

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/336272457>

# Traffic signal detection from in-vehicle GPS speed profiles using functional data analysis and machine learning

Article in *International Journal of Data Science and Analytics* · June 2020

DOI: 10.1007/s41060-019-00197-x

CITATIONS

4

READS

1,190

6 authors, including:



**Yann Meneroux**

Institut national de l'information géographique et forestière

9 PUBLICATIONS 12 CITATIONS

[SEE PROFILE](#)



**Guillaume Saint Pierre**

Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'am...

48 PUBLICATIONS 315 CITATIONS

[SEE PROFILE](#)



**Mohammad Ghasemi**

University of Tabriz

4 PUBLICATIONS 9 CITATIONS

[SEE PROFILE](#)



**Sebastien Mustiere**

Institut national de l'information géographique et forestière

80 PUBLICATIONS 789 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



ecoDriver [View project](#)



PHD thesis [View project](#)

# Traffic Signal Detection from in-vehicle GPS Speed Profiles using Functional Data Analysis and Machine Learning

Y. Méneroux · A. Le Guilcher · G. Saint Pierre · M. Ghasemi Hamed ·  
S. Mustière · O. Orfila

*International Journal of Data Science and Analytics* Received: 21 Dec 2018 / Accepted: 23 Sep 2019

**Abstract** The increasing availability of large-scale Global Positioning System (GPS) data stemming from in-vehicle embedded terminal devices enables the design of methods deriving road network cartographic information from drivers' recorded traces. Some machine learning approaches have been proposed in the past to train automatic road network map inference, and recently this approach has been successfully extended to infer road attributes as well, such as speed limitation or number of lanes. In this paper, we address the problem of detecting traffic signals from a set of vehicle speed profiles, under a classification perspective. Each data instance is a speed versus distance plot depicting over a hundred profiles on a 100-meter-long road span. We proposed three different ways of deriving features: the first one relies on the raw speed measurements; the second one uses image recognition techniques; and the third one is based on functional data analysis. We input them into most commonly used classification algorithms and a comparative analysis demonstrated that a functional description of speed profiles with wavelet transforms seems to outperform the other approaches with most of the tested classifiers. It also highlighted

that Random Forests yield an accurate detection of traffic signals, regardless of the chosen feature extraction method, while keeping a remarkably low confusion rate with stop signs.

**Keywords** Map Inference · GPS Speed Profiles · Functional Data Classification · Traffic Signal Detection

## 1 Introduction

### 1.1 Inferring road network

An exhaustive and accurate knowledge of road networks is a prerequisite to many applications such as route planning, traffic flow control or even risk management. Besides, roads are often considered as structural elements in maps, insofar as one can use them as a visual reference frame to get a quick grasp of one's current location. Accordingly, most mapping agencies consider road networks as a major cartographic theme, and put considerable effort in road data acquisition, update and quality review (Bonin 2002; Liu et al 2012; Zhang and Couloigner 2006).

This is traditionally done by photogrammetric restitution on couples of stereographic images, or directly by digitizing roads on aerial and satellite orthophotographs. Some approaches have been introduced to perform fully automatic road extraction, relying on multi-spectral and high resolution imagery (Bentabet et al 2003; Zhang and Couloigner 2006). These methods are effective for surveying a road network *ex-nihilo*, but most of the time, the updating process is bounded by the frequency of image release, which may turn out to be prohibitive when data currency is of utmost importance (Chen and Krumm 2010). Indeed aerial image

---

Y. Méneroux, A. Le Guilcher, S. Mustière  
Univ. Paris-Est, LASTIG MEIG, IGN, ENSG, F-94160 Saint-Mande, France  
E-mail: yann.meneroux@ign.fr Phone: +33 1 43 98 80 00 + 7125

G. Saint Pierre  
Cerema, Project-team STI, 1 avenue du Colonel Roche, F-31400, Toulouse, France

M. Ghasemi Hamed, O. Orfila  
French Institute of Science and Technology for Transport, Development and Networks, Laboratory for Vehicle Infrastructure Driver Interactions (IFSTTAR/LIVIC), 77, rue des Chantiers, 78000 Versailles, France

campaigns are typically conducted every several years, which is to be put in perspective with the fast evolution of cities. This is of course notwithstanding unavoidable field survey completions that may render the whole process slower and even more expensive (Zhang et al 2017). More recently, the *OpenStreetMap* project (Haklay and Weber 2008) aims at providing citizens with a platform for sharing geographic information, though with no quality guarantee and with large disparities in accuracy and completeness (Girres and Touya 2010; Liu et al 2012).

For the last few years, the availability of data stemming from Global Positioning System (GPS) embedded in mobile devices has given plenty of scope for technological developments in many fields. Indeed, as we are moving closer to having about as many cell-phone subscriptions as people on Earth (Lulli et al 2017), this new source of data is the most qualified to conduct studies where massive, crowdsourced and inexpensive information on individual mobility is required, though it is often acknowledged to be sometimes biased (Arai and Shibasaki 2013). As opposed to the traditional methods, *map inference* techniques aim at deriving cartographic information from traces recorded by in-vehicle GPS receivers (Biagioni and Eriksson 2012). Besides, while aerial images may not always be easily accessed in developing countries, or may only be available at a prohibitive cost, local fleets or collaborative transport smartphone applications, instead, produce large sets of GPS traces which could be used as input in map inference algorithms to offer a cheap surrogate for map construction.

A few years ago, some authors such as Liu et al (2012), introduced measures to assess the quality of maps produced by GPS traces, which opened the way for a full machine learning resolution of the problem (Biagioni and Eriksson 2012).

## 1.2 Inferring road attributes

Detailed road maps are of utmost importance for self-driving cars, whose trajectory planning system is essentially relying on accurate and up-to-date information. Embedded sensors are not always self-sufficient, and a typical solution to this issue is to rely on a prior map in order to identify beforehand the approximate location and the type of the object that should be detected (Lundgren et al 2014; Mu et al 2015). Sharing information between intelligent vehicles detections can therefore help maintaining accurate and up-to-date detailed maps. Such innovative cloud-based solutions are

currently under development among original equipment manufacturers (for example, the "static eHorizon" solution from Continental Automotive). Eco-driving, road safety (Chen et al 2016), vehicle-embedded driving assistance devices conception (Andrieu et al 2013) or accurate routing time estimation (Bonin 2002), are many other examples of fields where road maps need to be completed with up-to-date attributes.

Although geometry and topology are the base components of road maps, semantic information (*e.g.* road nature) and attributes (*e.g.* traffic calming infrastructure, speed limitation or number of lanes) are progressively getting captured in the scope of map inference (Van Winden 2014). For instance, Schroedl et al (2004) proposed a complete workflow relying on probe vehicles equipped with differential GPS to provide an accurate estimate of the road centerline and the number of lanes for the refinement of road map intersections. This idea has been extended to standard quality receiver by Chen and Krumm (2010), who modeled GPS traces on a road with multiple lanes as a Gaussian Mixture whose components are estimated with the Expectation-Maximization (EM) algorithm (Dempster et al 1977) to determine the most plausible number of lanes and their respective widths. In a similar way, Van Winden et al (2016) found that Support Vector Machines (SVM) (Vapnik 2013) and regression trees (Breiman et al 1984) are the most adequate algorithms for speed limit inference. In other approaches, GPS traces are combined with external sources of data, like in Li et al (2015) where Twitter data and SVM are used for an automatic mining of street names. Another work (Biljecki et al 2013) relies on an Expert System program to analyze GPS traces and geographic information data to infer users' transportation modes. This was extended by Endo et al (2016) and Dabiri and Heaslip (2018), where expert knowledge is replaced by deep neural networks (LeCun et al 1999). The outputs of these works may be a very useful source of information to provide roads with a set of attributes describing the accessibility by type of vehicle.

## 1.3 Inferring traffic signs

Surprisingly, very few research works address the issue of utilizing GPS data to infer punctual road infrastructure, such as traffic calming devices (traffic lights, stop signs, speed bumps...) or speed management devices (speed limits, speed enforcement cameras...). However, such detailed information is often missing in geographic databases produced by national mapping agencies (even

though it may be available at individual city scales, it is difficult to collect, especially in decentralized countries). Similarly, crowd-sourced project databases are usually not complete or detailed enough. For instance, *OpenStreetMap* contains 6746 traffic signal intersections in France (2018), which barely corresponds to 5% of the total number that is expected to be reported.

The most related work is certainly the one of Wang et al (2017), who used a set of GPS trajectories to detect and localize stop bars on each individual lane at crossings. Intersection spots are identified with entropy analysis of vehicle headings, then, in the second stage, the set of GPS stop points is modeled as a mixture of gaussians, and the EM algorithm is used to precisely estimate the position of the stop bar. The results are very accurate with a sub-meter level positional error. However, the proposed method assumes that all intersections are controlled by traffic signals, which means that it does not distinguish between traffic signals, stop signs and yield signs. Contrarily, with this approach, traffic signals located far from intersection nodes (*e.g.* traffic lights associated with pedestrian crossings) are not detected.

More recently Munoz-Organero et al (2018) used machine learning algorithms to detect in real-time several kinds of road infrastructures based on an analysis of speed and acceleration signals estimated from GPS positions of a single vehicle. Despite providing very good results, the performance scores clearly exhibit some limitation on traffic signal detection, with systematically lower precision and recall performance scores on almost all the tested classification algorithms, in comparison to street crossings and roundabouts. Moreover, a natural extension of this work would be to combine the information stemming from all vehicles which traveled on a specific road link to detect infrastructure elements. This was proposed by Zourlidou et al (2019), where predictions are performed on each individual trajectory based on its representation as spatial and temporal speed profiles. Subsequently, predicted labels are aggregated to form the overall prediction at the intersection level. However, we believe that machine learning should be able to deal with a set of trajectories and infer predictions based both on individual speed profiles and on their mutual variability. This is what we propose hereafter.

In this paper, we apply machine learning algorithms on in-vehicle GPS speed profiles to detect traffic signals. We believe that speed profiles, as defined by Andrieu et al (2013), are potentially more explicit than raw trajectories (*i.e.* sequences of timestamped geo-

graphic coordinates) and contain all the information needed to recognize different kinds of traffic calming infrastructures, provided that a sufficient number of traces are observed on the same road segment. However, as in any machine learning problem, the choice of explanatory variables has a significant impact on the prediction performance. We have applied 6 of the most commonly used algorithms to compare different modeling choices, depending on whether the set of speed profiles is considered as an image or as a functional dataset. These approaches lead to 2 different sets of explanatory variables, modeling the same reality. This has been completed with a third method, referred to as *direct approach* hereafter, which constitutes our baseline approach. Hence it amounts to train 18 models of classifiers (3 approaches times 6 algorithms) in similar conditions in order to compare their respective prediction performances on a validation sample. The goal of this paper is to estimate separately the fitness of each couple of one data description model and one machine learning algorithm; thus we do not investigate the interest of combining different sets of descriptors. As far as we know, we believe that this is the first attempt of using machine learning techniques for the detection of traffic signals on a collection of GPS speed curves.

The remaining of the paper is structured as follows: the dataset and its preparation are briefly described in the next section, while 3 different approaches along with the selected machine learning algorithms are detailed in section 3. After a summary of the performance indices, section 4 provides the results which are discussed in section 5. Eventually, section 6 concludes the paper.

## 2 Constitution of the dataset

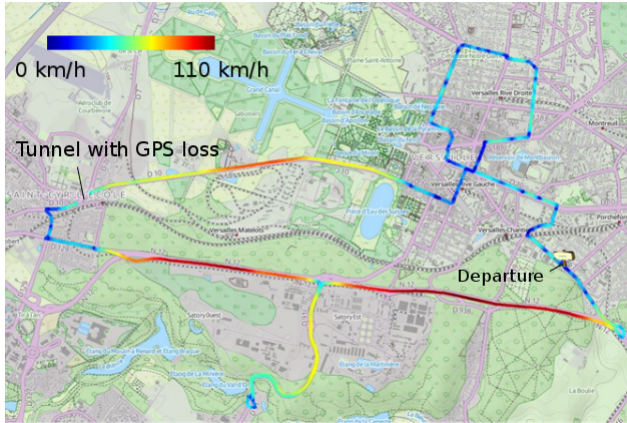
### 2.1 Data acquisition

GPS traces were collected for the EU-funded Eco-driver project<sup>1</sup> in 2014. A total of 30 different non-professional drivers recruited from the general public were requested to run a predefined 25-km-long circuit loop in the town of Versailles (France) and its outskirts. Each subject had to repeat the experiment (non consecutively) between 4 and 6 times, which amounts to a total of 170 trips. About half of them were recorded in normal driving condition, while the remaining trips were carried out with an eco-driving assistance system. The whole experimentation was conducted under the supervision of one of the authors.

---

<sup>1</sup> <http://www.ecodriver-project.eu/>

The experimentation route (depicted on Fig. 1) is driven clockwise in the upper loop, and anti-clockwise in the main loop. It mixes large urban conditions in the north with inter-urban and motorway conditions in the south. The whole trip takes between 45 minutes and 1 hour, depending on traffic conditions. Statistical representativeness of drivers and experiment modalities enable to assume naturalistic driving conditions.



**Fig. 1** Experimentation route. Average speed on the track is symbolized with rainbow color scale (from blue to red as speed increases).

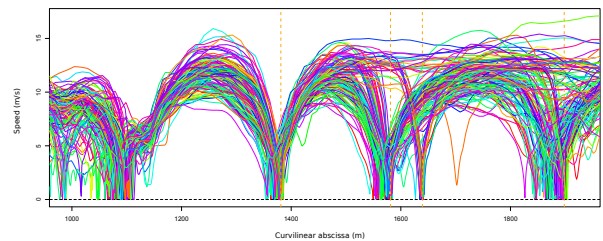
Two cars of the same model were used for the experiments. They were equipped with a data logger connected to both the vehicle CAN bus and the antenna of a Garmin GPS 16x LVC receiver. During each trip, 87 parameters were recorded, including general parameters such as position and its derivatives but also some more specific information like the remaining quantity of fuel or the steering wheel position. Most of these parameters were recorded at a 10 Hz sampling frequency, while GPS position was only measured at 1 Hz. Besides, starting times of all the sensors are usually different, leading to non-synchronized measurement cycles. For these reasons, data were post-processed in order to get a simpler file with one line by epoch, containing the corresponding timestamp and all the measured parameters. In all our study, 1 Hz data are used. It is important to note that speed measurements are not computed by positions finite difference as they are directly observed by Doppler measurement system integrated in the GPS (Kaplan and Hegarty 2005). This provides estimated speeds with one order of magnitude better accuracy ( $\pm 0.1$  m/s). It shall be noted however that not all modern in-vehicle receivers are able to provide such accurate speed, but as mentioned by Schroedl et al (2004), decreasing costs of sensor technology enables to assume

that in the future, cars will be undoubtedly equipped with instruments such as differential GPS, providing equivalent performances. Before any further processing, GPS longitude and latitude were projected into Lambert 93 planimetric coordinates.

To build ground truth, 235 infrastructure elements have been reported along the track, including 44 traffic lights, 5 stop signs, 9 speed bumps and 92 crosswalks. Each element has been geo-localized by pinpointing its position on Geoportail<sup>2</sup> ortho-images. Positional error is estimated to be less than 3 m.

## 2.2 Pre-processing

Speed profiles were mathematically modeled by Andrieu et al (2013) and are a classical tool to measure the impact of traffic calming devices (Moreno and García 2013). Let us denote by  $x \in \mathbb{R}$  the curvilinear abscissa of a vehicle along the track. Speed profile is defined as a function  $v : x \rightarrow v(x)$ , the instantaneous speed of the vehicle at location  $x$ . A collection of speed profiles is depicted on Fig. 2.

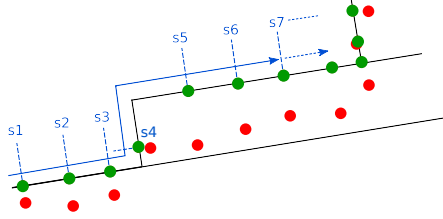


**Fig. 2** A set of speed profiles. Each color corresponds to an individual vehicle. Positions of traffic lights are depicted with vertical dashed lines.

Curvilinear abscissa theoretically correspond to the traveled distances provided that all vehicles followed exactly the same path. This assumption is obviously unrealistic, and the actual distance traveled by a car is most of the time strongly related to the driver's behavior. Moreover, the noise on the GPS receiver will be embodied by local expansions and contractions of speed curves. A way to deal with this issue is to map-match GPS points on a reference road network, as illustrated on Fig. 3. This operation enables a) to express

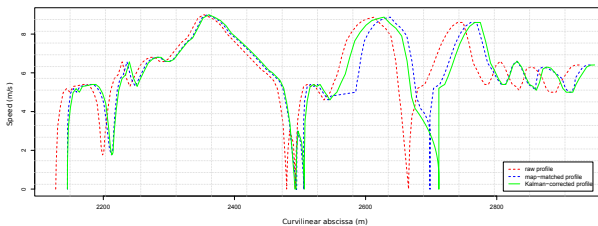
<sup>2</sup> French national web mapping service:  
<https://www.geoportail.gouv.fr/>

vehicle positions in a uni-dimensional reference coordinate system and b) to get more accurate positions (road network adds an external information for solving the localization problem). The second point is particularly important in the north-west part of the circuit, near Saint-Cyr l'École, where a tunnel obstructs GPS signal for approximately 200 m (Fig. 1).



**Fig. 3** Map-matching (green) of raw GPS points (red) on a reference road network (black lines) and curvilinear abscissa computation (blue dashed lines).

We performed map-matching on a reference road network with the method introduced by Newson and Krumm (2009). However, map-matching a smooth trajectory on a piecewise linear network may result in systematic artifacts which may turn out to be confusing for machine learning algorithms. For example, in Fig. 3, points 4 and 5 are much further apart after map-matching, resulting in an unnatural leap forward in the corresponding speed profile. Consequently, GPS positions sequence has been smoothed with a monotonic constrained Kalman filter (Tully et al 2011). This step enabled to produce more realistic speed profiles, as illustrated on Fig. 4, with Kalman processed profile mostly adjusted to the map-matched profile, except on the artifact around 2550 m where it has a shape more similar to the raw profile. Enforcing a monotonic constraint on curvilinear abscissas was motivated by the assumption that vehicles never moved backwards.



**Fig. 4** Kalman filtered speed curve (green line) compared to curves before (red dashed line) and after (blue dashed line) map-matching.

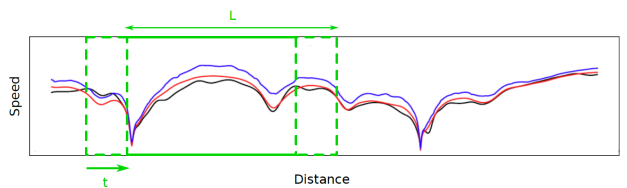
Eventually we computed functional box-plots (Sun and Genton 2011) on the set of curves to remove outliers (*i.e.* profiles whose behavior differs abnormally from central tendency, which turned out to correspond to sensor or logging failures during the experimentation).

As a result, 144 (over 170) profiles are remaining in the data, for a total experiment duration of 126 hours, and an approximate traveled distance of 3650 km.

### 2.3 Sliding window computation

Each window contains the 144 curves and has a length  $L$ . Two consecutive windows are separated by a distance  $t$ , chosen smaller than  $L$  so that windows are overlapping each other (Fig. 5). This artificial increase of the number of training data will enable to train algorithms to detect a traffic light, wherever it is located in the window. Obviously, since those new training data are deeply correlated with original ones, one should expect the gain in prediction performance to be incommensurate with the increase of the training sample size. Also, it will be important to take this correlation into account when separating training and validation datasets.

Each window stands for an individual instance, from which different approaches will extract a vector of features  $X \in \mathbb{R}^p$ . The dimension  $p$  of the vector depends as well on the selected approach. It remains to compute the target variable  $Y$ . Since we are facing a classification problem,  $Y$  takes its values in the binary set  $\mathcal{Y} = \{0, 1\}$ .

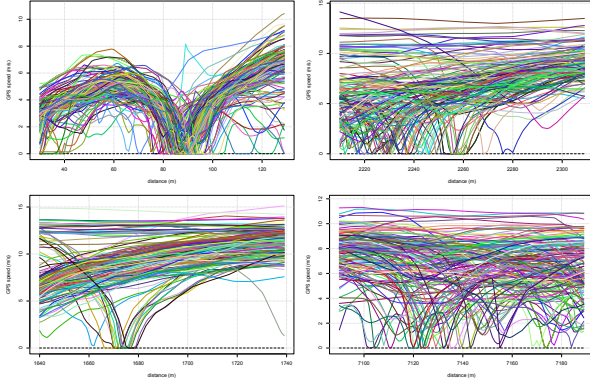


**Fig. 5** Sliding windows extraction on a set of 3 speed profiles. Each window covers a span  $L$  of the track and is a translation  $t$  of its immediate left neighbor.

For the experiment, we set  $L = 100$  m and  $t = 10$  m (90% overlap). This choice is mainly motivated by the fact that 100 m appeared to be a fair compromise to get sufficiently large window to *capture* as much traffic light signature as possible, and at the same time it enables to limit the risk of having two traffic lights in the same window. Besides, an overlap of 90% enables to get a dense cover of traffic light relative positions in the windows on the whole dataset.



With this method, a total of 2505 windows were extracted, including 402 (16%) positive samples. Four examples of sliding windows are depicted hereafter on Fig. 6. Note that the top two windows correspond to negative instances (a stop sign on the left and a crosswalk on the right), while bottom instances are positive.



**Fig. 6** Four examples of sliding windows : each color corresponds to an individual trip.

### 3 Methodology

In this section, we describe the methods used for extracting explanatory variables from sliding windows, then selected machine learning algorithms are briefly presented. Eventually, the training/test sample separation process is detailed.

#### 3.1 Data description

Even after the preprocessing phase, windows of GPS trajectories can not be immediately input into machine-learning algorithms. One needs a modeling approach to transform a set of curves in a vector of real features that will characterize each window for our algorithms. This modeling phase can greatly impact the performance of the process. For this reason, three different methods were experimented: 1) a straight-forward approach taking into account all speed measurements in a single ordered vector of features (referred to as *direct approach* hereafter). 2) A more traditional approach considering each sliding window as an image (*image approach*). 3) A specific approach dealing with speed profiles as continuous functions (*functional approach*). The first method was chosen out of the desire to evaluate the fitness of a model that considers, as much as possible, the data in their primitive form.

In the present section,  $X \in \mathbb{R}^p$  stands for a  $p$ -dimensional feature vector, describing a sliding window, whose response variable is denoted  $Y$ . The total number of speed profiles is noted  $N$ . Let us notice that our objective is to design an operational classifier, adequated to every GPS dataset, regardless of how many speed profiles it contains. This is important as the number of available traces on a given street is unknown beforehand, and training one algorithm for each possible size of dataset is not a reasonable solution. More formally, we state the following constraint:

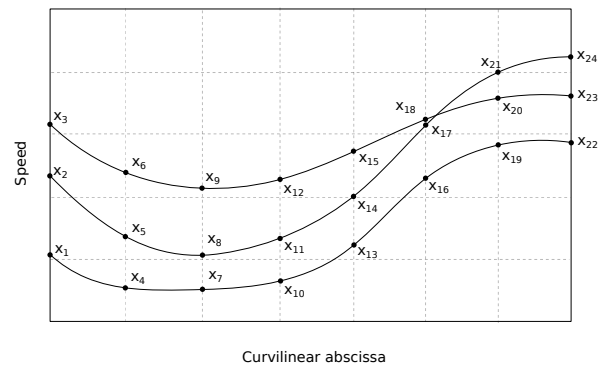
**Constraint 1.** *The structure of the explanatory vector must not depend on the number of available profiles.*

This constraint ensures that a unique classifier model is able to operate on every collection of curves. Obviously, for any consistent classifier, the expected prediction accuracy will increase with the number of vehicle traces recorded on a given road.

##### 3.1.1 Direct approach

The most direct and straight-forward approach lies in considering pointwise values of speed profiles as a finite real vector, whose dimension depends on a discretization step parameter.

For a discretization step of 1 m, the feature vector  $X$  is composed of  $p = 14400$  explanatory variables, each of them corresponding to one of the  $100 \times 144 = 14400$  speed measurements reported in a window.



**Fig. 7** An example of feature extraction with direct approach on 3 speed profiles.

In a given frame, the ordering of the variables is done using the lexicographical order on the timestamp-velocity couples. See Fig. 7 for an extraction of the ex-

planatory variables on a fictive sliding window with 3 speed profiles and discretization step set at  $1/7^{th}$  of total length  $L$ , hence generating  $8 \times 3 = 24$  features  $(X_1, X_2, \dots, X_{24})$ . This choice leads to the loss of the links between points of the same trajectories, but it guarantees that the distance function is not affected by an arbitrary ordering of trajectories.

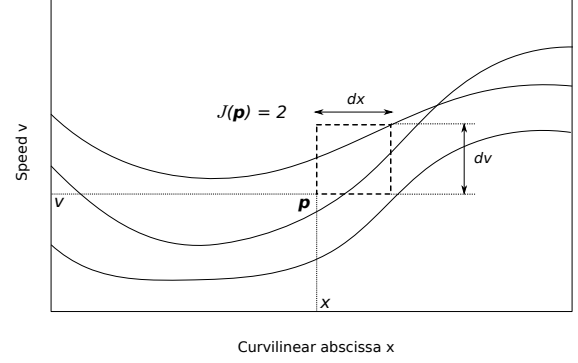
Moreover, the rearrangement inequality guarantees the following property: among the orderings abiding by the natural constraint that variables having the same timestamps share the same indices, this ordering always minimizes the distance between two frames. This property indicates that the distances we compute are at least consistent. As this ordering does not take the whole trajectories into account, but rather focus on the distribution of the velocities at each timestamp, it may be viewed as predominantly image-driven, even if it works on functional data.

Obviously, this approach does not abide by constraint 1, but it makes a useful basis to evaluate the performance of the other two models, which entail more complex modeling approaches

### 3.1.2 Image approach

This second approach is motivated by the fact that human recognition of a potential traffic light on graphics depicting speed curves is mostly visual and based on simple reference patterns (a traffic light is embodied by a null value of a fraction of curves on a short distance, while other curves do not seem to be much impacted). Consequently, we may believe that an image-oriented approach has some chance to be successful for solving the present classification task. To do so, we used the simple and efficient algorithm introduced by Ozuysal et al (2007), based on a computation of binary features randomly and evenly spread throughout the image. Despite its extreme succinctness, it has been proven to perform very well on complex image recognition tasks. The algorithm we present below is an adapted version, for the specific case of speed profile images.

Given a position  $\mathbf{p} = (x, v) \in [0, L] \times [0, v_{max}]$  in the window image, and a couple of dimension parameters  $(dx, dv)$ , let us define function  $\mathcal{J}$  such that  $\mathcal{J}(\mathbf{p}) \in \mathbb{N}$  returns the number of speed profiles intersecting a  $dx \times dv$  rectangular cell, whose lower left corner is positioned on  $\mathbf{p}$ . Figure 8 provides an example of computation of the function  $\mathcal{J}$  for a simple case with 3 curves.



**Fig. 8** Two curves intersecting cell positioned in  $(x, v)$ .

At the beginning of the experiment, a number  $p$  of position pairs  $(\mathbf{p}_j^1, \mathbf{p}_j^2)$  are chosen at random with a uniform sampling. Each feature  $X_j$  is then computed as the normalized difference of the number of curves intersecting cells positioned at  $\mathbf{p}_j^1$  and  $\mathbf{p}_j^2$ :

$$X_j = \frac{\mathcal{J}(\mathbf{p}_j^1) - \mathcal{J}(\mathbf{p}_j^2)}{N} \quad (1)$$

The normalizing factor  $N$  ensures that explanatory variables are insensitive to the total number of speed profiles. As a consequence, each feature extracted with this method corresponds to a difference of curves densities at two specific locations selected in the image.

It is important to note that even though positions are chosen at random, they are fixed during both the training and classification steps. Then each variable  $X_j$  corresponds to a specific couple of positions in the image, and this definition is the same for all sliding windows. For a sufficient number of features, the uniform sampling provides a complete and homogeneous cover of our image. Since this approach does not distinguish between individual curves, it automatically abides by constraint 1.

When the number of profiles is important, the extraction process may be time-consuming. However, the total computation time can be reduced by pre-arranging curves in a quad-tree structure, and then performing faster intersections research with each of the  $2p$  cells. Processing time gain is highly dependent on the quad-tree parameters, but with a quick trial and error settings, we divided the extraction time by 5 (from 4 hours to less than 45 minutes on our machine).



In our experiment, we set  $dx = 5$  m,  $dv = 5$  km/h and  $v_{max} = 80$  km/h (the model was evaluated for different values of  $v_{max}$  in the range  $[10; 130]$  km/h and we selected the value providing optimal results). A number  $p = 15000$  features were extracted.

### 3.1.3 Functional approach

The particularity of our classification task is that our data are inherently functional. Though it is always possible to consider a function as a  $\mathbb{R}^p$  vector of explanatory variables, regularly sampled along the abscissa, some research works have shown that this is seldom an optimal modeling for classification and regression problems. Indeed, generally, for functions whose behavior is not completely erratic, consecutive features will be significantly correlated, hence decreasing the performance of most classifiers. Some solutions rely on an irregular sampling of the function, in order to acquire more information on intervals where variations are the most important, and also on the contrary, saving memory where no meaningful signal has to be registered. However, this assumes that a unique and optimal sampling pattern can be found for all data, which is often impossible since high-variation intervals of functions have no reason to be colocated. For these reasons, a functional description of such data is often a better solution than a sequence of values taken in an arbitrary set of locations (Berlinet et al 2008; Ferraty and Vieu 2011).

A functional description of speed profiles aims at extracting a small number of meaningful parameters that optimally represent the set of functions we may be facing in the detection task. This is mathematically expressed by a basis in a functional space, with each data being described by its projection components on basis functions. There exist multiple choices of basis, and the most well-known is certainly the Fourier representation, specifically designed for periodic signals. Unfortunately, Fourier basis functions are not localized in space (or time), which makes them inappropriate in our classification task. On the reverse, Splines basis are completely localized, implying that information is *a priori* evenly distributed in all coefficients, then rendering impossible to reduce the number of features by series truncation. Between these two extremal solutions, wavelets are acknowledged to be localized both in space and frequency, making them a perfect compromise for functional classification problems (Berlinet et al 2008). For more information on basis expansions for functional data classification, refer to Gregorutti (2015).

Profile	Description
$f_1$	Pointwise mean
$f_2$	Pointwise standard deviation
$f_3$	Third standardized moment (skewness)
$f_4$	Fourth standardized moment (kurtosis)
$f_5$	Sarle's bimodality coefficient
$f_6, f_7, f_8$	Median, 15 <sup>th</sup> and 85 <sup>th</sup> pointwise percentiles
$f_9$	Pointwise dispersion : $f_8 - f_7$
$f_{10}, f_{11}$	Pointwise minimal and maximal values
$f_{12}$	Pointwise amplitude : $f_{11} - f_{10}$

**Table 1** Description of the 12 aggregated profiles used in our approach. The term *pointwise* refers to the fact that calculations are performed separately for each abscissa  $x \in [0, L]$ .

We propose to describe data with the discrete Haar wavelet transform (Dremin et al 2001) for its simplicity and computational efficiency. Using wavelet-based features is not new in machine learning field, and many authors reported them as being powerful descriptors for signal data-based learning, notably in acoustic (Daniels 2010; Morizet et al 2016), medical (Aydemir and Kayikcioglu 2011; Sumathi et al 2014) and image (Lotfi et al 2009) classification fields.

In order to abide by constraint 1, we propose to merge speed curves in a small number of *aggregated profiles*. The 12 aggregated profiles used for this approach are summarized in Table 1.

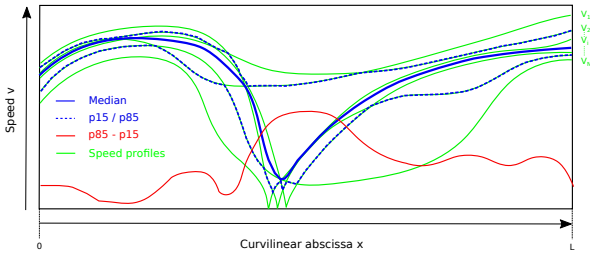
The profile  $f_1$  represents the central tendency of the speed curves collection. The average speed analysis is not sufficient to distinguish between traffic lights, stop signs or even speed bumps. Thus, we define the standard deviation profile  $f_2$ . The rationale behind  $f_2$  introduction is that stop signs and speed bumps are characterized by a very normative response from drivers, hence implying reduced speed variance at every location  $x$ , unlike traffic lights, which most of the time split drivers into two distinct sets depending on whether they have to stop or not. This second aggregated profile will be arguably less informative when the green waves phenomenon<sup>3</sup> is particularly important, for example in major avenues. Accordingly, we believed that only poor results may be extracted from the analysis of these first two profiles, and we decided to add higher order moments  $f_3$  and  $f_4$  (assuming  $f_2$  is strictly positive, *i.e.* there exists no point in  $[0, L]$  where all profiles would intersect), which can be combined to derive a bi-modality profile with Sarle's coefficient (Ellison 1987):

$$f_5(x) = \frac{f_3(x)^2 + 1}{f_4(x)} \quad (2)$$

<sup>3</sup> Series of coordinated traffic lights on major road axis.

Hopefully,  $f_5$  is very informative to discriminate traffic lights from other road infrastructure elements, based on the observation that speeds distribution in some neighborhood of the sign is inherently bi-modal.

Dispersion  $f_9$  is calculated by subtracting 85<sup>th</sup> and 15<sup>th</sup> percentiles, and forms a surrogate measure for the dispersion, possibly more robust than standard deviation profile  $f_2$ . The motivation for selecting these percentile values, instead of the classically used interquartile range, stems from the fact that the 85<sup>th</sup> percentile of speed is a commonly used indicator in transportation studies. Note that if outliers speed profiles were not priorly removed from the data,  $f_{10}$  and  $f_{11}$  can be simply and efficiently estimated as  $Q(\cdot, \varepsilon)$  and  $Q(\cdot, 1 - \varepsilon)$ , where  $Q(x, \cdot)$  is the pointwise generalized inverse of the speed cumulative distribution function at location  $x$  and  $\varepsilon \in \mathbb{R}^+$  is a small value like 0.05 for example.



**Fig. 9** Example of  $f_6$  (median in blue),  $f_7$ ,  $f_8$  (percentiles in dashed lines) and  $f_9$  (percentiles difference in red) aggregated profiles computation, the green curves being the 6 original speed profiles considered.

Figure 9 depicts an example of aggregating profiles on a simple example with 6 speed profiles. Median aggregated profile, percentiles and difference of percentiles are computed at each position.

Each of these 12 profiles contains 100 values, hence implying a total of 1200 explanatory variables at the end of this step. However, these variables are highly correlated. First of all, speed variations of individual profiles are considerably limited by vehicle inertia, thus producing smooth curves, where values taken at neighbor locations are precluded from being very dissimilar. Secondly, profiles are computed by merging individual values in a single curve, which decreases as much local variability. For these reasons, the 1200 explanatory variables are significantly redundant and Haar wavelet

basis expansion is used to reduce the number of features while keeping most of relevant information.

Haar wavelet system may be thought of as a multi-resolution analysis tool, designed to build a ladder of closed subspaces corresponding to different levels of details (Berlinet et al 2008). Each function is then expanded on a basis of orthogonal functions and only meaningful coefficients are retained. Depending on the desired refinement level, one can select the most adequate number of dimensions. As a result, each function is characterized by a unique set of parameters, each of them describing the behavior of the function at a given scale and position. To simplify without loss of generality, we will assume that the length of sliding windows is normalized, thus enabling to consider speed profiles as individual elements of the square-integrable function space  $L_2([0, 1])$ .

Given Haar scaling function  $\phi = \mathbb{1}_{[0,1]}$  and its associated mother wavelet function  $\psi = \mathbb{1}_{[0,1/2]} - \mathbb{1}_{[1/2,1]}$ , basis functions are scaled and translated versions of the mother wavelet function :  $\psi_{jk}(x) = 2^{j/2}\psi(2^j x - k)$ .

It can be proved that the set  $\{\psi_{jk}\}_{j \geq 0, k=0..2^j-1}$  completed with  $\phi$  forms an orthonormal basis of  $L_2([0, 1])$ , therefore aggregated profiles can be written as linear combinations of Haar functions (Dremin et al 2001).

$$\forall x \in [0, 1] \quad f_z(x) = \omega_0^z \phi(x) + \sum_{j=0}^{\infty} \sum_{k=0}^{2^j-1} \omega_{jk}^z \psi_{jk}(x) \quad (3)$$

where  $\omega_0$  and  $\omega_{jk}^z$  coefficients are computed by projection on each basis function:

$$\omega_0^z = \int_0^1 f_z(x) \phi(x) dx \quad \text{and} \quad \omega_{jk}^z = \int_0^1 f_z(x) \psi_{jk}(x) dx \quad (4)$$

We decided to discard all coefficients above  $j = 3$ , which corresponds to a most refined level of 6 meters:

$$f_z(x) = \omega_0^z \phi(x) + \sum_{j=0}^3 \sum_{k=0}^{2^j-1} \omega_{jk}^z \psi_{jk}(x) + \varepsilon_z(x) \quad (5)$$

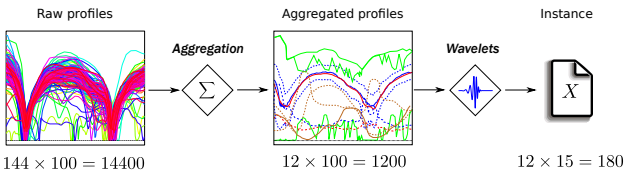
where  $\varepsilon_z$  is the approximation error function containing the smaller levels of details.

We also removed the first component  $\omega_0$  since we believe that average values of aggregated profiles are too informative. Suppressing them compels the training algorithm to search discrimination criteria in most refined levels of details components. Then, we are left with a total of 15 coefficients to describe each one of the 12 profiles *i.e.*  $p = 180$  explanatory variables:

$$X = \{\omega_{jk}^z\} \quad \forall z \in [1, 12] \quad \forall j \in [0, 3] \quad \forall k \in [0, 2^j - 1]$$

For example,  $\omega_{23}^5$  is modeling the behavior of the bi-modality aggregated profile ( $z = 5$ ) at the scale level 12.5 meters ( $j = 2$ ) on the right ( $k = 3$ ), *i.e.* for  $x$  in  $[75, 100]$ , of any sliding window. A simplified diagram of the overall workflow is depicted on Fig. 10.

Wavelet transform was performed by matrix multiplication of resampled aggregated profiles with a  $1024 \times 1024$  normalized Haar matrix. Refer to Stanković and Falkowski (2003) for further information on the discrete Haar wavelet transform. In our implementation, *Wave-tresh* R package was used for computing the transform matrix (Nason and Maechler 2006; R Core Team 2015).



**Fig. 10** Functional instance computation process. From left to right: input sliding window (144 raw profiles sampled at 100 locations), aggregation into 12 profiles  $f_i$  and projection on a basis of 15 wavelet functions.

### 3.2 Classifiers

Given a dataset  $\mathcal{D}$  of training samples  $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ , where  $\mathcal{X} \subseteq \mathbb{R}^p$  and  $\mathcal{Y} = \{0, 1\}$  denote input and output spaces, respectively, and a previously unseen sliding window whose feature vector  $X_{new} \in \mathcal{X}$  has been extracted, the task of a classifier is to estimate the posterior probability of a traffic signal presence in the window  $\mathbb{P}(Y_{new} = 1 | X_{new}, \mathcal{D})$ .

There exist a large number of classification algorithms, and given the fact that we do not have many data instances, we restrained to simple classifiers that do not require too much tuning. For this reason, Artificial Neural Network, despite being very popular, will not be included in the following comparison. In return, in most cases, tuning step is not necessary since we use default parameters recommended in the literature, which enabled to save data for training and test steps.

#### Naive Bayes

Naive Bayes (NB) classifier relies on the often unrealistic assumption that explanatory variables are conditionally independent given the class label (Rish et al 2001). Yet, it provides a simple way of assessing posterior probabilities as a product of conditional probabilities of each individual feature, which is much more easily tractable. It appears that despite its simplicity, Naive Bayes classifier usually achieves decent classification performance even on moderately difficult problems (Hand and Yu 2001). The input features being real-valued here, we modeled their class-conditional probabilities as normal distributions, whose means and variances are estimated from the training dataset, as described more in depth in Ng and Jordan (2002).

#### K Nearest Neighbors

The k-Nearest Neighbors (kNN) (Fix and Hodges Jr 1951) is a simple algorithm, only requiring the definition of an integer parameter  $k$  and a metric on the feature space. It classifies a new instance as the dominant label among the  $k$  nearest training data. In our experimentation,  $k$  was set to 7 and the  $p$ -dimensional euclidean distance was used.

#### Decision Tree

Introduced by Breiman et al (1984) under the name CART for Classification and Regression Trees, decision trees can be seen as an elaborate version of k-NN algorithm. Among many advantages, they are simple, computationally-efficient and easy to interpret. In our experimentation, we set maximum depth to 25. Splits homogeneity are evaluated with Gini criteria.

#### Support Vector Machine

Originating from Vapnik (2013) theoretical works on the *separability* of instances in the feature space, Support Vector Machine (SVM) finds a discrimination

hyperplane separating positive and negative instances with a maximal margin in order to optimize the generalization capabilities of the model. When data are not linearly separable, soft-margin SVM is used to allow for errors in the training dataset. It can be demonstrated that the traditional formulation of SVM is equivalent to a Tikhonov regularization, hence implying that SVM is naturally robust against overfitting (Evgeniou et al 2000). In our experimentation, the misclassification cost parameter was set independently for each approach by selecting the value providing optimal results in the range  $[0.1 - 100]$ .

### *Random Ferns*

Random Ferns have been introduced by Ozuysal et al (2007). Since then they have been used successfully in multiple research works, mostly in the field of image recognition (Aniruddha and Babu 2014; Villamizar et al 2012). Random Ferns can be considered as a semi-naive Bayes (SNB) classifier, insofar as they rely on the more realistic assumption of feature group conditional independence. We used the extension proposed by Kursu (2012) to handle real-valued features. In our experimentation, 1000 ferns of 10 variables each were trained.

### *Random Forests*

Introduced by Breiman (2001), Random Forests algorithm is a statistically robust version of CART, relying on the ensemble method concept to reduce the prediction variance of individual decision trees. This makes Random Forests a simple, fast and efficient classification and regression tool, often considered as robust to over-fitting and particularly useful in high-dimensional problems where one has no strong reason to believe that all features will be helpful for discriminating instances. Moreover, in his foundation paper, Breiman also introduced parameters setting empirical rules, which makes the tuning process quite effortless and straight-forward. For more detailed information about Random Forests, we recommend the complete and extensive works of Louppe (2014) for the theoretical background or Criminisi et al (2011) for a presentation of some of its capabilities in a wide range of practical problems. In our implementation,  $T = 500$  trees were grown and  $\sqrt{p}$  dimensions were randomly chosen at each split (Breiman 2001).

Those 6 algorithms were selected for multiple reasons. First of all, they represent a large variety of methods: 3 of them are based on hyperplanes (CART, SVM and RF), one is based on the feature space distance (kNN) and another one is fully probabilistic and generative (NB). Besides, the ensemble method class of algorithms is represented, with RF and SNB (the latter being included to bridge the gap between NB and RF). The selected panel of methods is well-balanced between parametric and non-parametric models, though the separation is not always clear (Walsh 1962) and two of them (SVM and RF) are particularly robust to high-dimensional instances (Friedman et al 2001). In our comparison, we could have also used logistic regression (see Reunanen et al (2019) for example, for a detailed formulation of the model), which is a popular classification algorithm. However, considering that NB and logistic regression form a generative/discriminative pair, for the sake of conciseness, we decided to report only one of them in our results. Besides preliminary experimentations revealed that logistic regression, despite being much faster in the inference step, was markedly longer to train and provided slightly lower performance scores overall. This may be explained by the fact that while they exhibit higher asymptotic error rate, generative classifiers have faster convergence rate (Ng and Jordan 2002), hence requiring fewer training data instances and making them particularly suitable for our experimentation.

### 3.3 Experimentation protocol

For each of the 18 classifiers, prediction performance was assessed with a 10-fold cross validation process. Since sliding windows are overlapping each other, a mere random split in 10 subsets is impossible. We decided to split the 25-km-long track in 10 parts, so that about the same number of traffic lights are included in each of them. Then, a classifier is built from 9/10<sup>th</sup> of the data and validated on the remaining 1/10<sup>th</sup>. This process was repeated 10 times, until all data instances were classified. In order to avoid that training and validation sets share some common parts of the data, all sliding windows overlapping the training and the validation portions of the loop are removed. Then, the whole 10-fold cross validation process itself is repeated 10 times with randomly selected fold splits in order to perform a sensibility analysis of output results regarding geometric separation of training and test datasets.

Due to the fact that data are significantly imbalanced (16 % of positive instances), some algorithms

may be under-performing. To overcome this issue, at each training process, the training dataset is balanced with over-sampling (*i.e.* positive instances are randomly replicated until their number matches the number of negative instances). The validation dataset is left imbalanced since there is no reason to think that any given proportion is more realistic than another. For a comprehensive study on binary classification from imbalanced datasets, refer to Menardi and Torelli (2014). All predictions of each validation subset is then compiled in a unique confusion table on which most commonly used performance indices are computed.

Note that the full set of drivers is included in both training and validation sample. A more rigorous solution would have been to split drivers as well, so that prediction performance is assessed on completely unseen data. However, it is usually acknowledged that speed profiles are mostly determined by road infrastructure, implying that individual behavior of drivers should not be of any further help in the detection process.

The experimental process was implemented in R with most commonly used machine learning packages: *e1071* (Dimitriadou et al 2009), *knn package* (Schliep et al 2007), *rpart* (Therneau et al 1997), *rFems* (Kursa 2012) and *randomForest* (Liaw and Wiener 2002).

## 4 Results

The experiment was launched on an Intel Core(TM) i7-3770 processor (3.40 GHz RAM 8 Go) with a total computation time just over 12 h 30 min. For each of the 18 classifiers, the following indices are computed :

- *Sensitivity* (or *recall*) **STV**, the probability for a traffic light in a given window to be detected:  $TP/(TP + FN)$ .
- *Specificity* **SPC**, the probability for an empty window to be classified as negative:  $TN/(TN + FP)$ .
- *Precision* **PPV**, the probability of a positive window to contain a traffic light in ground truth:  $TP/(TP + FP)$ .
- *F-measure* **F1M**, the harmonic mean of recall and precision indices:  $2TP/(2TP + FP + FN)$ .
- *Accuracy* **ACC**, the proportion of correctly classified instances:  $(TP + TN)/(TP + TN + FP + FN)$ .
- *Area Under Curve* **AUC** of Receiver Operating Characteristics (ROC) is the integral of  $TPR = f(FPR)$

where *TPR* and *FPR* stand for true and false positive rates respectively.

- *Stop signs confusion* **STP** is the ratio of stop signs incorrectly classified as traffic lights (out of 50 windows containing a stop sign).
- *Offline time* **OFT** is the feature extraction and learning process computation time (measured in seconds per window).
- *Online time* **ONT** is the feature extraction and classification computation time (measured in seconds per window).

*TP*, *TN*, *FP* and *FN* indicate respectively the ratios of true and false positive and negative instances in the prediction out of the total number of tested instances. The first 7 indices above should be as high as possible, while the last 3 ones are expected to be as small as possible.

Table 2 provides complete results where all fields are expressed in % except last two columns which are in sec/window. Confidence intervals are computed at 95 % level over the 10 training/test split replications (Macskassy and Provost 2004).

Table 3 contains Student’s two-sample paired test p-values over Area Under Curve index for 3 out of the 6 machine learning algorithms. Here, we shall note that Student’s test is designed for normal distribution of values. Even though one may assume that AUC indices are normally distributed according to Central Limit Theorem, Wilcoxon signed rank paired test p-values (which do not require normality assumption) were computed in order to confirm Student’s test results. Overall consistency ratio (for all couple of classifiers) is equal to 97% (according to Rand index). This enables to state that AUC normality assumption is realistic, though Wilcoxon test results may be more robust, especially on couples of series containing outliers. Given the agreement between both test results, we will restrain hereafter to Student’s test decision.

Receiver Operating Characteristic (ROC) curves are depicted on Fig. 11 (for each classifier, 10 curves are averaged with threshold averaging method, and confidence bands have been computed at 95% level with fixed-width band method, *i.e.* upper and lower bands are translated versions of the ROC curve, along a line with slope  $-\sqrt{m/n}$  where *m* and *n* denote, respectively, numbers of positive and negative instances (Macskassy

Methods	Performance								
	STV	SPC	PPV	F1M	ACC	AUC	STP	OFT	ONT
Direct approach									
NB	88.30 $\pm 0.10$	69.13 $\pm 1.78$	36.26 $\pm 0.24$	51.41 $\pm 0.25$	72.31 $\pm 0.14$	78.31 $\pm 0.35$	90.12 $\pm 0.22$	0.69	<b>0.30</b>
CART	86.56 $\pm 0.41$	87.68 $\pm 0.20$	58.29 $\pm 0.49$	69.66 $\pm 0.45$	87.50 $\pm 0.22$	88.14 $\pm 1.04$	46.53 $\pm 2.14$	0.72	<b>0.30</b>
kNN	62.18 $\pm 0.22$	93.07 $\pm 0.10$	64.10 $\pm 0.45$	63.13 $\pm 0.27$	87.95 $\pm 0.53$	82.54 $\pm 0.36$	37.79 $\pm 0.45$	1.24	<b>0.30</b>
SVM	83.58 $\pm 0.41$	83.08 $\pm 0.37$	49.55 $\pm 0.28$	62.22 $\pm 0.41$	83.16 $\pm 0.32$	91.94 $\pm 0.26$	44.37 $\pm 1.02$	6.77	<b>0.30</b>
SNB	95.52 $\pm 0.14$	64.19 $\pm 0.34$	34.65 $\pm 0.23$	50.86 $\pm 0.25$	69.38 $\pm 0.27$	88.07 $\pm 0.59$	97.85 $\pm 2.27$	0.40	<b>0.30</b>
RF	72.88 $\pm 0.27$	95.99 $\pm 0.08$	78.34 $\pm 0.23$	75.51 $\pm 0.16$	92.16 $\pm 0.06$	94.69 $\pm 0.99$	4.70 $\pm 0.53$	2.54	<b>0.30</b>
Image approach									
NB	87.56 $\pm 0.23$	67.95 $\pm 2.90$	35.20 $\pm 1.74$	50.21 $\pm 1.94$	71.20 $\pm 2.47$	76.68 $\pm 1.48$	92.05 $\pm 0.53$	1.87	1.49
CART	82.33 $\pm 1.29$	87.09 $\pm 0.61$	55.91 $\pm 1.29$	66.59 $\pm 1.01$	86.30 $\pm 0.49$	85.92 $\pm 1.26$	56.00 $\pm 4.74$	1.87	1.49
kNN	68.40 $\pm 0.73$	91.24 $\pm 0.20$	60.84 $\pm 1.00$	64.40 $\pm 0.69$	87.45 $\pm 0.18$	83.05 $\pm 0.27$	54.12 $\pm 2.47$	3.10	1.49
SVM	70.64 $\pm 0.41$	87.53 $\pm 0.37$	52.98 $\pm 0.28$	60.55 $\pm 0.41$	84.73 $\pm 0.32$	82.98 $\pm 0.26$	68.27 $\pm 1.02$	9.23	1.49
SNB	94.27 $\pm 1.56$	62.11 $\pm 2.37$	33.10 $\pm 2.56$	48.99 $\pm 2.33$	67.45 $\pm 0.37$	86.59 $\pm 0.33$	99.02 $\pm 1.59$	1.50	1.49
RF	64.17 $\pm 0.45$	96.34 $\pm 0.18$	77.71 $\pm 0.99$	70.29 $\pm 0.55$	91.00 $\pm 0.22$	94.14 $\pm 0.23$	6.13 $\pm 1.86$	5.16	1.49
Functional approach									
NB	88.30 $\pm 0.22$	90.60 $\pm 0.06$	65.13 $\pm 0.22$	74.97 $\pm 0.18$	90.22 $\pm 0.06$	91.88 $\pm 0.21$	67.95 $\pm 0.47$	<b>0.31</b>	0.31
CART	88.30 $\pm 0.22$	91.54 $\pm 0.31$	67.49 $\pm 0.69$	76.50 $\pm 0.53$	91.00 $\pm 0.27$	88.61 $\pm 0.58$	57.52 $\pm 1.08$	<b>0.31</b>	0.31
kNN	79.60 $\pm 0.22$	94.41 $\pm 0.04$	73.90 $\pm 0.27$	76.64 $\pm 0.20$	91.95 $\pm 0.06$	91.5 $\pm 0.32$	48.21 $\pm 0.53$	<b>0.31</b>	0.31
SVM	82.33 $\pm 0.42$	88.41 $\pm 0.53$	58.28 $\pm 0.92$	68.24 $\pm 0.72$	87.28 $\pm 0.39$	90.74 $\pm 0.22$	36.40 $\pm 1.47$	0.32	0.31
SNB	<b>97.51</b> $\pm 0.10$	73.88 $\pm 0.41$	42.60 $\pm 0.45$	59.30 $\pm 0.45$	77.80 $\pm 0.35$	95.59 $\pm 0.25$	82.23 $\pm 1.33$	<b>0.31</b>	0.31
RF	82.58 $\pm 0.23$	<b>97.23</b> $\pm 0.06$	<b>85.56</b> $\pm 0.69$	<b>84.05</b> $\pm 0.27$	<b>94.80</b> $\pm 0.08$	<b>97.28</b> $\pm 0.22$	<b>2.81</b> $\pm 0.80$	0.32	0.31

**Table 2** Prediction performance indices for the 5 classifiers on the 3 different approaches with 95% confidence level. Bold type indicates the best performance in each column.

and Provost 2004). The displacement distance is chosen such that a fraction  $r$  of curves is completely included within the confidence bands ( $r$  is the confidence level). This method is particularly relevant here since it does not depend on the actual numbers of instances, provided that the ratio of positive over negative instance numbers is known. In our case study, since windows are overlapping each other, it is very difficult to estimate the exact number of instances used for the validation process (data instances are very correlated to each other, thus implying that this number is lying somewhere between the number of non-overlapping windows and 10 times this number). However, estimating the ra-

tio of positive instances (16%) is easy and makes the fixed-width band method well adapted to our situation. We shall note, though, that unlike some other approaches, this method is global, *i.e.* at 95% level for example, 19 times out of 20, the real unknown ROC curve is completely included in the confidence bands. This often results in larger bands than with local methods.

The results in Table 2 and Fig. 11 highlight that the functional approach seems to achieve better performance than its image and direct counterparts on 4 out of 6 investigated algorithms. There is an exception

		Approaches $\times$ Algorithms								
		Direct			Image			Functional		
		NB	SNB	RF	NB	SNB	RF	NB	SNB	RF
Direct	NB									
	SNB	+								
	RF	+	+							
Image	NB	8.91	-	-						
	SNB	+	1.85	-	+					
	RF	+	+	33.5	+	+				
Func.	NB	+	+	-	+	+	-			
	SNB	+	+	16.9	+	+	+	+		
	RF	+	+	+	+	+	+	+	+	
		NB = Naive Bayes			SNB = Random Ferns			RF = Random Forest		

**Table 3** Student’s paired test p-values (in %) for each couple of classifiers. The symbol + (resp. -) indicates a p-value is below 1 % and that row classifier is more (resp. less) performant (with respect to AUC index) than column classifier. The test is symmetric, thus only lower triangular part is provided.

for decision tree classifier, where the direct approach area under curve index ( $88.14\% \pm 1.04$ ) is very close to the functional approach ( $88.61\% \pm 0.58$ ). However, this can hardly be a strong conclusive evidence against the adequateness of CART algorithm to classify functional objects, since ROC curves (depicted on Fig. 11) do not seem to be very precise, with large confidence bands. This high variance may be partly explained by the fact that CART algorithm is often acknowledged as not being extremely robust to noise in observations, and slightly different training data may often result in markedly different models of decision trees. However, interestingly, SVM algorithm provides significantly better results on the direct approach, and ranks in second position behind Random Forest on that approach. These results seem to indicate to CART and SVM do not take advantage of the low-dimensional description of the functional approach, and perform better or as well on the coarser, high-dimensional image approach, which confirm the idea that these algorithms are well-suited for high-dimensional data.

Conversely, on the 4 remaining algorithms, the functional approach is clearly outperforming the other approaches and this is all the more noticeable in kNN, NB and SNB. Contrary to CART and SVM, these algorithms are more adapted to detect complex data structures in lower dimensions. We may think that a sophisticated classifier like Random Forest should combine the strengths of both sets of algorithms and be able to get decent results, even with a poor set of features, which results in slighter differences between curves in ROC space for RF algorithm. Still, considering only last row in Fig. 11, functional features give significantly better results, with an AUC equal to 97.28 % *i.e.* more

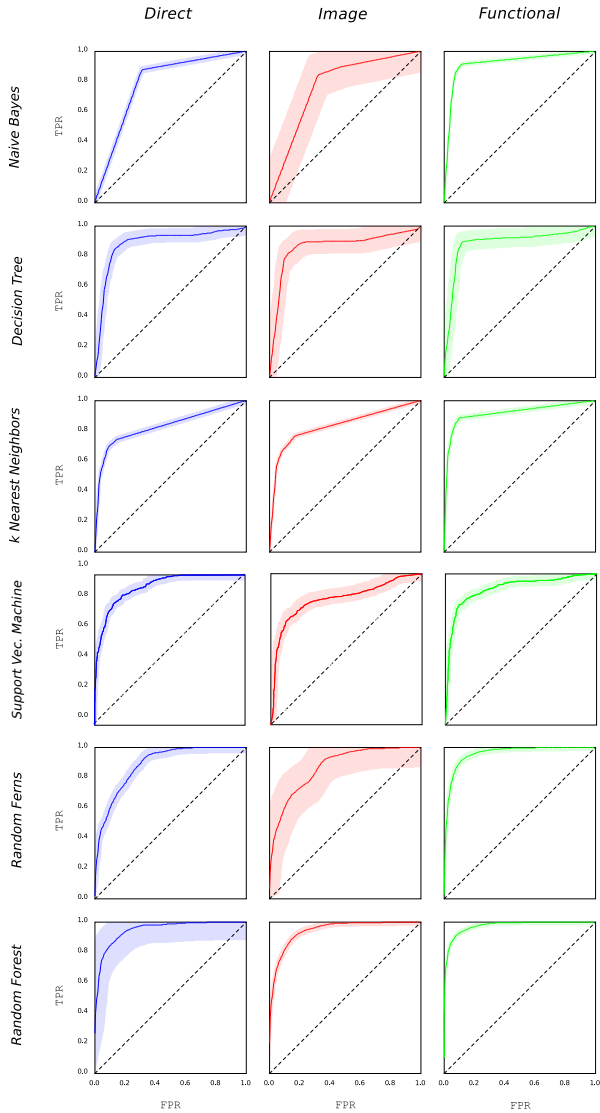
than 2% above direct approach and 3% above image approach. These AUC differences are illustrated on the left part of Fig. 12 where it can be observed that even though direct and image methods are equivalent for a specific threshold, functional approach is undoubtedly better on a wide range of thresholds.

Besides, the gap with other approaches is especially large in the operational part of the ROC space, *i.e.* where TP and FP rates have reasonable values for an application to a real problem. More specifically in our case study, we do not want to take the risk of updating a road infrastructure database with some wrong information. Hence, we have some reason to think that keeping a low FP rate is the most important. As a result, optimal operating point of any given classifier will be computed by intersecting its ROC curve with a line whose slope angle is higher<sup>4</sup> than  $\pi/4$ . This part of the ROC space, is especially where the gap between green curve and other curves is the most prominent. This comes from the fact that we have imbalanced data in favor of negative instances, thus entailing larger confidence bands along  $y$ -axis direction.

The widths of ROC curve confidence bands depend on the variance introduced by selecting splits at random on 10 different replications. For this reason, they mostly emphasize the uncertainties of classifiers due to variations in training-validation dataset inputs. As mentioned before, we discarded the variance due to the lack of data by using a fixed-width band method.

<sup>4</sup> This holds provided that the class proportions are not too much imbalanced in favor of positive instances. Even in urban conditions, we may consider that this is a realistic assumption.

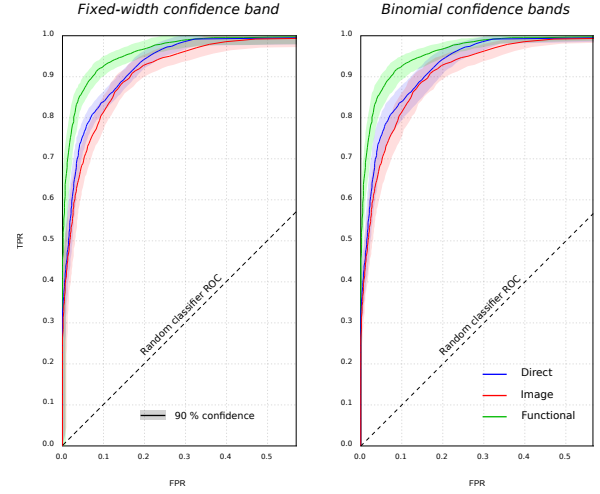




**Fig. 11** ROC curves at 95 % confidence level for each method

However, for Random Forest comparison on each approach, we also wanted to assess result uncertainties in ROC curves that may be attributed to this second factor. To do so, we computed a second set of confidence bands at 90% confidence level, with threshold averaging method under binomial distribution (Wilson 1927) assuming overlapping windows are independent. These bands are depicted on the right part of Fig. 12, on which we may observe that the functional approach ROC curve prominence remains significant.

Differences between some of combinations (with 3 out of the 6 classification algorithms) may be analyzed with paired tests p-values (Table 3). Given that we do not have a very large number of data instances, it is de-



**Fig. 12** Comparison of Random Forest ROC curves at 90 % confidence level

cided hereafter to consider a p-value as significant when it is below 1%. It can be clearly seen that functional approach is quite often better than other approaches, especially when combined with Random Forest, and this preeminence is most of the time below 1% threshold. It is worth noticing as well that despite its simplicity, Random Forest on direct features (sixth row in the tables) is performing very well.

More anecdotally, the functional approach is much faster than the direct and the image approaches (Table 2, columns OFT and ONT), especially as far as training time is concerned. The difference in processing time mostly comes from the fact that there are significantly fewer functional features. However, on large feature set, it may be seen, as might be expected, that Random Ferns (SNB), by selecting only some subsets of interdependent explanatory variables, are able to perform comparatively fast training.

On any given approach, Random Forest is the most accurate in terms of stop signs confusion (between 3 to 6% of stop signs are incorrectly classified as traffic lights). This suggests that RF classifier is not blindly scanning speed profiles at low speed (looking for stop points) but has some higher discriminating power, giving confidence in future works, for multinomial classifier specifically trained to detect and identify road infrastructure elements. As a drawback, in our experimentation RF required longer training time. This issue can be addressed with parallel computing techniques. Training time may also be reduced by decreasing the number of trees to grow in the model (500 in our application).

A more refined analysis would enable to set this parameter equal to the number of trees needed to assume convergence has been practically reached.

On functional approach, all classifiers appear to need about same training time (preparation time *i.e.* profile aggregation and wavelet transforms, is comparatively much more time-consuming). Even though Random Forest is significantly the best classifier for our task, Random Ferns are not lagging far behind. To a lesser extent, this is also the case for CART algorithm (on the optimal point of ROC curve in terms of overall accuracy). However, both of these methods score poorly on stop signs confusion index. Besides, RF natural threshold (SPC over STV) is more adequate to the problem we would like to solve.

We must notice that, while the functional approach has access to some extent to the acceleration data (via the low order wavelet coefficients), it is not the case for the direct approach (where speeds are ordered separately at each location, hence breaking the relationship among individual profiles) and the image approach (where only a group average acceleration can be seized by the features). For the sake of consistency, we derived the spatial acceleration profiles directly from the speed profiles with chain rule derivative and centered finite difference (or forward finite difference for the first record in the windows):

$$a(x) = \frac{dv}{dt}(x) = \frac{dv}{dx} \frac{dx}{dt} = v'(x)v(x)$$

$$a_k = \begin{cases} v_k(v_{k+1} - v_k) & \text{if } k = 1 \\ \frac{1}{2}v_k(v_{k+1} - v_{k-1}) & \text{otherwise.} \end{cases}$$

where  $k \in \llbracket 1; 100 \rrbracket$  stands for the relative position in the sliding window.

The vector of accelerations is then ordered at each location and concatenated to the feature vector extracted with the direct approach.

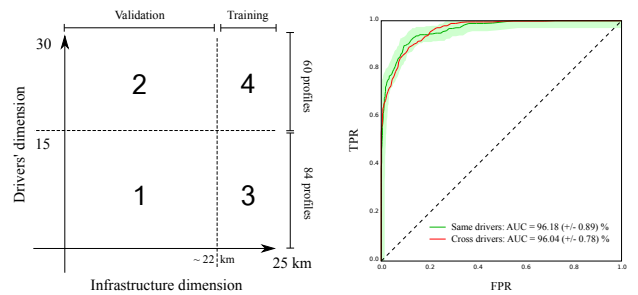
According to Table 4, taking into account pointwise acceleration values in the classification process does not seem to improve significantly the results, except on the traffic signal / stop confusion (STP) performance indices.

Eventually, we must validate an hypothesis formulated in section 3.3: we assumed that speed profiles

Algo.	ACC	AUC	STP	OFT	ONT
NB	71.31	77.12	85.05	1.96	0.44
CART	86.61	88.39	41.33	2.11	0.44
kNN	86.55	83.51	37.08	3.65	0.44
SVM	83.31	91.51	40.18	19.79	0.44
SNB	68.48	88.21	97.49	1.78	0.44
RF	91.79	95.72	4.50	7.55	0.44

**Table 4** Prediction performance indices for the 6 classifiers on the enhanced direct method with accelerations.

are mostly dependent upon the road infrastructure and that the drivers' specificities may not be of any help in the detection process. As a consequence, we decided to include all the drivers in both validation and training datasets. As illustrated in Fig. 13, this amounts to train a model on datasets 1 and 2, and test it on 3 and 4. In order to confirm this hypothesis, we replicated one of the experiments (RF classifier on functional features) twice with two different protocols. In the first one, we randomly sampled 15 drivers and the 84 speed profiles generated by these drivers were used to create both the training and the validation dataset. This process corresponds to the protocol detailed in section 3.3, except that here, half of the drivers are discarded. In the second protocol, 84 speed profiles (stemming from 50% of the drivers) are used to create the training dataset, while the 60 remaining profiles (stemming from the remaining half of the drivers) have been used for the validation dataset. In Fig. 13, this schematically corresponds to training a model with dataset 1 and validating with 4. Note that in both these experiments, the split of circuit loop (and then of traffic lights) is operated in a similar way with the one described in the main part of this paper. Hence, this additional experimentation aims at capturing, if it exists, the difference between training and validating a model with the same or different drivers.



**Fig. 13** Left: split training and validation dataset along two dimensions: road infrastructure and drivers. Right: ROC curves for a model trained and validated on the same (resp. different) drivers in green (resp. red).

The results of these two experiments confirmed our hypothesis, since no significant difference was observed in the respective ROC curves produced by the models, as illustrated in Fig. 13. This means that, as we anticipated, training and validating our classifier models with speed profiles stemming from the same drivers (for the sake of simplicity) does not produce a bias in the classification scores that would lead to an overestimation of the performances. We may notice as well that the AUC index of the two models are slightly lesser than the AUC of the original experiment (97.3%). This may be explained by the fact that we used a smaller number of speed profiles in this additional experiment.

## 5 Discussion

From the experimentation results, we clearly observed that a functional description of profiles, apart from being remarkably faster, leads to better results in classification performance, especially when used in combination with Random Forest, which proved to be the best algorithm regarding stop sign-traffic light discrimination. From a thematic perspective, this point is particularly important as future works will aim not only at detecting but also classifying infrastructure elements. We also found that Random Forest seems less sensitive to the selected approach for extracting features from speed profiles and is able to perform decent detection task even on a poor set of features.

However, the lack of data is a strong limitation of our experimentation. By considering only 44 instances of traffic lights, even with an artificial increase of this number via window overlap, it can hardly be assumed that we have been dealing with a fully representative set of instances. We tried to take this shortage into account by randomly bootstrapping folds to compute an estimate of confidence interval on ROC curves and performance indices. While, overlapping data instances was necessary to ensure that any given classifier is able to detect an infrastructure element regardless of its relative position in the window, it also entailed challenging issues in terms of validation procedure. Indeed, as data are highly redundant, a sheer random split of data between training and validation sets is no longer possible - by doing so we would risk to end up with a validation set whose almost all instances have been used (in a very similar version) during training phase. Unfortunately, as far as we know, there is no proper methodology in machine learning literature to address the issue of cross validating continuous and overlapping instances. This compelled us to perform a geometric separation of folds,

thus needing to reiterate the experiment multiple times with different split patterns to ensure that our results were not obtained by a singular split.

While the number of drivers is as well problematic, we believe that by having each of them running the circuit multiple times, we have a sufficient number of speed profiles in each window to perform detection. In future works, we will aim at assessing the influence of this parameter on the performance classification and try to estimate the minimal number of profiles required to assume that convergence has been reached. If we can confirm that this number is not so large (*e.g.* 20 curves or so), we will orient our research to non-experimental large-scale GPS datasets (with fewer drivers but more traffic light instances). Such a *real-life data* based experimentation will be the opportunity as well to conduct sensitivity analysis of our algorithms to GPS frequency and precision (especially when speed is no longer observed with Doppler system but roughly estimated by positions finite difference). It may as well ensure that we are not overlooking any specific infrastructure element that might be easily confused with traffic lights.

One of the main interests of using speed profiles to detect road infrastructure, is that it does not require stop points, which are very specific to traffic lights and stop signs. By learning detection directly on the complete profile patterns, we give opportunities to our method for the detection of other infrastructure elements, including those that do not require the vehicle to stop (like speed bumps or yield signs for example). These experiments show that building an operational method based on these algorithm needs a larger dataset to tune and assess the algorithms more thoroughly. The functional approach seems to be both the most precise and the most computationally efficient, it thus makes a good basis to develop with more extended studies. As for the machine-learning algorithms, random forest is both efficient and easy enough to tune.

With more data, we will tune our more promising approach to find other aggregated profiles, which may help in classification task. Another significant improvement may be found in leveraging spatial correlations between instances. In our application case, labels are positively correlated at short scale: when a given sliding window is classified as containing a traffic light, then due to overlapping between instances, it is more likely that immediate neighbor instances are also labeled as positive. At a larger scale (neighbor but non-overlapping, or slightly overlapping windows), spatial

correlation becomes negative, since it is relatively unlikely to find two successive traffic signals separated by a short distance on a same road. In the more general case where GPS trajectories are not located on a linear path, spatial correlation may also be significantly positive at short distance between different streets leading to the same intersection. All these considerations may motivate instance modeling as a Markov Random Field to detect traffic signals consistently and simultaneously on the whole road network.

## 6 Conclusion

In this paper, we proposed a model for traffic signal detection on speed profiles with machine learning techniques and we derived it in 3 different approaches. Each of them puts the stress on a different aspect of speed curves and has been input to the most common classifiers. One of these approaches has been adapted from previous works in image recognition. We also used a methodology combining paired tests and ROC curves confidence bands on 10 repetitions of a 10-fold cross validation to overcome data scarcity and compare the different combinations of approaches and algorithms.

Experimental results analysis revealed that considering functional nature of profiles for extracting relevant features enabled to build the fastest and the most efficient classifiers with most of the investigated algorithms. Besides, when combined with Random Forest, which proved to be the best algorithm on each of the 3 approaches, the resulting classifier turned out to be sensitive for discriminating traffic signals and stop signs, with a low confusion rate. These results instilled confidence in combining wavelet transforms and Random Forest for detecting and locating any kind of road infrastructure element from a set of GPS speed curves.

The results obtained with random forests and a functional modelization indicate that this combination can be a sound basis to build an operational method to detect traffic lights (and eventually general punctual road infrastructure) using GPS datasets. The automatic construction of a large scale database of such information would be extremely useful for urban planning, navigation application, and self-driving cars. Our next research steps will aim at further investigating its capabilities as well as its robustness to change of environment, number of curves and sensor measurement rate and positional accuracy. We will also try to enhance the results through the combination of the different data modeling approaches in a single framework.

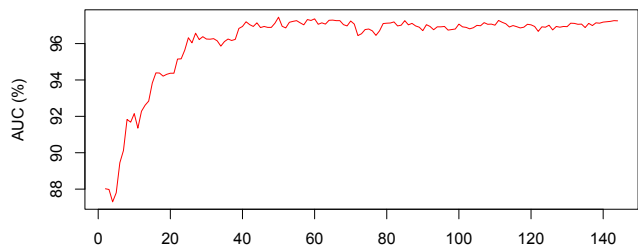
On behalf of all authors, the corresponding author states that there is no conflict of interest.

## 7 Annexes: sensitivity analysis

This supplementary material provides a sensibility analysis of our results (for the case of Random Forest learning on functional features) upon the number of available profiles (7.1) and the choice of features (7.2).

### 7.1 Number of available profiles

An important issue in map inference is to determine the size of the dataset needed to get the desired results. The answer to this question lies in the sensitivity analysis of the detection algorithm to the available number of profiles. In our experiment, we are taking advantage of the fact that we have a large number of tracks (144 in total) on all the sliding windows, to reduce this number and evaluate the impact on the AUC index. We therefore selected a number  $N$  of profiles from 2 to 144 (the algorithm is undefined for  $N = 1$ ), in steps of one unit. For each experiment, both the training and validation sets contain  $N$  profiles. Fig. 14 illustrates the results.



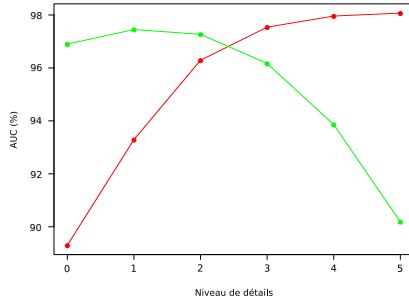
**Fig. 14** Available number of profiles versus Area Under Curve (AUC) index.

The results clearly show the convergence of the AUC score from 30 to 40 profiles. Satisfactory performance (*i.e.* about 92 %) may be obtained with as few as 10 profiles. A more detailed study in the future could attempt to assess the impact of the number of available profiles separately on the training and validation datasets.

### 7.2 Importance of features

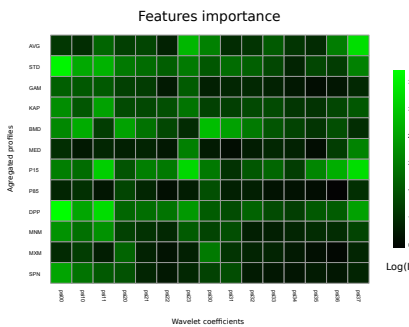
It seems natural to assess the influence of the level of representation of functional data on the quality of detection, measured in terms of AUC index.

Figure 15 hereafter illustrates the increase in detection performance with the level of detail of the wavelet database (red curve). Convergence seems to be reached for level 3 (6 m), but levels 4 (3 m) and 5 (1.5 m) allow to obtain slightly better results, at the expense of an extra computation time (each level doubles the number of explanatory variables to be processed). On the other hand, the green curve shows that the algorithm performance is optimal after removing the coarsest level.



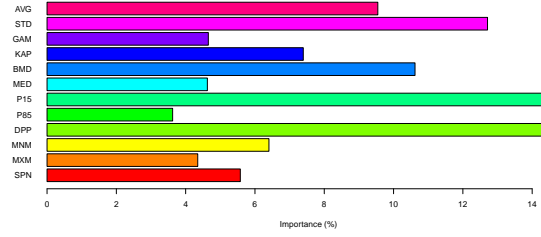
**Fig. 15** Influence of the level of both the father wavelet (in green) and the most refined mother wavelet (in red) on the AUC index of the Random Forest model. The horizontal scale is logarithmic, with a level  $h$  representing a metric scale equal to  $L/2^{h+1}$  with  $L = 100$  m.

Figure 16 provides the relative importance of individual features at different scales and positions. We used the empirical importance index calculated by Monte-Carlo simulation (Gregorutti 2015).



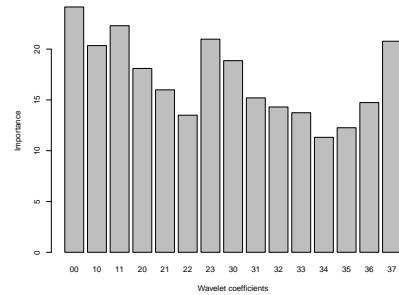
**Fig. 16** Importance computed by permutation for the 12 aggregated profiles, projected on the 15 base wavelets.

Figure 17 represents an integrated version of the matrix shown on Fig. 16, revealing the significant importance of mean profiles (AVG), standard deviation (STD), bimodality (BMD), 15th percentile (P85) and P85-P15 difference (DPP). The 85th percentile is seem-



**Fig. 17** Relative importance of aggregated profiles (integrated on all level of details of the wavelet description).

ingly not so important, probably because of the redundancy of the information provided by the 3 aggregated profiles P15, P85 and DPP. The profiles featuring extremal vales (MNM, MXN and SPN) seem marginal, suggesting that the data are too noisy for them to bring decisive information. Order moments 3 (GAM) and 4 (KAP) also seem marginal with regard to their BMD combination. Surprisingly, the median profile (MED) is relegated to the background by the average profile.



**Fig. 18** Relative importance of level of details of the wavelet description (integrated on all aggregated profiles).

Figure 18 reveals interestingly a more significant importance for coefficients of basis wavelets located close to the window edges. Further analysis is required to understand the underlying process of this observation.

## References

- Andrieu C, Saint Pierre G, Bressaud X (2013) Estimation of space-speed profiles: A functional approach using smoothing splines. In: Intelligent Vehicles Symposium (IV), 2013 IEEE, IEEE, pp 982–987
- Aniruddha AK, Babu RV (2014) Visual object tracking via random ferns based classification. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE, pp 6533–6537

- Arai A, Shibasaki R (2013) Estimation of human mobility patterns and attributes analyzing anonymized mobile phone cdr: Developing real-time census from crowds of greater dhaka. *Proceedings 2nd AGILE PhD School*
- Aydemir O, Kayikcioglu T (2011) Wavelet transform based classification of invasive brain computer interface data. *Radioengineering* 20(1):31–38
- Bentabet L, Jodouin S, Ziou D, Vaillancourt J (2003) Road vectors update using sar imagery: a snake-based method. *IEEE Transactions on Geoscience and Remote Sensing* 41(8):1785–1803, DOI 10.1109/TGRS.2003.813850
- Berlinet A, Biau G, Rouviere L (2008) Functional supervised classification with wavelets. In: *Annales de l'ISUP*, vol 52, p 19
- Biagioni J, Eriksson J (2012) Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation Research Record: Journal of the Transportation Research Board* (2291):61–71
- Biljecki F, Ledoux H, Van Oosterom P (2013) Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science* 27(2):385–407
- Bonin O (2002) Modèle d'erreurs dans une base de données géographiques et grandes déviations pour des sommes pondérées ; application à l'estimation d'erreurs sur un temps de parcours
- Breiman L (2001) Random forests. *Machine learning* 45(1):5–32
- Breiman L, Friedman J, Stone CJ, Olshen RA (1984) *Classification and regression trees*. CRC press
- Chen Q, Song X, Yamada H, Shibasaki R (2016) Learning deep representation from big and heterogeneous data for traffic accident inference. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI Press, AAAI'16, pp 338–344
- Chen Y, Krumm J (2010) Probabilistic modeling of traffic lanes from gps traces
- Criminisi A, Shotton J, Konukoglu E (2011) Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. *Microsoft Research Cambridge, Tech Rep MSRTR-2011-114* 5(6):12
- Dabiri S, Heaslip K (2018) Inferring transportation modes from gps trajectories using a convolutional neural network. *Transportation Research Part C: Emerging Technologies* 86:360 – 371, DOI <https://doi.org/10.1016/j.trc.2017.11.021>
- Daniels M (2010) Classification of percussive sounds using wavelet-based. *CCRMA, Stanford University thesis*
- Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39(1):1–22
- Dimitriadou E, Hornik K, Leisch F, Meyer D, Weingessel A, Leisch MF (2009) Package e1071. R Software package, available at <http://cran.rproject.org/web/packages/e1071/index.html>
- Dremin IM, Ivanov OV, Nechitailo VA (2001) Wavelets and their uses. *Physics-Uspekhi* 44(5):447
- Ellison AM (1987) Effect of seed dimorphism on the density-dependent dynamics of experimental populations of *atriplex triangularis* (chenopodiaceae). *American Journal of Botany* 74(8):1280–1288
- Endo Y, Toda H, Nishida K, Ikeda J (2016) Classifying spatial trajectories using representation learning. *International Journal of Data Science and Analytics* 2(3-4):107–117
- Evgeniou T, Pontil M, Poggio T (2000) Regularization networks and support vector machines. *Advances in computational mathematics* 13(1):1
- Ferraty F, Vieu P (2011) Richesse et complexité des données fonctionnelles. *Revue Modélad* 43:25–43
- Fix E, Hodges Jr JL (1951) Discriminatory analysis-nonparametric discrimination: consistency properties
- Friedman J, Hastie T, Tibshirani R (2001) *The elements of statistical learning*, vol 1. Springer series in statistics New York
- Girres JF, Touya G (2010) Quality assessment of the french openstreetmap dataset. *Transactions in GIS* 14(4):435–459, DOI 10.1111/j.1467-9671.2010.01203.x
- Gregorutti B (2015) Forêts aléatoires et sélection de variables: analyse des données des enregistreurs de vol pour la sécurité aérienne. PhD thesis, Paris 6
- Haklay M, Weber P (2008) Openstreetmap: User-generated street maps. *IEEE Pervasive Computing* 7(4):12–18
- Hand DJ, Yu K (2001) Idiot's bayesnot so stupid after all? *International statistical review* 69(3):385–398
- Kaplan E, Hegarty C (2005) *Understanding GPS: principles and applications*. Artech house
- Kursa MB (2012) rferns-random ferns method implementation for the general-purpose machine learning. *Journal of Statistical Software*
- LeCun Y, Haffner P, Bottou L, Bengio Y (1999) Object recognition with gradient-based learning. In: *Shape, contour and grouping in computer vision*, Springer, pp 319–345
- Li J, Qin Q, Han J, Tang LA, Lei KH (2015) Mining trajectory data and geotagged data in social media for road map inference. *Transactions in GIS* 19(1):1–18

- Liaw A, Wiener M (2002) Classification and regression by randomforest. *R news* 2(3):18–22
- Liu X, Biagioni J, Eriksson J, Wang Y, Forman G, Zhu Y (2012) Mining large-scale, sparse gps traces for map inference: Comparison of approaches. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, NY, USA, KDD '12, pp 669–677, DOI 10.1145/2339530.2339637
- Lotfi M, Solimani A, Dargazany A, Afzal H, Bandarabadi M (2009) Combining wavelet transforms and neural networks for image classification. In: *System Theory, 2009. SSST 2009. 41st Southeastern Symposium on*, IEEE, pp 44–48
- Louppe G (2014) Understanding random forests: From theory to practice. *arXiv preprint arXiv:14077502*
- Lulli A, Gabrielli L, Dazzi P, DellAmico M, Michiardi P, Nanni M, Ricci L (2017) Scalable and flexible clustering solutions for mobile phone-based population indicators. *International Journal of Data Science and Analytics* 4(4):285–299
- Lundgren M, Stenborg E, Svensson L, Hammarstrand L (2014) Vehicle self-localization using off-the-shelf sensors and a detailed map. In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, IEEE, pp 522–528
- Macskassy S, Provost F (2004) Confidence bands for roc curves: Methods and an empirical study. *Proceedings of the First Workshop on ROC Analysis in AI*. August 2004.
- Menardi G, Torelli N (2014) Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery* pp 1–31
- Moreno AT, García A (2013) Use of speed profile as surrogate measure: Effect of traffic calming devices on crosstown road safety performance. *Accident Analysis & Prevention* 61:23–32
- Morizet N, Godin N, Tang J, Maillet E, Fregonese M, Normand B (2016) Classification of acoustic emission signals using wavelets and random forests: Application to localized corrosion. *Mechanical Systems and Signal Processing* 70:1026–1037
- Mu G, Xinyu Z, Deyi L, Tianlei Z, Lifeng A (2015) Traffic light detection and recognition for autonomous vehicles. *The Journal of China Universities of Posts and Telecommunications* 22(1):50–56
- Munoz-Organero M, Ruiz-Blaquez R, Snchez-Fernndez L (2018) Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on gps traces while driving. *Computers, Environment and Urban Systems* 68:1 – 8
- Nason G, Maechler MM (2006) The wavethresh package
- Newson P, Krumm J (2009) Hidden markov map matching through noise and sparseness. In: *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, ACM, pp 336–343
- Ng AY, Jordan MI (2002) On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In: *Advances in neural information processing systems*, pp 841–848
- Ozuysal M, Fua P, Lepetit V (2007) Fast keypoint recognition in ten lines of code. In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, Ieee, pp 1–8
- R Core Team (2015) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL <https://www.R-project.org/>
- Reunanen N, Rätty T, Jokinen JJ, Hoyt T, Culler D (2019) Unsupervised online detection and prediction of outliers in streams of sensor data. *International Journal of Data Science and Analytics* pp 1–30
- Rish I, et al (2001) An empirical study of the naive bayes classifier. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol 3, pp 41–46
- Schliep K, Hechenbichler K, Schliep MK (2007) The kkn package. Unknown
- Schroedl S, Wagstaff K, Rogers S, Langley P, Wilson C (2004) Mining GPS traces for map refinement. *Data mining and knowledge Discovery* 9(1):59–87
- Stanković RS, Falkowski BJ (2003) The haar wavelet transform: its status and achievements. *Computers & Electrical Engineering* 29(1):25–44
- Sumathi S, Beaulah HL, Vanithamani R (2014) A wavelet transform based feature extraction and classification of cardiac disorder. *Journal of medical systems* 38(9):98
- Sun Y, Genton MG (2011) Functional boxplots. *Journal of Computational and Graphical Statistics* 20(2):316–334
- Therneau TM, Atkinson EJ, et al (1997) An introduction to recursive partitioning using the rpart routines. Tech. rep., Technical report Mayo Foundation
- Tully S, Kantor G, Choset H (2011) Inequality constrained kalman filtering for the localization and registration of a surgical robot. In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, IEEE, pp 5147–5152
- Van Winden K (2014) Automatically deriving and updating attribute road data from movement trajectories
- Van Winden K, Biljecki F, Van der Spek S (2016) Automatic update of road attributes by mining gps tracks. *Transactions in GIS*



- Vapnik V (2013) The nature of statistical learning theory. Springer science & business media
- Villamizar M, Garrell A, Sanfeliu A, Moreno-Noguer F (2012) Online human-assisted learning using random ferns. In: Pattern Recognition (ICPR), 2012 21st International Conference on, IEEE, pp 2821–2824
- Walsh JE (1962) Handbook of Nonparametric Statistics. Van Nostrand
- Wang C, Hao P, Wu G, Qi X, Lyu T, Barth M (2017) Intersection and stop bar position extraction from crowdsourced gps trajectories. Tech. rep.
- Wilson EB (1927) Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association* 22(158):209–212
- Zhang Q, Couloigner I (2006) Automated road network extraction from high resolution multi-spectral imagery. In: Proceedings of ASPRS 2006 Annual Conference, pp 01–05
- Zhang Y, Liu J, Qian X, Qiu A, Zhang F (2017) An automatic road network construction method using massive gps trajectory data. *ISPRS International Journal of Geo-Information* 6(12), DOI 10.3390/ijgi6120400
- Zourlidou S, Fischer C, Sester M (2019) Classification of street junctions according to traffic regulators. In: Agile proceedings