

Research Article

Driver and Path Detection through Time-Series Classification

Mario Luca Bernardi ¹, Marta Cimitile ², Fabio Martinelli,³ and Francesco Mercaldo³

¹*Giustino Fortunato University, Benevento, Italy*

²*Unitelma Sapienza, Rome, Italy*

³*National Research Council of Italy (CNR), Pisa, Italy*

Correspondence should be addressed to Marta Cimitile; marta.cimitile@unitelma.it

Received 7 September 2017; Revised 17 January 2018; Accepted 12 February 2018; Published 22 March 2018

Academic Editor: Aboelmegeed Noureldin

Copyright © 2018 Mario Luca Bernardi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Driver identification and path kind identification are becoming very critical topics given the increasing interest of automobile industry to improve driver experience and safety and given the necessity to reduce the global environmental problems. Since in the last years a high number of always more sophisticated and accurate car sensors and monitoring systems are produced, several proposed approaches are based on the analysis of a huge amount of real-time data describing driving experience. In this work, a set of behavioral features extracted by a car monitoring system is proposed to realize driver identification and path kind identification and to evaluate driver's familiarity with a given vehicle. The proposed feature model is exploited using a time-series classification approach based on a multilayer perceptron (MLP) network to evaluate their effectiveness for the goals listed above. The experiment is done on a real dataset composed of totally 292 observations (each observation consists of a given person driving a given car on a predefined path) and shows that the proposed features have a very good driver and path identification and profiling ability.

1. Introduction

Driver identification allows discovering the identity of a vehicle driver using what he possesses and/or his physical and behavioral characteristics [1]. This topic is recently becoming very critical for the automobile industry, with an increasing number of always more sophisticated and accurate car sensors and monitoring systems able to extract information about the driver (i.e., hand geometry, keystroke dynamics, and voiceprint). The reason for the interest towards the topic is that driver identification may improve driver experience allowing (i) a safer driving and an intelligent assistance in case of emergencies, (ii) a more comfortable driving, and (iii) a reduction of the global environment problems [2]. Looking for the driver safety, the driver identification may support detecting some changes in the driver (due to possible indisposition or state of being drunk) and activate any security procedures (e.g., a ring may invite the driver to stop). The study of the driver behavior, for each segment of a road, also allows profiling each road section supporting the activation of alerts signals when more caution is required (e.g., a vocal message may alert the driver to reduce the brake

pressure in a dangerous curve) [3]. Concerning the driver comfort, for instance, the driver identification may discover which member of the family is currently driving the car and, consequently, perform an automatic setting of the car equipment (i.e., radio volume and frequency, temperature, or even speed limit) [4, 5]. Finally, the driver identification can be useful to suggest new car improvements based on the driver preferences or new systems with the aim of reducing the car consumption and pollution based on the driving characteristics. Based on the above-discussed advantages deriving by the driver profiling and identification, several studies have been proposed in the last years focusing on the identification of driver physical and behavioral features. The physical features [6, 7] are stable human characteristics that have been largely diffused in banking and forensic domain to guarantee a higher safeness concerning the more traditional authentication system based on the ownership of a key (this authentication system can be easily bypassed when someone comes in handy of the key). The behavioral features are used to detect individual personality features and are becoming the target of several studies in recent years that

are mainly focused on speaker recognition [8]. The limit of these approaches is that they are based on the analysis of only one feature. This may cause a high uncertainty in driver identification especially if there is a noisy sensor. For this reason, new approaches based on multimodal identification systems are introduced [9, 10]. They provide more accurate driver identification based on the detection and analysis of a higher set of behavioral features.

In this work, we propose a set of behavioral features to perform driver identification and path kind identification (i.e., dirt road, road with bumps, and highway) and to evaluate driver's familiarity with a given vehicle. Moreover, we exploit a time-series classification (TSC) approach based on a multilayer perceptron (MLP) network to evaluate the effectiveness of the proposed set of features extracted by using a CAN bus monitoring system.

We have identified three research questions (RQs):

RQ₁: To what extent does the set of extracted features allow driver identification?

RQ₂: To what extent does the set of extracted features allow path kind identification?

RQ₃: Is it possible, using the available features, to identify the driver's familiarity with the vehicle (e.g., ownership)?

The rest of the paper is organized as follows.

Section 2 discusses related work. Section 3 describes the background of our study. Section 4 presents the proposed approach discussing the overall classification process. Sections 5 and 6 discuss a set of experiments to answer the proposed research questions using three datasets (made of 292 driving sessions performed by ten drivers on four cars in five different paths). Finally, Section 7 provides conclusive remarks and future directions.

2. Related Work

In the past, the real-world automotive data retrieving was limited due to the difficulty to equip the sensors in the cars. Since the introduction of CAN protocol (<http://www.can.bosch.com>), this limit is overcome and driving style identification is becoming a very appealing scenario. CAN protocol defines a generic communication standard for all the vehicle electronic devices. CAN protocol can cooperate with the OBD II diagnostic connector (<http://obd2-elm327.com/>) that provides a candidate list of vehicle parameters to monitor along with how to encode their data.

As a matter of fact, researchers in [11] discuss a driver identification approach that is based on the driving behavior signals observed while the driver is following another vehicle. The analyzed signals (they were measured using a driving simulator) are the following: accelerator pedal, brake pedal, vehicle velocity, and distance from the vehicle in front. The approach obtains an identification rate equal to 81% for twelve drivers and equal to 73% for thirty drivers.

The accelerator and the steering wheel as characteristics to discriminate between different drivers are analyzed in [12].

They employ hidden Markov model (HMM) on the considered features to model the driver characteristics. They build two models for each driver: one trained from accelerator data and the other one learned from steering wheel angle data. Consequently, the models are used to identify different drivers obtaining an accuracy rate equal to 85%.

Another HMM-based approach to model driver human behavior is proposed in [13]. This method employs a simulated driving environment to evaluate the effectiveness of the proposed solution. Van Ly [14] explores the possibility of using the inertial sensors of the vehicle from the CAN bus to build a profile of the driver observing braking and turning events to characterize an individual compared to acceleration events.

Authors in [15, 16] represent gas and brake pedal operation patterns with the Gaussian mixture model (GMM). They obtain an identification rate equal to 89.6% using data extracted by a driving simulator and equal to 76.8% for a field test with 276 drivers, resulting in 61% and 55% error reduction, respectively, over a driver model based on raw pedal operation signals without spectral analysis. Considering data from steering wheel angle, brake status, acceleration status, and vehicle speed, researchers in [17] model the driver behavior through HMMs and GMMs with the aim of capturing the sequence of driving characteristics acquired from the CAN bus information. They obtain 69% of the accuracy of action classification and 25% of accuracy for driver identification.

Authors in [18] classify real-world mechanical features from the CAN bus with four different classification algorithms: they obtain an accuracy equal to 0.939 using decision tree, equal to 0.844 using KNN, equal to 0.961 for Random-Forest, and equal to 0.747 using MLP algorithm. Researchers in [19] classify a set of features extracted from the powertrain signals of the vehicle, showing that the learned classifier is able to recognize the human driving style based on the power demands placed on the vehicle powertrain with an overall accuracy equal to 77%.

In [20], the features extracted from the accelerator and brake pedal pressure are used as inputs to a fuzzy neural network (FNN) system to ascertain the identity of the driver. Two fuzzy neural networks, namely, the evolving fuzzy neural network (EFuNN) and the adaptive network-based fuzzy inference system (ANFIS), are used to demonstrate the viability of the two proposed techniques.

Summarizing the results obtained from the above-described studies, we can conclude that the obtained identification rate is ranging from 25% [17] to 0.961 [18]. The method we propose is able to reach a precision rate equal to 99%, overcoming the current literature in terms of precision. Furthermore, the existing methods are often tested in simulated environments: a plethora of variables (like the traffic jam and the number of the cars involved in the scenarios) are set a priori.

Differently, we conduct experiments in the real-world environment, in order to take into account real-world variables that cannot be predicted. In addition, we perform a set of experiments aiming to identify the car owner regardless of the car and path. Differently, the other discussed methods

usually perform the experiments considering a single setting: for example, in the experiment proposed in [18] (it is the method for which the best precision values are obtained) the drivers under analysis perform the same path on the same car.

3. Background: Time-Series Classification Approaches

Machine Learning (ML) explores the study and implementation of some algorithms aiming to learn from some monitored data and make predictions about their future values. Here, we focus on TSC algorithms used to classify sequences of observations of a phenomenon [21], as effectively experimented in [22] in another context.

Time Series (TS) consists of a sequence of discrete-time observations evaluated at successive equally spaced points in time. They are widely diffused to describe the time course of a phenomenon. TS is useful to predict its future trend and relate it to other phenomena under study. For this reason, they are adopted in several domains with different aims. A problem recurring in several domains is TS classification (TSC) which requires training a classifier on a set of cases, where each case contains an ordered set of real-valued attributes and a class label (qualifying its kind or nature). TSC problems arise in a wide range of fields including environmental sciences, computational biology, image processing, and software engineering.

ML approaches are widely adopted to perform such classification. There are two main ML algorithms families:

- (i) *Supervised Learning (SL)*. The classifier is presented with example inputs and their desired outputs, given by an oracle, and the goal is to learn a general rule that maps inputs to outputs. The learning phase is the process of building a model able to discriminate the classes from a set of records that contain class labels.
- (ii) *Unsupervised Learning*. No labels are given to the learning algorithm, leaving it to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data), or a means towards an end (feature learning).

In this paper, the focus is on supervised algorithms to perform classification of TS data.

A well-known class of classifier is based on decision trees. A decision tree can be used to predict the future values of an item or to perform classification tasks. The classification consists of splitting a dataset into smaller sets organized as a binary tree-like structure obtaining an easier description of the distribution of the data [23].

RandomForest [24] is a widely adopted decision trees algorithm proposing a way of averaging multiple deep decision trees (trained on different parts of the same training set) with the aim of reducing the variance. It starts by constructing several decision trees at training time. For each tree, the class (which can be the mode of the classification or the mean of prediction) is obtained. Random decision forests are used to correct decision trees habit of overfitting to their training set.

Another well-known method for TS classification approach is based on Dynamic Time Warping (DTW) that allows an elastic shifting of the time axis, to accommodate sequences that are similar, but out of phase. DTW is a classic technique for time-series classification. DTW is normally used with instance-based classifiers and can be incorporated into a decision trees algorithm. For constructing the decision trees, the J48 algorithm was exploited [23]. It starts by building decision trees from a set of training data and then visits all decision-node. At each iteration, it chooses the most active node and splits until each leaf is reached and not any more splits are obtainable.

Other effective approaches to classification problems exploit neural networks. MLPs are among the most popular kinds of neural networks and have been used in a wide variety of modeling, classification, prediction, and optimization problems. They are subclasses of more general structures called feedforward neural networks that are capable of approximating generic classes of functions (including continuous and integrable functions).

In this paper, an MLP network architecture is used.

An MLP neural network is based on multiple layers of nodes in a directed graph where each layer is fully connected to the next one [25]. In particular, the simplest MLP network model describes a fully connected network with three layers (one input layer, one hidden layer, and an output layer) in which each node is a neuron that uses a nonlinear activation function.

An activation function in MLP network is a (typically nonlinear) function transforming the weighted sum of inputs to a neuron into an output value. Usually, it maps the range of input signals into a reference interval (such as $[-1, 1]$ or $[0, 1]$). The activation functions used in typical neural networks are classified into threshold, linear, and nonlinear activation functions. For MLP networks the hidden layers have nonlinear activation functions whereas the output layer could have both linear and nonlinear ones. The MLP models adopt the following bipolar sigmoidal function (i.e. hyperbolic tangent sigmoid) as activation function:

$$\phi(v) = \frac{2}{(1 + e^{-2v})} - 1 \quad (1)$$

whereas a linear activation function is used for output layer. Looking for example to a single network model, it contains at least three fully connected layers (one hidden layer and one output layer). The hidden layer neurons are connected to the nodes of input layer by W weights, whereas V weights are used to connect the neurons of the output layers with the neurons of the hidden layer. Each layer applies the same activation function. Weights in the network are used to connect neurons between layers.

MLP behavior and performances depend upon three fundamental aspects: (i) activation functions of the units; (ii) network architecture; and (iii) the weight of each input connection. Since the first two aspects are fixed, the behavior of the MLP is mainly defined by the current values of the weights. The weights are defined during the training process: they are initially set to random values, and then instances

of the training set are repeatedly exposed to the network. The values for the input of an instance are placed on the input units and the output of the net is compared with the desired output for this instance. Then, all the weights in the net are adjusted slightly in the direction that would bring the output values of the net closer to the values for the desired output. There are several learning algorithms with which a network can be trained but the most well-known and widely used learning algorithm to estimate the values of the weights is the Backpropagation (BP) algorithm. An MLP network is trained using some form of gradient descent and the gradients are calculated using backpropagation. For classification, the learning algorithm minimizes the cross-entropy loss function (E_L) and weights are updated using Stochastic Gradient Descent (SGD) evaluating the gradient of the loss function as follows:

$$w = w - \eta \left(\alpha \frac{\delta R(w)}{\delta w} + \frac{\delta E_L}{\delta w} \right), \quad (2)$$

where η is the learning rate parameter which controls the step-size in the parameter space search and the cross-entropy loss function E_L is given by

$$E_L(\hat{y}, y, W) = -y \ln \hat{y} - (1 - y) \ln (1 - \hat{y}) + \alpha \|W\|_2^2, \quad (3)$$

where $\alpha \|W\|_2^2$ is a regularization factor that penalizes complex models and $\alpha > 0$ is a positive parameter that controls the magnitude of the imposed penalty.

For binary classification, the function computed by the network passes through the logistic function:

$$g(z) = \frac{1}{(1 + e^{-z})} \quad (4)$$

to obtain output values between zero and one. In this case a threshold t would assign samples of outputs larger than or equal to t to the positive class and the rest of the negative one. When there are $n > 2$ classes, the output of the network is a vector of size n . In this case, in order to assign a single class to the input sample, the softmax function is used, which can be defined as

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{l=1}^k \exp(z_l)}, \quad (5)$$

where z_i represents the i th element of the input to softmax, which corresponds to class i , and n is the number of classes. The result is a vector containing the probabilities that the sample x belongs to each class and the output class is the one with the highest probability.

In this paper, a dataset that has n classes and an architecture based on n joint models (each with one output layer neuron) has been adopted. All the models are jointly trained across K -fold cross-validation over the classes on the whole input w , and hence they are all characterized by the same best parameters (i.e., the same number of neurons in hidden layer, the same learning rates, and the same number of iterations). As already observed since this architecture, with $n > 2$ models, is suitable to detect more than two classes, it requires the softmax stage that takes the n inputs (the probabilities that the input sample belongs to each class) and provides the best class as output.

4. The Methodology

This section presents the proposed approach for classification of drivers and paths using data extracted from vehicle sensors. Figure 1 summarizes the overall mining process structured as two main subprocesses: (a) Datasets Generation and (b) MLP Training and Time-Series Classification. The remaining part of the section will describe in more detail these two subprocesses.

4.1. Datasets Generation. The Datasets Generation steps are reported in Figure 1(a). The first step consists of the dataset cleaning (by performing the removal of incomplete and wrong data values) and normalization. The cleaning and normalization activity is necessary since real-world data tend to be rather noisy, not complete, or even inconsistent. It applies techniques aimed at filling missing values, filters out the noise, and corrects the inconsistent values (or removes them) from the dataset. We adopted the following data cleaning subprocess to polish data produced by the car monitors in order to obtain a consistent dataset that is suitable for statistical inference:

- (i) fixing missing values
- (ii) removing noise
- (iii) removing special character or values
- (iv) verifying semantic consistency
- (v) normalization.

In the last step numerical attributes are normalized using a min-max normalization that performs a linear transformation of the original data. If \min_X and \max_X are the minimum and maximum values for the attribute X , the min-max normalization maps a value v_i of X to a v'_i in the range $\{\text{newMin}_X, \text{newMax}_X\}$ by computing

$$v'_i = \frac{v_i - \min_X}{\max_X - \min_X} (\text{newMax}_X - \text{newMin}_X) + \text{newMin}_X. \quad (6)$$

The normalized data is split into two sets (Training and Test Set Generation step): (i) a training set used to train the classifier and (ii) a test set used to assess the performance of the classifier. The dataset partitioning is performed by using a K -Fold Cross-Validation approach [26] consisting in splitting the data into k equally sized folds. Subsequently, k iterations of training and validation are performed (for each iteration a different fold of the data is used for validation and the remaining folds are used for training). Moreover, to ensure that each fold is representative, the data are stratified prior to being split into folds. This model selection method, according to [27], provides less biased estimation of the accuracy.

4.2. MLP Training and Time-Series Classification. The MLP Training and Time-Series Classification process is reported in Figure 1(b) and is based on an MLP network for learning [28].

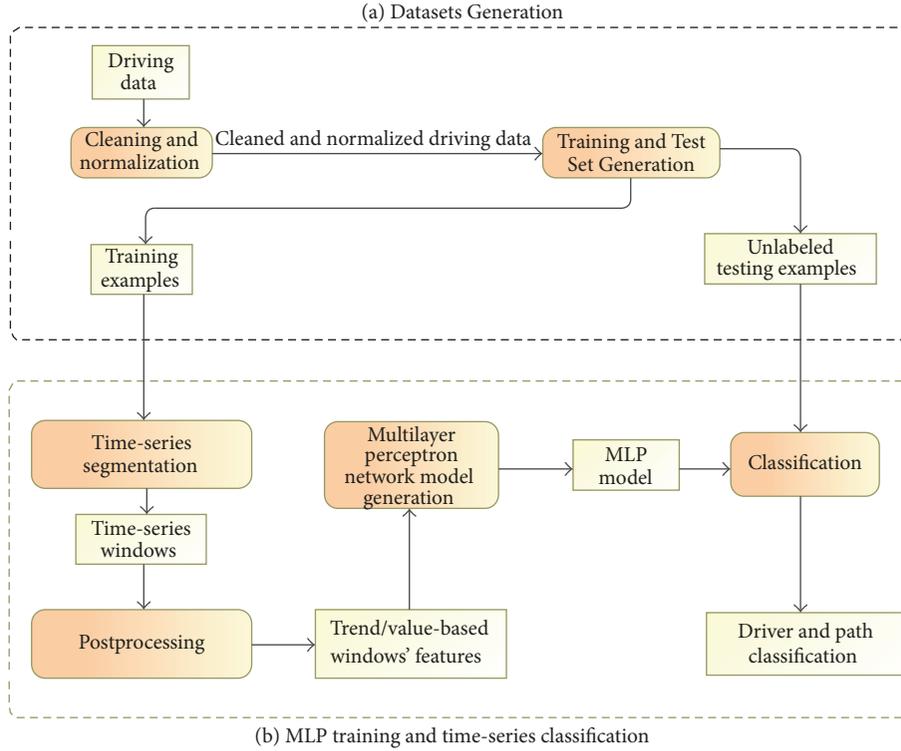


FIGURE 1: The overall classification process.

The adopted time-series classification approach is based on the following main steps:

- (i) Time-Series Segmentation, in which the time series are analyzed and divided into segments (i.e., windows)
- (ii) Postprocessing, in which for each window a representation based on values and trends features is generated
- (iii) Multilayer Perceptron Network Model Generation, in which an MLP network is trained
- (iv) Classification, in which the trained classifier is tested on new time-series samples to perform the classification step.

In the first step, the multivariate time series is firstly divided into a sequence of segments by sliding a window (which can be fixed or threshold-based) incrementally across the time-series values. In this paper, we experimented two time-series sliding window approaches: fixed windows and threshold-based windows (we refer to it as Start&Stop). The former exploits windows of fixed sizes. In particular, we defined five windows of increasing sizes ranging from 1 sec to 60 sec. Outside of this range, results or performances were not acceptable. The latter segmentation approach defines time-series regions based on signal variation thresholds. A Start&Stop region is defined as the interval in which the vehicle moves off from a steady state until it reaches another steady state. This means that only transitory parts of the signals are captured. The rationale for using just these portions

of the time series instead of the entire signal is intuitive: when signals became constants, they give little information (especially for what concerns the driving behavior). This, in effect, has proven to be very effective in filtering out useless information that wastes classifier memory but provides no useful information to help discriminating driving behavior or path kind.

After the segmentation, in the Postprocessing step, a set of features is evaluated for each time-series window. In particular, each window w_i is represented by a sequence of features $(Fv_1, \dots, Fv_n, Ft_1, \dots, Ft_m)$ containing n value-based features and m trend-based features. The first n features represent the discretized values of the time series and take one value in a discretized set (in the $[0, 1]$ range). The second set of m features describes the trend of the time series local to the window and is usually represented using shape-based metrics (including moments of various orders, mean, standard deviation, average energy, entropy, skewness, and kurtosis). In this paper, only a single feature is used as a value-based feature and two moments (standard deviation and skewness) are used to represent trend-based features. The approach to generating these representations is similar to the trend and the value-based analysis proposed in [29].

In Multilayer Perceptron Network Model Generation step, an MLP network, the ensemble of n networks, has been used. It is instantiated for each dataset and exploits a number of networks model equal to the number of the classes to be detected.

The input vectors for the MLP network contains the window representation components (value and the two

TABLE 1: Features involved in the study with the relative description and features source (i.e., a feature extracted by the OBD connector or gathered by the device GPS).

#	Feature	Description
#1	CO ₂ ing/km (average) (g/km)	CO ₂ consumption computed in average
#2	Cost per mile/km (Instant) (€/km)	Instantaneous cost for kilometer
#3	Cost per mile/km (Trip) (€/km)	Trip cost for kilometer
#4	Engine Load (%)	Percentage load of the engine
#5	Engine RPM (rpm)	Revolutions per minute of the engine
#6	Fuel cost (trip) (cost)	Cost of the consumed fuel for trip
#7	$G(x)$	x -axis gyroscope, i.e, pith
#8	$G(y)$	y -axis gyroscope i.e., yaw
#9	$G(z)$	z -axis gyroscope i.e., roll
#10	GPS Speed (meters/second)	Speed as measured from GPS unit
#11	GPS versus OBD Speed difference (km/h)	Speed difference between GPS chipset and OBD
#12	Kilometers per Liter (Instant) (kpl)	Instantaneous number of kilometers traveled for fuel liter
#13	Kilometers per Liter (Long Term Average) (kpl)	Average number of kilometers traveled for fuel liter
#14	Liters Per 100 Kilometer (Instant) (l/100 km)	Instantaneous liters employed for 100 Km
#15	Liters Per 100 Kilometer (Long Term Average) (l/100 km)	Average liters employed for 100 Km
#16	Trip Average KPL(kpl)	Distance traveled (in km)/fuel consumed by the vehicle

moments) representing data for each feature reported in Table 1.

In this step, the MLP network is trained using the class labels available for each set of value-based and trend-based information of each window.

In the classification step, the trained classifier can be used to classify new data and is validated on new samples to assess its performances.

Features Selection. The proposed classification approach is based on some assumptions that are confirmed by the experimental results: (i) each person has a different driving style that can be recognized irrespective of the path or vehicle; (ii) there exist features whose distribution of values is well separated from women and men allowing an effective genre identification; (iii) there exist features, related to how the vehicle is solicited by the road, that allow recognizing road kind (among several types). Based on such considerations, we selected an initial set of sixteen features that are reported in Table 1.

The set of features reported in Table 1 is analyzed performing a feature selection step to reduce the dimensionality of the dataset. This step requires analyzing and understanding the feature's impact on the classification model. The most relevant features are selected by evaluating correlations among the features (the selection of a subset of an orthogonal and independent set of features allows discarding redundant information). To this aim, we used a correlation-based features selection (CFS) algorithm exploiting correlation matrix filters as discussed in [30]. CFS approach allows rating the features using a correlation-based heuristic evaluation function. This function is biased towards those subsets of features largely correlated with the class to be detected and uncorrelated with each other. All the unnecessary features can be ignored given their low correlation with the class and the redundant features can be eliminated given their

high correlation with one or more of the remaining features. A feature is approved if it gives an extended and efficient classification in regions of the examples space not already covered by other features. The CFS evaluation function can be expressed as follows:

$$M_S = \frac{kr_{cf}}{\sqrt{k + k(k-1)r_{ff}}} \quad (7)$$

M_S represents the heuristic goal function of the subset S containing k features. The mean feature-class correlation is indicated as r_{cf} while r_{ff} is the average feature-feature intercorrelation. The numerator of the equation indicates the capability of a set of features to classify an example while the denominator indicates how much redundancy there is among the features set.

5. Experiment Setting

In this section, we describe the features extraction process and the dataset realized to perform the experiments.

5.1. Features Extraction. We constructed a dataset gathering data from the CAN bus of a set of real vehicles. In order to collect data, the Torque Pro (<https://play.google.com/store/apps/details?id=org.prowl.torque>) application and Mini Bluetooth ELM327 OBD 2 Scanner were used.

The OBD scanner was installed on the vehicles to produce a self-diagnostic report generated by the onboard monitoring system.

The data is recorded every second during driving using Torque Pro application by an Android smartphone fixed in the car using an adequate support.

5.2. Datasets Definition. In this paper three different datasets (some further details and the download of the

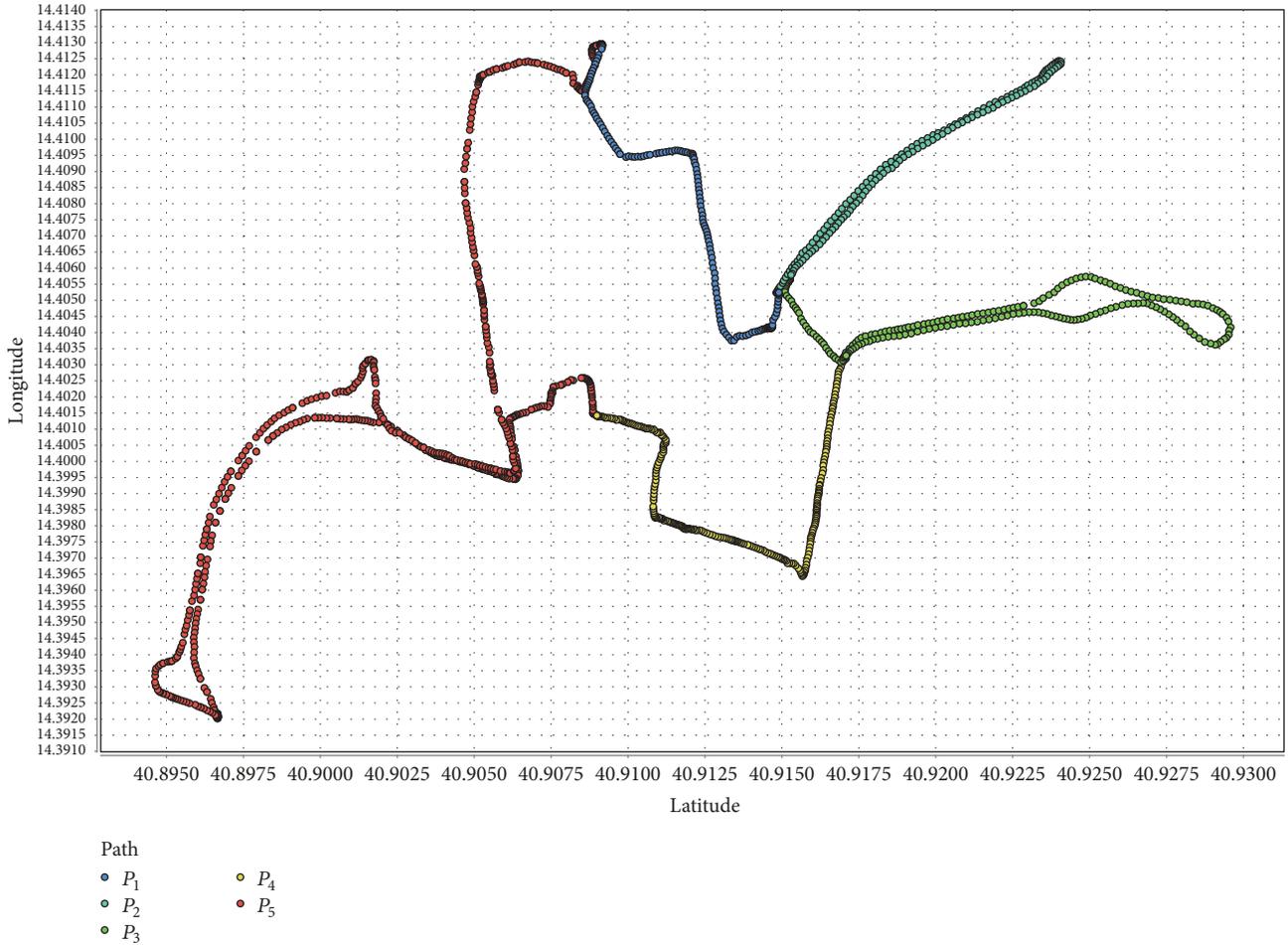


FIGURE 2: An overview of the paths under study: 30 km in five paths of different kind in the Naples area.

datasets are available at <https://github.com/martinacimitile/Car-Data-Mining/wiki> (D_A , D_B , and D_C) are exploited to answer the proposed research questions. All the datasets refer to the area shown in Figure 2, where the study has been executed: x - and y -axes of the figure are associated with longitude and latitude whereas the color represents one of five paths.

Each dataset contains two replicas of the same observation obtained with different conditions to avoid bias.

D_A is composed of two replicas of 16 observations and has the following characteristics:

- (i) Each observation describes the driving on the entire track reported in Figure 2.
- (ii) The observations are performed by four drivers (DF_1 , DM_1 , DM_2 , and DM_3) on 4 cars (Hyundai i20, Lancia y, Fiat Punto, and Nissan Note).
- (iii) For each replica, four persons drive four cars on the entire track.

D_B is composed of two replicas of 50 observations having the following characteristics:

- (i) Each observation describes one of the paths (i) composing the track of Figure 2. As shown by the figure we

consider five consecutive paths (called, resp., P_1 , P_2 , P_3 , P_4 , and P_5).

- (ii) Five men (DM_1, \dots, DM_5) and five women (DF_1, \dots, DF_5) drive the same car (Hyundai i20).

D_C is composed of two replicas of 80 observations having the following characteristics:

- (i) Each observation describes one of the paths reported in Figure 2.
- (ii) The observations are performed by four drivers (DF_1 , DM_1 , DM_2 , and DM_3) on 4 cars (Hyundai i20, Lancia y, Fiat Punto, and Nissan Note) and on 5 paths.
- (iii) For each replica, each person drives all the 4 cars one time on each path.

5.3. Descriptive Statistics. In this section, descriptive statistics are used to describe the features used in this study (a quantitative analysis of the distributions of features has been performed). The analysis shows that several groups of features are well separated and median values do not fall into interquartile ranges of the distributions.

In order to provide statistical evidence that the features can be considered as characteristics of the driver

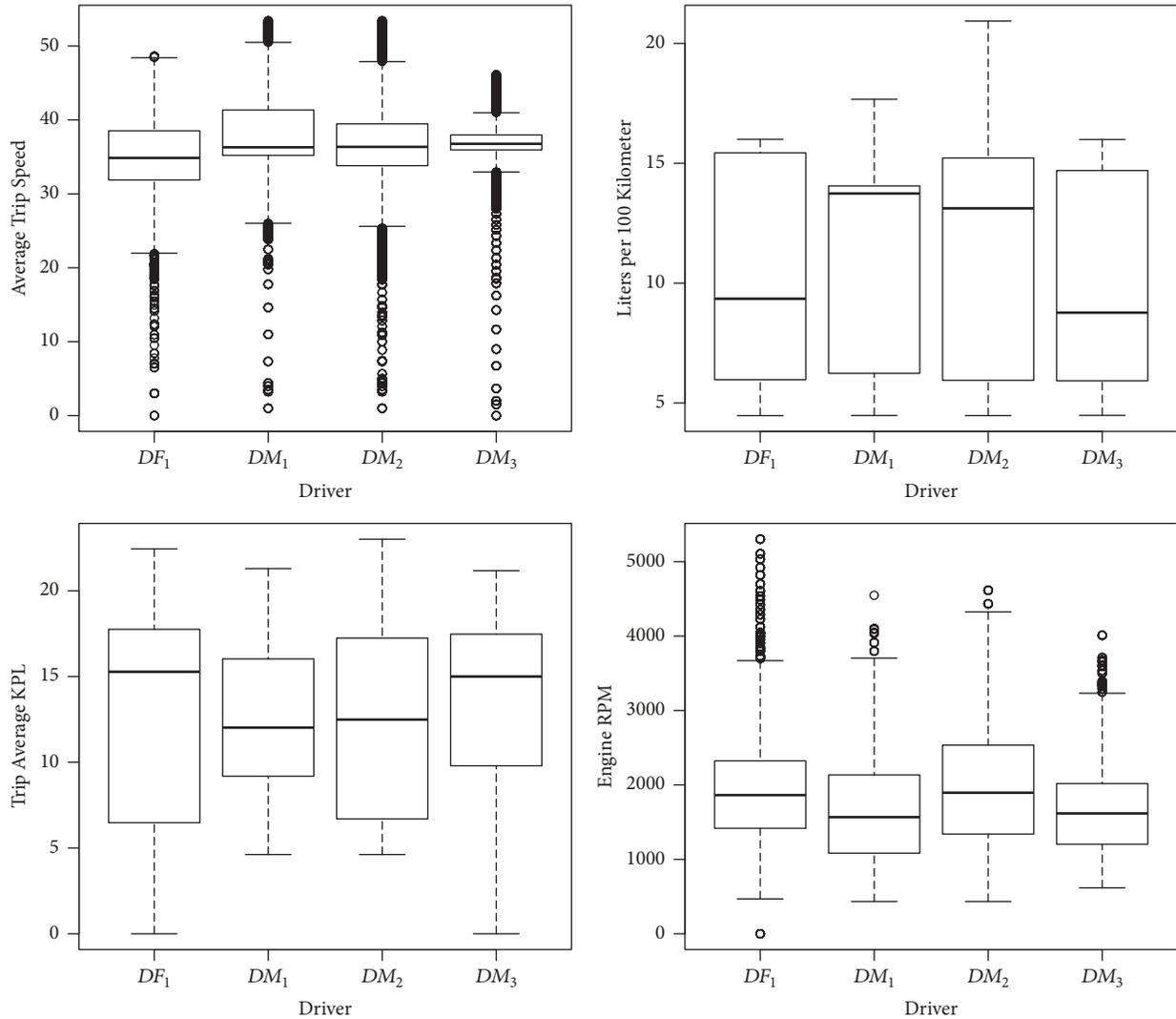


FIGURE 3: Box plots related to the distribution of the features Average Trip Speed, Liters Per 100 Kilometers, Trip Average KPL, and Engine RPM for the drivers DF_1 , DM_1 , DM_2 , and DM_3 .

behavior we show the box plot related to four drivers (i.e., DF_1 , DM_1 , DM_2 , and DM_3) features distribution shown as box plots.

Figure 3 shows the distributions of four drivers related to Average Trip Speed, Liters Per 100 Kilometers, Trip Average KPL, and Engine RPM.

As shown in the figure, the Average Trip Speed of the four drivers is the same but it is interesting to observe that the drivers DM_2 and DF_1 do not present peaks of speed (they keep the same speed variation both in acceleration and in deceleration during the entire driving session).

The figure also shows that even if the interquartile ranges are comparable among the drivers, the median values are quite different for different features.

Moreover, for the feature ‘‘Liter per 100 Kilometer,’’ distributions present different medians for the drivers. This is symptomatic of the different fuel consumption between the drivers involved in the experiment.

The driver DM_1 is confirmed to be the more aggressive relating to the driving style: indeed his median is very close to the 3rd quarter, and this is confirmed by the fact that he is the driver that reaches the higher speed (as confirmed by the box plots in Figure 3). The driver DF_1 , on the other side, presents a fuel consumption very close to the one exhibited by DM_1 but, as we observe from the box plot in Figure 3, she reaches an average speed very close to the one of DM_2 , consuming less fuel than DF_1 and DM_1 .

The driver DM_3 exhibits an average fuel consumption slightly lower than DM_2 : this confirms the balanced driving style of the driver.

As shown by the box plots, the medians of the Trip Average KPL (i.e., the ratio between distance traveled and fuel consumed) for DM_2 and DM_3 exhibit the best values; moreover DM_1 and DF_1 present almost the same medians. This means that DM_2 and DM_3 can travel more kilometers with less fuel with respect to DM_1 and DF_1 .

The distributions relating to the Engine RPM box plot show that the drivers DM_2 and DF_1 exhibit the higher value related to the Engine RPM: probably they change gear too late, and hence the Engine RPM rises. From the other side, we observe that DM_1 and DM_3 drivers present a lower media value for the considered feature: this means that they do not stress the engine.

This is also confirmed looking at the scatter plots of the distributions shown in Figure 4. Figure 4(a) shows a long term average of the kilometers per liter that are done by drivers (shown in different colors). The scatter plot, in this case, shows that the distributions are well separated. The same considerations can be done for the distribution of velocity concerning path kind that is shown in Figure 4(b). In this case, the scatter plot reveals that different path kinds have quite different velocity distributions.

6. Experiments Description

In this section, the description of the experiments conducted to answer the RQs introduced in Section 1 is reported.

6.1. Experiments Design. The set of conducted experiments is synthesized in Table 2. The table shows, for each RQ, the list of the conducted experiments (the experiment label, its goal, and the involved dataset).

Looking at the table, RQ₁ is explored with five experiments (E_{11} , E_{12} , E_{13} , E_{14} , and E_{15}). E_{11} , E_{12} , and E_{13} aim to evaluate the effectiveness of the approach in identifying the driver in three cases:

- (i) Regardless of the car but fixing the path (E_{11})
- (ii) Regardless of the path but fixing the car (E_{12})
- (iii) Regardless of both car and path (E_{13}).

Moreover, E_{14} and E_{15} aim to identify the gender of the driver:

- (i) Regardless of the car but fixing the path (E_{14})
- (ii) Regardless of the path but fixing the car (E_{15}).

Similarly, RQ₂ is explored with a single experiment (E_{21}) aiming to evaluate if it is possible to identify the path kind fixing the car but regardless of the path and the driver. RQ₃ is explored by means of two experiments. The first experiment (E_{31}) aims at evaluating if it is possible to identify the driver familiarity with the vehicle fixing the car but regardless of the path and the driver. The second experiment (E_{32}) aims at evaluating if it is possible to identify the driver familiarity with the vehicle when car, driver, and path can change.

Each experiment consists in applying the classification method described in Section 4 on a specific dataset (Table 2 reports for each experiment the considered dataset).

In particular in E_{11} the classification method is applied to the dataset D_A , since it contains several drivers and a single path. For what concerns E_{12} , the dataset D_B can be used to study the influence of the path, fixing the car (since it was built on a single car). In E_{13} the dataset D_C is used since it is based on four cars and contains five paths. E_{14} is conducted using the dataset D_A since we need multiple

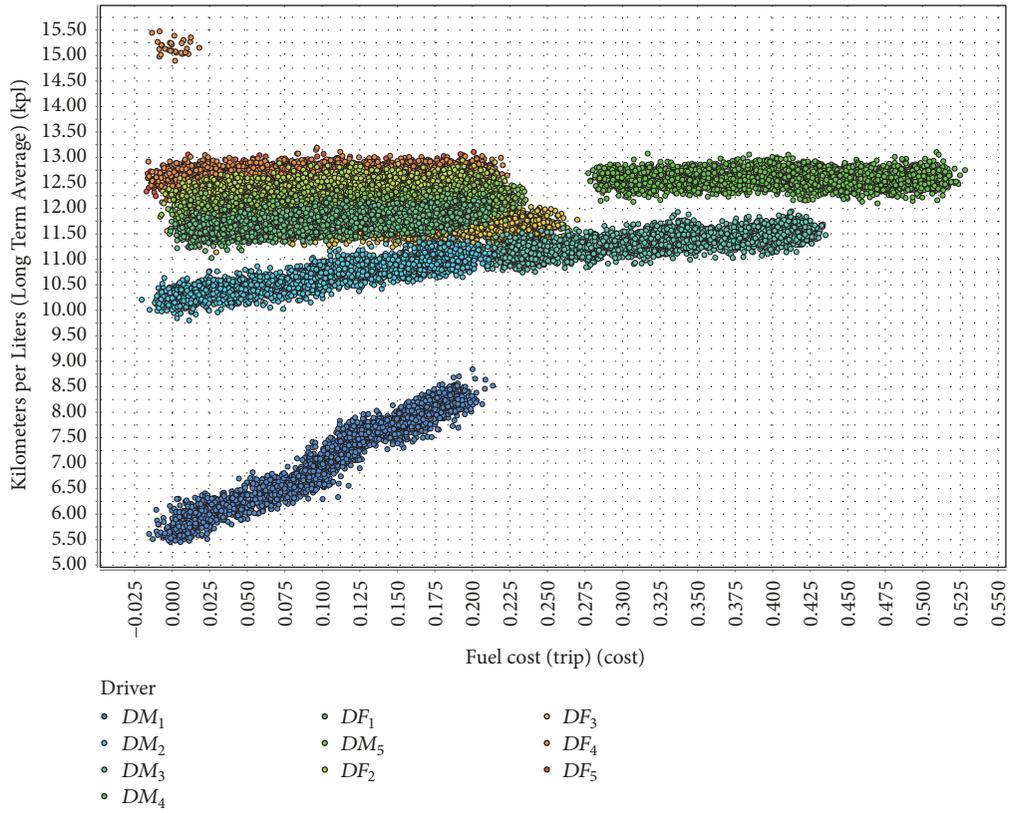
cars involving drivers of both genres. Conversely, E_{15} is performed on the dataset D_B since it is based on 5 paths for each driver and contains a well-balanced set of male and female drivers (five men and five women) on a single car. For E_2 we exploited the dataset D_B since it contains twelve drivers and five paths and it was performed on a single car. RQ₃ is explored in the experiments E_{31} and E_{32} involving, respectively, the datasets D_A and D_B . Based on the classification process described in Figure 1, each considered dataset is cleaned and normalized performing the Datasets Generation steps. The normalized dataset is used to generate a training set and a test set. Each experiment is performed using the MLP classifier as described in Figure 1. Moreover, the experiment E_{11} is performed using two alternative classifiers (RF and DWT) in order to compare the effectiveness of the MLP classifier with respect to other approaches often used in literature. Finally, each experiment is also repeated using different window segmentation strategies (1 s, 10 s, 30 s, 60 s, and the Start&Stop method) in order to evaluate the impact of the sliding window approach on the classifier performances.

Another consideration can be made on the training dataset. It is obtained as a partition of the normalized dataset and it is augmented with a column that specifies the classification labels associated with the evaluated instance. This column was derived by an expert looking at the evaluated instance in the area considered for the study. This classification is then used to both train the classifier and perform the evaluation. In the context of RQ₁, for the experiments E_{11} , E_{12} , and E_{13} , the considered datasets are augmented with a column that specifies the driver identity. For the E_{11} and E_{13} the possible drivers identity labels can be DF_1 , DM_1 , DM_2 , and DM_3 (corresponding to all the possible drivers involved in D_A and D_C). For E_{12} , looking at the explored dataset D_B the drivers identity label can assume the following values: DM_1 , DM_2 , DM_3 , DM_4 , and DM_5 and DF_1 , DF_2 , DF_3 , DF_4 , and DF_5 . Similarly, for the experiments corresponding to RQ₁₄ and RQ₁₅ the considered datasets are augmented with a column that specifies the driver gender (driver is labeled as “Male” or “Female”). In the experiment conducted to answer the RQ₂, the dataset D_B was augmented with a column that specifies the kind of the path (the path is labeled as “City Street”; “Highway”; or “Dirt road”) needed to perform the training and the validation. Similarly, in the context of RQ₃, the explored datasets were augmented with a column that specifies the ownership (“Owner”; “Not Owner”).

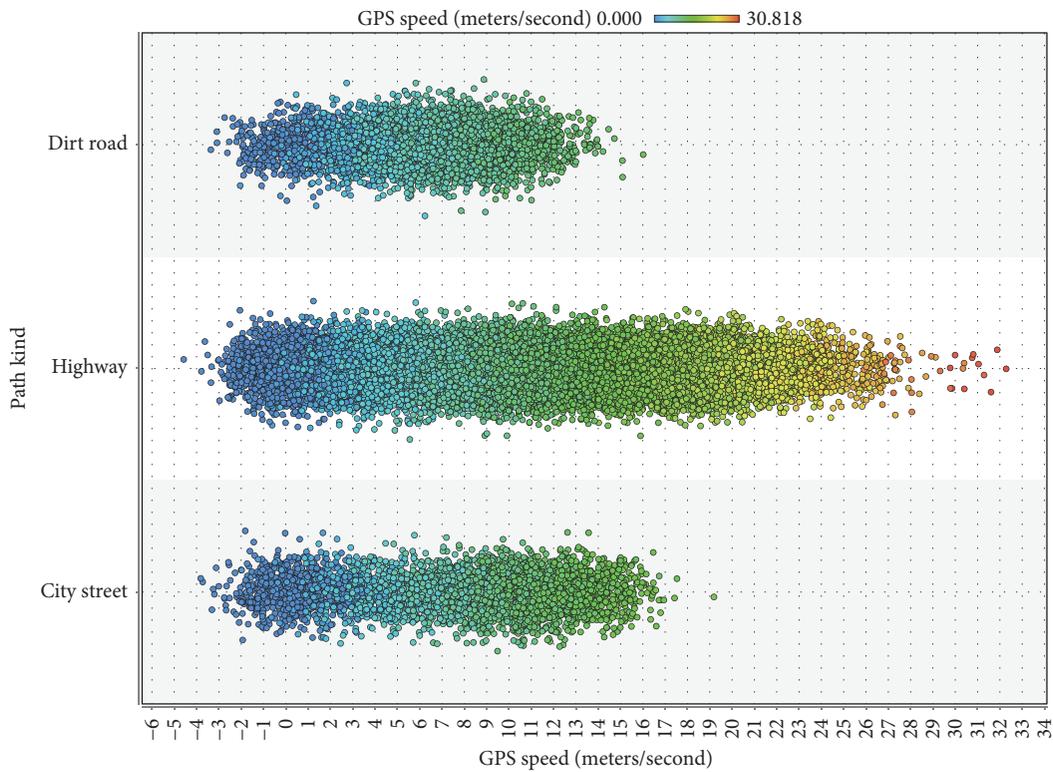
6.2. Evaluation Strategy. The validation has been performed using classification quality metrics. The three metrics used to evaluate the performance of our approach for the research questions are precision, recall, and accuracy.

Precision has been computed as the proportion of the observations that truly belong to investigated class (e.g., driver, driver genre, driver path, and driver’s familiarity) among all those which were assigned to the class. It is the ratio of the number of records correctly assigned to a specific class to the total number of records assigned to that class (correct and incorrect ones):

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}, \quad (8)$$



(a) Kilometers per Liter (long average) with respect to fuel cost, for different drivers



(b) Speeds with respect to different path kinds (city street, highway, and dirt road)

FIGURE 4: Scatter plots of the data distribution on dataset D_A .

TABLE 2: Experiments overview.

RQ	Experiment label	Dataset	Experiment goal
RQ ₁ : to what extent do OBD II features allow driver identification?	E_{11}	D_A	Evaluate the effectiveness of the proposed approach to identify the driver regardless of the car but fixing the path
	E_{12}	D_B	Evaluate the effectiveness of the proposed approach to identify the driver regardless of the path but fixing the car
	E_{13}	D_C	Evaluate the effectiveness of the proposed approach to identify the driver regardless of both car and path
	E_{14}	D_A	Evaluate the effectiveness of the proposed approach to identify the driver gender regardless of the car but fixing the path
	E_{15}	D_B	Evaluate the effectiveness of the proposed approach to identify the driver gender regardless of the path but fixing the car
RQ ₂ : to what extent do OBD II features allow path kind identification?	E_{21}	D_B	Evaluate the effectiveness of the proposed approach to identify the path kind fixing the car but regardless of the path and the driver
RQ ₃ : is it possible, using the features available in OBD II, to identify the driver's familiarity with the vehicle?	E_{31}	D_A	Evaluate the effectiveness of the proposed approach to identify the driver familiarity with the vehicle fixing of the car but regardless of the path and the driver
	E_{32}	D_B	Evaluate the effectiveness of the proposed approach to identify the driver familiarity with the vehicle when car, driver, and path can change

TABLE 3: E_{11} , precision, recall, accuracy, and training time obtained by using the MLP classifier for different windows sizes.

Average window size	Driver	Driver recall	Driver precision	Overall precision	Overall recall	Accuracy	Training time (sec)
1 s 14592 training segments	DF1	0,66	0,68	0,68	0,68	0,93	1048
	DM1	0,71	0,66				
	DM2	0,69	0,7				
	DM3	0,65	0,69				
10 s 1459 training segments	DF1	0,76	0,87	0,90	0,90	0,90	564
	DM1	0,86	0,81				
	DM2	0,98	0,93				
	DM3	0,98	0,98				
30 s 486 training segments	DF1	0,69	0,91	0,82	0,83	0,82	177
	DM1	0,7	0,76				
	DM2	0,92	0,85				
	DM3	0,97	0,78				
60 s 243 training segments	DF1	0,25	0,94	0,49	0,53	0,44	78
	DM1	0,21	0,26				
	DM2	0,51	0,54				
	DM3	0,98	0,36				
Start&Stop 7562 training segments	DF1	0,96	0,82	0,96	0,94	0,94	762
	DM1	0,86	0,94				
	DM2	1	0,99				
	DM3	1	1				

where tp indicates the number of true positives and fp indicates the number of false positives.

The recall has been computed as the proportion of observations that were assigned to a given class, among all the observations that truly belong to the class. It is the ratio of the number of relevant records retrieved to the total number of relevant records:

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}, \quad (9)$$

where tp indicates the number of true positives and fn indicates the number of false negatives.

Accuracy is defined as a statistical measure of how well a binary classification is able to evaluate correctly the instances under analysis with respect to the considered features. Basically the accuracy is the proportion of true results (both true positives and true negatives) among the total number of instances evaluated.

In the following, for each RQ, the results of experimentation have been described and discussed. The datasets were all replicated once in different conditions of traffic and timing, to avoid the bias of such variables. The results were consistent with the ones obtained on the first replica and hence are not reported in detail. In the online repository (<https://github.com/martinacimitile/Car-Data-Mining/wiki>) all the datasets, along with replicas, are provided to allow experiments replication.

6.3. *Discussion of Results.* Table 3 reports results for the experiment E_{11} performed by using the MLP classifier.

In the first column of the table, the adopted window sizes are reported. For each window size, we also specified the number of observations that are collected (e.g., considering a 1-second window, the number of collected training segments is 14592). Starting from the third column, the table reports—for each segmentation choice and for each driver (column two)—the values of precision, recall, accuracy, and training times obtained by using the MLP classifier. Precision and recall are evaluated for each class (the driver in this context) and as averages. The results of the experiment E_{11} performed by using the RF and the DTW classifiers are shown in Table 4 (for brevity, they are shown only for this first experiment). Comparing Tables 3 and 4, we can conclude that the best results are obtained for MLP on almost all the sizes. Based on the results, we can also conclude that the best segmentation choice on MLP is the threshold-based segmentation. However, MLP is also the slowest classifier in terms of training time. Moreover, MLP and RF perform better than DTW on medium window sizes and for threshold-based segmentation.

The E_{11} results are also synthesized by Figure 5. Figure 5(a) shows the trend of the classification accuracy for MLP, RF, and DTW with respect to the adopted window strategy. Figure 5(b) shows the training times: RF and DTW are comparable but several times (6x) faster than the MLP network.

TABLE 4: E_{11} , precision, recall, accuracy, and training time obtained by using the RF and DTW classifiers for different windows sizes.

Average window size	RandomForest (RF)						DTW					
	Driver recall	Driver precision	Overall precision	Overall recall	Accuracy	Training time (sec)	Driver recall	Driver precision	Overall precision	Overall recall	Accuracy	Training time (sec)
1 s	DFI 0,84	0,82					0,81	0,72				
14592 training segments	DMI 0,73	0,87	0,83	0,85	0,84	228	0,81	0,77	0,71	0,77	0,72	151
	DM2 0,81	0,89					0,75	0,81				
	DM3 0,88	0,89					0,76	0,83				
10 s	DFI 0,76	0,87					0,73	0,85				
1459 training segments	DMI 0,84	0,80	0,87	0,88	0,89	48	0,81	0,78	0,83	0,87	0,81	48
	DM2 0,96	0,90					0,91	0,87				
	DM3 0,93	0,97					0,89	0,96				
30 s	DFI 0,67	0,89					0,65	0,87				
486 training segments	DMI 0,67	0,74	0,79	0,80	0,80	21	0,63	0,70	0,75	0,77	0,76	21
	DM2 0,88	0,83					0,84	0,81				
	DM3 0,92	0,76					0,89	0,71				
60 s	DFI 0,24	0,93					0,21	0,89				
243 training segments	DMI 0,19	0,26	0,47	0,51	0,52	15	0,17	0,25	0,44	0,49	0,52	15
	DM2 0,48	0,52					0,43	0,47				
	DM3 0,97	0,35					0,94	0,34				
Start&Stop	DFI 0,78	0,78					0,74	0,73				
7562 training segments	DMI 0,83	0,91	0,89	0,89	0,85	192	0,79	0,91	0,85	0,87	0,82	192
	DM2 0,96	0,96					0,91	0,94				
	DM3 0,99	0,92					0,95	0,88				

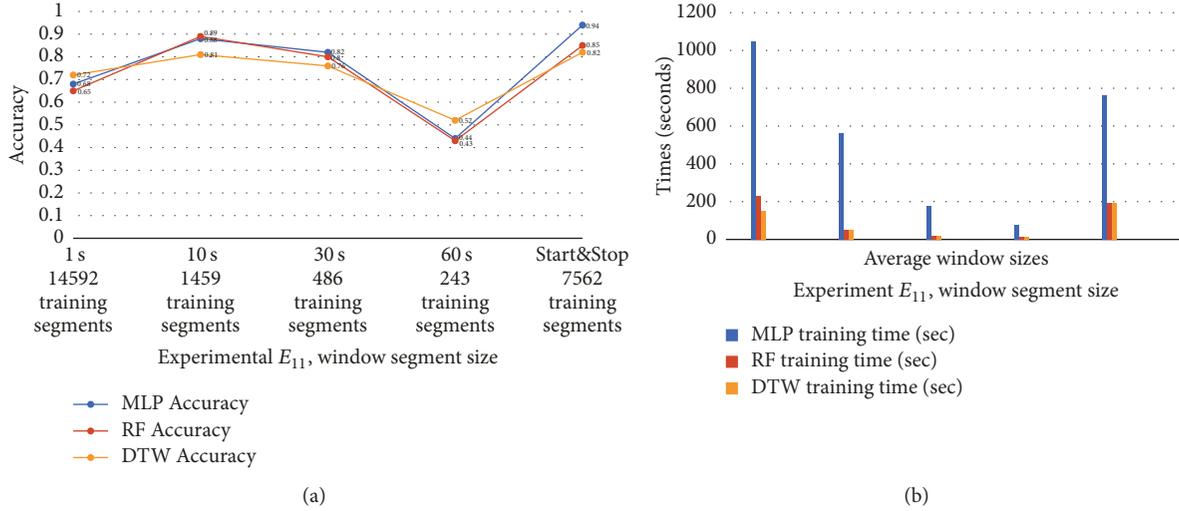


FIGURE 5: E_{11} , trend of accuracy (a) and training times (b) evaluated by using MLP, RF, and DTW classifiers for different windows sizes.

For what concerns experiments E_{12} and E_{13} , the results are reported in Tables 5 and 6. As we can see, data follows the same trends observed for E_{11} .

The following considerations can be done:

- (i) Driver classification on the dataset D_A , on a single path, gives better results for the same identification on dataset D_B that is performed on a fixed car but varying the path.
- (ii) The effectiveness obtained in E_{13} (i.e., regardless of the car and the path) is, as expected, lower than E_{11} and E_{12} effectiveness. This is important to estimate the quality of classification for applications where the path or the car is fixed. In those cases, a RF or a DTW approach could be adopted since precision and recall are higher and could be acceptable (leading to faster training times). Conversely, MLP approach remains the best classifier in our experimentation in terms of classification quality, and it is the best one to choose for the general case.

Results for precision and recall of the driver gender identification experiments (E_{14} and E_{15}) for different size windows are reported, respectively, in Tables 7 and 8.

The tables show that even if the best results have been obtained for the threshold-based segmentation, the fixed windows of sixty seconds provide quite reasonable results with a much more reduced training time. This means that, for applications more sensitive to training time, it could be preferred.

Table 9 shows the results of experiment E_2 . It reports, for each segmentation choice, the results (precision, recall, accuracy, and training times) of the runs on the MLP classifier for path kind identification among three classes (highway, city street, and dirt road). Precision and recall are evaluated for each class (the path kind in this context) and as averages. The best results are also obtained in this case for threshold-based segmentation on the MLP. In particular, the threshold-based segmentation provides the best accuracy of 0.94 (which is

also the best accuracy obtained for path kind identification) whereas, for fixed windows segmentation, the best value of accuracy was 0.75 (obtained for the fixed windows size of 30 s). Fixed windows segments too small (around 1 second) and too wide (more than 60 seconds) provided very bad results limiting useful windows sizes in the range (10 s, 60 s).

Tables 10 and 11, respectively, report results of experiments E_{31} and E_{32} . They report, for each segmentation choice, the results for MLP classifier in terms of precision, recall, accuracy, and training times. The evaluated instance is the familiarity detection and it is labeled as “Owner” and “Not Owner.” Precision and recall are evaluated for each class and as averages.

Threshold-based segmentation on the MLP confirms to be the best classifier. For E_{31} , the best value for accuracy (which was also the best achieved overall accuracy for familiarity detection) was 0.97. The performance trend among fixed windows sizes is consistent with other classifiers. For experiment E_{32} , the best value of accuracy is 0.9 showing that when car, driver, and path can change, the classifier has worse performance.

Finally, even if it is not possible to directly compare the obtained results with the results obtained in the work discussed in Section 2 (each approach is tested on a different dataset), we can observe that the obtained precision rate is very encouraging. However, we reach a precision rate equal to 0.99 with respect to the precision rate of 0.961 obtained in [18] (it is the best precision described in related work).

We conclude this section reporting the trend of accuracy metric for all the experiments and all the segments. Figure 6 highlights that the adoption of threshold-based segmentation improves with respect to using fixed windows for all the groups of features.

6.4. Threats to Validity. In this section the main threats to the validity of our research are discussed. Construct validity represents the quality of choices about the particular forms of the variables (i.e., the choice of outcome measure or the

TABLE 5: E_{12} , precision, recall, accuracy, and training time evaluated by using MLP for different windows sizes.

Average window size	Driver	Driver recall	Driver precision	Overall precision	Overall recall	Accuracy	Training time (sec)
1 s 12960 training segments	DF1	0,56	0,58	0,48	0,50	0,52	3268
	DF2	0,58	0,51				
	DF3	0,51	0,52				
	DF4	0,7	0,68				
	DF5	0,6	0,4				
	DM1	0,54	0,51				
	DM2	0,21	0,67				
	DM3	0,6	0,5				
	DM4	0,55	0,5				
	DM5	0,45	0,67				
10 s 1296 training segments	DF1	0,67	0,58	0,63	0,47	0,61	302
	DF2	0,66	0,54				
	DF3	0,71	0,58				
	DF4	0,72	0,61				
	DF5	0,62	0,7				
	DM1	0,57	0,58				
	DM2	0,4	0,45				
	DM3	0,63	0,41				
	DM4	0,59	0,38				
	DM5	0,76	0,45				
30 s 432 training segments	DF1	0,71	0,77	0,74	0,74	0,76	112
	DF2	0,73	0,75				
	DF3	0,76	0,79				
	DF4	0,74	0,71				
	DF5	0,77	0,75				
	DM1	0,79	0,78				
	DM2	0,7	0,7				
	DM3	0,81	0,67				
	DM4	0,73	0,79				
	DM5	0,67	0,7				
60 s 216 training segments	DF1	0,84	0,95	0,89	0,90	0,95	56
	DF2	0,92	0,91				
	DF3	0,89	0,89				
	DF4	0,91	0,92				
	DF5	0,83	0,91				
	DM1	0,89	0,84				
	DM2	0,91	0,84				
	DM3	0,9	0,95				
	DM4	0,92	0,92				
	DM5	0,93	0,88				
Start&Stop 7562 training segments	DF1	0,89	0,99	0,95	0,94	0,95	1430
	DF2	0,99	0,96				
	DF3	0,9	0,91				
	DF4	1	0,99				
	DF5	0,81	0,92				
	DM1	1	1				
	DM2	0,93	0,84				
	DM3	0,94	0,99				
	DM4	0,97	0,99				
	DM5	1	0,89				

TABLE 6: E_{13} , precision, recall, accuracy, and training time evaluated by using MLP for different windows sizes.

Average window size	Driver	Driver recall	Driver precision	Overall precision	Overall recall	Accuracy	Training time (sec)
1 s 14592 training segments	DF1	0,65	0,55	0,53	0,56	0,58	4320
	DM1	0,45	0,54				
	DM2	0,49	0,6				
	DM3	0,52	0,53				
10 s 1459 training segments	DF1	0,56	0,61	0,58	0,57	0,63	458
	DM1	0,58	0,53				
	DM2	0,62	0,55				
	DM3	0,55	0,59				
30 s 486 training segments	DF1	0,62	0,66	0,65	0,67	0,68	178
	DM1	0,64	0,69				
	DM2	0,66	0,62				
	DM3	0,68	0,71				
60 s 243 training segments	DF1	0,86	0,78	0,84	0,83	0,82	88
	DM1	0,83	0,81				
	DM2	0,81	0,83				
	DM3	0,86	0,88				
Start&Stop 7562 training segments	DF1	0,94	0,81	0,91	0,92	0,92	1740
	DM1	0,84	0,93				
	DM2	0,92	0,97				
	DM3	0,93	0,95				

TABLE 7: E_{14} , precision, recall, accuracy, and training time evaluated by using MLP for different windows sizes.

Average window size	Ownership	Precision genre	Recall genre	Overall precision	Overall recall	Accuracy	Training time (sec)
1 s 14592 training segments	FEMALE	0,44	0,45	0,46	0,43	0,49	4750
	MALE	0,48	0,41				
10 s 1459 training segments	FEMALE	0,52	0,55	0,52	0,58	0,55	578
	MALE	0,51	0,61				
30 s 486 training segments	FEMALE	0,65	0,69	0,67	0,70	0,71	216
	MALE	0,68	0,71				
60 s 243 training segments	FEMALE	0,71	0,78	0,69	0,79	0,74	132
	MALE	0,67	0,8				
Start&Stop 7562 training segments	FEMALE	0,97	0,99	0,98	0,97	0,97	1789
	MALE	0,98	0,95				

TABLE 8: E_{15} , precision, recall, accuracy, and training time evaluated by using MLP for different windows sizes.

Average window size	Ownership	Precision genre	Recall genre	Overall precision	Overall recall	Accuracy	Training time (sec)
1 s 12960 training segments	FEMALE	0,44	0,45	0,46	0,43	0,51	3450
	MALE	0,48	0,41				
10 s 1296 training segments	FEMALE	0,52	0,55	0,52	0,58	0,56	369
	MALE	0,51	0,61				
30 s 432 training segments	FEMALE	0,65	0,69	0,67	0,70	0,72	124
	MALE	0,68	0,71				
60 s 216 training segments	FEMALE	0,71	0,78	0,69	0,79	0,79	76
	MALE	0,67	0,8				
Start&Stop 5920 training segments	FEMALE	0,93	0,94	0,92	0,93	0,91	1345
	MALE	0,92	0,91				

TABLE 9: E_2 , precision, recall, accuracy, and training time evaluated by using MLP for different windows sizes.

Average window size	Path kind	Path kind recall	Path kind precision	Average path kind precision	Average path kind recall	Overall accuracy	Training time (sec)
1 s 14592 training segments	Highway	0,56	0,67	0,54	0,62	0,6	3858
	City street	0,57	0,61				
	Dirt road	0,49	0,59				
10 s 1459 training segments	Highway	0,76	0,78	0,71	0,71	0,69	542
	City street	0,72	0,7				
	Dirt road	0,64	0,66				
30 s 486 training segments	Highway	0,78	0,72	0,77	0,76	0,75	204
	City street	0,73	0,77				
	Dirt road	0,81	0,78				
60 s 243 training segments	Highway	0,55	0,67	0,55	0,63	0,59	122
	City street	0,59	0,6				
	Dirt road	0,51	0,63				
Start&Stop 7562 training segments	Highway	0,92	0,93	0,95	0,94	0,94	1542
	City Street	0,97	0,94				
	Dirt road	0,95	0,95				

TABLE 10: E_{31} , precision, recall, accuracy, and training time evaluated by using MLP for different windows sizes.

Average window size	Ownership	Ownership recall	Ownership precision	Overall precision	Overall recall	Accuracy	Training time (sec)
1 s 14592 training segments	OWNER	0,54	0,55	0,56	0,54	0,55	4750
	Not Owner	0,58	0,53				
10 s 1459 training segments	Owner	0,65	0,58	0,64	0,61	0,66	578
	Not Owner	0,62	0,64				
30 s 486 training segments	Owner	0,67	0,66	0,70	0,70	0,72	216
	Not Owner	0,72	0,74				
60 s 243 training segments	Owner	0,62	0,58	0,58	0,57	0,6	132
	Not Owner	0,54	0,56				
Start&Stop 7562 training segments	Owner	0,96	0,98	0,96	0,97	0,97	1789
	Not Owner	0,95	0,96				

TABLE 11: E_{32} , precision, recall, accuracy, and training time evaluated by using MLP for different windows sizes.

Average window size	Ownership	Ownership recall	Ownership precision	Overall precision	Overall recall	Accuracy	Training time (sec)
1 s 12960 training segments	Owner	0,4	0,45	0,41	0,43	0,4	3450
	Not Owner	0,42	0,41				
10 s 1296 training segments	Owner	0,5	0,55	0,49	0,58	0,45	369
	Not Owner	0,48	0,61				
30 s 432 training segments	Owner	0,6	0,69	0,61	0,70	0,68	124
	Not Owner	0,62	0,71				
60 s 216 training segments	Owner	0,49	0,51	0,50	0,60	0,62	76
	Not Owner	0,51	0,68				
Start&Stop 5920 training segments	Owner	0,8	0,9	0,92	0,93	0,9	1345
	Not Owner	0,88	0,92				

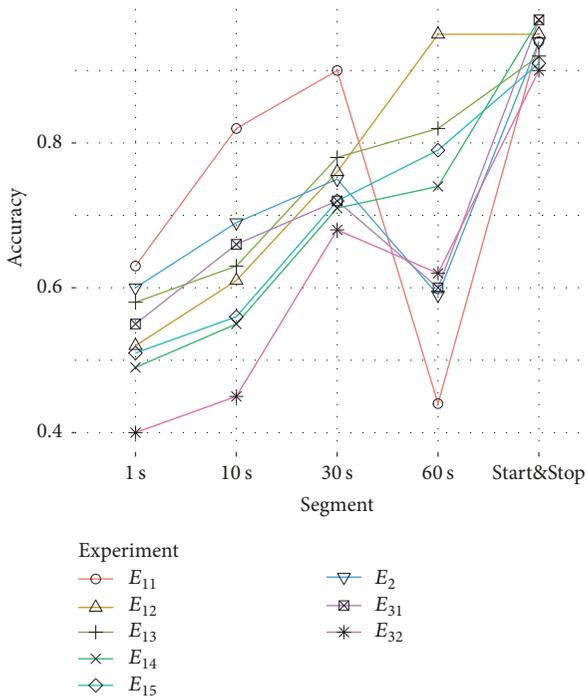


FIGURE 6: Accuracy trend over segments and experiments.

choice of treatment). They concern the relationship between theory and observation. In our proposal, some problems can be introduced by the hypothesis guessing of all the involved drivers. Drivers can know or guess the desired end-result, and they can change their behavior. This risk is mitigated by omitting the scope of the study and what are the monitored features to the drivers. Moreover, a bias in experimental design can be introduced by the path knowledge. If some drivers well know the tested path they can have a different behavior if compared to the drivers that never ride along that path. This risk is mitigated by training all the drivers on the paths so that their knowledge is similar.

Internal validity is concerned with the possibility that some factors would be more suitable for the proposed features to perform classification. To exclude this eventuality, we performed a specific feature selection step studying correlation and independence for all features available in OBD II standard. Moreover, in order to best validate the training of the classifier, we adopted a k -fold cross-validation.

Conclusion validity regards the degree to which the conclusions we state about the relationship (between the treatment and the outcome) are reasonable.

Threats to external validity concern the generalization of our findings. Of course, replication on further projects to confirm or contradict the obtained results is always desirable.

7. Conclusion

This paper proposes an approach to identify the driver, the familiarity of the driver with the vehicle, and the kind of the road based on the study of the behavior of a person during

the driving. It is based on the assumptions that a proper set of behavioral features can be used to (i) recognize different drivers by capturing their different driving style; (ii) detect their familiarity with the car; and (iii) detect the road kind on which they are driving (among several types). Based on these assumptions, we extracted, using a monitoring system placed in the cars, an effective set of features whose samples are sent to a time-series classification approach. The classifier exploits a supervised learning approach and is based on an MLP network; its performances were compared with classic decision tree classifiers. The proposed time-series classifier has been proved to be effective at identifying the driver, the driver genre, and the road kind after being trained on the proposed set of behavioral features. Specifically, the approach has been evaluated with eight experiments on three datasets made from real data logged on four cars driven by ten drivers in the Naples area. Each experiment allows exploring a specific aspect of the proposed research questions, evaluating the effectiveness of the proposed approach to

- (i) identify the driver regardless of the car but fixing the path;
- (ii) identify the driver regardless of the path but fixing the car;
- (iii) identify the driver regardless of both car and path;
- (iv) identify the driver gender regardless of the car but fixing the path;
- (v) identify the driver gender regardless of the path but fixing the car;
- (vi) identify the path kind fixing the car but regardless of the path and the driver;
- (vii) identify the driver familiarity with the vehicle fixing the car but regardless of the path and the driver;
- (viii) identify the driver familiarity with the vehicle when car, driver, and path can change.

The obtained results show high accuracy for all the performed experiments. In particular, the proposed approach is very effective in identifying driver. The best accuracy (0.97) is obtained to identify the gender of the driver regardless of the car but fixing the path. Good accuracy is also obtained in all the other experiments aiming to perform driver identification (the accuracy value is never less than 0.92). Looking for the driver genre identification (it is evaluated regardless of the car but fixing the path), the proposed approach shows an accuracy of 0.91 revealing that man and woman have different driving style. Moreover, the proposed approach can be also used to identify the road kind (highway, city street, and dirt road) by fixing the car but regardless of the path and the driver. The obtained accuracy value is, in this case, equal to 0.94. Effective results are also obtained in the detection of driver familiarity. Here we have a higher accuracy (0.97) in identifying the driver familiarity when the vehicle is fixed. A slightly lower accuracy (0.91) is obtained when all cars, drivers, and paths can change. Further experimentation in this general case should be performed to see if the size and the number of hidden layers of a single network and the number of networks positively influence the resulting accuracy.

Finally, we also compare the proposed MLP classifier with tree classic decision classifiers. The obtained results show how even if all the classifiers are characterized by high values of accuracy, the best performances are obtained by using the proposed ensemble MLP classifier. Training times however also show that our ensemble classifier is the slowest during the learning phase. This means that our approach is perfect for applications that do not require real-time identification of changing drivers (e.g., the owner can perform continuous training on its car and can be detected with very high levels of accuracy). As future work, we are extending the study adding a behavioral characterization of the driver using both formal approaches (using model checking) and fuzzy rule extraction from the example dataset. This not only allows performing the identification of driver and paths but also will provide an explanation of how the identified driver is behaving for predefined classes (e.g., polluting driver, aggressive driver, and cheap driver). Moreover, the evaluation can be extended to a higher number of drivers, cars, and paths. Finally, the application of the proposed approach in the road kind identification can be further explored considering a more accurate road classification model.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been partially supported by H2020 EU-funded projects NeCS and C3ISP and EIT-Digital Project HII and PRIN “Governing Adaptive and Unplanned Systems of Systems” and the EU Project CyberSure 734815.

References

- [1] K. Igarashi, C. Miyajima, K. Itou, K. Takeda, F. Itakura, and H. Abut, “Biometric identification using driving behavioral signals,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, pp. 65–68, Taipei, Taiwan.
- [2] M. Okamoto, S. Otani, Y. Kaitani, and K. Uchida, “Identification of driver operations with extraction of driving primitives,” in *Proceedings of the 2011 20th IEEE International Conference on Control Applications, CCA 2011*, pp. 338–344, USA, September 2011.
- [3] M. Itoh, D. Yokoyama, M. Toyoda, and M. Kitsuregawa, “Visual interface for exploring caution spots from vehicle recorder big data,” in *Proceedings of the 3rd IEEE International Conference on Big Data, IEEE Big Data 2015*, pp. 776–784, USA, November 2015.
- [4] A. Wahab, C. Quek, C. K. Tan, and K. Takeda, “Driving profile modeling and recognition based on soft computing approach,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 20, no. 4, pp. 563–582, 2009.
- [5] F. Martinelli, F. Mercaldo, A. Orlando, V. Nardone, A. Santone, and A. K. Sangaiyah, “Human behavior characterization for driving style recognition in vehicle system,” *Computers & Electrical Engineering*, 2018.
- [6] J. Daugman, “How iris recognition works,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, 2004.
- [7] S. González, C. M. Travieso, J. B. Alonso, and M. A. Ferrer, “Automatic biometric identification system by hand geometry,” in *Proceedings of the 37th Annual International Carnahan Conference on Security Technology*, pp. 281–284, twm, October 2003.
- [8] D. A. Reynolds, “An overview of automatic speaker recognition technology,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. IV-4072–IV-4075, Orlando, FL, USA, May 2002.
- [9] S. Ribarić, D. Ribarić, and N. Pavešić, “Multimodal biometric user-identification system for network-based applications,” *IEE Proceedings Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 409–416, 2003.
- [10] R. W. Frischholz and U. Dieckmann, “BioID: A multimodal biometric identification system,” *The Computer Journal*, vol. 33, no. 2, pp. 64–68, 2000.
- [11] T. Wakita, K. Ozawa, C. Miyajima et al., “Driver identification using driving behavior signals,” *IEICE Transaction on Information and Systems*, vol. E89-D, no. 3, pp. 1188–1194, 2006.
- [12] X. Zhang, X. Zhao, and J. Rong, “A study of individual characteristics of driving behavior based on hidden markov model,” *Sensors & Transducers Journal*, vol. 167, no. 3, pp. 194–202, 2014.
- [13] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, “Automobile driver fingerprinting,” in *Proceedings on Privacy Enhancing Technologies*, vol. 2016, pp. 34–50, 2016.
- [14] M. Van Ly, S. Martin, and M. M. Trivedi, “Driver classification and driving style recognition using inertial sensors,” in *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, IEEE IV 2013*, pp. 1040–1045, Australia, June 2013.
- [15] C. Miyajima, Y. Nishiwaki, K. Ozawa et al., “Driver modeling based on driving behavior and its evaluation in driver identification,” *Proceedings of the IEEE*, vol. 95, no. 2, pp. 427–437, 2007.
- [16] Y. Nishiwaki, K. Ozawa, T. Wakita, C. Miyajima, K. Itou, and K. Takeda, “Driver identification based on spectral analysis of driving behavioral signals,” in *Advances for In-Vehicle and Mobile Systems*, pp. 25–34, Springer, 2007.
- [17] S. Choi, J. Kim, D. Kwak, P. Angkititrakul, and J. H. Hansen, “Analysis and classification of driver behavior using in-vehicle can-bus information,” *Biennial Workshop on DSP for In-Vehicle and Mobile Systems*, pp. 17–19, 2007.
- [18] B. I. Kwak, J. Woo, and H. K. Kim, “Know your master: Driver profiling-based anti-theft method,” in *Proceedings of the 14th Annual Conference on Privacy, Security and Trust, PST 2016*, pp. 211–218, New Zealand, December 2016.
- [19] G. Kedar-Dongarkar and M. Das, “Driver classification for optimization of energy usage in a vehicle,” in *Proceedings of the Conference on Systems Engineering Research, CSER 2012*, pp. 388–393, USA, March 2012.
- [20] X. Meng, K. K. Lee, and Y. Xu, “Human driving behavior recognition based on hidden Markov models,” in *Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics, ROBIO 2006*, pp. 274–279, China, December 2006.
- [21] P. Esling and C. Agon, “Time-series data mining,” *ACM Computing Surveys*, vol. 45, no. 1, article 12, 2012.
- [22] M. L. Bernardi, M. Cimitile, F. Martinelli, and F. Mercaldo, “A time series classification approach to game bot detection,” in *Proceedings of the 7th International Conference on Web*

- Intelligence, Mining and Semantics, (WIMS '17)*, pp. 1–6, New York, NY, USA, June 2017.
- [23] N. K. Al-Salihy and T. Ibrikci, “Classifying breast cancer by using decision tree algorithms,” in *Proceedings of the the 6th International Conference (ICSCA '17)*, pp. 144–148, Bangkok, Thailand, February 2017.
- [24] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer New York Inc., New York, NY, USA, 2001.
- [25] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, “Time series classification using multi-channels deep convolutional neural networks,” in *International Conference on Web-Age Information Management*, F. Li, G. Li, S.-W. Hwang, B. Yao, and Z. Zhang, Eds., vol. 8485 of *Lecture Notes in Computer Science*, pp. 298–310, 2014.
- [26] M. Stone, “Cross-validatory choice and assessment of statistical predictions,” *Journal of the Royal Statistical Society*, vol. 36, pp. 111–147, 1974.
- [27] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 2 of *IJCAI95*, pp. 1137–1143, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, <http://dl.acm.org/citation.cfm>.
- [28] A. Nanopoulos, R. Alcock, and Y. Manolopoulos, “Feature-based classification of time-series data,” *International Journal of Computer Research*, vol. 10, pp. 49–61, 2001.
- [29] B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser, *Multivariate Time Series Classification by Combining Trend-Based and Value-Based Approximations*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [30] N. Sánchez-Maróño, A. Alonso-Betanzos, and M. Tombilla-Sanromán, *Filter Methods for Feature Selection – A Comparative Study*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.



Hindawi

Submit your manuscripts at
www.hindawi.com

