On Developing a Driver Identification Methodology Using In-Vehicle Data Recorders

Luis Moreira-Matias, *Member, IEEE*, and Haneen Farah

***Abstract*— Recently, cutting edge technologies to facilitate data collection have emerged on a large scale. One of the most prominent is the in-vehicle data recorder (IVDR). There are multiple ways to assign the IVDR's data to the different drivers who share the same vehicle. Irrespective of the level of sophistication, all of these technologies still suffer considerable limitations in their accuracy. The purpose of this paper is to propose a methodology, which can identify the driver of a given trip using historical trip-based data. To do so, an advanced machine learning pipeline is proposed. The main goal is to take advantage of highly available data—such as driver-labeled floating car data collected by a IVDR—to build a pattern-based algorithm able to identify the trip's driver category when its true identity is unknown. This stepwise process includes feature generation/selection, multiple heterogeneous explanatory models, and an ensemble approach (i.e., stacked generalization) to reduce their generalization error. Our goal is to provide an inexpensive alternative to existing driver identification technologies, which can serve as their complement and/or validation purposes. Experiments conducted over a real-world case study from Israel uncover the potential of this idea: it obtained an accuracy of ∼88% and Cohen's Kappa agreement score of ∼74%.**

***Index Terms*— Identification methods, in-vehicle data recorders, classification, stacked generalization, supervised learning, machine learning.**

I. Introduction

In THE last decade, significant advances have been made in sensoring and communication technologies. Such advances led to a considerable growth in the development and use of Intelligent Transportation Systems (ITS). One of the most widely used technologies to collect the driver behaviour is the In-Vehicle Data Recorder (IVDR). IVDR is a system able to measure the driver's actions, as well as the vehicle's movement performance. This data can help to develop control-focused ITS able to adapt to each driver's unique driving characteristics. Early usage of these systems was targeted towards fuel efficiency and vehicle location tracking purposes. Several researchers developed algorithms on driver behavioural characteristics to detect abnormal driving behaviours for automotive control applications [1]–[3].

Manuscript received October 28, 2015; revised March 31, 2016 and October 14, 2016; accepted December 2, 2016. Date of publication January 16, 2017; date of current version August 28, 2017. The Associate Editor for this paper was P. Ye.

L. Moreira-Matias is with NEC Laboratories Europe, 69115 Heidelberg, Germany (e-mail: luis.matias@neclab.eu).

H. Farah is with the Department of Transport and Planning, Delft University of Technology, 2600 GA Delft, The Netherlands (e-mail: h.farah@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2639361

Usually, a vehicle can be shared by multiple drivers (two or more). Thus, one of the main challenges in this context is the **driver's identification**. Several identification technologies exist. However, their usage does not solve this issue adequately (i.e. either due to their high cost and/or low accuracy levels). Consequently, it is critical to develop affordable methodologies able to deal with this problem.

The availability of Global Positioning System (GPS) data faced an explosive growth. This data is now widely used among transportation industry [4]–[6]. Recently, Wallace *et al.* [7] used GPS and OBDII logs (on-board diagnostics) from a preliminary sample of 100 trips and 4 drivers to test the potential of several features - such as *time of day, road choice, velocity* and *acceleration* - on distinguishing drivers of a shared vehicle. Hence, this sample size is insufficient to generalize significant conclusions on this topic. At the best of our knowledge, this is the only research work using this approach.

The main purpose of this study is to develop a methodology which can identify the driver for a given trip of interest using historical trip-based data. This data is not more than a high level aggregation of Floating Car Data (FCD) collected through an user-identified device, e.g. smartphone. Per opposition to most of existing IVDR-based driver identification methodologies, we aim to leverage in the **daily seasonalities** inherent to the human behavioural routines (e.g. wake up, go to the school and get back home for lunch). Moreover, the collection of this type of data is easier due to the amount of devices that already exist with capabilities of storing and/or broadcasting this type of data. Such availability makes the information about the driver's identification easier and cheaper to get than from any other IVDR device (e.g. video cameras).

To do it so, an advanced Machine Learning (ML) methodology is proposed. Its first step is to map the original feature space into a higher (and sparser) dimensional space which is then cleverly prunned to reduce the features' redundancy and, simultaneously, keep the relevant ones. Secondly, four baseline models are constructed using four distinct algorithms which are then combined in a Ensemble approach (i.e. Stacking). The main goal is to take advantage of the driver-labelled trip data to build a pattern-based algorithm able to identify the trip's driver where its true identity is unknown. Data collected from a particular case study in Israel [8] is used to validate the applicability of the proposed methodology. The contributions of this study are twofold: (1) the suggestion and exploitation of supervised learning approaches over trip-based data (e.g. Floating Car Data) to serve as complement to existing IVDR technologies through an exhaustive comparison of different

1524-9050 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

TABLE I
LIMITATIONS AND ADVANTAGES OF STATE-OF-THE-ART IVDR TECHNOLOGIES

Technology	Strengths	Weaknesses
	Physical Systems	
Dallas Keys / iButtons	Low Cost; Easy to implement [14];	Requires drivers' activation; Can be transferable among drivers [26];
RFID (Radio Frequency IDentification)	Low Cost; Easy to implement;	Requires to attach the tag directly to the driver and not to the vehicle; Drivers need to remember to wear it with them when driving;
	Sensing Systems	
In-vehicle video cameras	It does not require drivers' activation;	More expensive than the typical physical systems; Camera's lens can be highly sensitive to illumination conditions and driver orientation;
Biometric fingerprint systems	Relatively more accurate and reliable; Moderately cheap; Easy to use; Unique identification;	Highly intrusive [27];
Voice recognition	Quite accurate; Can be easily bypassed by using a pre-recorded voice of another driver;	Relatively expensive [26];
Iris technology	Quite accurate; Not that easily bypassed;	Drivers may need to remove eyewear; Scans may not work with people with cataract or glaucoma [26]; Relatively expensive [26];

types of off-the-shelf ML algorithms; (2) a straightforward ML methodology that can obtain high prediction accuracy in this task, independently of the dependence structure in place between the driver category and the remaining explanatory variables.

This paper is structured as follows: next section presents a comprehensive literature review about the topic. Section III presents a description of the real-world case study used to evaluate this method. The fourth Section introduces the proposed ML methodology in its diverse steps namely: data sampling, feature generation and selection, base learners and a Stacking approach. This is followed by an Experimental section, which describes and discusses in detail our test bed and the obtained results. Finally, conclusions are drawn.

II. LITERATURE REVIEW

Earlier research such as the "100 cars naturalistic study" [9] and PROLOGUE [10] used IVDR to observe drivers' naturalistic driving behaviour. Later, its insights allowed to reduce risky behaviours by providing feedback to drivers or to their supervisors (e.g. parents, fleet managers in [11] and [12]). One of IVDR's drawbacks is the driver identification. Farah *et al.* [8], [12] requested all members of a participating family to identify themselves at the beginning of each trip using Dallas keys (personal magnetic identification keys). However, only 78% of these trips end up being driver-labelled.

The following section provides an overview of the main IVDR technologies, as well as a comparison of their related weaknesses and strengths.

A. Identification Technologies

The existing identification technologies can be classified into two main categories: Physical or Sensing systems.

1) Physical Systems:

- **Dallas Keys/iButtons** are personal magnetic identification keys (chip-based data carrier) which were used in [8], [10], and [12].
- **iRFID (Radio Frequency IDentification)** technology is based on the use of radio waves to read and capture information stored on tags attached to persons, vehicles and/or other objects.

2) Sensing Systems:

- **In-vehicle video cameras** [13]. The most sophisticated technologies on this research line are the Apple iPhoto and the Google Picasa [14], which are based on face detection and identification. In the context of driving, this technology takes a single snapshot of the driver's face at the trip's start to identify him/her.
- **Biometric fingerprint systems** [15]. Upon vehicle start, drivers need to identify themselves by a pre-authorized fingerprint.
- **Voice recognition and iris technology** [16]. The voice recognition is done through an in-vehicle microphone combined with a biometric speech identification software. Iris technology relies on two basic types of eye scans: iris scanning and retinal scanning.

B. Shortcomings of Existing Identification Technologies

Each of the previously mentioned technology has its own strengths and weaknesses. Table I summarizes them.

Reference [17] considered several driver identification methods including key fobs or entry codes. However, both still require a prior activation. Other options which were considered include the use of wearable devices [18] or mobile applications [19]. Yet, drivers are obliged to carry these devices on.

The approach more closely related with our own is the In-Vehicle Video Camera. Typically, the identification of the drivers using their face [20] goes through a ML pipeline and therefore, the two problems synergies since they share the same goal (i.e. driver identification) with identical approaches - even from a very high level perspective. In fact, pattern recognition from video cameras installed in road vehicles is a wide studied topic in ITS community - namely by providing relevant insights for Advanced Driver Assistance Systems such as alarms for relevant hazards in the road, pedestrians or even identification of road signs [21].

Typically, this approach is also divided on two stages: feature engineering/selection and model induction. Earlier landmark research on transforming faces' video frames into

suitable feature spaces were mainly focused into reducing its very high dimensionality through Principal Component Analysis (PCA) - a technique which is able to find a low-dimensional orthogonal linear projection which typically assures a representation of the majority of the variance in a smaller dimension [22]. Then, in the second stage, a simple classifier such as boosted trees (e.g. [23]) is put in place.

Common approaches to this problem suffer several limitations when compared to our own, namely: (i) they require an installation of a specific equipment (i.e. a video camera) in each vehicle with a setup specific for this purpose; (ii) they are typically designed to solve a much more complex problem of identifying each individual rather than its category, which results on target variable with a very high cardinality and consequently, a requirement of possessing a considerable large amount of samples to produce accurate models; (iii) its accuracy is highly dependent of the absence of *noise*, which can be introduced by variations on the illumination conditions and/or face orientation (see, for instance, [24]); (iv) the PCA-like approaches guarantee the absence of redundancy in the feature subset but not the presence of relevancy as they are commonly carried out on a completely unsupervised setting (i.e. without taking into account the target variable). Recent promising advances on Neural Networks community aimed to address this challenge by joining the two stages into a single learning process (e.g. Convolution Neural Networks [25]). Yet, per contrast with our approach, the typical application of this latter type of methodology discards the regularities of the human behavior as input to their feature space.

C. Main Contributions

The analysis performed throughout the previous section uncovered that the state-of-the-art technologies for driver's identification still have multiple limitations to overcome. They are related with the relationship between their identification accuracy and their cost - which are still far unbalanced in most of the cases. The main motivation for this paper is to provide a way of generalizing the driver's behavior regarding their **mobility seasonalities** - e.g., when and where they go along the day/week/month and/or patterns of vehicle usage such as parents at daylight and children at evening.

In this paper, we propose to leverage on inexpensive FCD acquired from each individual to **identify who is driving** the vehicle through an advanced and **highly** accurate ML pipeline. The key advantage of using FCD is that it can be collected and recorded by devices as simple as smartphones. The proposed ML methodology operates in two stages: (1) firstly, it transforms the original data into meaningful **explanatory non-redundant features**. This new feature space is then used to (2) build explanatory models about the driver's identification by **combining a series of models** learned by heterogeneous induction methods into an ensemble using a straightforward technique named Stacked Generalization [28].

The usage of GPS-enabled mobile phones and other individual identification mechanisms (e.g. transit smartcards [29]) to generalize mobility behaviour is not new (see, for instance, [30], [31]). However, at the best of our knowledge, this is the first large-scale work to propose an accurate methodology to generalize the behaviour of categories of driver's at a micro-level (such as a family role) using FCD standalone.

TABLE II
NUMBER OF TRIPS AND DRIVING HOURS BY CATEGORY

Category	No. of Trips	Driving Hours (h)
Young driver	108191	34074
Father	78963	31325
Mother	102120	33701
Other family members	20070	7187
Unidentified	87181	28489
Total	396525	134776

The next two sections detail our case study and the methodology that we proposed to carry out the abovementioned task.

III. CASE STUDY

The IVDR system used was the GreenRoad technology [8], [12]. It is a G-force based system which tracks all trips made by the vehicle and records the following information: 1) Trip start and end times; 2) Driver identification using Dallas keys; 3) Vehicle location; 4) Events of excessive manoeuvres defined by patterns of G-forces measured in the vehicle. This data was collected throughout one year.

A. Participants and Recruitment Process

A rolling recruitment procedure was carried out between July 2009 and November 2010. The data collection process was already in place throughout this period. 217 families participated in this study throughout an one year period. Participants received a monetary compensation of $\sim$ \$250. The recruitment process, the characteristics of the drivers and their families is described in detail in [8] and [12].

B. Data Collection

Table II contains a brief description of the data collected. It is possible to conclude that roughly 22% of the trips are unidentified. Such ratio constitutes a significant portion of the total data which should not be discarded at any case to carry out any data driven analysis, regardless its goal. When further analysing this ratio per family, it was found that many families present highly unbalanced identification ratios. In particular, 62% of the families had an identification ratio above 0.8, 25% had an identification ratio between 0.6-0.8. However, there are some families, 8%, that had relatively similar number of identified and unidentified trips (between 0.4-0.6), and others 5% that had an identification ratio between 0-0.4.

The set of variables which describe a given trip are detailed in Table III. As postprocessing task, four types of trips were defined: HH (home to home), which are trips that start and end in the area around home; HO (home to other), which are trips that start at home area toward a more distant location; OH (other to home), which are trips that start from a distant location toward the home area, and OO (other to other), are trips that start and end from locations distant from the home area. In this study, a home area is defined as the length of the radius of a circular area around the location of its exact

TABLE III
FEATURE DESCRIPTION

Variable	Type	Domain	Acronym
Family ID	Categorical	Unique ID for each family	VID
Weekday	Categorical	{SUN, MON, ..., SAT}	WD
Departure time	Numeric	$[0, 1440]$ (min.)	ToD
Trip duration	Numeric	$\mathbb{N}$ (min.)	Dur
Trip aggressiveness level	Categorical	{Moderate, Intermediate, High}	TSL
Solo or accompanied	Categorical	{Solo, Accompanied}	SON
Number of events (IVDR)	Numeric	$\mathbb{N}$ (count number)	Ev
Cluster ID of trip origin	Categorical	{1,2,3,...,263}	OrigClusID
Cluster ID of trip destination	Categorical	{1,2,3,...,263}	DestClusID
Cluster ID of home	Categorical	{1,2,3,...,80}	HomeCluster
Trip type	Categorical	HH (home to home); HO (home to other); OH (other to home); OO (other to other);	TripType
Previous category	Categorical	{Father, Mother, Young driver, Other}	PR

address. An algorithm was developed in [32] to define a specific radius of a circular area around the home location of each participating driver. The average radius across families in the dataset was 1034 ± 346 meters.

IV. METHODOLOGY

The regularities of the human behavior have been providing important advances in many transportation topics. Some successful examples on applying such insights are the passenger demand prediction problem [33], origin-destination matrices [34] or transit planning [35]. In this particular application, the authors intend to take advantage on such type of seasonal patterns and trends to address the driver identification failures presented by most of the state-of-the-art methodologies.

Let $T = \{t_1, t_2..., t_n\}$ denote the n trips recorded by the members of all participant families (i.e. *data samples*). Each trip can be expressed as a pair $t_i = (\vec{x_i}, y_i)$ where $\vec{x_i} \rightarrow X^N$ denotes a vector of N features while $y_i \in Y = \{'F', 'M', Y', 'O'\}$ denotes a target variable (i.e. driver category). The target has some sort of dependence of the feature vector which is *unknown*. Classification denotes the problem of estimate an approximation of such mathematical dependence levering on labelled training examples (i.e. a finite set of data samples for which the value of y_i is known) representative of the underlying joint distribution $p(y|\vec{x})$ (in a probabilistic definition):

$$\hat{f} : \vec{x_i}, \vec{\theta} \rightarrow Y \tag{1}$$

$$\hat{f}(\vec{x_i}, \vec{\theta}) = f(\vec{x_i}) = y_i, \forall \vec{x_i} \in X, y_i \in Y \tag{2}$$

where $f(\vec{x_i})$ denotes the true dependence between X and Y, $\hat{f}$ our estimated approximation of f while $\vec{\theta}$ represents a generic parameter set (which representation will naturally change along with the functional form of $\hat{f}$). $\hat{f}, \vec{\theta}$ have to be learned using the training examples in T in order to minimize some sort of loss function $L(y_i, \hat{y_i})$ - which expresses the classifier *risk* to misclassifying novel unlabelled samples. To know more about Classification, the reader can consult [36]. To carry out this task, we devised a stepwise methodology with two main parts: I) Preprocessing and II) Model Induction. The first one comprises the tasks necessary to prepare the dataset to apply any type of ML algorithms over it while the II) comprise the tasks necessary for estimate $\hat{f}, \vec{\theta}$ from a set of training examples. There are mainly three types of preprocessing tasks [37]: (I-A) *Sampling* - the selection of a subset of examples from the original training set; (I-B) *Feature Generation* - the definition of the values to be used at each feature and (I-C) *Feature Selection* - the selection of the features to be used by the ML algorithm. For the (II) Model Induction part, we decided to adopt a **Stacking** [28] approach - where (II-D) we build multiple distinct explanatory models from the original dataset (typically using different types of learners) which (II-E) outputs/predictions are then combined using another learner. An overview of the proposed methodology is depicted in Fig. 1 and formally explained throughout this Section.

A. Sampling

Firstly, we started by assuming statistical independence among the families. Thus, the original dataset was *splitted* into 216 datasets and the feature VID was eliminated. Consequently, one model is generated over each dataset/family.

Secondly, we simply removed from the study families without a minimum number of samples $n_{\min} \in \mathbb{N}$. This was done so we can not only guarantee that there is a fair comparison among different classifiers (where some may require large samples to do a good generalization effort), but as well as there is good chances of doing a good generalization effort.

It should be noted that not every family have four different categories of drivers (e.g. some families just have one father or no sons). Consequently, the problem may vary from a binary classification to a multi-class one from dataset to dataset. Similarly, datasets which have no variation on the target class (i.e. just one family member driving) were also discarded from the study. Finally, categories/labels which possess less than $N_{\min} \in \mathbb{N}$ samples were also removed from the respective dataset by removing such samples.

B. Feature Generation

From Table III, we can conclude that 9 out of the total 11 features are categorical. Many powerful ML classifiers are unable to deal with such type of features. Consequently, we use a common preprocessing technique to turn them into numerical ones: one-hot encoding (OHE) [38]. OHE consists into mapping each one of the feature's possible values into a binary features (1 for samples which have such value assigned; 0 for the remaining ones). Missing values were mapped as a binary feature as well, including for the numerical variables - where those were simply replaced by zero. This results on a high sparse multidimensional feature space for each dataset to be used with the base learners (see Section IV-D) where $\vec{x_i} \rightarrow \mathbb{R}^M : M >> N$ (i.e. $M \simeq 650$ for this case study). Additionally, we also scaled each one of the two numerical features by subtrating their sample-based mean and dividing them by their sample-based standard deviation (discarding the samples with missing data to compute such statistics). Finally, we leveraged on a domain expert to select a feature subset to be used on the meta level (see Section IV-E) of our II) Model

Induction Component: Weekday, Previous Category, Departure Time and Trip Duration (WD, PR, ToD and Dur, respectively).

C. Feature Selection

The last step created a highly sparse feature space that needs to be reduced before conducting any supervised learning algorithm over it. The majority of these new features are not relevant to determine the value of our target variable. To do it so, we firstly aim to remove nearly invariant features. We remove all the features which present a ratio between the two most frequent values of, at least, $\phi \in [0, 1]$ or that possess a subset of, at least, $\tau \in [0, 1]$ samples with the same value. Then, we remove high correlated features using the absolute coefficient of Pearson Correlation and a maximum threshold of $r_{\max}$. Finally, we try to assess the predictive power of each feature using a Supervised Feature Selection procedure. This task is highly important on several dimensions of a ML such as to reduce computational time needed to learn an explanatory model or to avoid overfitting irrelevant aspects in the data. These techniques can be categorized into two types: classifier-dependent (such as Wrappers and Embedded methods) and classifier-independent (Filters) [39], [40].

Here, we propose to use a Filter method known as Permutation Importance or Mean Decreasing Accuracy (MDA) - commonly used with Random Forests (RF) (see Section IV-D.3 to know more about this method). MDA assigns a score to each feature by evaluating the impact of removing the association between each feature and the target. In pratice, the procedure operates as follows: 1) the error of each tree is assessed using the Out-of-Bag data - the data that was not part of the bootstrap used to build that tree; 2) The same procedure is done after permuting individually each feature out of the trees. 3) The differences for each pair (feature j,tree i), i.e. $v_{i,j}$ can then be used to compute MDA as follows

$$\mu_{MDA_j} = \frac{1}{\rho}\sum_{i=1}^{\rho} v_{i,j}\,;\, \sigma_{MDA} = \sqrt{\frac{1}{\rho}\sum_{i=1}^{\rho}(v_i - \mu_{MDA})^2} \tag{3}$$

$$MDA_j = \frac{\mu_{MDA_j}}{\sigma_{MDA_j}} \tag{4}$$

where ρ denotes the number of trees hyperparameter used in Random Forests. 4) The MDA is normalized/scaled into a new score $NMDA_j \in [0, 1]$ for each feature j and they are selected based on a minimum importance threshold $\text{NMDA}_{\min}$. As the feature selection process is made without considering the model that will be used later on, it is considered a filter approach. Note that even if there is a classifier involved in the process (i.e. learned by RF), the error rate/counts of the classifier are not used directly in the MDA computation - per contrast with the Wrapper methods, which typically search the space of feature subsets with some optimization procedure where the cost/utility depends on the error produced by a given classifier [39]. MDA operates as a sort of statistical test to assess if the nodes constructed by a feature j are useful or not for each individual tree. The main advantage of MDA facing many other Filter methods (such as minimal-redundancy-maximal-relevancy [41]) is that it considers the effects on redundancy reduction around the target variable of a given feature on sample subsets generated by partial domains of other features (i.e. as the one generated by decision tree nodes). To know more about feature importance measures using RF, the reader can consult [40].

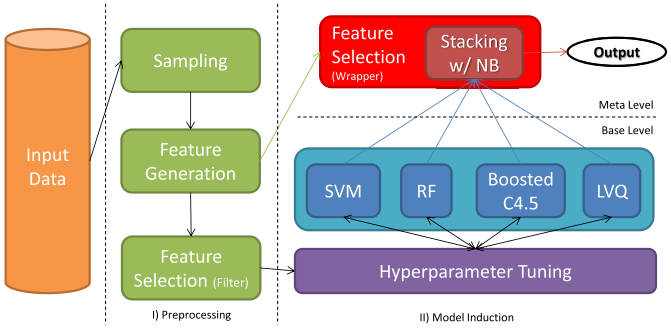

Fig. 1. Illustration of the proposed ML methodology.

D. Base Supervised Learners

As Fig. 1 points, we used four base learners: a baseline method - Learning Vector Quantization (LVQ) [42] - and three powerful off-the-shelf learners: Boosted C4.5 [43], Random Forests [44] and Support Vector Machines [45]. These methods were used in parallel to construct four distinct explanatory models over the training examples.

All these methods have hyperparameters whose values must be tuned for each dataset before learning any model from it. Typical approach includes a time consuming heuristic of testing a manual edited grid of possible values over some sort of cross validation (CV) procedure. As alternative, we used a popular optimization procedure: Random Search [46].

The *modus operandi* of these methods is briefly explained throughout this Section. An interested reader should refer to the references of each method for a formal explanations.

1) Learning Vector Quantization (LVQ): LVQ was selected to be our baseline method. LVQ is a faster alternative to k Nearest Neighbours (kNN) algorithm. kNN is *hanged* up to our entire training set to find the k nearest neighbours of each test sample from it accordingly to some distance metric (e.g. Euclidean). In alternative, LVQ learns a set of M (i.e. hyperparameter) *codebook* vectors which contain N values. These vectors are supposed to represent *neighbourhoods*. To train such vectors, we start by passing one sample by our *codebooks*. The *codebooks* with target similar to the input sample are updated where as each feature value is *moved closer* (i.e. using delta rule from Stochastic Gradient Descent) to the feature values of the given instance. The procedure is repeated to all the samples multiple times (i.e. epochs) till a convergence criteria (e.g. ϵ minimum threshold for the error reduction from epoch to epoch) is met. Predictions are operated the same way than kNN (i.e. using k as hyperparameter) but over the M learned *codebooks*.

Similarly to kNN, LVQ also suffers from the *tie* problem (where there are multiple codebooks at the same distance). To avoid such ties, we perturbed the feature's input to LVQ

by adding them a very small amount of white noise defined as $x_{i,j} + \epsilon : \epsilon \sim \mathcal{N}(0, \sigma_{\mathrm{LVQ}})$.

2) Boosted C4.5: C4.5 is one of the most well-known algorithms to learn decision trees for Classification. It builds its decision tree by recursively selecting the most informative attributes with respect to their capacity of reducing entropy around the target variable (i.e. Information Gain (IG)). C4.5 build maximum depth trees by stopping only when a *pure* leaf is found (i.e. a leaf with samples of the same class). Then, a "bottom-up" pruning stage takes place where the near leaf nodes are statistically tested to be replaced by a single leaf, avoiding irrelevant decision nodes with a low support and the consequent overfitting effects. Additionally, C4.5 can also build rule sets and it also allows the trees to be built with samples with different weights (which affect the IG calculation as well the splitting points). This characteristic allows its combination with an Ensemble technique known as Boosting [47], where a weak classifier can be improved by iteratively learning other classifiers on the top of it where the samples that contribute more to the generalization error are given an higher weight. Finally, at each iteration, a feature selection technique named as winnowing [48] is often employed. The type of model (i.e. rules/trees; $model$), the number of boosting iterations (i.e. $trials \in \mathbb{N}$) and the $win = \{0, 1\}$ are the main hyperparameters to be tuned in this method.

3) Random Forests: RF [44] is an ensemble method based on classification and regression trees (CART [49]). Several trees are grown by randomly choosing a set of candidate predictors at every node for a sample of the data and then producing the split by choosing the best splitter available. RF combines this with a random selection of samples (i.e. bootstraps) to train each tree to then produce decisions by averaging all trees outputs (i.e. bagging). RF's hyperparameters are (i) the number of randomly selected predictors used at each split $mtry$ and (ii) the number of grown trees $ntree$.

4) Support Vector Machines: SVM are binary classifiers that perform their task by constructing hyperplanes in a multidimensional space able to separate instances either linearly or non-linearly. In ϵ-SVM, these hyperplanes are constructed in a way to ensure the largest minimum distance to the training examples. This distance (ϵ) is denominated as *margin*. To allow examples to be in the margin or to be misclassified in order to increase the classifier robustness to noise, slack variables $\xi_i >= 0$ are often introduced. The optimization problem becomes as follows:

$$\arg\min_{w,b} \frac{||w||^2}{2} + C \times \sum_{i=1}^{N} \xi_i \tag{5}$$

where $C \in \mathbb{N}^+$ is a hyperparameter that sets the relative importance of maximizing the margin and minimizing the amount of slack. Kernels are typically used in SVMs to map the data points into higher dimensional feature space where the classification problem is likely to be linearly separable. Typical kernels include polynomial and radial basis functions. Hereby, we use the latter one, which possesses an hyperparameter σ_{SVM} that controls the smoothness of the Gaussian kernel. In order to adapt SVMs to multiclass problems, we take a *one-against-all* approach where one model is built per class.

5) Hyperparameter Tuning Using Random Search: Grid Search is a typical tool for ML hyperparameter tuning tasks. It exhaustively tests all the hyperparameter combinations specified from an user grid of values. Hence, it requires a high computational effort. A valid alternative is Random Search [46]. It consists on conducting independent draws from a uniform distribution using the same configuration space as the one defined by a regular grid. This approach only evaluates a random subsample of grid points - a global hyperparameter named gp - and presents similar results on an efficient manner.

E. Stacked Generalization

Stacked Generalization (i) [28] is an ensemble method that consists on using the prediction outputs of multiple ML models learned over a given training set as features input to another (meta)-learning algorithm that can learn dependences among those predictions and the true output. One of the first *Stacking* methods were proposed in [50], where the Stacking algorithm was Logistic Regression - thus, consequently, the resulting meta-model consisted on a linear combination of the base learners outputs with respect to the logarithm of their odd ratios. Breiman observed in [50] (ii) that many of these base learners are often excluded from the final decision meta-model. Latter research on the meta-learning (i.e. ML methods that relate the characteristics of the problems with the performance of the methods) community pointed out that (iii) the inclusion of base features on the meta-learning process can help to reduce the generalization error further [51].

Leveraged on these three base ideas, we propose a Stacking method as follows: (1) a new meta feature space is created, containing the predictions of the four base learners plus a four additional base features: WD, PR, ToD and Dur. (2) a feature selection stage takes place to ensure which is the best subset of meta-features to use with each family dataset. This step is known to be important to avoid either under and overfitting on the learning stage at meta level [52]. We do it so by taking a well-known wrapper method - Recursive Feature Elimination (RFE) [53]. A typical implementation into two stages process: evaluate the generalization error over a CV process in the training set using the current feature subset; rank the features with respect to their relevance and eliminate the last. The process is repeated for a pool of (small) cardinalities of the feature space. The feature subset with lowest error is used in the final model. (iii) A Naive Bayes (NB) classifier was used as meta learner due to its simplicity and low computational requirements. It can be (probabilistically) defined as

$$\arg\max_{y_i} \; p(y_i|\vec{x}) = p(y) \sum_{j=1}^{N} p(x_j|y_i), \forall y_i \in Y \tag{6}$$

As NB just admit categorical features, ToD and Dur were discretized using equal-frequency histograms. The experiments conducted to evaluate this methodology are described in the following Section.

TABLE IV
GENERAL HYPERPARAMETER SETTING

	Value	Description
$n_{\min}$	500	minimum number of samples/trips to consider a family;
$N_{\min}$	10	minimum admissible sample size of a given class label;
ϕ	1%	minimum admissible ratio between the 2nd/1st more frequent values of a relevant feature;
τ	95%	maximum admissible ratio of invariant values of a relevant feature;
$r_{\max}$	0.8	maximum admissible Pearson correlation coefficient between two relevant features;
$\text{NMDA}_{\min.}$	2.5%	minimum Normalized Mean Decreasing Accuracy of a relevant feature;
σ_{LVQ}	10^{-5}	standard deviation of the Gaussian noise feature perturbations used as preprocessing for LVQ;
gp	60	number of iterations of Random Search;

TABLE V
TUNED METHOD-SPECIFIC HYPERPARAMETERS

	Method	Description
M	LVQ	number of codebooks/neighbourhoods trained;
k	LVQ	number of neighbours used in test phase;
win	BoC4.5	Binary Flag to use or not winnowing [48]
$trials$	BoC4.5	number of boosting trials/iterations;
$model$	BoC4.5	Model type: decision tree or rule set;
$mtry$	RF	cardinality of the feature subset to be selected at each decision node;
C	SVM	misclassification budget to maximize the margin;
σ_{SVM}	SVM	variance of the Gaussians to be used on the kernel transformations;

V. EXPERIMENTS

This section starts by introducing the Experimental Setup followed by our experiments, passing by the used evaluation metrics to then point out the obtained results as well as a brief discussion on their insights.

A. Experimental Setup

The experiments were conducted using the R Software [54]. This methodology possess two types of hyperparameters: general and model-specific. They are briefly described in Tables IV and V, respectively. The first ones were user-defined while the second ones were tuned for each dataset as described in Section IV-D.5 using a stratified 5-fold CV process with 3 repetitions. The ML methods LVQ, BoC4.5, RF, SVM and NB were used following the implementations of the R packages `class`, `C50`, `randomForest`, `kernlab` and `RWeka`, respectively, while RFE used the base implementation included in the `caret` package. The non-tuned methods hyperparameters followed its defaults for these packages with exception of the number of trees in RF - which was 1000. RFE used 1 series of a stratified 5-fold CV process.

B. Evaluation

From the preprocessing described in Sections IV-A,IV-B, we obtained 196 datasets (196 out of the initial 213 families described in Section III). We designed two evaluation test-beds: A) *Hold-Out* (HO), where our datasets were divided on two stratified partitions of 70%/30% for training and testing tasks, respectively; B) one series of a stratified 10-fold cross validation (10CV). Two evaluation metrics were employed:

TABLE VI
EXPERIMENTAL RESULTS

Metric	*Testbed*	NB_STA	BOC4.5	SVM
ACC	**HO**	$\mathbf{86.87\% \pm 8.25}$	$86.55\% \pm 8.30$	$86.32\% \pm 8.47$
	10CV	$\mathbf{87.75\% \pm 8.09}$	$87.24\% \pm 8.38$	$86.98\% \pm 8.52$
κ	**HO**	$\mathbf{71.86\% \pm 15.53}$	$70.14\% \pm 17.36$	$69.69\% \pm 17.08$
	10CV	$\mathbf{73.38\% \pm 16.58}$	$71.22\% \pm 18.91$	$70.89\% \pm 18.42$
$\mathbf{ACC_2}$	**HO**	$\mathbf{90.01\% \pm 8.48}$	$89.59\% \pm 8.15$	$89.35\% \pm 8.68$
	10CV	$\mathbf{90.38\% \pm 8.30}$	$90.14\% \pm 8.30$	$89.66\% \pm 8.76$
κ_2	**HO**	$\mathbf{72.38\% \pm 20.27}$	$67.04\% \pm 25.50$	$69.05\% \pm 23.62$
	10CV	$\mathbf{72.85\% \pm 21.38}$	$68.23\% \pm 26.53$	$69.57\% \pm 24.45$
$\mathbf{ACC_*}$	**HO**	$\mathbf{84.41\% \pm 7.71}$	$84.11\% \pm 7.60$	$83.54\% \pm 8.12$
	10CV	$\mathbf{85.51\% \pm 7.60}$	$84.97\% \pm 7.98$	$84.36\% \pm 8.03$
κ_*	**HO**	$\mathbf{71.96\% \pm 12.30}$	$70.82\% \pm 12.45$	$69.61\% \pm 12.77$
	10CV	$\mathbf{73.65\% \pm 12.70}$	$72.39\% \pm 13.58$	$70.91\% \pm 14.02$
Metric	*Testbed*	**RF**	**LVQ**	
ACC	**HO**	$\mathbf{87.23\% \pm 8.00}$	$82.83\% \pm 9.89$	
	10CV	$\mathbf{87.76\% \pm 8.02}$	$83.63\% \pm 9.95$	
κ	**HO**	$\mathbf{71.97\% \pm 16.22}$	$60.17\% \pm 21.78$	
	10CV	$72.92\% \pm 17.32$	$62.06\% \pm 22.25$	
$\mathbf{ACC_2}$	**HO**	$\mathbf{90.18\% \pm 8.22}$	$86.66\% \pm 10.09$	
	10CV	$\mathbf{90.39\% \pm 8.26}$	$87.20\% \pm 10.36$	
κ_2	**HO**	$\mathbf{71.53\% \pm 22.61}$	$59.45\% \pm 27.09$	
	10CV	$71.70\% \pm 23.50$	$61.07\% \pm 28.00$	
$\mathbf{ACC_*}$	**HO**	$\mathbf{84.91\% \pm 7.47}$	$78.82\% \pm 9.53$	
	10CV	$\mathbf{85.62\% \pm 7.47}$	$80.24\% \pm 9.40$	
κ_*	**HO**	$\mathbf{72.76\% \pm 12.03}$	$58.91\% \pm 16.58$	
	10CV	$73.65\% \pm 12.65$	$61.71\% \pm 16.78$	

Accuracy and Cohen's Kappa. The first one is simply a normalized version of the success rate on classifying the examples in the test set, while the second one aims to express a likelihood of agreement between two raters (i.e. ground truth and a classifier). These metrics can be defined as follows:

$$ACC = \frac{1}{Z}\sum_{i=1}^{Z}\epsilon_i, \text{ where } \epsilon_i = \begin{cases} 1 & \text{if } y_i = \hat{y}_i \\ 0 & \text{if } y_i \neq \hat{y}_i \end{cases} \quad (7)$$

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (8)$$

where Z denotes the number of testing examples, p_e the hypothetical probability of chance agreement (computed over the marginals of each predictor/rater) and $p_o = ACC$. To know about more about κ, the reader is referred to [55]. An evaluation is also done for particular partitions of the data with respect to the number of family members/categories for binary, i.e. ACC_2, κ_2 and multiclass problems, i.e. ACC_*, κ_*, respectively.

C. Results

The results of our experiments are presented in four different axis: (i) Table VI resumes the results obtained by 4 base learners (RF, BoC4.5, SVM and LVQ) and IV-C) while (ii) Fig. 2 contains the boxplot of the κ scores of each one of the 4 baseline predictors + the Stacking method (i.e. NB_STA) on the two defined test beds. As the reported metric values are close to each other, a Statistical Test was conducted to assess if the 5 predictors are Statistically equivalent to each other for this task: the Friedman Test (following one of the methodologies proposed by [56]). This hypothesis was rejected with a p-value $\simeq 0$. Fig. 4 (iii) contains the respective Post-Hoc analysis of these test results, conducting paired tests among all the classifiers. Finally, (iv) Fig. 3 contains a Barplot with the number of datasets where each original feature (or one of its transformation) were included on the base modelling part (i.e. after performing the preprocessing stages described in Sections IV-A, IV-B and IV-C).

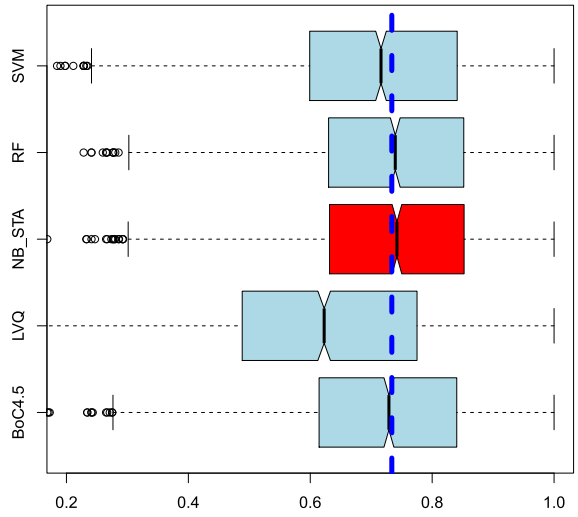

Fig. 2. Boxplots of the κ scores obtained from 10CV by all the ML methods on every dataset.

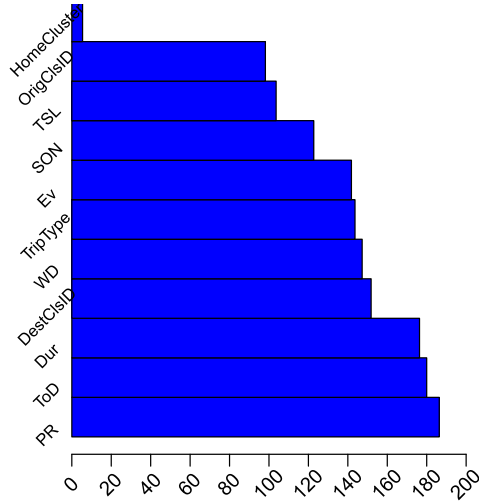

Fig. 3. Barplot reporting the feature usage frequency after performing the preprocessing tasks.

D. Discussion

All the 5 predictive methods present a reasonable predictive capacity, with an $ACC > 80\%$ for every of them. However, the kappa scores uncover the poor capacity of our baseline method (LVQ) - easily confirmed in Fig. 2. From observing both Table VI and Fig. 2, it is possible to conclude that all the remaining methods seem to be have a very similar performance either in HO or 10CV setups. Given such small differences, a statistical significance test such as the Friedman Test [57] is important to assess if such absence of difference means that the produced classifiers are *statistically* equivalent among them. Although the hypothesis was rejected in our experiments, we do not know exactly which are the methods responsible for such rejection. Such interpretability comes with the PostHoc Analysis depicted in Fig. 4: only two of the two-paired tests results on non-significant differences among the classifiers: BoC4.5-SVM and NB_STA-RF. In fact, the boxplot of the differences from the latter one is absolutely symmetric around 0.

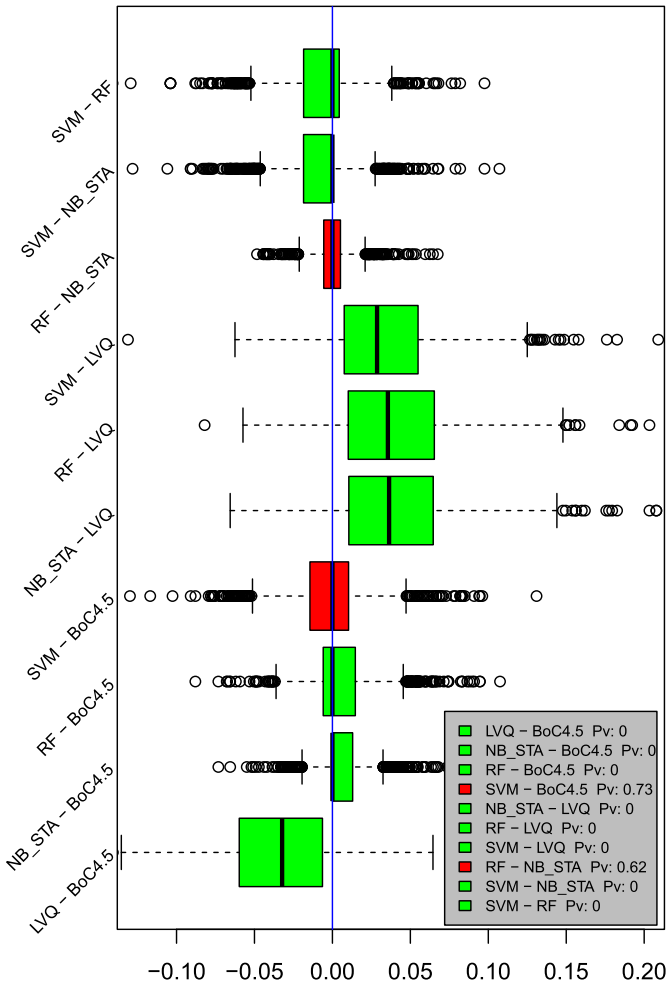

Fig. 4. Post-hoc analysis of the Friedman Rank Test for every pair of methods with their boxplots of the differences (ACC). Only two pairs have no statistically significant differences.

Even if NB_STA shows some improvements facing other methods with respect to κ-scores - specially in 10CV -, we can conclude that our Stacking approach fails on building models significantly stronger than the ones presented by the baseline methods. It is difficult to explain how we could change it without further experiments. However, hypothesis would pass through the following two steps: (i) add up additional features in the meta-level either by adding new types of baseline models (such as SVMs with different kernels or other powerful predictors such as Gradient Boosting machines [58]) or by including special features describing the model performance such as their confidence about the generalization error (as suggested in [51]); (ii) change the meta learner algorithm to other powerful Stacking model such as Logistic Model Trees [59] (used by the two top-winners of the famous Netflix competition, as described in [60]). Nevertheless, it is important to point that our schema (independently on using RF or a Stacking approach) **guarantees a high predictive capacity with a low generalization error** (illustrated by the evaluation obtained for the two proposed metrics). Our stacking approach returns an higher accuracy on binary classification problems (two family members) rather than in multiclass ones.

A solution to solve this issue may pass by add an intermediate step which adds up undersampling/oversampling techniques to deal with imbalanced classification training sets.

Regarding the features' usage, the Fig. 3 ends up pointing out that the most important factors on identifying the driver are the departure date/time, its duration, its destination as well as the previous driver. On the other hand, information about the home cluster (only relevant if family members live in different addresses), driving style or the trip origin were filtered out from $\sim$ 50% of the datasets even before the modelling part. **This observation goes along our hypothesis that the regularities of human behavior represent a set of meaningful explanatory factors around the driver's identification problem**. Nevertheless, note that this analysis do not take any interpretability about the ranking made by the Feature Selection method proposed in Section IV-C neither on the role that those features played (if any) on the modelling part.

VI. Conclusions

This study aimed at developing and testing an identification methodology using trip-based data. The main motivation to do it so is to complement the deficiencies of the existing IVDR technologies. For this purpose, an advanced ML methodology, including feature selection, multiple base learners and an Ensemble of them (i.e. Stacking), was proposed to take advantage of the underlying patterns of the human behavior. Trip-based data collected in Israel was used to test the usefulness of the proposed methodology.

The results of this paper provide a first glimpse on which are the most promising ML techniques for the applications of driver's identification. A high accuracy was achieved in predicting the driver category using basic trip information ($\sim$ 88%). Therefore, the authors believe that this methodology is worth to be further investigated in future studies when using IVDR or similar identification devices. It should be noted, however, that the proposed methodology is recommended to be used as a support method to the different identification technologies (e.g. Dallas Key, Face Identification) and not as a standalone methodology for driver identification. Furthermore, the assumption, in the proposed methodology, is that the training examples are reliable and trustworthy to a certain extent. Further research is required to overcome such limitations.

In order to carry out such future work, possible directions are proposed as follows: (1) testing the information gained by other trip features that were not included in this study, such as route choice and type of roads (rural, urban, suburban); (2) testing unsupervised learning approaches to derive the number of driver categories when it is not possible to know it *apriori*; (3) testing this methodology as a complement and/or a validation technique to the deficiencies of various identification technologies, such as the face recognition ones (e.g. [23]). In particular, it would be interesting to explore the feature space proportioned by the present approach can improve and/or complement the ones typically explored by computer vision approaches to this problem and vice-versa; (4) testing the proposed methodology on larger datasets and (5) explore additional ways of improving the prediction accuracy within the context of the present methodology by (5-a) improving the feature engineering at the meta learning level, (5-b) the feature selection at the base learning level (e.g. by including wrappers for each of the methods) or the ML algorithms in any of those modelling parts (e.g. to use Logistic Model Trees for the Stacked Generalization).

Acknowledgments

The authors would like to thank Dr. Tsippy Lotan from Or Yarok Association for providing the data of "The First Year Study" and Prof. Hans Van Lint from TU Delft for the opportunity to collaborate in this work.

References

[1] N. Lin, C. Zong, M. Tomizuka, P. Song, Z. Zhang, and G. Li, "An overview on study of identification of driver behavior characteristics for automotive control," *Math. Problems Eng.*, vol. 2014, Mar. 2014, Art. no. 569109.

[2] A. Silver and L. Lewis, "Automatic identification of a vehicle driver based on driving behavior," U.S. Patent 9 201 932, Dec. 1, 2015.

[3] K. J. Sanchez, A. S. Chan, M. R. Baker, M. Zettinger, B. Fields, and J. A. Nepomuceno, "Systems and methods to identify and profile a vehicle operator," U.S. Patent 8 738 523, May 27, 2014.

[4] R. Nunes, L. Moreira-Matias, and M. Ferreira, "Using exit time predictions to optimize self automated parking lots," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2012, pp. 302–307.

[5] L. Moreira-Matias, J. Gama, J. Mendes-Moreira, and J. F. de Sousa, "An incremental probabilistic model to predict bus bunching in real-time," in *Advances in Intelligent Data Analysis XIII*. Cham, Switzerland: Springer, 2014, pp. 227–238.

[6] J. Khiari, L. Moreira-Matias, V. Cerqueira, and O. Cats, "Automated setting of bus schedule coverage using unsupervised machine learning," in *Advances in Knowledge Discovery and Data Mining*. Cham, Switzerland: Springer, 2016, pp. 552–564.

[7] B. Wallace, R. Goubran, F. Knoefel, S. Marshall, and M. Porter, "Measuring variation in driving habits between drivers," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Jun. 2014, pp. 1–6.

[8] H. Farah *et al.*, "The first year of driving: Can an in-vehicle data recorder and parental involvement make it safer?" *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2327, pp. 26–33, 2013.

[9] T. A. Dingus *et al.*, "The 100-car naturalistic driving study, phase II—Results of the 100-car field experiment," U.S. Department of Transportation - NHTSA, USA, Tech. Rep. DOT HS 810 593, 2006.

[10] T. Lotan, G. Albert, T. Ben-Bassat, and D. Ganor, "Potential benefits of in-vehicle systems for understanding driver behaviour," European Comission, Brussels, Belgium, Tech. Rep. Grant 233597-Deliverable D3.2, 2010.

[11] C. G. Prato, T. Toledo, T. Lotan, and O. Taubman-Ben-Ari, "Modeling the behavior of novice young drivers during the first year after licensure," *Accident Anal. Prevention*, vol. 42, no. 2, pp. 480–486, 2010.

[12] H. Farah *et al.*, "Can providing feedback on driving behavior and training on parental vigilant care affect male teen drivers and their parents?" *Accident Anal. Prevention*, vol. 69, pp. 62–70, Aug. 2014.

[13] B. G. Simons-Morton *et al.*, "Crash and risky driving involvement among novice adolescent drivers and their parents," *Amer. J. Public health*, vol. 101, no. 12, pp. 2362–2367, 2011.

[14] D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. He, and C. Miao, "Learning to name faces: A multimodal learning scheme for search-based face annotation," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2013, pp. 443–452.

[15] V. L. Washington, "Vehicle security system including fingerprint and eyeball part identification," U.S. Patent 5 686 765, Nov. 11, 1997.

[16] P. Rosenzweig, A. Kochems, and A. Schwartz, "Biometric technologies: Security, legal, and policy implications," *Legal Memorandum*, vol. 12, pp. 110, 2004

[17] J. Thompson, M. Baldock, J. Mathias, and L. Wundersitz, "The benefits of measuring driving exposure using objective GPS-based methods and subjective self-report methods concurrently," in *Proc. ACRS*, 2013, pp. 1–11.

[18] P. Stopher, C. FitzGerald, and J. Zhang, "Search for a global positioning system device to measure person travel," *Transp. Res. C, Emerg. Technol.*, vol. 16, no. 3, pp. 350–369, 2008.

[19] C. Inbakaran and A. Kroen, "Travel surveys—Review of international survey methods," in *Proc. Australasian Transp. Res. Forum*, 2011, pp. 1–15.
[20] H. Ekenel, M. Fischer, H. Gao, L. Toth, and R. Stiefelhagen, "Face recognition for smart interactions," in *Proc. 8th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Sep. 2008, pp. 1–2.
[21] N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 920–939, Sep. 2011.
[22] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
[23] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
[24] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, "Face recognition using HOG–EBGM," *Pattern Recognit. Lett.*, vol. 29, no. 10, pp. 1537–1543, Jul. 2008.
[25] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proc. 5th ACM Int. Conf. Multimedia Retr.*, 2015, pp. 643–650.
[26] N. Lerner *et al.*, "An exploration of vehicle-based monitoring of novice teen drivers," Nat. Highway Traffic Safety, Washington, DC, USA, Final Rep. DOT HS 811 333, 2010.
[27] S. T. Bhosale and B. S. Sawant, "Security in E-banking via card less biometric ATMS," *Int. J. Adv. Technol. Eng. Res.*, vol. 2, no. 4, pp. 457–462, 2012.
[28] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.
[29] L. Moreira-Matias and O. Cats, "Towards an AVL-based demand estimation model," in *Proc. 95th Annu. Meeting Transp. Res. Board*, 2016, pp. 1–16, paper 16-2001.
[30] J. C. Herrera, D. B. Work, R. Herring, X. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via GPS-enabled mobile phones: The *Mobile Century* field experiment," *Transp. Res. C, Emerg. Technol.*, vol. 18, no. 4, pp. 568–583, Aug. 2010.
[31] X. Ma, Y.-J. Wu, Y. Wang, F. Chen, and J. Liu, "Mining smart card data for transit riders' travel patterns," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 1–12, Nov. 2013.
[32] O. Musicant and Y. Benjamini, "Driving patterns of novice drivers—A temporal spatial perspective," in *Proc. 91st Annu. Meeting Transp. Res. Board*, 2012, pp. 1–21, paper 12-1388.
[33] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "On predicting the taxi-passenger demand: A real-time approach," in *Progress in Artificial Intelligence* (Lecture Notes in Computer Science), vol. 8154. Berlin, Germany: Springer, 2013, pp. 54–65.
[34] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Time-evolving O-D matrix estimation using high-speed GPS data streams," *Expert Syst. Appl.*, vol. 44, pp. 275–288, Feb. 2016.
[35] S. M. Hassan, L. Moreira-Matias, J. Khiari, and O. Cats, "Feature selection issues in long-term travel time prediction," in *Proc. Int. Symp. Intell. Data Anal.*, 2016, pp. 98–109.
[36] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, no. 1, pp. 249–268, 2007.
[37] T. Reinartz, "A unifying view on instance selection," *Data Mining Knowl. Discovery*, vol. 6, no. 2, pp. 191–210, Apr. 2002.
[38] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2755–2763.
[39] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 27–66, Jan. 2012.
[40] B. Gregorutti, B. Michel, and P. Saint-Pierre, "Correlation and variable importance in random forests," *Statist. Comput.*, vol. 2013, pp. 1–20, Oct. 2013.
[41] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[42] T. Kohonen, "Learning vector quantization," in *Self-Organizing Maps*. Berlin, Germany: Springer, 1997, pp. 203–217.
[43] J. R. Quinlan, *C4.5: Programs for Machine Learning*, vol. 1. San Mateo, CA, USA: Morgan Kaufmann, 1993.
[44] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[45] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[46] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 281–305, 2012.
[47] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
[48] N. Littlestone, "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Mach. Learn.*, vol. 2, no. 4, pp. 285–318, Apr. 1988.
[49] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.
[50] L. Breiman, "Stacked regressions," *Mach. Learn.*, vol. 24, no. 1, pp. 49–64, 1996.
[51] L. Todorovski and S. Džeroski, "Combining classifiers with meta decision trees," *Mach. Learn.*, vol. 50, no. 3, pp. 223–249, 2003.
[52] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. De Sousa, "Ensemble approaches for regression: A survey," *ACM Comput. Surv.*, vol. 45, no. 1, 2012, Art. no. 10.
[53] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.
[54] R Core Team. (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. [Online]. Available: http://www.R-project.org
[55] M. Banerjee, M. Capozzoli, L. McSweeney, and D. Sinha, "Beyond kappa: A review of interrater agreement measures," *Can. J. Statist.*, vol. 27, no. 1, pp. 3–23, 1999.
[56] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[57] M. Friedman, "A comparison of alternative tests of significance for the problem of *m* rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, 1940.
[58] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.
[59] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Mach. Learn.*, vol. 59, nos. 1–2, pp. 161–205, May 2005.
[60] J. Sill, G. Takács, L. Mackey, and D. Lin. (2009). "Feature-weighted linear stacking." [Online]. Available: https://arxiv.org/abs/0911.0460

Luis Moreira-Matias (M'14) received the M.Sc. degree in informatics engineering and the Ph.D. degree in machine learning from University of Porto in 2009 and 2015, respectively.

He served in the Program Committee of multiple high-impact research venues, such as ECML/PKDD, the IEEE ITSC or TRB, among others. He was a recipient of an International Data Mining competition held during a research summer school at TU Dortmund in 2012.

He is currently a Senior Researcher with the Intelligent Transportation Systems Group, NEC Laboratories Europe, Heidelberg, Germany. He authored over 30 publications on his research interests, which include machine learning, intelligent public transports, and big data analytics applied to improve urban mobility in general.

Haneen Farah has been an Assistant Professor with the Transport and Planning Department, Faculty of Civil Engineering and Geosciences, Delft University of Technology, since 2014. She has been a Lecturer with different European institutes, such as Technion-Israel Institute of Technology and the KTH-Royal Institute of Technology, Stockholm, for over 10 years. She has experience in several European projects and COST Actions, such as COOEPRS, Cooperative Systems for Intelligent Road Safety, and MULTITUDE (Methods and Tools for Supporting the Use Calibration and Validation of Traffic Simulation Models). She is currently involved in several projects related to vehicles automation in relation to road design, road user behavior, and traffic safety, such as HFAuto (Human Factors of Automated Vehicles), WEPod (Realization of a Pilot with Two Self-Driving Vehicles), and STAD Project (Spatial and Transport Impacts of Automated Driving). Her research interests focuses on traffic and road safety, road user behavior modeling, road geometric design, and vehicle technology. Her research combines elements from the transport systems analysis, behavioral and human factors sciences, and econometrics. She has over 30 scientific papers published in leading refereed international journals, such as *Accident Analysis & Prevention*, *Transportation Research Part B*, *Transportation Research Part C*, and *Transportation Research Part F*.