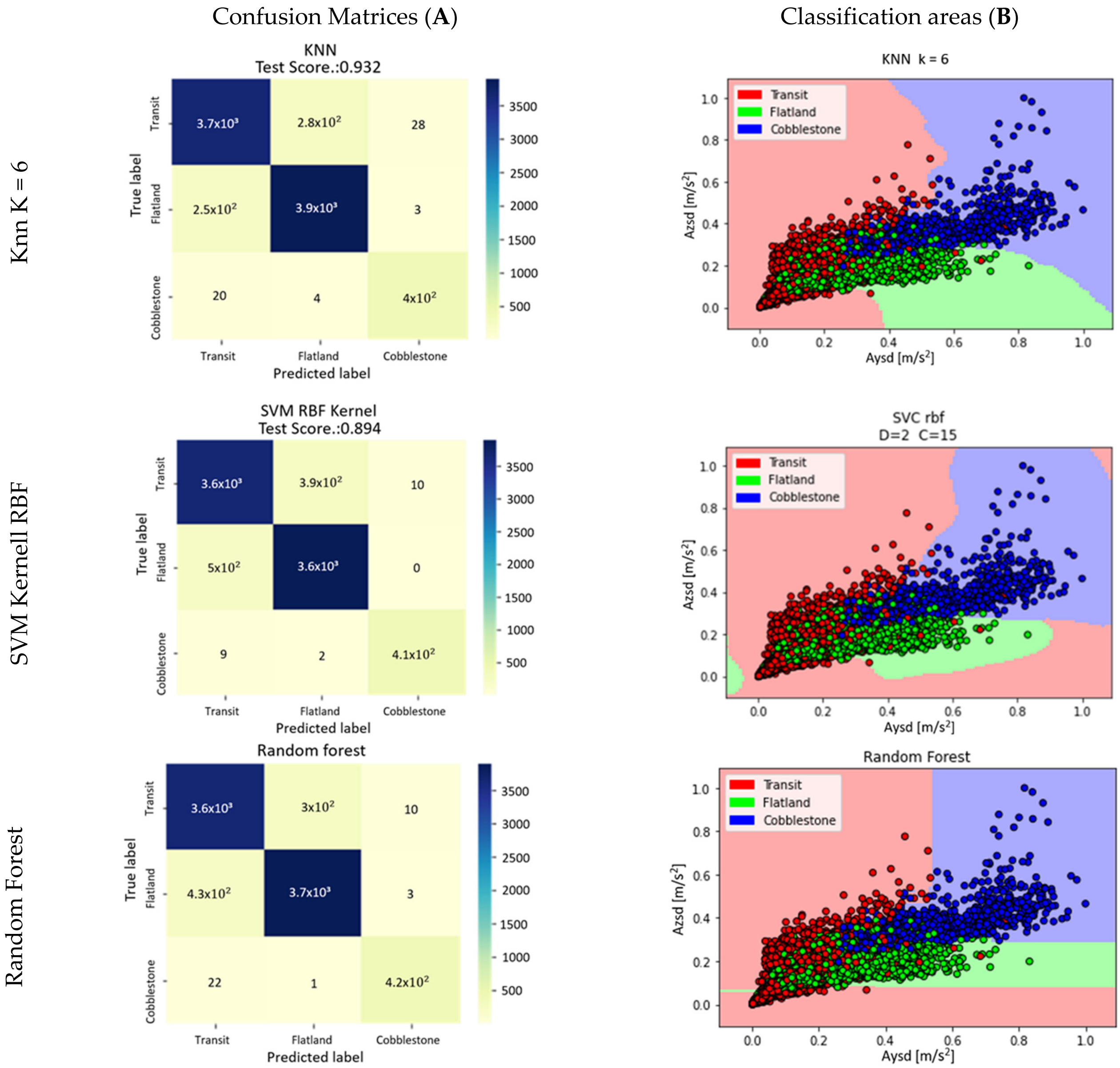

Figure 13. (**A**) Confusion matrices for KNN, SVM:RBF, and RF. (**B**) Class zones examples created by each classifier for two dimensions.

The separation is more visible at higher dimensions (Figure 14). Considering that the selected features were eight, the high-dimensional classification allows us to achieve better classification scores.

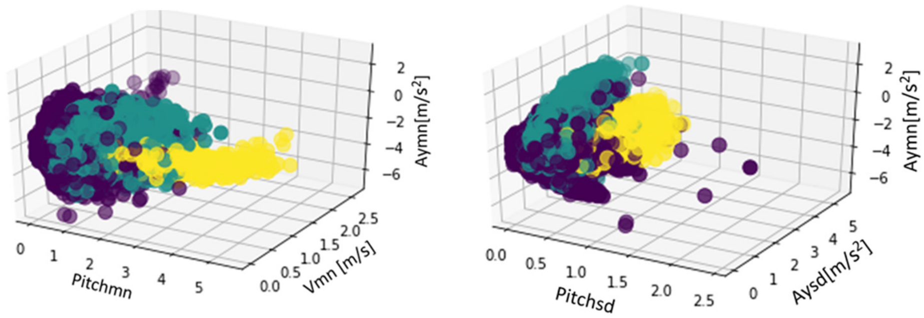

Figure 14. Visualization of the tree data classes in 3-dimensional graphs for 5 different features.

8. Conclusions

From the 23 originally proposed features, only some of the mean and standard deviation features were found to be important for this work's classification problem. This does not mean that the RPA and Jerk could not be of use in these kinds of problems, but rather that the particular characteristics of the three driving scenarios that were considered in this work were more easily classified by the other available features. The former affirmation can also be applied to the energy features that were not selected in the end.

Although the classification result scores of the driving scenarios were high (99%), when analyzing datasets of each pure class (Section 6.1), this was not the objective of the work, and it only served the purpose of providing a baseline of the ideal scenario in which only the pure data of each class were grouped and statistically processed together. After that, the discretization of the data into 2-s samples was closer to the actual purpose of analyzing the driving environment continuously and to make continuous predictions of the current driving environment. In this case, the classification results were over 93.2% and 90.7% accurate for KNN and RF, respectively, while using only eight features to do so.

Although the SVM:RBF classifier also achieved high scores (89%), the elevated calculation time to perform predictions and classifications makes it not suitable for real-time applications. It could be used for off-line training and vehicle data analysis, but not directly in real-time decision making.

9. Future Work

There is still plenty of research to be conducted to help autonomous driving systems to better perceive and process their environment and trying new approaches will help to bring the next improvements. Some propositions on this matter that can be studied further are:

- The integration of ML classifiers into intelligent driving systems for real-time awareness in pre-defined scenarios and the use of this information for calculating optimal energy-use strategies (Figure 15).
- The automatic classification of new scenarios, using ML strategies such as unsupervised learning for the clustering of classes with common characteristics in terms of drivability and energetic-related driving-style requirements.
- The same methodology exposed in this work can be tested again and improved by including more driving scenarios and routes with more complex characteristics and interactions with other players of the driving environment.

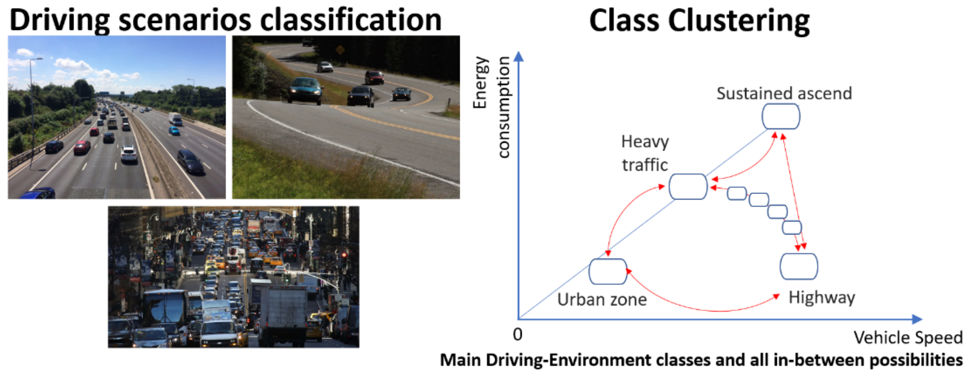

Figure 15. Application and future development of driving environment classification systems.

Author Contributions: Conceptualization, A.S.-D., M.R.B.-B. and J.d.C.J.-R.; methodology, A.S.-D., C.A.R.-R. and J.d.C.J.-R.; software, J.d.C.J.-R. and C.A.R.-R.; validation, J.d.C.J.-R., C.A.R.-R., R.A.R.-M. and A.S.-D.; formal analysis, M.R.B.-B., A.S.-D. and R.A.R.-M.; investigation, J.d.C.J.-R. and C.A.R.-R.; resources, A.S.-D.; data curation, C.A.R.-R. and J.d.C.J.-R.; writing—original draft preparation, J.d.C.J.-R. and C.A.R.-R.; writing—review and editing, J.d.C.J.-R., C.A.R.-R., M.R.B.-B., R.A.R.-M. and A.S.-D.; supervision, M.R.B.-B., A.S.-D. and R.A.R.-M.; project administration, A.S.-D.; funding acquisition, A.S.-D. and M.R.B.-B. All authors have read and agreed to the published version of the manuscript.

Funding: Funding was provided by Tecnológico de Monterrey (Grants No. a01368435 and a01368743) and Consejo Nacional de Ciencia y Tecnologia (CONACYT) by the scholarships 867164 and 892071.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available at request. Please write the corresponding authors.

Acknowledgments: The authors would like to thank the Research Center in Automotive Mechatronics (CIMA) for providing the facilities and equipment for the testing and data gathering.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. List of acronyms.

Acronym	Definition
ML	Machine Learning
KNN	k-nearest neighbors
RF	Random Forest
SVC	Support Vector Classifier
SVM	Support Vector Machine
RBF	Radial Base Function
IMU	Inertial Measurement Unit

Table A2. List of features.

Feature Name	Description
Vmn [m/s]	Mean Longitudinal Velocity [44]
Vsd [m/s]	Standard deviation of longitudinal velocity [44]
Amn(+) [m/s^2]	Mean of the positive acceleration in X. It is related to the use of the throttle [44]
Asd(+)	Standard deviation of positive acceleration at X [44]
Axsd [m/s^2]	Standard deviation of longitudinal acceleration [44]
Axmn [m/s^2]	Mean longitudinal acceleration [44]
Aymn [m/s^2]	Mean of vertical acceleration
Aysd [m/s^2]	Standard deviation of vertical acceleration

Table A2. *Cont.*

Feature Name	Description
Azmn [m/s^2]	Average lateral acceleration
Azsd [m/s^2]	Standard deviation of lateral acceleration
Abrmn [m/s^2]	Mean of braking acceleration (negative values of longitudinal acceleration) [44]
Abrsd	Standard deviation of braking acceleration [44]
RPA	Relative positive acceleration [38]
Potmn	Average power during the test, measured from batteries
Potsd	Standard deviation of electrical power
Jerk (X, Y, Z) [m/s^3]	Statistical value for comfort, relates the standard deviation and the average of the signal derived from the acceleration in each of the coordinate axes [40]
Rollmn	Roll average
Rollsd	Roll standard deviation
Pitchmn	Pitch average
Pitchsd	Pitch standard deviation
Energy [Wh/km]	Electric energy measured at the motor, given for each kilometer travelled

References

1. Qi, X.; Barth, M.J.; Wu, G.; Boriboonsomsin, K.; Wang, P. Energy Impact of Connected Eco-Driving on Electric Vehicles. In *Road Vehicle Automation 4*; Springer: Cham, Switzerland, 2018; pp. 97–111. [CrossRef]
2. Asher, Z.D.; Trinko, D.A.; Bradley, T.H. Increasing the Fuel Economy of Connected and Autonomous Lithium-Ion Electrified Vehicles. In *Green Energy and Technology*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 129–151. [CrossRef]
3. Shao, Y.; Sun, Z. Optimal Speed Control for a Connected and Autonomous Electric Vehicle Considering Battery Aging and Regenerative Braking Limits. In Proceedings of the ASME 2019 Dynamic Systems and Control Conference, Park City, UT, USA, 8–11 October 2019; Volume 1. [CrossRef]
4. Urbanik, T.; Tanaka, A.; Lozner, B.; Lindstrom, E.; Lee, K.; Quayle, S.; Beaird, S.; Tsoi, S.; Ryus, P.; Gettman, D.; et al. *Signal Timing Manual*, 2nd ed.; Transportation Research Board: Washington, DC, USA, 2015. [CrossRef]
5. Katrakazas, C.; Quddus, M.; Chen, W.H.; Deka, L. Real-Time Motion Planning Methods for Autonomous on-Road Driving: State-of-the-Art and Future Research Directions. *Transp. Res. Part C Emerg. Technol.* **2015**, *60*, 416–442. [CrossRef]
6. Hellström, E.; Åslund Jan, J.; Nielsen, L. Design of an Efficient Algorithm for Fuel-Optimal Look-Ahead Control. *Control. Eng. Pract.* **2010**, *18*, 1318–1327. [CrossRef]
7. Kamal, M.A.S.; Mukai, M.; Murata, J.; Kawabe, T. Ecological Vehicle Control on Roads with up-down Slopes. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 783–794. [CrossRef]
8. Ding, F.; Jin, H. On the Optimal Speed Profile for Eco-Driving on Curved Roads. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 4000–4010. [CrossRef]
9. Elamrani Abou Elassad, Z.; Mousannif, H.; al Moatassime, H.; Karkouch, A. The Application of Machine Learning Techniques for Driving Behavior Analysis: A Conceptual Framework and a Systematic Literature Review. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103312. [CrossRef]
10. Tian, J.; Chin, A.; Yanikomeroglu, H. Connected and Autonomous Driving. *IT Prof.* **2018**, *20*, 31–34. [CrossRef]
11. Chen, Y.; Feng, G.; Wu, S.; Tan, X. A New Hybrid Model Predictive Controller Design for Adaptive Cruise of Autonomous Electric Vehicles. *J. Adv. Transp.* **2021**, *2021*, 6626243. [CrossRef]
12. Kanjee, R.; Bachoo, A.K.; Carroll, J. Vision-Based Adaptive Cruise Control Using Pattern Matching. In Proceedings of the 2013 6th Robotics and Mechatronics Conference (RobMech), Durban, South Africa, 30–31 October 2013; pp. 93–98. [CrossRef]
13. Bingham, C.; Walsh, C.; Carroll, S. Impact of Driving Characteristics on Electric Vehicle Energy Consumption and Range. *IET Intell. Transp. Syst.* **2012**, *6*, 29–35. [CrossRef]
14. Ajanović, Z.; Stolz, M.; Horn, M. Energy Efficient Driving in Dynamic Environment: Considering Other Traffic Participants and Overtaking Possibility. In *SpringerBriefs in Applied Sciences and Technology*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 61–80. [CrossRef]
15. Bae, S.; Kim, Y.; Guanetti, J.; Borrelli, F.; Moura, S. Design and Implementation of Ecological Adaptive Cruise Control for Autonomous Driving with Communication to Traffic Lights. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 4628–4634. [CrossRef]
16. Shao, Y.; Mohd Zulkefli, M.A.; Sun, Z.; Huang, P. Evaluating Connected and Autonomous Vehicles Using a Hardware-in-the-Loop Testbed and a Living Lab. *Transp. Res. Part C Emerg. Technol.* **2019**, *102*, 121–135. [CrossRef]
17. Hellström, E.; Ivarsson, M.; Åslund, J.; Nielsen, L. Look-Ahead Control for Heavy Trucks to Minimize Trip Time and Fuel Consumption. *Control. Eng. Pract.* **2009**, *17*, 245–254. [CrossRef]
18. Škugor, B.; Deur, J. Synthetic Driving Cycles-Based Modelling of Extended Range Electric Vehicle Fleet Energy Demand. In Proceedings of the 30th International Electric Vehicle Symposium & Exhibition (EVS30), Stuttgart, Germany, 9–11 October 2018.

19. Teng, G.; Xiao, J.; He, Y.; Zheng, T.; He, C. Use of Group Method of Data Handling for Transport Energy Demand Modeling. *Energy Sci. Eng.* **2017**, *5*, 302–317. [CrossRef]
20. Brand, C.; Anable, J.; Morton, C. Lifestyle, Efficiency and Limits: Modelling Transport Energy and Emissions Using a Socio-Technical Approach. *Energy Effic.* **2019**, *12*, 187–207. [CrossRef]
21. De Cauwer, C.; Verbeke, W.; Coosemans, T.; Faid, S.; van Mierlo, J. A Data-Driven Method for Energy Consumption Prediction and Energy-Efficient Routing of Electric Vehicles in Real-World Conditions. *Energies* **2017**, *10*, 608. [CrossRef]
22. Delogu, M.; del Pero, F.; Pierini, M. Lightweight Design Solutions in the Automotive Field: Environmental Modelling Based on Fuel Reduction Value Applied to Diesel Turbocharged Vehicles. *Sustainability* **2016**, *8*, 1167. [CrossRef]
23. Topić, J.; Škugor, B.; Deur, J. Neural Network-Based Modeling of Electric Vehicle Energy Demand and All Electric Range. *Energies* **2019**, *12*, 1396. [CrossRef]
24. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In *Computer Vision—ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2014; Volume 8689, pp. 818–833. [CrossRef]
25. Tian, Y.; Pei, K.; Jana, S.; Ray, B. DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars. In Proceedings of the 2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE), Gothenburg, Sweden, 27 May–3 June 2018; pp. 303–314. [CrossRef]
26. Min, K.; Yeon, K.; Jo, Y.; Sim, G.; Sunwoo, M.; Han, M. Vehicle Deceleration Prediction Based on Deep Neural Network at Braking Conditions. *Int. J. Automot. Technol.* **2020**, *21*, 91–102. [CrossRef]
27. Jonas, T.; Hunter, C.D.; Macht, G.A. Quantifying the Impact of Traffic on Electric Vehicle Efficiency. *World Electr. Veh. J.* **2022**, *13*, 15. [CrossRef]
28. Wang, J.; Meng, Q.; Shang, P.; Saada, M. Road surface real-time detection based on Raspberry Pi and recurrent neural networks. *Trans. Inst. Meas. Control.* **2021**, *43*, 2540–2550. [CrossRef]
29. Lei, T.; Mohamed, A.A.; Claudel, C. An IMU-based traffic and road condition monitoring system. *HardwareX* **2018**, *4*, e00045. [CrossRef]
30. Trogh, J.; Botteldooren, D.; de Coensel, B.; Martens, L.; Joseph, W.; Plets, D. Map Matching and Lane Detection Based on Markovian Behavior, GIS, and IMU Data. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 2056–2070. [CrossRef]
31. Orfila, O.; Gruyer, D.; Hamdi, K.; Glaser, S. Safe and Ecological Speed Profile Planning Algorithm for Autonomous Vehicles Using a Parametric Multiobjective Optimization Procedure. Accepted. *Int. J. Automot. Eng.* **2019**, *10*, 26–33. [CrossRef]
32. Hu, Y.; Li, W.; Xu, K.; Zahid, T.; Qin, F.; Li, C. Energy Management Strategy for a Hybrid Electric Vehicle Based on Deep Reinforcement Learning. *Appl. Sci.* **2018**, *8*, 187. [CrossRef]
33. Jeoung, H.; Lee, K.; Kim, N. Methodology for finding maximum performance and improvement possibility of rule-based control for parallel type-2 hybrid electric vehicles. *Energies* **2019**, *12*, 1924. [CrossRef]
34. Kazemi, H.; Mahjoub, H.N.; Tahmasbi-Sarvestani, A.; Fallah, Y.P. A Learning-based Stochastic MPC Design for Cooperative Adaptive Cruise Control to Handle Interfering Vehicles. *IEEE Trans. Intell. Veh.* **2018**, *3*, 266–275. [CrossRef]
35. Rasool, E.; Liaquat, M.; Bhatti, A.I.; Mehmood, A. Cooperative Adaptive Cruise Control with Fuel Efficiency Using PMP Technique. In Proceedings of the 2019 5th International Conference on Control, Automation and Robotics (ICCAR 2019), Beijing, China, 19–22 April 2019; pp. 396–400. [CrossRef]
36. Delnevo, G.; di Lena, P.; Mirri, S.; Prandi, C.; Salomoni, P. On combining Big Data and machine learning to support eco-driving behaviours. *J. Big Data* **2019**, *6*, 64. [CrossRef]
37. Zhu, J.; Ngo, C.; Sciarretta, A. Real-Time Optimal Eco-Driving for Hybrid-Electric Vehicles. *IFAC-PapersOnLine* **2019**, *52*, 562–567. [CrossRef]
38. Tutuianu, M.; Bonnel, P.; Ciuffo, B.; Haniu, T.; Ichikawa, N.; Marotta, A.; Pavlovic, J.; Steven, H. Development of the World-Wide Harmonized Light Duty Test Cycle (WLTC) and a Possible Pathway for Its Introduction in the European Legislation. *Transp. Res. Part D: Transp. Environ.* **2015**, *40*, 61–75. [CrossRef]
39. Van de Weijer, C.J.T. Heavy-Duty Emission Factors: Development of Representative Driving Cycles and Prediction of Emissions in Real-Life. Ph.D. Thesis, Technischen Universität Graz, Graz, Austria, 1997.
40. Murphey, Y.L.; Milton, R.; Kiliaris, L. Driver's Style Classification Using Jerk Analysis. In Proceedings of the 2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems, Nashville, TN, USA, 30 March–2 April 2009; pp. 23–28. [CrossRef]
41. MATLAB Documentation. Available online: https://www.mathworks.com/help/matlab/ (accessed on 4 April 2022).
42. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
43. Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Müller, A.C.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; et al. API Design for Machine Learning Software: Experiences from the Scikit-Learn Project. *arXiv* **2013**, arXiv:1309.0238. [CrossRef]
44. Constantinescu, Z.; Marinoiu, C.; Vladoiu, M. Driving Style Analysis Using Data Mining Techniques. *Int. J. Comput. Commun. Control* **2010**, *5*, 654. [CrossRef]