AGILE: GIScience Series, 3, 22, 2022. https://doi.org/10.5194/agile-giss-3-22-2022
Proceedings of the 25th AGILE Conference on Geographic Information Science, 2022.
Editors: E. Parseliunas, A. Mansourian, P. Partsinevelos, and J. Suziedelyte-Visockiene.
This contribution underwent peer review based on a full paper submission.

Traffic Regulation Recognition using Crowd-Sensed GPS and Map Data: a Hybrid Approach

Stefania Zourlidou, Jens Golze, and Monika Sester
Institute of Cartography and Geoinformatics, Leibniz University, Hannover, Germany

Correspondence: Stefania Zourlidou (zourlidou@ikg.uni-hannover.de)

Abstract. This article presents a method for traffic control recognition at junctions (traffic lights, stop, priority and right of way rule) using crowd-sensed GPS data (vehicle trajectories), as well as features extracted from OpenStreetMap. Traffic regulators are not mapped in most maps, although the way they regulate traffic at intersections affects the traffic flow and therefore the vehicle idle time at intersections, the fuel consumption, the CO_2 emissions, and the arrival time at a destination. Because of the controlled interaction that road users have with each other at intersections, driving safety or assistance applications can be enabled if intersection regulators are mapped. In order to verify the proposed method two sets of trajectories were used, one of which is an open dataset, from two different cities, Hannover and Chicago. Two classification methods were tested, random forest and gradient boosting, using exclusively either dynamic features (trajectories), or static (only data from OSM) or a combination of the dynamic and static features (*hybrid* model). The results show that the gradient boosting classification with hybrid features can predict traffic regulations with high accuracy (93% in Chicago and 94% in Hannover), outperforming the other detection models (static and dynamic). At the end directions for further research on this topic are proposed.

Keywords. traffic regulator detection, traffic signs, GPS trajectories, crowd-sensing, road-maps, classification

1 Introduction

In recent years, the analysis of motion sequences (GPS trajectories) has become increasingly important for companies and individuals. Especially with the advent of smartphones as low-cost sensors, recording sensor data such as GPS trajectories has become trivial. Thus, a large amount of GPS trajectories are generated around the globe on a daily basis. These trajectories can be analyzed for various purposes. One of them is for the automatic map updating or map enhancement. In this process, additional map information is generated from the collected GPS trajectories with the goal of either adding contextual information that is not yet available, or updating existing but outdated information. Such kind of information can be, e.g., the automatic detection of road network changes (Shan et al., 2015; Tang et al., 2019; Gao et al., 2021), the estimation of traffic flow (Li et al., 2021; Tu et al., 2021), the traffic signal waiting time (Lian et al., 2021; Yoshioka et al., 2022) and the determination of road roughness (Wage and Sester, 2021; Hiremath et al., 2021).

Another example of such contextual information is traffic control systems, which are used to control the traffic of road users such as vehicles, bicycles and pedestrians at intersections. Traffic regulators at intersections contain important navigational information and could, for example, help in deriving more accurate travel time estimates. Traffic regulators, such as traffic signals, have a significant impact on traffic flow at intersections, which in turn contribute to increased fuel consumption because they involve queuing conditions and many stop-and-go events. According to Alshayeb et al. (2021), intersections are one of the main spots where excessive fuel is consumed. In addition, traffic signals contribute more to air pollution compared to other types of controls due to excessive vehicle emissions at these locations (Gastaldi et al., 2014). Furthermore, the type of regulators is also critical for the development of autonomous vehicles and decision making regarding their behaviour as road users. Therefore, it is important that this information related to traffic regulators is present on maps and is up-to-date.

When examining public map databases, such as OpenStreetMap (OSM), it is observed that many intersections are missing information on traffic regulations. The Figure 1 shows the OSM from a very central and busy area of a large sized city in northern Germany, where traffic regulations are only available for five of the fifty illustrated intersections. This is remarkable, as in the city there is a very active OSM community, leading to very detailed OSM maps in general. The task of automatically detecting and identifying (the type of) traffic regulators can be

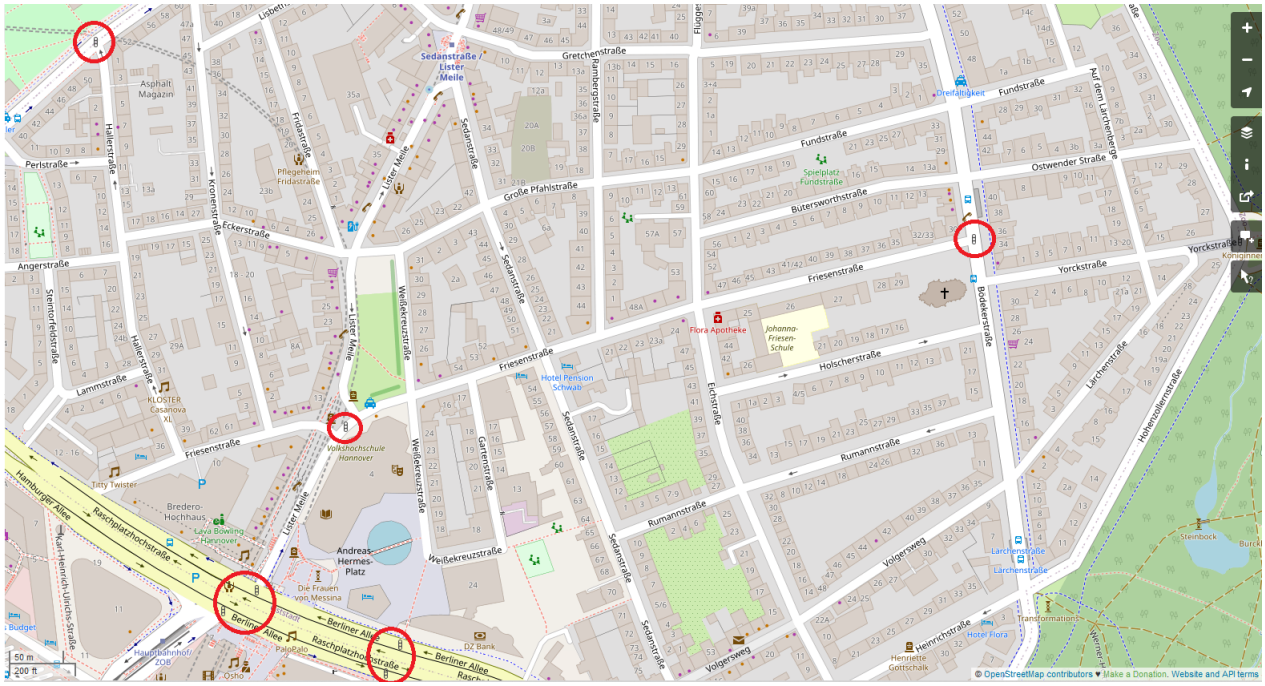


Figure 1. In red are depicted the intersections for which OSM contains regulators information. For all the other intersections there is no such information available.

solved by using different data sources. First, images from cameras or mobile mapping systems can be used to create a traffic sign inventory system. However, acquiring images with such equipment has high time and operational costs. Even using Google Street View images for such purposes would have cost constraints, since access to the associated API for large-scale usage would require a license. A second more time- and cost-friendly solution for tracking traffic regulators is GPS trajectories, which can be easily collected from GPS-enabled devices such as smartphones, and reveal the distinct behaviour of drivers when approaching intersections. GPS trajectories can reveal the spatiotemporal behavior of moving objects, they have already been used in various applications in this context, such as understanding the spatiotemporal behavior of tourists (Yao et al., 2021), mining medical periodic patterns by identifying periodic visits to medical centers or health professionals (Zhang et al., 2018), and extracting spatiotemporal routine patterns of people's (typical) commute patterns (Qin et al., 2018). Finally, another free solution for detecting intersection regulators is to use intersection connectivity features, such as the length of the road an intersection belongs to and the distance of an intersection from neighboring intersections. This information can be extracted without changes from OSM.

The work presented in this paper uses vehicle GPS trajectories combined with open data derived from OSM to identify the types of traffic regulators that control intersections. The regulator types are predicted using machine learning (random forest and gradient boosting) for each intersection approach (arm). The proposed method is tested on two datasets, each containing different traffic regulators.

The paper is organized as follows. In Section 2, the related research work is presented. In Section 3, the datasets used to test the proposed approach are introduced. The methodology is described in Section 4. In Section 5 the results of the experiments conducted on the two datasets are presented. A discussion of the main findings as well as future research directions can be found in Section 6 and 7, respectively.

2 Related Work

In this section, we briefly describe the main existing work on traffic rule recognition from GPS trajectories (Section 2.1). Based on their limitations that we identify, we explain the motivation of the work presented in this paper and enumerate its contribution to the existing body of research (Section 2.2).

2.1 Existing Research Work

A comprehensive systematic literature review was conducted by Zourlidou and Sester (2019) on the methods and datasets investigating the detection and identification of different traffic regulators at traffic intersections based on GPS trajectory data from the population. They analyzed several available articles and found that GPS trajectories provide high predictive potential (over 80%). However, no study examined all available regulator types at intersections using the same methodology or dataset. The various traffic regulator types found in the literature are summarized in the following categories: traffic signals (TS), stop

sign (SS), priority sign (PS), yield sign (YS), uncontrolled intersections (UN), roundabout (RB) and turning restrictions (TR). It should be noted, however, that most studies cover only a subset of the available regulator types due to limitations in the datasets.

The most recent study to our knowledge is that of Liao et al. (2021), which presents a framework for detecting and assessing traffic signals using a Deep Long Short-Term Memory (DLSTM) network that achieves an Area Under the ROC Curve (AUC) value of 0.95. Their goal is to detect traffic signals and provide an estimation of the potential area of influence based on the corresponding GPS trajectories and other intersection contextual characteristics such as intersection type, road type, and traffic flow information. Another solution to the same binary classification problem was proposed by Méneroux et al. (2020), where they investigated traffic signal detection using speed profiles. They found that a random forest classifier and a feature extraction technique where functional analysis of speed measurement series is combined with a wavelet transform performed better among other tested approaches (95% accuracy).

Golze et al. (2020) achieved 88% accuracy in predicting intersection regulation types (traffic signals, priority signs and unregulated junctions) by using a random forest classifier in conjunction with oversampling and enabled bagging booster. They used physical features such as standstill events (number of events and the duration of closest event to the junction), distance to the approaching intersection, vehicle speed, and percentage of trajectories with at least one standstill event. In addition, statistical features (minimum, maximum, mean, variance) were calculated for each physical feature similarly to Hu et al. (2015) and Saremi and Abdelzaher (2015). On the same multiclass classification problem, Cheng et al. (2020) trained a deep learning classifier using speed profiles as classification features. They used a conditional variational autoencoder that achieved a prediction accuracy of 90%.

Among the older works, the innovative works of Saremi and Abdelzaher (2015) and Hu et al. (2015) stand out. The methodology of Saremi and Abdelzaher (2015) is noteworthy because it is the only approach that has used features extracted from maps (OSM) to detect traffic controllers. Motivated by the fact that regulators in the USA are positioned based on certain criteria, such as the number and angle of intersection arms and vehicle speeds, they proposed an inference method for regulators that uses such static features that can be extracted from OSM. Moreover, on availability of dynamic crowd-sensed information (GPS tracks), the inference model is enhanced by incorporating the additional dynamic information and indeed improves classification performance. A random forest classification is used to predict three types of regulators: traffic signals, stop signs and unregulated intersections. The overall classification accuracy, with a confidence level of 80% in the prediction, is reported as 97%. Unfortunately, no detailed classification report is given describing the results per regulator class, nor quantitative information on the datasets used. Only the total number of intersection approaches is given, not e.g. the number of regulators per regulator class. Although we acknowledge this work as the one that is methodologically closest to the research presented here, we cannot directly compare our results with it.

Finally, Hu et al. (2015) using the duration of the last stop, minimum crossing speed, number of delays, number of stops, and distance from the intersection of the last stop as features (min, max, mean and variance). They investigate supervised and unsupervised methods with different feature and implementation settings to address a classification problem with three classes. Their results show that a random forest classifier with enabled active- and self-learning adapters achieves over 90% accuracy with 28% of the training data.

2.2 Contributions

As mentioned at the beginning of the previous section, no research work has so far been applied to different datasets with different classes of regulators to evaluate the ability to generalize the method to different classes of rules. A method may be good at predicting traffic signals (lights), stop signs and uncontrolled intersections, but is it equally good at predicting priority signs? Is a single classification feature *sufficiently* descriptive to distinguish between different groups of regulators? In addition, some methods are applied in the context of two-class classification (traffic signals, not traffic signals) problems providing excellent performance, but would they be equally good if they are applied to three-class problems?

It becomes clear that open reference datasets are missing so far, in which both the trajectories and the regulators' ground-truth map (label information) are available and researchers can use it as a benchmark when it comes to comparing the classification performance of the proposed approaches. Since for the scope of this paper we manually mapped the regulators of the road network of an open trajectory dataset, as explained in Section 3, we placed this dataset in an open repository so that other researchers can test their methodologies and compare the classification results with others.

The work presented in this paper differs from existing research in the following ways 1) a feature vector is proposed that includes the number and duration of deceleration events in addition to vehicle stop events. Moreover, in addition to the duration of the last stop event, the total duration of all stops and decelerations before crossing an intersection is also considered. 2) The proposed methodology is tested on two datasets with different regulators, and 3) a powerful classification model is tested, along with random forest, which has dominated many Kaggle competitions recently, the XGBoost, that is an implementation of the Gradient Boosted Decision Trees algorithm. Finally, an additional contribution is: 4) the ground-truth regulators'

map of an open trajectory dataset created for the needs of this work is made publicly available for other researchers.

3 Data

3.1 Data description

Two datasets from the cities Hannover and Chicago were used to test the proposed method for traffic regulator identification. The trajectories of the Hannover (Germany) dataset were collected by the authors themselves with an average sampling rate of 1Hz. The Chicago dataset (USA) is publicly available by Ahmed et al. (2015) and has an average sampling rate of 0.28Hz. Necessary to this work is the availability of ground-truth information of intersection regulators which is required both for the learning part of the method (training of a classifier) and for evaluating its predictive performance. The ground-truth information for the Hannover dataset was created manually by on-site observation (visiting the intersections and mapping their regulator types). For the Chicago dataset, we used the Mapillary street imagery data (Mapillary, 2022), verified by other sources, and manually extracted the intersection rules.

Each intersection is divided in intersection arms or approaches. Therefore, a four-way intersection has four arms, whereas a T-junction has three. Often not all arms of the same intersection are controlled by the same regulator (e.g. priority/yield controlled intersections), the notion of traffic regulator in the context of this work refers to the regulator of each intersection arm. As such, the Chicago dataset has 156 intersections with 499 arms (regulators) and the Hannover dataset 1064 intersections with 3523 arms. Regarding the regulator types, in Chicago dataset 93 arms are uncontrolled (UN), 139 are stop sign (SS) and 264 traffic signal (TS) controlled. In Hanover, 850 are UN, 1199 are priority-controlled (PS), 558 are yield controlled (YS), 26 are SS and 890 are TS. Figure 2 gives an overview of the extent of the collected GPS trajectories in the examined cities. The Chicago dataset has fewer junctions compared to Hannover (156 vs. 1064) but each junction arm is on average more densely sampled by the available trajectories. The Hannover dataset has 1204 trajectories and the Chicago 889. The minimum, the 1st, 2nd and 3rd percentiles, as well as the average and maximum of trajectory crossings per intersection arm is 1, 3, 11, 84, 54, 373 for Chicago and 1, 1, 3, 9, 12, 213 for Hannover.

As for the regulator rule classes, the Hannover dataset contains traffic signals, priority, yield and stop signs, as well as unregulated intersections (right-of-way rule). Since there are only 26 stop-controlled intersection arms, we excluded this regulator type from the classification due to insufficient samples for training and testing. Similarly, the yield-controlled intersection category was excluded because the trajectories crossing yield-controlled intersections were very sparse. Of the 558 yield-controlled arms, only 122 were sampled with at least one trajectory, and most of these arms had very few trajectories. 436 yield-controlled arms were not crossed by any trajectory. Therefore, this category was also omitted. With respect to the Chicago dataset, the roads are controlled by three types of regulations: traffic signals, stop signs, and right-of-way priority (uncontrolled intersections).

OpenStreetMap (OSM contributors, 2022) provides freely usable spatial data for almost all places around the globe and has become known as the most popular Volunteered Geographic Information (VGI) system (Jokar Arsanjani et al., 2015). In this work, given the trajectories that cross certain intersections in Chicago and Hannover datasets, we exported road network distance related information for those intersections from OSM.

That way we have for each of the two cities, two datasets: one that contains *dynamic* information (trajectories) and another with *static* information (intersection related distances). More details about the feature extraction from the two datasets are given in Section 4.2.

3.2 Data and Software Availability

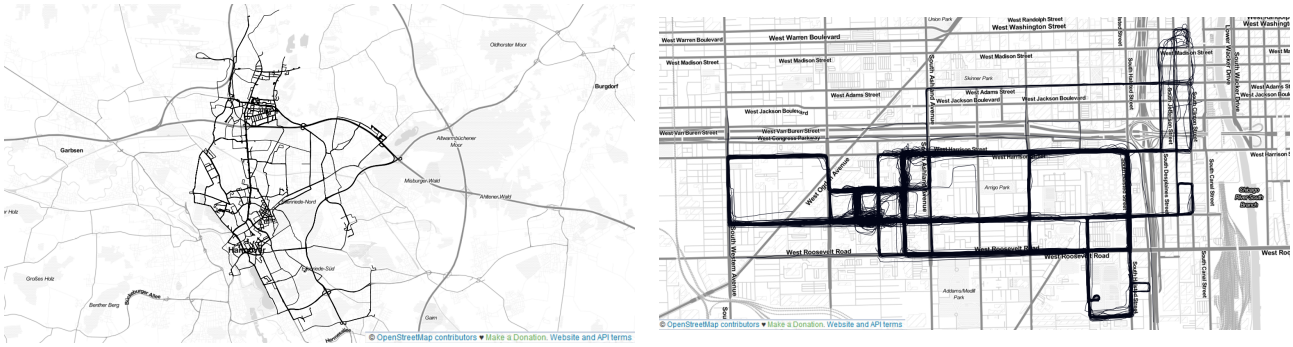
The groundtruth map from the Chicago dataset, is accessible in a public repository (Zourlidou and Golze, 2022) and has an open data licence. The Chicago trajectory dataset, as already mentioned, is publicly available by Ahmed et al. (2014). The Hannover dataset is not available due to privacy restrictions. The trajectories have been collected from the authors, capturing the daily car drives from home to work, shopping centers, kids' schools, etc., therefore the containing information could compromise the privacy of research participants. The Python code of all experiments is the intellectual property of the first author of this paper and intends to use and extend it for further investigations in the context of on-going doctoral studies. Therefore, the code will be available after completion of the relevant studies and publications.

4 Methodology

The proposed approach can be divided into three main steps, which are explained in the following sections. First, the trajectory data is pre-processed so that each trajectory is associated with the intersection approaches it crosses. Afterwards, features are extracted from the pre-processed trajectories (*dynamic* model) and from the OSM (*static* model), for each intersection arm. In the last step, the classification process is performed under different experimental settings.

4.1 Data Pre-processing

The input data, as previously mentioned, are twofold: trajectory data and OSM. Each trajectory is a time-ordered sequence of GPS data and must be associated with the in-



(a) Hannover dataset.

(b) Chicago dataset.

Figure 2. The two trajectory datasets used in this study, (a) Hannover dataset and (b) Chicago dataset.

tersections it crosses. Therefore, in the first step of data pre-processing, trajectories are associated with all the intersections they cross from the beginning to the end of their journey. This way each intersection arm of the dataset is associated with all the crossing trajectories and each trajectory is associated with all the intersection arms that crosses.

An important aspect of the methodology is that the classification process applies to each intersection approach of an intersection and not to the entire intersection. Due to this fact, not all arms of the same intersection are always regulated by the same rule. For example, a T-intersection may be controlled by one-way stop, or a four-way intersection may be controlled by priority/yield. Therefore, the classification features describe each intersection arm. For the *dynamic* model, classification features are calculated from the trajectories within the road segment connecting the respective intersection arm to the adjacent intersection arm. This is done so that the implementation is not parameter-depended and is valid for each dataset and for all intersections in a dataset. For example, we could compute the features of an arm at a distance of 50 meters from the center of the intersection, but this distance might be large for some intersections (it could also capture nearby intersections within this radius, not just the road segment connecting two junctions) or very small for large intersections.

In addition, for each trajectory that crosses an intersection arm, the direction of crossing is calculated, i.e. whether the vehicle is going straight or turning left or right. This is done by calculating the angle of entry and exit from the respective intersection arm. This information is needed to be able to select, e.g. only the straight trajectories from the dataset for feature calculation. Similarly to Hu et al. (2015) and Golze et al. (2020), we ignore the turning trajectories because the turning maneuver affects the crossing speed and causes deceleration and stopping events related to the maneuver itself rather than to the traffic regulators.

For the *static* model, where the features are calculated from OSM, the map is first downloaded from the area of interest and then the centers of the intersections are estimated. This information can not be obtained directly from OSM and is therefore estimated for each intersection as the point where different road segments intersect each other. For each intersection approach the features are then calculated as explained in Section 4.2.2.

4.2 Feature Definition

In this section we explain the features we derive from trajectories (*dynamic* model) and OSM (*static* model). The hybrid approach uses the extracted features both from trajectories and OSM.

4.2.1 Dynamic model

The set of features resulting from GPS trajectories is called *dynamic* (Saremi and Abdelzaher, 2015), as trajectories are not static entities. These dynamic features are estimated for each individual intersection arm from the trajectories that cross it and are summarized in Table 1. They concern various aspects of the stopping and deceleration events that take place within half the distance of the road segment connecting each intersection arm to its neighbour, as well as the speed at which vehicles pass.

Figure 3a depicts a four-way intersection with four approaches, namely $j-arm$, $k-arm$, $i-arm$ and $l-arm$. Each of these arms is connected to an intersection arm from a nearby junction via a road segment depicted by green line. Afterwards, for each trajectory that crosses an intersection approach, the trajectory-features are calculated within half the green distance which is depicted by red dashed line in the same Figure 3a and is called a *semi-approach segment*. The per trajectory features are called physical (Hu et al., 2015) and are enlisted in the first column of Table 1. The per intersection arm features, calculated within the semi-approach segment are called statistical and are enlisted in the second and third column of the Table 1. More specifically the stopping/deceleration related features are the:

1. **number of stop/decel. events** identified in a trajectory $traj$ while crossing an intersection approach arm_i within the semi-approach segment of arm_i

(a) Dynamic model

(b) Static model.

Figure 3. (a) Illustration of the road segments along which the features of the dynamic model are calculated (red-dashed lines). (b) Illustration of the road segments along the approaches of an intersection which are used as features in the static model (red, green and red lines).

Table 1. Overview of the features derived from the *dynamic* model.

	Physical Feature*	Statistical Features derived from Physical**	
Stop events	Number of stops	avg_nstops	var_nstops
	Duration of last stop	avg_dur_lstop	var_dur_lstop
	Duration of all stops	avg_dur_all_stops	var_dur_all_stops
	Mean Duration of all stops	avg_mean_dur_all_stops	var_mean_dur_all_stops
	Median Duration of all stops	avg_med_dur_all_stops	var_med_dur_all_stops
	Distance of last stop	avg_dist_lstop	var_dist_lstop
	Mean Distance of all stops	avg_mean_dist_all_stops	var_mean_dist_all_stops
	Median Distance of all stops	avg_med_dist_all_stops	var_med_dist_all_stops
Decel. events	Number of decel. events	avg_ndecel	var_ndecel
	Duration of last decel. event	avg_dur_ldecel	var_dur_ldecel
	Duration of all decel. events	avg_dur_all_decels	var_dur_all_decels
	Mean Duration of all decel. events	avg_mean_dur_all_decels	var_mean_dur_all_decels
	Median Duration of all decel. events	avg_med_dur_all_decels	var_med_dur_all_decels
	Distance of last decel. event	avg_dist_ldecel	var_dist_ldecel
	Mean Distance of all decel. events	avg_mean_dist_all_decels	var_mean_dist_all_decels
	Median Distance of all decel. events	avg_med_dist_all_decels	var_med_dist_all_decels
Speed	Minimum speed	avg_min_speed	var_min_speed
	Maximum speed	avg_max_speed	var_max_speed
	Average speed	avg_avg_speed	var_avg_speed

* Derived per trajectory, ** Derived per intersection arm, i.e. from all trajectories that cross the intersection approach

2. **duration of the last stop/decel. event** (closest to intersection junction center) of $traj$ within the semi-approach segment of arm_i

3. **duration of all stop/decel. events** of $traj$ within the semi-approach segment of arm_i

4. **mean duration of all stop/decel. events** of $traj$ within the semi-approach segment of arm_i

5. **median duration of all stop/decel. events** of $traj$ within the semi-approach segment of arm_i

6. **the distance of the last stop/decel. event** of $traj$ within the semi-approach segment arm_i from the intersection center

7. **mean distance of all stop/decel. events** of $traj$ within the semi-approach segment arm_i from the intersection center

8. **the median distance of all stop/decel. events** of $traj$ within the semi-approach segment arm_i from the intersection center

The speed-related features are:

1. **minimum speed** of the vehicle of $traj$ while approaching the intersection through the intersection arm arm_i

2. **maximum speed** of the vehicle of $traj$ while approaching the intersection through the intersection arm arm_i

3. **mean speed** of the vehicle of $traj$ while approaching the intersection through the intersection arm arm_i.

Therefore, 19 physical features describe how a trajectory crosses an intersection arm. In order to describe the intersection approaches we compute the average and variance of the physical feature values of all trajectories crossing an intersection arm. Hence, in the *dynamic* model, each arm is represented by a feature-vector with 19x2= 38 values.

The algorithmic approach to realize the estimation of the dynamic features is described in Algorithm 1.

Algorithm 1 Traffic Regulation Detection from GPS Data

Data: $TrajSet$: GPS tracks, Y: ground truth regulators' labels
Result: Y_{pred}: Predicted regulator labels
for $\forall$ $traj$ *in* $TrajSet$ **do**
  Detect all stop events within $traj$
  Detect all deceleration events within $traj$
end
for $\forall$ *intersection arm* arm_i **do**
  Find the arm_j that is connected with arm_i
  Find the trajectories that cross arm_i
  for $\forall$ $traj$ *that crosses* arm_i **do**
    Find the turn direction (straight, left, right)
    Find the stop events of $traj$ within the $semi-approach$ segment defined between arm_i and arm_j
    Find the decel events of $traj$ within the $semi-approach$ segment defined between arm_i and arm_j
    Compute the crossing speed
  end
  Add feature vector of arm_i to $FeaturesDBT$
end
Classification with data from $FeaturesDBT$
Print classification report

4.2.2 Static model

The second set of features is extracted from OSM and referred to as static. They are proposed by Saremi and Abdelzaher (2015) and similar to the dynamic features, they are calculated for each intersection arm. The five static features are depicted in Figure 3b; their description is as follows:

1. **maxspeed** is the maximum allowed speed along the arm_i. Intersections controlled by a traffic signals are expected to be on roads with a higher maximum speed than, for example, intersections controlled by a stop sign.

2. **category** refers to the road type category of the arm_i (e.g. primary, secondary, tertiary, residential). The rationale for using this feature is similar to 1.

3. **e2e** is the end-to-end distance of the road that the arm_i belongs to (blue line in Figure 3b). The length of a road can reflect its importance in the road network. For example, an avenue may serve more traffic than a 50-metre-long road and therefore their intersection regulations may differ accordingly. The same rationale applies also to the other distance-based features (4, 5).

4. **semi_dist** of an approach arm_i is the distance from the center of the junction that the arm_i belongs to, until the center of the most distant junction that the arm_i is connected to (green line in Figure 3b).

5. **close_dist** of an approach arm_i is the distance from the center of the junction that the arm_i belongs to, until the center of the nearest junction that the arm_i is connected to (red line in Figure 3b).

In this work we examine two feature settings for the *static* model. One is along with the work of Saremi and Abdelzaher (2015), referred from now as *default setting static* model, containing an extended feature set. The other one is a simplification of it, referred to as *simplified-setting* and results to fewer features. The knowledge representation for arm_i under the *default setting* is:

$$id\_i, (speed_i, cat_i, e2e_i, semi_i, close_i), (speed_l, cat_l, e2e_l, semi_l, close_l), (speed_j, cat_j, e2e_j, semi_j, close_j), (speed_k, cat_k, e2e_k, semi_k, close_k)$$

where for the description of an arm_i is also included the information from the adjacent arms (arm_l, arm_j and arm_k) of the same junction. Under the *simplified setting*:

$$id\_i, (speed_i, cat_i, e2e_i, semi_i, close_i)$$

where only the information for the arm_i is included.

The classification steps for the *static* model are described in Algorithm 2.

Algorithm 2 Traffic Regulation Detection from OSM Extracted Features

Data: OSM, Y: ground truth regulator labels
Result: Y_{pred}: Predicted regulator labels
$interSet \leftarrow$ extracted Intersection Centers
for $\forall$ *arm*$_i$ *of* $inter_i \in$ *interSet* **do**
  Compute maxspeed, category, e2e_dist, semi_dist, close_dist
  Add feature vector of arm_i *to* $FeaturesOSM$
end
Classification with data from $FeaturesOSM$
Print classification report

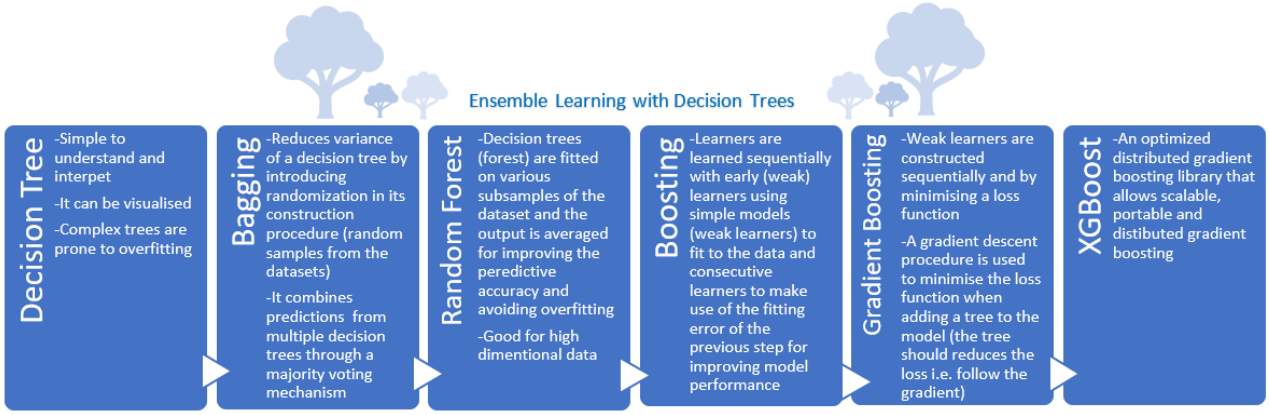


Figure 4. The evolution of a single Decision Tree to the gradient boosting approach XGBoost.

4.2.3 Hybrid model

The *hybrid* model is a composite of the *dynamic* and *static* model, using all the features of both models. As shown in Section 5, although the *hybrid* model outperforms the two models, additionally a feature selection is applied to identify the most important features as well as a tuning of some important parameters of the classifier. In the next section we explain the classification settings for testing the proposed methodology.

4.3 Classification Settings

The classification of traffic regulator types is done using supervised learning techniques. Three different settings, one for each model (*static*, *dynamic*, *hybrid*) are tested for each of the two datasets. In addition, two classification methods are tested for each experiment. The first method is the random forest classifier, which consists of several individual decision trees. These decision trees are trained on a random subset of the available training data using the bagging approach. The result of a random forest is the sum over all its predictors.

The second method, is the gradient boosting approach. In the context of the problem it can be considered as a gradient descent minimization procedure over a loss function estimated each time a tree is added to the model (weak learner). For the implementation we used the XGBoost (XGBoost Python, 2022) library which has recently dominated many Kaggle competitions. The evolution of a simple decision tree up to XGBoost is illustrated in Figure 4, where the main features of the ensemble tree-based learning are given.

To evaluate the classification process, the *accuracy*, the *F-measure* and the *true positive (TPR)*, sensitivity, and *false positive rate (FPR)*, fall-out, are estimated. The accuracy is the percentage of correctly classified samples, while the F-measure is the harmonic mean of precision and recall, which are calculated from true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). These four elements are represented in a confusion matrix, where the (per class) performance can be visually evaluated. The TPR and FPR are defined as:

$$TPR = \frac{TP}{TP+FN}; \quad FPR = \frac{FP}{FP+TP} \tag{1}$$

5 Results

5.1 Static Model

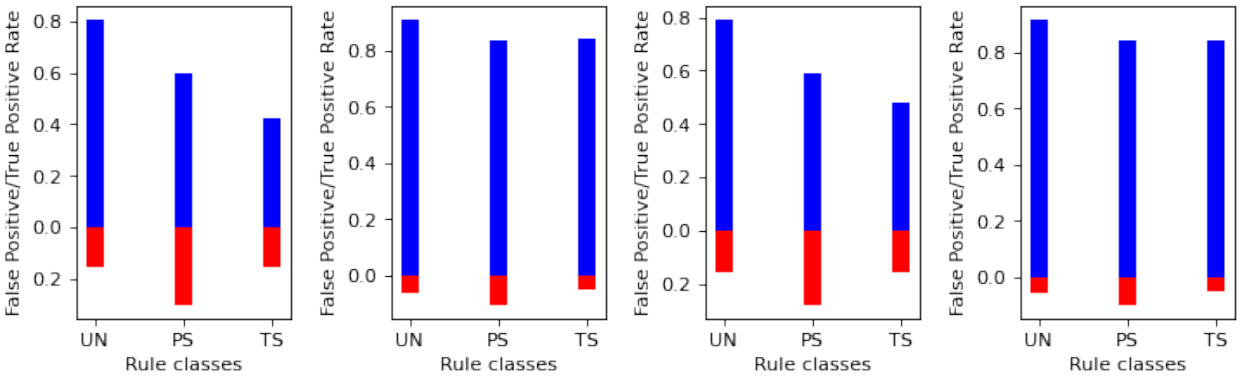
The Table 2 shows the classification results for the *static* model in the two datasets using both classification methods. A more detailed classification report with the performance by regulator type can be found in Appendix A1.

Regarding the simplified setting *static* model, in Hannover, gradient boosting (GB) performs slightly better than random forest (RF), with an F-Measure of 0.61 and 0.60 respectively. In Chicago, both classifiers show similar performance with an F-measure of 0.71. Although the performance in Chicago is better than in Hannover, it is still low. This low performance is also highlighted by the true positive (TPR) and false positive (FPR) rates depicted in Figure 5. In Hannover, priority signs (PS) have the highest FPR (see Figure 5a and 5c) compared to uncontrolled intersections (UN) and traffic signals (TS), while in Chicago, TS have the highest FPR, which is justified by the very low TPR of UN (compare Figure 5e and 5g). The average TPR and FPR in Hannover for the GB model are 0.66 and 0.16 and in Chicago 0.62 and 0.2, respectively.

Regarding the default setting *static* model, in Hannover, gradient boosting performs slightly better than random forest, with F-Measure 0.87 and 0.86 respectively. In Chicago, RF performs better than GB, with an F-measure of 0.85 versus 0.83 for GB. Performance, as depicted in the TPR/FPR plots (see Figure 5), shows no differences between the regulator types, as in both datasets TPR and FPR have similar per class values (Figure 5b, 5d, 5f, 5h).

Table 2. Classification results of the *static* model.

Dataset	Classifier	Recall		Precision		F-Measure		Accuracy	
		simpl. setting	default setting	simpl. setting	default setting	simpl. setting	default setting	simpl. setting	default setting
Hannover	RF	0.61	0.86	0.60	0.86	0.60	0.86	0.61	0.86
	GB	0.61	0.87	0.61	0.87	0.61	0.87	0.61	0.87
Chicago	RF	0.73	0.85	0.71	0.86	0.71	0.85	0.73	0.85
	GB	0.72	0.84	0.71	0.84	0.71	0.83	0.72	0.84



(a) Simplified (RF), Hanover. (b) Default (RF), Hanover. (c) Simplified (GB), Hanover. (d) Default (GB), Hanover.

(e) Simplified (RF), Chicago. (f) Default (RF), Chicago. (g) Simplified (GB), Chicago. (h) Default (GB), Chicago.

Figure 5. True positive (blue) and false positive (red) rates of the *static* model (simplified and default feature settings) for the Hannover (first row) and Chicago datasets (RF: random forest, GB: gradient boosting).

In Hannover, UN has a slightly better TPR than the other classes (Figure 5b and 5d), while in Chicago, TS has the highest TPR, although not much higher than the other classes (compare Figure 5f and 5h). The average TPR and FPR in Hannover for the GB classifier are 0.87 and 0.07 and in Chicago for the RF model 0.84 and 0.08 respectively.

Therefore, for both the simplified and default setting *static* models, the choice of classifier (RF versus GB) makes a small difference. However, the default setting outperforms the simplified one, which means that indeed for the classification of a intersection approach, information from the neighboring arms belonging to the same intersection is an important aspect. Hence, for the remaining experiments the default setting features are used.

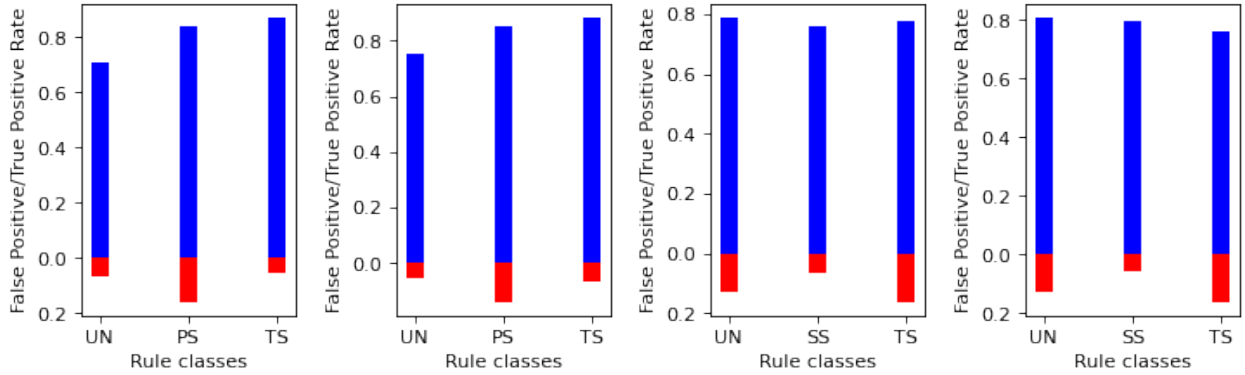
5.2 Dynamic Model

Table 3 shows the classification results for the *dynamic* model on the two datasets. A more detailed classification report with performance by regulator type can be found in the Appendix B1. For the Hannover dataset, GB performs better than RF (F-measure 0.85 vs. 0.83). Similarly, in Chicago, GB has an F-measure of 0.78 while RF has an F-measure of 0.77.

In Hannover, for both classifiers, the highest FPR is observed in the PS category, which can be explained by the low TPR of the UN category (see Figure 8a and 8b). This implies that UN intersection arms are erroneously classified more often as PS controlled. In addition, the PS and TS categories show a higher TPR than 80%. The average

Table 3. Classification results of the *dynamic* model.

Dataset	Classifier	Recall	Precision	F-Measure	Accuracy
Hannover	RF	0.83	0.84	0.83	0.83
	GB	0.85	0.86	0.85	0.85
Chicago	RF	0.78	0.79	0.77	0.78
	GB	0.78	0.79	0.78	0.78



(a) RF, Hanover. (b) GB, Hanover. (c) RF, Chicago. (d) GB, Chicago.

Figure 6. True positive (blue) and false positive (red) rates of the *dynamic* model for the Hannover and Chicago datasets (RF: random forest, GB: gradient boosting).

TPR and FPR for the RF and GB classifier in Hannover are 0.81 and 0.09 (RF) and 0.83 and 0.085 (GB).

In Chicago, for both classifiers, the highest FPR is observed in the TS category, which is close to the corresponding measure for the UN class (see Figure 6c and 6d). TPRs are similar within the three regulator classes, with the lowest FPR observed in the SS category. However, the FPR for the UN and TS samples is quite high, which given the high FPR of SS, means that the UN class is more often misclassified as TS and TS as UN (this fact is easily visible in the confusion matrices in the Appendix C1). Interestingly, such behavior is not noticeable in the results of the Hannover dataset. The average TPR and FPR for the RF and GB models are 0.77 and 0.12 (RF) and 0.79 and 0.11 (GB), respectively.

When comparing the *dynamic* to the *static* model, it outperforms the simplified setting *static* model but falls short of the default setting for both datasets. In Hannover, the default setting *static* model has an F-measure of 0.87, while the dynamic model has 0.85. In Chicago, the difference between the *static* and *dynamic* models is larger, with the *default setting static model* having an F-measure of 0.85, while the *dynamic* model has an F-measure of 0.78. One explanation for the poor performance of the *dynamic* model on the Chicago dataset may be the low sampling rate of the GPS samples, which may affect the detection of short duration deceleration/stop events.

As previously reported in Section 3, the Chicago dataset has an average sampling rate of 0.28Hz, which is approximately one sample every three seconds. This means that the detected movement events as well as the speed estimated from the GPS points are "smoothed out" and no events happened within those 3 sec can neither be recovered nor detected. What we can estimate is only the difference in speed from time t to time $t + 3sec$ (on average) and the movement events that have a duration longer than the time interval between at least two GPS samples.

5.3 Hybrid Model

As mentioned in Section 4.2.3, by combining the features of the *static* and *dynamic* models, we create a new model (*hybrid* model). We tested this model in both of the datasets, with the RF and GB classifiers. The results are presented in Table 4. On the Hannover dataset, as shown in Exp.1 and Exp.2, GB performs better than RF, with an F-measure of 0.92 (GB) versus 0.89 (RF). Similarly in Chicago, Exp.5 and Exp.6, GB performs 0.91 versus 0.89 for RF. Interestingly, the performance of the *hybrid* model in the two datasets is similar, regardless of the classifier, compared to the dynamic model, where a significantly higher performance in Hannover than in Chicago was observed (Table 3).

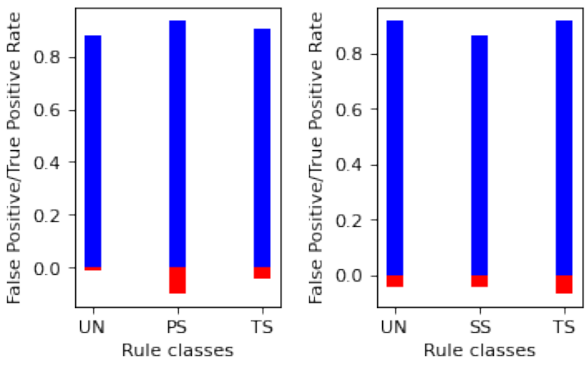
Regarding the TPR/FPR of the GB classifier on the two datasets, as seen in Figure 7, the TPRs of the three regulator classes are similar. In Hannover, the TPRs for UN, PS and TS are 0.88, 0.94, 0.90 and in Chicago for UN, SS and TS, 0.92, 0.86 and 0.92 respectively. In Chicago, FPSs are also similar across regulator categories (UN: 0.04, SS: 0.04, PS: 0.064). In contrast, in Hannover, the FPR of PS is

Table 4. Classification results of the *hybrid* model.

Dataset	Exp.	Classifier	Feature Selection	Tuning	Recall	Precision	F-Measure	Accuracy
Hannover	Exp.1	RF	No	No	0.89	0.89	0.89	0.89
	Exp.2	GB	No	No	0.92	0.92	0.92	0.92
	Exp.3	GB	Yes	No	0.94	0.94	0.94	0.94
	Exp.4	GB	Yes	Yes	0.94	0.94	0.94	0.94
Chicago	Exp.5	RF	No	No	0.89	0.91	0.89	0.89
	Exp.6	GB	No	No	0.91	0.92	0.91	0.91
	Exp.7	GB	Yes	No	0.93	0.92	0.92	0.93
	Exp.8	GB	Yes	Yes	0.93	0.92	0.92	0.93

remarkably higher than in the other regulator classes (FPR in UN:0.01, PS: 0.096, TS: 0.04). The average values of TPRs/FPRs in Hannover are 0.91 and 0.05 and in Chicago 0.90 and 0.05. Compared to the average TPR/FPR of the best default setting *static* model (GB in Hannover and RF in Chicago), the *hybrid* model has an increase/decrease of 4%/2% in Hannover and 6%/3% in Chicago. Compared to the *dynamic* model, the *hybrid* model has a corresponding increase/decrease of 8%/4% in Hannover and 11%/6% in Chicago.

In addition, we examined whether selecting the most important features would further increase the performance of the GB model. Figure 8 illustrates the importance of each feature in the classification process. The first observation from the two graphs is that the important features in each dataset are different. This could come from the fact that each dataset contains different traffic regulators. Another possibility could be the influence of the sampling rate on the detection of movement events mentioned in a previous section. A second observation is that in the Hannover dataset only a few features are distinct in importance from the total of 58 features.



(a) Hanover. (b) Chicago.

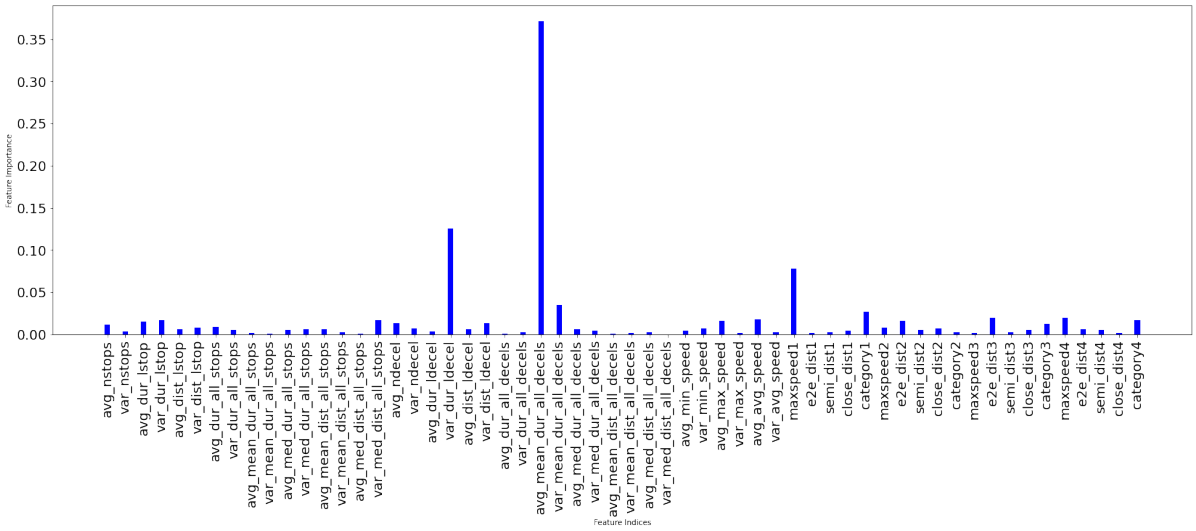
Figure 7. True positive (blue) and false positive rates (red) for the Hannover and Chicago datasets with gradient boosting classification (Exp.2 and Exp.6 from Table 4).

The $avg\_mean\_dur\_all\_decels$, var_dur_ldecel, $maxspeed1$, $var\_mean\_dur\_all\_decels$ and $category1$. In the Chicago dataset, overall, there are more important features compared to the few most important features in Hannover: var_mean_speed, $e2e\_dist4$, $category1$, $category2$, avg_min_speed, $semi\_dist1$, $avg_dist\_l\_stop$, $e2e\_dist1$, $close\_dist3$, avg_nstops, etc. Furthermore, the dynamic features in the Hannover dataset are mostly related to the deceleration events, whereas in Chicago to stop events. This finding is rational given that the Chicago dataset has stop signs in contrast to the priority signs in Hannover. Stop signs cause mostly stop events whereas at priority sign locations, deceleration events are mostly observed when e.g. the vehicle in front wants to turn in the intersection and decelerate in order to complete the turn maneuver (or stop if turn left). Furthermore, in Chicago, static features are more important than dynamic features. Given the low performance of the *dynamic* model in the Chicago dataset, we can see that in the *hybrid* model the static features make up for this difference and combined with the few important dynamic features the performance of the *hybrid* models outperforms the individual models.

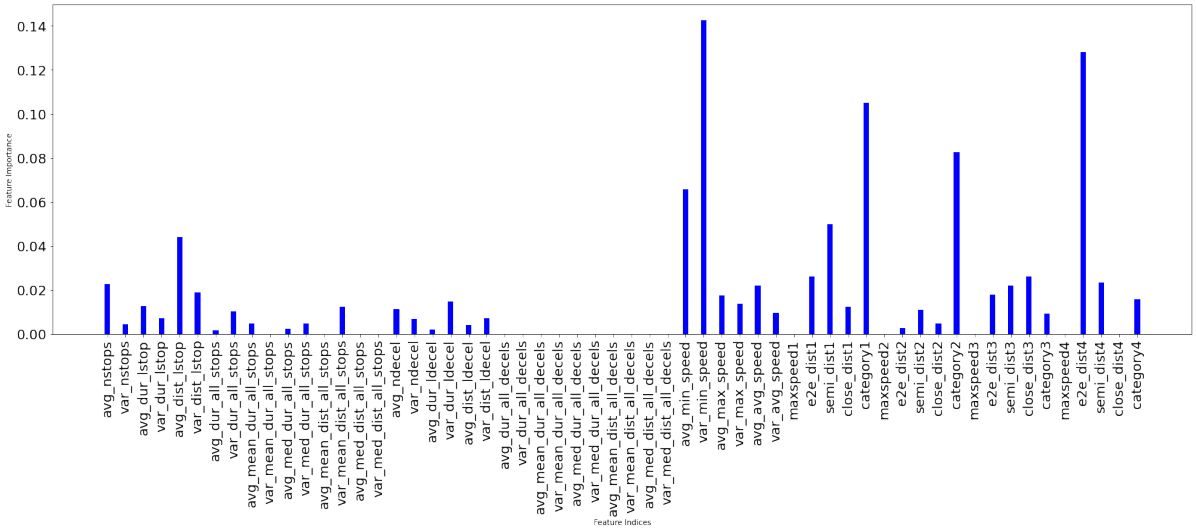
The results of the classification using only the most significant features are shown in Table 4, Exp.3 and Exp.7. In the Hannover dataset there is an improvement in F-measure and accuracy of 0.02, (from 0.92 in Exp.2 to 0.94) and in Chicago the F-measure increases from 0.91 to 0.92 and the accuracy from 0.91 to 0.93.

Finally, we experimented with tuning the parameters of the GB classifier, namely the number of decision trees, their size (depth), row subsampling, column subsampling (per tree and split) and learning rate. We omit to report the results of the tuning experiments due to space limitation. An example is illustrated in Figure 9, where we tuned two parameters together, the number and size of decision trees. As shown, the negative loss function is maximized for a maximum depth equal to 4 (red line), for 25 decision trees ($n\_estimators$).

Several experiments were carried out, tuning various combinations of the above-mentioned parameters, as well as tuning each parameter separately. The use of tuned row and column subsampling values did not improve performance on any of the datasets. By tuning the size and number of trees, as well as the learning rate, as shown in Ta-



(a) Hanover.



(b) Chicago.

Figure 8. Feature importance for the Hannover and Chicago datasets.

ble 4 (Exp.4 and Exp.8), the classification performance remained unchanged.

Regarding the TPRs and FPRs of the GB classifier (Exp.4 and Exp.8) where feature selection and parameter tuning are enabled, Figure 10 illustrates the increase in TPR and decrease in FPR for both datasets, across all categories, compared to the GB default model (Exp.2 and Exp.6, Figure 7).

In the Hannover dataset the average TPR/FPR is 0.93/0.04, while per regulator class it is in UN 0.91/0.007, PS 0.95/0.08 and TS: 0.96/0.03. The higher FPR in PS and lower TPR in UN indicates that UN is often misclassified as PS (compare confusion matrix Figure E1a in the Appendix). Compared to the *dynamic* model, the increase/decrease in the average TPR/FPR is 0.1/0.05, i.e., an increase in TPR of 12% and a decrease in FPR of 55%. Compared to the default setting *static* model, the in-

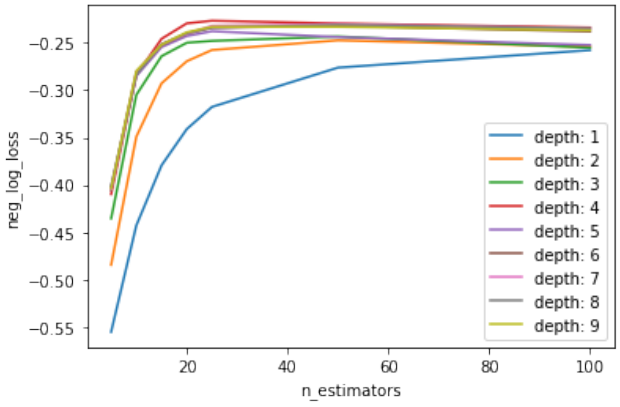
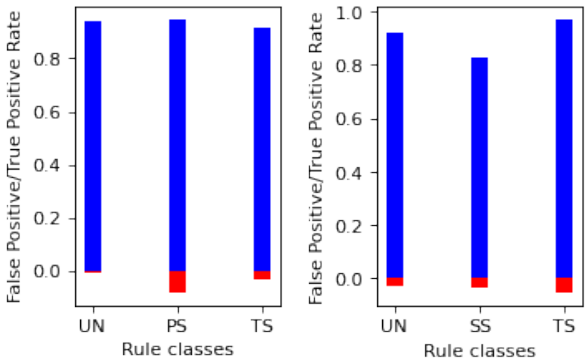


Figure 9. Tuning parameters of the gradient boosting classifier for the Hannover dataset: the number of decision trees $n\_estimators$ and the size of each tree, $depth$.



(a) Hannover. (b) Chicago.

Figure 10. GB models, with feature selection and parameter tuning enabled.

crease/decrease in the average TPR/FPR is 0.06/0.03, i.e. an increase in TPR by 7% and a decrease in FPR by 43%.

In the Chicago dataset the average TPR/FPR is 0.91/0.04, while by regulator class it is UN 0.92/0.03, SS 0.83/0.03 and TS: 0.97/0.05. Of interest is the lower TPR of SS compared to the other categories and the slightly higher FPR of TS. Compared to the *dynamic* model, the increase/decrease in the average TPR/FPR is 0.12/0.07, i.e. an increase in TPR of 15% and a decrease in FPR of 64%. Compared to the default setting *static* model, the increase/decrease of the average TPR/FPR is 0,07/0,04, i.e. an increase of TPR by 8% and a decrease of FPR by 50%.

6 Discussion

The main finding of this work can be summarized as follows:

1. The *dynamic* model has comparable performance to the default setting *static* model on the Hannover dataset, but significantly lower on the Chicago dataset. We believe that this problem is not a weakness of the proposed *dynamic* model, but is instead inherited due to the limitations of the dataset (low sampling rate).

2. The *hybrid* model outperforms the two individual models. Even in the Chicago dataset with the *weak* dynamic features, the map features combined with these weak features not only compensate the performance of the *dynamic* model, but reach comparable levels of accuracy/F-measure to the Hannover dataset, where the *static* and *dynamic* models perform similarly.

3. The feature selection, as well as the model tuning (number and size of trees, learning rate) of the gradient boosting classifier either improves or leaves the performance untouched.

4. The choice of classifier (random forest vs. gradient boosting) in the *static* model does not seem to affect the performance. However, in the other two models, where the number of features is larger than in the *static* model, gradient boosting outperforms random forest.

5. The *hybrid* model provided a performance between 0.92 and 0.94 of F-measure. In Hannover, it improved the TPR by 12% compared to the *dynamic* model and by 7% compared to the *static* model. The FPR also decreased by 55% and 43% respectively. In Chicago, it also improved TPR by 15% compared to the *dynamic* model and by 8% compared to the *static* model. FPR also decreased by 64% and 50% respectively.

6. When the performance is compared to existing works under similar data settings (e.g. same regulator types), the performance of the proposed model outperforms them, e.g. Cheng et al. (2020) and (Golze et al., 2020).

Finally, a future work topic is to examine the possibilities of *transferability* of the *hybrid* model from one city to another, i.e. training the model in one city and testing it in another one (e.g. training in Hannover and testing in Chicago). This aspect of the problem is motivated by the labeled data limitation discussed earlier.

Furthermore, semi-supervised techniques, such as *co-training*, which exploits different *views* of the data (i.e. two different feature sets that provide complementary information about the instance and are conditionally independent), are worth considering in the context of the problem, as they can provide good performance under small amounts of labeled data and large amounts of unlabeled data.

In addition, an important aspect of the problem to consider is whether the tested approaches (static, dynamic and hybrid) would work in smaller cities, where the distance from one intersection to another is shorter and therefore the extracted classification features are calculated over a shorter road length. Moreover, driving behaviour in such cities may be additionally influenced by sources other than traffic regulators, such as pedestrians who, knowing that cars are moving at low speed, may cross streets more freely. In addition, smaller cities may have an irregular street layout unlike Chicago which has an almost perfect grid-like network consisting of rectangular street blocks. The same question arises also for larger cities with more complex street networks than those considered in this research paper, such as Berlin and Athens.

Finally, another important aspect of the problem is whether the proposed methodology could give equally good results when the trajectory data is less dense or when the trajectory densities of the dataset are irregular, e.g. one part of a city is sampled from hundreds of trajectories and other parts from only a few tracks. Therefore, further investigation is needed to clarify whether the number of trajectories affects the performance and if so, how much.

7 Conclusion

The motivation for this article is the very small percentage of intersection regulators found in public available map databases. Although traffic regulators, such as traffic signals and stop signs, affect the traffic flow and could therefore be used in navigation applications to optimise travel time or estimate fuel consumption, they are still largely absent from map databases. Moreover, intersections are locations that require drivers' attention, so mapped traffic regulators could be used in driver assistance applications in various safety scenarios. However, mapping and updating this information for each intersection of a road network is an expensive task if traditional surveying procedures are followed.

In this paper, we present a method to identify traffic regulators using vehicle trajectories that can be easily recorded with low-cost mobile phone devices, as well as information extracted from the public OpenStreetMap, which is freely available. The method is being tested on two datasets from two different cities, Hannover and Chicago, where their intersections are controlled by different regulator types. Two classification methods, random forest and gradient boosting, are tested using either dynamic features (trajectories), static (OSM data only) or a combination of both. The results show that gradient boosting classification with static-dynamic features can predict traffic regulators with high accuracy (93% in Chicago and 94% in Hannover) and with per class predictions balanced. The *hybrid* model outperforms the other inference models (*static* and *dynamic*), highlighting the role that crowd-sourced information can play in detection scenarios such as the one considered here, which can increase classification performance when used in combination with easily extracted and free information derived from OSM.

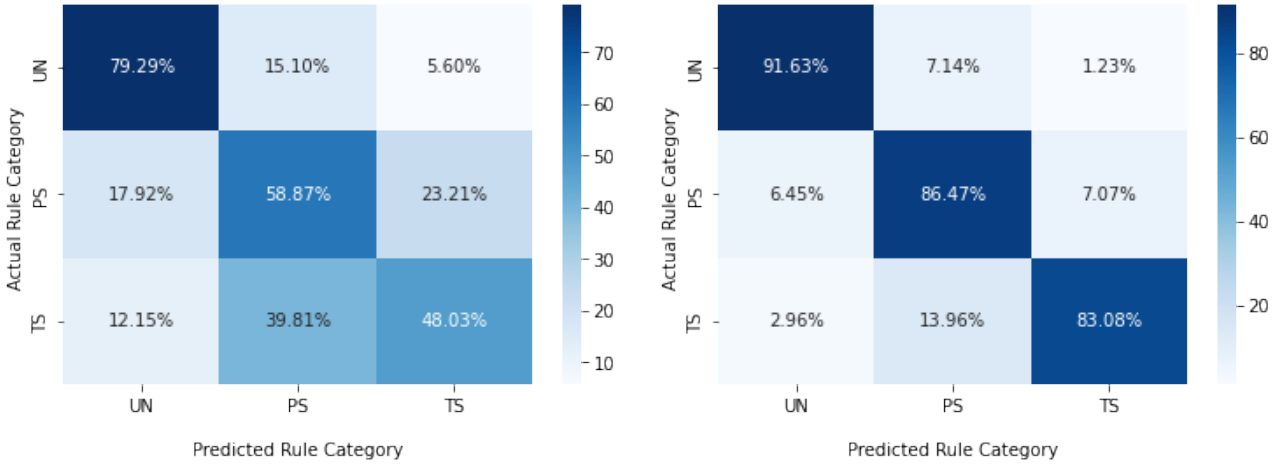
Author contributions. All authors contributed to writing-review, editing and project administration. The first author contributed to conceptualization, methodology, investigation, validation, formal analysis, writing-original draft preparation and visualization. The second author contributed to resources and writing-original draft preparation. The third author contributed to supervision and funding acquisition.

Competing interests. The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Acknowledgements. This research was funded by the German Research Foundation (Deutsche Forschungsgemeinschaft (DFG)) with grant number 227198829/GRK1931.

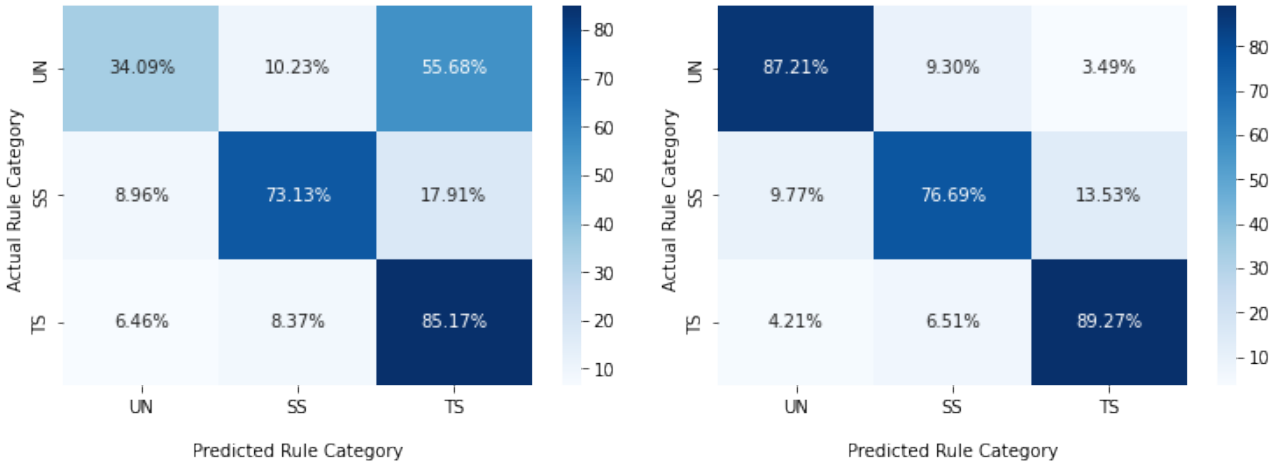
Table A1. Classification results of the static models.

Dataset	Classifier	Label	Recall		Precision		F-Measure		Accuracy		Support	
			simpl. setting	default setting	simpl. setting	default setting	simpl. setting	default setting	simpl. setting	default setting	simpl. setting	default setting
Hannover	RF	UN	0.81	0.92	0.68	0.87	0.74	0.89			821	812
		PS	0.60	0.84	0.58	0.86	0.59	0.85			1172	1131
		TS	0.43	0.84	0.54	0.87	0.48	0.86			864	845
		Weigh.Avg.	0.61	0.86	0.60	0.86	0.60	0.86			2857	2788
									0.61	0.86		
Hannover	GB	UN	0.79	0.92	0.67	0.88	0.73	0.90			821	812
		PS	0.59	0.86	0.60	0.85	0.59	0.86			1172	1131
		TS	0.48	0.83	0.57	0.89	0.52	0.86			864	845
		Weigh.Avg.	0.61	0.87	0.61	0.87	0.61	0.87			2857	2788
									0.61	0.87		
Chicago	RF	UN	0.34	0.87	0.48	0.77	0.39	0.81			88	86
		SS	0.73	0.76	0.77	0.80	0.75	0.78			134	133
		TS	0.85	0.89	0.76	0.92	0.80	0.91			263	261
		Weigh.Avg.	0.73	0.85	0.71	0.86	0.71	0.85			485	480
									0.73	0.85		
Chicago	GB	UN	0.44	0.78	0.49	0.76	0.46	0.76			88	86
		SS	0.72	0.74	0.76	0.75	0.73	0.74			134	133
		TS	0.81	0.90	0.77	0.92	0.79	0.91			263	261
		Weigh.Avg.	0.72	0.84	0.71	0.84	0.71	0.83			485	480
									0.72	0.84		



(a) Simplified setting (GB), Hanover.

(b) Default setting (GB), Hanover.



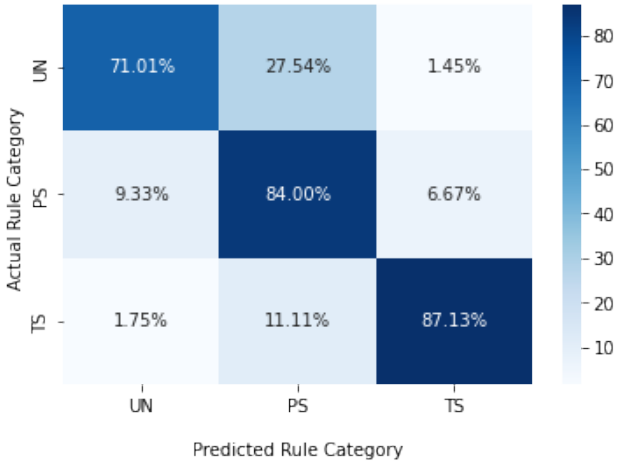
(c) Simplified setting (RF), Chicago.

(d) Default setting (RF), Chicago.

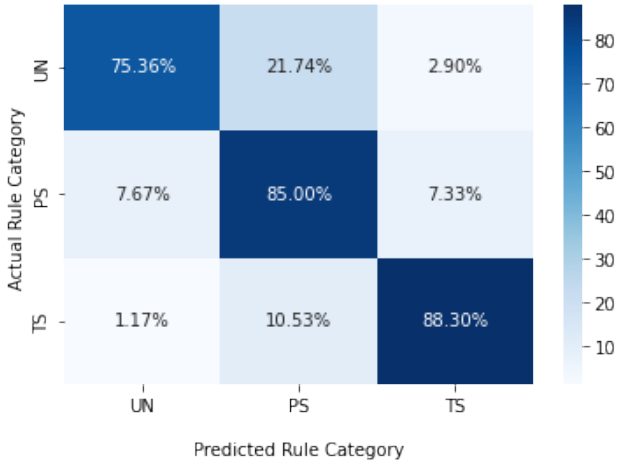
Figure A1. Confusion matrices of the *static model* for simplified and default feature settings, for Hannover and Chicago datasets.

Table B1. Classification results of the *dynamic* model.

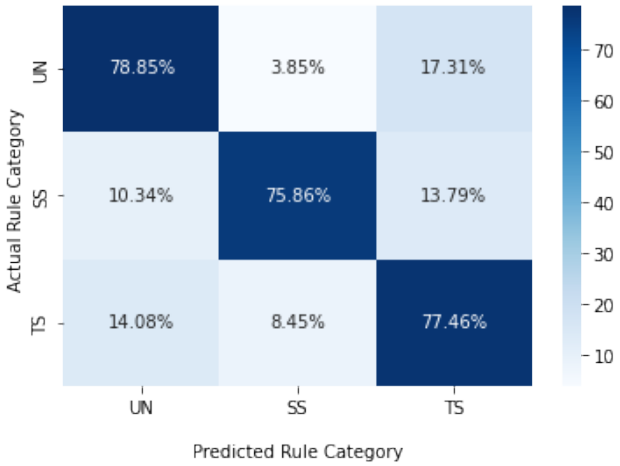
Dataset	Classifier	Label	Recall	Precision	F-Measure	Accuracy	Support
Hannover	RF	UN	0.71	0.62	0.65		69
		PS	0.84	0.87	0.85		300
		TS	0.87	0.88	0.87		171
		Weigh.Avg.	0.83	0.84	0.83		540
						0.83	
	GB	UN	0.75	0.69	0.71		69
		PS	0.85	0.89	0.87		300
		TS	0.88	0.86	0.87		171
		Weigh.Avg.	0.85	0.86	0.85		540
						0.85	
Chicago	RF	UN	0.79	0.77	0.77		52
		SS	0.75	0.74	0.72		29
		TS	0.78	0.82	0.79		71
		Weigh.Avg.	0.78	0.79	0.77		152
						0.78	
	GB	UN	0.81	0.79	0.79		52
		SS	0.80	0.78	0.77		29
		TS	0.76	0.81	0.78		71
		Weigh.Avg.	0.78	0.79	0.78		152
						0.78	



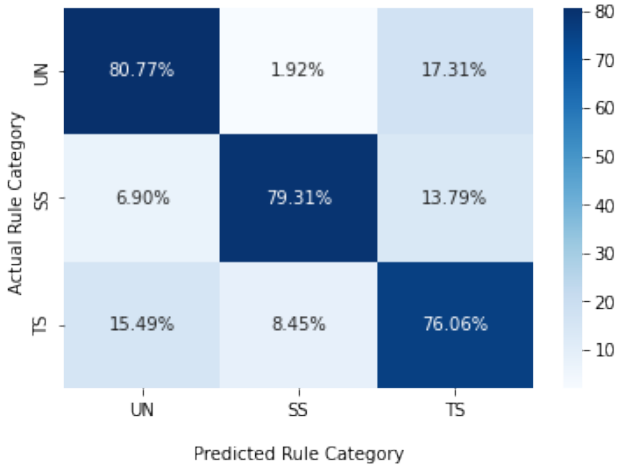
(a) RF, Hanover.



(b) GB, Hanover.



(c) RF, Chicago.

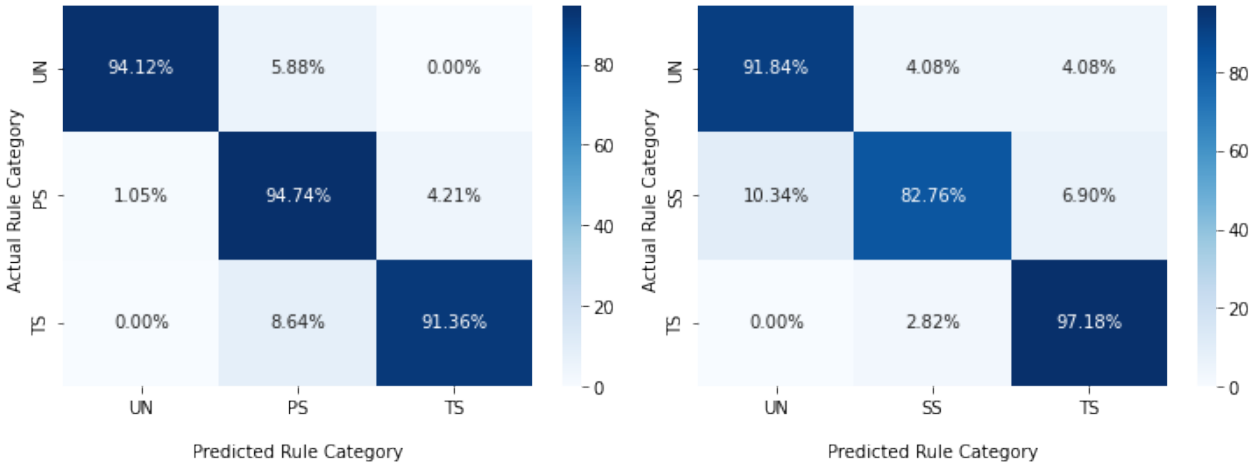


(d) GB, Chicago.

Figure C1. Confusion matrices of the *dynamic* models for the Hannover and Chicago datasets.

Table D1. Classification results of the *hybrid* model.

Dataset	Classifier	Feature Selection	Tuning	Label	Recall	Precision	F-Measure	Accuracy	Support
Hannover	RF	No	No	UN	0.75	0.91	0.81		68
				PS	0.92	0.89	0.90		285
				TS	0.88	0.89	0.88		162
				W.Avg.	0.89	0.89	0.89		515
								0.89	
	GB	No	No	UN	0.88	0.93	0.90		68
				PS	0.94	0.93	0.93		285
				TS	0.90	0.91	0.90		162
				W.Avg.	0.92	0.92	0.92		515
								0.92	
	GB	Yes	No	UN	0.94	0.95	0.94		68
				PS	0.95	0.94	0.94		285
				TS	0.92	0.93	0.93		162
				W.Avg.	0.94	0.94	0.94		515
								0.94	
	GB	Yes	Yes	UN	0.94	0.96	0.95		68
				PS	0.95	0.94	0.94		285
				TS	0.91	0.93	0.92		162
				W.Avg.	0.94	0.94	0.94		515
								0.94	
Chicago	RF	No	No	UN	0.88	0.93	0.89		49
				PS	0.80	0.89	0.82		29
				TS	0.94	0.90	0.92		71
				W.Avg.	0.89	0.91	0.89		149
								0.89	
	GB	No	No	UN	0.92	0.93	0.92		49
				SS	0.87	0.88	0.86		29
				TS	0.91	0.93	0.92		71
				W.Avg.	0.91	0.92	0.91		149
								0.91	
	GB	Yes	No	UN	0.94	0.91	0.92		49
				SS	0.83	0.84	0.83		29
				TS	0.96	0.95	0.95		71
				W.Avg.	0.93	0.92	0.92		149
								0.93	
	GB	Yes	Yes	UN	0.92	0.95	0.93		49
				SS	0.83	0.79	0.80		29
				TS	0.97	0.95	0.96		71
				W.Avg.	0.93	0.92	0.92		149
								0.93	



(a) Hanover. (b) Chicago.

Figure E1. Confusion matrices of *hybrid* model for the Hannover and Chicago datasets (Exp.4 and Exp.8 from the Table D1).

References

Ahmed, M., Karagiorgou, S., Pfoser, D., and Wenk, C.: A comparison and evaluation of map construction algorithms using vehicle tracking data, GeoInformatica, 19, 601–632, https://doi.org/10.1007/s10707-014-0222-6, 2014.

Ahmed, M., Karagiorgou, S., Pfoser, D., and Wenk, C.: A comparison and evaluation of map construction algorithms using vehicle tracking data, GeoInformatica, 19, 601–632, https://doi.org/10.1007/s10707-014-0222-6, 2015.

Alshayeb, S., Stevanovic, A., and Effinger, J. R.: Investigating impacts of various operational conditions on fuel consumption and stop penalty at signalized intersections, International Journal of Transportation Science and Technology, https://doi.org/10.1016/j.ijtst.2021.09.005, 2021.

Cheng, H., Zourlidou, S., and Sester, M.: Traffic Control Recognition with Speed-Profiles: A Deep Learning Approach, ISPRS International Journal of Geo-Information, 9, https://doi.org/10.3390/ijgi9110652, 2020.

Gao, S., Li, M., Rao, J., Mai, G., Prestby, T., Marks, J., and Hu, Y.: Automatic Urban Road Network Extraction From Massive GPS Trajectories of Taxis, pp. 261–283, Springer International Publishing, Cham, https://doi.org/10.1007/978-3-030-55462-0_11, 2021.

Gastaldi, M., Meneguzzer, C., Rossi, R., Lucia, L. D., and Gecchele, G.: Evaluation of Air Pollution Impacts of a Signal Control to Roundabout Conversion Using Microsimulation, Transportation Research Procedia, 3, 1031–1040, https://doi.org/10.1016/j.trpro.2014.10.083, 17th Meeting of the EURO Working Group on Transportation, EWGT2014, 2-4 July 2014, Sevilla, Spain, 2014.

Golze, J., Zourlidou, S., and Sester, M.: Traffic Regulator Detection Using GPS Trajectories, KN - Journal of Cartography and Geographic Information, https://doi.org/10.1007/s42489-020-00048-x, 2020.

Hiremath, R., Malshikare, K., Mahajan, M., and Kulkarni, R. V.: A Smart App for Pothole Detection Using Yolo Model, in: ICT Analysis and Applications, edited by Fong, S., Dey, N., and Joshi, A., pp. 155–164, Springer Singapore, Singapore, 2021.

Hu, S., Su, L., Liu, H., Wang, H., and Abdelzaher, T. F.: SmartRoad: Smartphone-Based Crowd Sensing for Traffic Regulator Detection and Identification, ACM Trans. Sen. Netw., 11, 55:1–55:27, https://doi.org/10.1145/2770876, 2015.

Jokar Arsanjani, J., Mooney, P., Zipf, A., and Helbich, M.: An introduction to OpenStreetMap in GIScience: Experiences, Research, Applications, 2015.

Li, J., Boonaert, J., Doniec, A., and Lozenguez, G.: Multi-models machine learning methods for traffic flow estimation from Floating Car Data, Transportation Research Part C: Emerging Technologies, 132, 103 389, https://doi.org/10.1016/j.trc.2021.103389, 2021.

Lian, F., Chen, B., Zhang, K., Miao, L., Wu, J., and Luan, S.: Adaptive traffic signal control algorithms based on probe vehicle data, Journal of Intelligent Transportation Systems, 25, 41–57, https://doi.org/10.1080/15472450.2020.1750384, 2021.

Liao, Z., Xiao, H., Liu, S., Liu, Y., and Yi, A.: Impact Assessing of Traffic Lights via GPS Vehicle Trajectories, ISPRS International Journal of Geo-Information, 10, https://doi.org/10.3390/ijgi10110769, 2021.

Mapillary: Mapillary: a street-level imagery platform, https://www.mapillary.com/, accessed: 2022-04-20, 2022.

Méneroux, Y., Guilcher, A., Saint Pierre, G., Hamed, M., Mustiere, S., and Orfila, O.: Traffic signal detection from in-vehicle GPS speed profiles using functional data analysis and machine learning, International Journal of Data Science and Analytics, 10, 101–119, https://doi.org/10.1007/s41060-019-00197-x, 2020.

OSM contributors: https://www.openstreetmap.org, accessed: 2022-04-19, 2022.

Qin, T., Shangguan, W., Song, G., and Tang, J.: Spatio-Temporal Routine Mining on Mobile Phone Data, ACM Trans. Knowl. Discov. Data, 12, https://doi.org/10.1145/3201577, 2018.

Saremi, F. and Abdelzaher, T. F.: Combining Map-Based Inference and Crowd-Sensing for Detecting Traffic Regulators, 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems, pp. 145–153, 2015.

Shan, Z., Wu, H., Sun, W., and Zheng, B.: COBWEB: A Robust Map Update System Using GPS Trajectories, in: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '15, p. 927–937, Association for Computing Machinery, New York, NY, USA, https://doi.org/10.1145/2750858.2804286, 2015.

Tang, J., Deng, M., Huang, J., Liu, H., and Chen, X.: An Automatic Method for Detection and Update of Additive Changes in Road Network with GPS Trajectory Data, ISPRS International Journal of Geo-Information, 8, https://doi.org/10.3390/ijgi8090411, 2019.

Tu, W., Xiao, F., Li, L., and Fu, L.: Estimating traffic flow states with smart phone sensor data, Transportation Research Part C: Emerging Technologies, 126, 103 062, https://doi.org/10.1016/j.trc.2021.103062, 2021.

Wage, O. and Sester, M.: Joint Estimation of Road Roughness From Crowd-Sourced Bicycle Acceleration Measurements, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, V-4-2021, 89–96, https://doi.org/10.5194/isprs-annals-V-4-2021-89-2021, 2021.

XGBoost Python: XGBoost Python Library, https://xgboost.readthedocs.io/en/stable/python/index.html, accessed: 2022-02-15, 2022.

Yao, Q., Shi, Y., Li, H., Wen, J., Xi, J., and Wang, Q.: Understanding the Tourists' Spatio-Temporal Behavior Using Open GPS Trajectory Data: A Case Study of Yuanmingyuan Park (Beijing, China), Sustainability, 13, https://doi.org/10.3390/su13010094, 2021.

Yoshioka, T., Sakakibara, H., Tenhagen, R., Lorkowski, S., and Oguchi, T.: Traffic Signal Control Parameter Calculation Using Probe Data, International Journal of Intelligent Transportation Systems Research, pp. 1–11, 2022.

Zhang, D., Lee, K., and Lee, I.: Mining Medical Periodic Patterns from Spatio-Temporal Trajectories, in: Health Information Science, edited by Siuly, S., Lee, I., Huang, Z., Zhou, R.,

Wang, H., and Xiang, W., pp. 123–133, Springer International Publishing, Cham, 2018.

Zourlidou, S. and Golze, J.: Ground-truth Intersection Regulators for Chicago, https://doi.org/10.25835/0vifyzqi, 2022.

Zourlidou, S. and Sester, M.: Traffic Regulator Detection and Identification from Crowdsourced Data—A Systematic Literature Review, ISPRS International Journal of Geo-Information, 8, https://doi.org/10.3390/ijgi8110491, 2019.