



Article Traffic Control Recognition with Speed-Profiles: A Deep Learning Approach

Hao Cheng^D, Stefania Zourlidou *^D and Monika Sester ^D

Institute of Cartography and Geoinformatics, Leibniz University, Appelstrasse 9a, 30167 Hanover, Germany; cheng@ikg.uni-hannover.de (H.C.); sester@ikg.uni-hannover.de (M.S.)

* Correspondence: zourlidou@ikg.uni-hannover.de; +49-511-762-19435

Received: 13 August 2020; Accepted: 27 October 2020; Published: 30 October 2020



Abstract: Accurate information of traffic regulators at junctions is important for navigating and driving in cities. However, such information is often missing, incomplete or not up-to-date in digital maps due to the high cost, e.g., time and money, for data acquisition and updating. In this study we propose a crowdsourced method that harnesses the light-weight GPS tracks from commuting vehicles as Volunteered Geographic Information (VGI) for traffic regulator detection. We explore the novel idea of detecting traffic regulators by learning the movement patterns of vehicles at regulated locations. Vehicles' movement behavior was encoded in the form of speed-profiles, where both speed values and their sequential order during movement development were used as features in a three-class classification problem for the most common traffic regulators: *traffic-lights, priority-signs* and uncontrolled junctions. The method provides an average weighting function and a majority voting scheme to tolerate the errors in the VGI data. The sequence-to-sequence framework requires no extra overhead for data processing, which makes the method applicable for real-world traffic regulator detection tasks. The results showed that the deep-learning classifier Conditional Variational Autoencoder can predict regulators with 90% accuracy, outperforming a random forest classifier (88% accuracy) that uses the summarized statistics of movement as features. In our future work images and augmentation techniques can be leveraged to generalize the method's ability for classifying a greater variety of traffic regulator classes.

Keywords: traffic regulators; traffic-signs; trajectories; crowdsensing; crowdsourcing; GPS tracks; speed-profiles; intersections; classification; sequence-to-sequence; conditional generative model

1. Introduction

Under the umbrella concept of *smart city*, the development of smart transportation and mobility has been on the agenda of many government departments and institutions [1]. Nowadays, as the growth of urbanization [2] and traffic congestion in European cities [3] is increasing, the need for daily fast commuting between one's place of residence and place of work, or that of less periodical recurring traveling, has motivated a lot of research on how this demand can be efficiently facilitated [4,5]. For this navigation task, maps are an elementary basis. These maps include the geometry and semantics of the route segments, as well as additional information such as restrictions or traffic regulators. Although mapping with aerial images or surveying equipment is an accurate and effective way of collecting features of a city or Earth's surface in general, this procedure is highly time-consuming and cost-expensive. This is especially relevant, as this data is subject to frequent changes.

A solution to the above problem has been given through the concept of *citizens as sensors* [6], where geographic information can be created and shared through individuals that can act as sensors of their environment. Under this concept, citizens can collect various measurements or data of their activity environment, such as temperature and noise values or images. These data coupled or

geo-tagged with the geographical coordinates of the location they are taken from, can be collectively processed for extracting information about certain aspects of a phenomenon at a geographical level. To this end, Volunteered Geographic Information (VGI) generated by individual users has been largely used to enrich the features in maps, e.g., OpenStreetMap (OSM) [7]. However, the information of traffic regulators as a map feature is still largely missing, incomplete or out-of-date.

1.1. From GPS-Tracks to Traffic-Regulator Detection

The efficient and "cheap" solution for traffic regulator detection is through "Crowdsourcing". It describes how a big or expensive task, e.g., the enrichment of traffic regulators for a city in digital maps, can be carried out by a group of volunteers via some participative online activity [8], e.g., collecting GPS tracks when they are commuting from different places in the city. This can be easily done with the equipment of modern mobile phone devices that are capable of performing enormous processing, storage and sensing tasks. Numerous location-based crowdsourcing applications have emerged through mobile crowd sensing (MCS) [9], such as road network generation from GPS tracks [10–12], parking spot recommendation [13], traffic congestion detection [14], speed-limit violations [15], energy-efficient map applications [16] and flood-risk management by harnessing crowdsourced data [17–19]. In this study, we were particularly interested in exploring the potential of using VGI [20], which is generated by city road users using mobile devices to collect GPS tracks, for enriching the traffic-regulator feature in maps.

City maps are created to enable the fastest possible spatial orientation in the urban space and the map feature providing traffic regulator information at intersections is very beneficial for effective commuting, e.g., precisely estimating the travel time. Moreover, self-driving cars also need such intersection-related information for evaluating risks, inferring other participants' intentions and overall for path planing along intersections. Surprisingly, this information is still largely missing. The work presented in this article was therefore motivated by this observation and primarily aimed at exploring how traffic regulators can be detected and identified by harnessing the VGI data generated by city commuters.

Images or GPS tracks are commonly used for regulator detection and classification [21]. Traffic-sign detection is a popular topic within the computer-vision community, with some studies focusing on the traffic-sign classification such as [22] and others on related topics such as the prediction of traffic-signal phases [23]. From a crowdsensing point of view, images generate a large amount of data that consume many resources, e.g., storage, bandwidth, energy and the consumption may affect the Quality-of-Service of applications that are built upon such data [24]. In addition, there are also privacy issues when individuals or number plates are recognizable in the images. Also, the GPS tracks work independent of occlusion, illumination or other situations deteriorating visibility. For these reasons, we opted to focus on *light-weight* data: GPS tracks instead of images. GPS tracks are time-ordered triplets {latitude, longitude and timestamp} as shown in Figure 1a, which are often connected in time sequence and stored as trajectories, as shown in Figure 1b. Schematically, the goal of predicting the traffic regulator of each arm in junctions from GPS tracks is depicted in Figure 1c,d.

In this paper, we propose a *sequence-to-sequence* deep learning model using GPS tracks as a time sequence for realizing the goal mentioned above. The majority of previous works suggests using the statistical features extracted from GPS tracks, such as stop times and stop duration [5,25,26], slowdown and standstill events [27–29] or speed-profiles [30–32]. These types of features summarize the dynamics of vehicles' motion in the relevant junctions over time. However, the detailed information of how a vehicle changes its motion from the previous time-step to the next is lost. On the other hand, the timing and location of the approaching vehicles can also lead to very different maneuver behaviors and speed profiles against the same traffic regulator. The statistical features require pre-processing of the GPS tracks as sequences for capturing the fine-grained dynamics of driving behaviors, in order to recognize traffic regulators. To overcome the overfitting problem, we use the relative GPS position to the junction

center and a sliding window along the GPS track to automatically learn the timing and location of the approaching vehicle. To the best of our knowledge, we are the first to use a sequence-to-sequence deep learning model, as suggested and applied in this paper, for traffic regulator detection.



(**a**) GPS tracks. Each point represents a location that a vehicle had at a certain time.



(**b**) A trajectory derived from connecting GPS tracks on time order.



(c) Prediction of the traffic regulator of a junction for (d) GPS tracks, depicted as trajectories, are the input each arm.data of the classifiers.

Figure 1. Geographical data collected from a mobile phone application, represented as points (**a**,**b**) trajectories. Map enriched with traffic regulators (**c**) from crowdsourced vehicles' GPS tracks (**d**).

1.2. Related Work

Recently, Zourlidou and Sester [21] conducted a systematic literature review on traffic-control detection from GPS tracks, highlighting (1) the importance of the topic itself, (2) the need for open data (GPS tracks and ground truth map) that researchers can use to test their methods and compare their results with others, (3) the low diversity of the predicted classes within each study and (4) the low percentage of studies that examine the cross-city applicability of their proposed methods (i.e., trained in city A and tested in city B). In this section, we briefly describe GPS-based methods for traffic regulator detection and identification. All the studies, along with the regulator classes they examine, are listed in Table 1.

The first category of methods are the studies that use map-extracted features. Such intersectionrelated features can be derived from open maps, e.g., OSM. The study by Saremi and Abdelzaher is unique in this regard [33]. They export features such as speed rating of road segments, distance of one junction to the closest one, end-to-end distance of the road that a junction belongs to, semi distances of a junction to the two ends of the road that belongs to and category of the street segments. They also test the combination of map-based features with trace-derived attributes (number of stops, traverse speed and stop duration), achieving improvement of the classification accuracy 97%.

a/a	Ref.	Author(s)	Year	Regulators *
1	[33]	Seremi & Abdelzaher	2015	SS, TL
2	[25]	Pribe & Rogers	1999	SS, TL
3	[29]	Carisi et al.	2011	SS, TL
4	[5]	Hu et al.	2015	SS, TL, RW
5	[28]	Aly et al.	2017	SS, TL
6	[34]	Wang et al.	2017	SB
7	[27]	Qiu et al.	2018	SS
8	[35]	Méneroux et al.	2018	TL
9	[26]	Golze et al.	2020	TL, PR, RW
10	[32]	Méneroux et al.	2020	TL
11	[36]	Munoz-Organero et al.	2018	TL
12	[31]	Zourlidou et al.	2019	TL, YS, PS, RW

Table 1. Studies that detect traffic regulators from crowdsourced data.

* SS: Stops-Signs, TL: Traffic-Lights, RW: Right-of-Way rule, SB: Stop-Bars, YS: Yield-Signs, PS: Priority-Signs.

The second category includes approaches that use various features mainly related with stop or deceleration events. Here we find the earliest published study on the topic, proposed by Pribe and Rogers [25]. A Neural Network (NN) is trained to learn to associate the driver behavior with two types of traffic rules, traffic-lights and stop-signs. As input data, they use the average and standard deviation of stop-event related features: the number of times a vehicle stops before crossing a junction, the total duration of all stops and the last three stops that are closest to the junction. An additive feature is the percentage of traversals that include at least one stop for each road segment. A similar approach has been suggested by Hu et al. [5]. They use two types of features, namely *physical* and *statistical*. As physical features, they compute the final stop duration, minimum crossing speed, number of times the vehicle decelerates, number of stops and distance of last stop event from intersection. By computing the minimum, maximum, mean and variance of the physical features, four new statistical features are defined for each physical one. They test a random forest classifier, as well as spectral clustering, for a three-class classification problem (stop-sign, traffic-light and uncontrolled junctions). Their method reports an accuracy over 90% for various features used for training and testing. Carisi et al. [29] propose a simple heuristic method for a binary classification problem (stop-signs and traffic-signals) using the slowdown and standstill events observed in the traces. They explain how to enrich digital maps with the location and timing of the aforementioned regulators. Their method achieves over 90% accuracy for the binary classification problem on a small dataset. Qui et al. [27] detect stop-signs based on a prevalent characteristic of stopping at a stop-sign: a deceleration followed by an acceleration. Besides, they use some heuristic rules on crowdsourced GPS tracks to distinguish between four-way and two-way stop-signs and between stop-signs and traffic-lights; There is no stop-sign if only one stop segment at an intersection is detected in a single trace, where other traces do not have such segments. Méneroux et al. [35] present a supervised-learning algorithm (random forests and regression) for detecting and localizing traffic-signals based on the spatial distribution of vehicle stop-points along the road. One-month GPS traces collected from a city in Japan are used to train the algorithm and their method reaches up to 85% in detection score and approximately 5-meter positional accuracy. A similar method for intersection and stop bar position extraction has been proposed in [34]. Aly and Basalamah [28] harness pedestrians' trajectories for detecting stop-signs and traffic-lights. They recognize locations where pedestrians stay over a time threshold (dwelling time) and categorize the regulators to the two categories accordingly. Similar to the study [5], the most recent work by Golze et al. [26] uses the speed-related statistics (e.g., mean and maximum crossing speed) extracted from GPS tracks for traffic regulator classification.

The third category includes approaches that use speed-profiles as classification features. For example, Zourlidou et al. [31] and Kuntzsch et al. [30] explore the effect of high-quality speed profiles derived from CAN-Bus for training a tree-classifier to distinguish traffic-light controlled junctions from priority and yield controlled junctions. The study of [31] is the first one to use

speed-profiles for regulator detection. It reports high recall but low precision and F-measure for predicting traffic-light regulator. Similarly, Méneroux et al. [32] detect traffic-signals by using speed-profiles. They test three different ways of deriving features: functional analysis of speed logs, raw speed measurements and image recognition technique. They demonstrate the functional description of speed profiles with wavelet transforms. Among different classifiers, random forests scored the best accuracy (95%). Last, Munoz-Organero et al. [36] detect in real-time various street infrastructure elements, such as traffic-lights, street crossings and roundabouts, by classifying speed and acceleration time series with a deep-learning approach. Although the combined precision and recall are relatively high, compared to the other two classes, the performance score of the traffic-light regulator exhibits a clear limitation on all the tested classification settings.

This article proposes a novel mechanism of manipulating the descriptive ability of speed-profiles that represent vehicles' movement behavior at regulated locations. Different from all the aforementioned approaches, we present a sequence-to-sequence conditional generative classifier for traffic regular recognition using GPS tracks crowdsourced from vehicles in a fashion of time-series with varying sequence lengths.

The rest of the paper is structured as follows: In Section 2, we describe the proposed method and the data we used for experimenting and in Section 3, we present the results. We discuss the results and future directions in Section 4 and summarize all findings in Section 5.

2. Materials and Methods

2.1. Dataset

The GPS tracks were collected using the mobile phone application *Geo Tracker* developed in the Android operating system. The duration for the data acquisition started in December 2017 and ended in March 2019. The data were naturalistic, in the sense that no instruction was given to the driver from an external person while driving regarding where to go or how to drive. All the trajectories/trips were part of drivers' daily travel, e.g., from home to work or shopping center, and vice versa. In total, 1204 trajectories that cross 1064 junctions (Figure 2) regulated by 3538 junction arm regulators were accumulated. For the classification task, we considered one regulator for each junction arm. Table 2 gives a description of the dataset that we used for testing the proposed method.

Most of the trajectories have length between 1 and 12 km (Appendix A Figure A1a), trip duration lasts mostly between 0 and 28 min (Appendix A Figure A1b) and the most common junction types are three-way and four-way (Appendix A Figure A1d). Regarding the number of trajectories per regulator type, traffic-lights, priority-signs and uncontrolled rules have the biggest number of crossings (Appendix A Figure A1e), with yield-signs, stop-signs and roundabouts being ignored indeed from the classification due to data limitation as discussed later in the next section. Last, most junctions (689 out of 1064) are sampled from 1–10 trajectories, following by 141 junctions having between 11 and 20 trajectories and only two junctions having between 421 and 460 trajectories (Appendix A Figure A1f).

Table 2. Dataset used for testing the proposed methods.

City \diamond	Junc. Rules		Traj.	Rules *	
Hannover (DE)	1064	3538	1204	PS, YS, TL, UC	

° Country name: DE (Germany); * PS: Priority-Signs, YS: Yield-Signs, TL: Traffic-Lights, UC: Uncontrolled.



Figure 2. Hannover Dataset. Vehicles trajectories are the blue lines and red points symbolize junctions. Data by © OpenStreetMap [7].

2.2. Methodology

2.2.1. Problem Formulation

The task of traffic regulator recognition is defined as a classification problem. For the given junction arm, the recognition task is mathematically formulated as $Y_n = f(X_n^{(T)})$, where Y_n is the traffic regulator (one of the classes *priority-signs*, *traffic-lights* and *uncontrolled*) regulating the given GPS track X_n , and n belongs to N denoting the total number of GPS tracks recorded in the given junction arm. The GPS track stores the timely-ordered observed signals $X_n = \{x^1, ..., x^T\}, x^i \in \mathbb{R}^d$ and d denotes the dimension of the feature vector, which contains the location and speed information at each signal point.

By the definition above, f(.) is a sequence-to-one classifier. We slightly change the form of Y_n to $Y_n = \sum_{i=1}^T \lambda(Y_i^t)$, where $\lambda(.)$ is the *weighting function* that summarizes the signal-wise prediction to the sequence-wise prediction for the traffic regulator for the give GPS track. Hence, the sequence-to-one classification now turns into a sequence-to-sequence classification.

In most cases for the given junction arm, $N \ge 1$ GPS tracks are available. The arm regulator is the *majority vote* of all the GPS tracks traversed alone the given arm. Equations (1) and (2) denote the prediction process using the traversed GPS tracks. Note that the signal-wise classification provides fine-gained feedback at each signal point for a single track, while the arm-regulator classification result is the crowdsourced feedback from all the GPS tracks.

$$\mathbf{Y} = \arg\max\frac{1}{N}\sum_{n=1}^{N} Y_{n},\tag{1}$$

where

$$\boldsymbol{Y}_{\boldsymbol{n}} = \sum_{i=1}^{T} \lambda(f(\boldsymbol{X}_{\boldsymbol{n}}^{(t)})).$$
⁽²⁾

One remaining problem that the sequence-to-sequence model has to tackle is the varying sequence length. In other words, *T* is not fixed due to the different duration of the tracks and availability of

GPS signals. We propose to use a sliding window with a fixed window size (w) for varying sequence lengths [37]. First, a sequence is divided into small sub-sequences, which capture both long and short dependencies and circumvent the problem of varying sequence lengths across different GPS tracks. Second, as we discussed in the previous section, the location and timing for a track regarding the traffic control is important. However, it is not known where and when the track might be exactly regulated by the traffic control. Besides, the exact location and timing may differ from one track to another. The sliding window exhausts through each time-step and automatically learns the location and timing impacted by the traffic control. Compared with a fixed location or timing, this method ($w \ll T$) is less likely to be overfitted to a particular junction. Equation (3) denotes the sliding window method with a stride being the same size as the sliding widow size. The overlap between two consecutive windows is allowed when the stride is set to be smaller than w (see the left part of Figure 3).



$$X_{n}^{(T)} = \{x_{w1}, ..., x_{wm}\}, \text{ where } m = \frac{T}{w}.$$
 (3)

Figure 3. The pipeline of the Conditional Variational Auto-Encoder (CVAE) model.

2.2.2. Conditional Generative Model

We propose to use a conditional generative model parameterized by neural networks for the classification function f(...), namely, the Conditional Variational Auto-Encoder (CVAE) [38,39]. The CVAE framework has been proven to be very successful for solving many complex problems, for instance, image classification and generation [39,40] and trajectory prediction [41]. The choice of this model is made by considering the following aspects: GPS tracks against the traffic regulators are stochastic due to (1) the uncertain driving behavior of the car drivers and (2) the location and timing for traversing along the given junction arm. The CVAE learns a recognition model that encodes the input into some stochastic variables, the so-called latent random variables, following some prior distribution such as Gaussian distribution. Then, it learns a generative model that is conditioned on the stochastic variables for the probabilistic prediction task [39].

In the following we briefly revisit the CVAE framework. Given the input **X** and the output **Y**, the CVAE model is defined as:

$$p(\mathbf{Y}|\mathbf{X}) = \mathcal{N}(f(\mathbf{z}, \mathbf{X}), \, \sigma^2 * I). \tag{4}$$

The conditional probability of the output is an isotropic Gaussian distribution, whose mean $\mu = f(\mathbf{z}, \mathbf{X})$, is a function of the input **X** and the latent variables **z**, and the covariance matrix $\Sigma = \sigma^2 * I$, is an identity matrix *I* multiplied by some scalar σ^2 .

Due to the intractable true posterior $q_{\theta}(\mathbf{z}|\mathbf{X},\mathbf{Y})$, the equation cannot be solved analytically. A variational posterior $q_{\phi}(\mathbf{z}|\mathbf{X}, \mathbf{Y})$ is introduced to approximate the true posterior. The model then can be trained using Stochastic Gradient Variational Bayes (SGVB) [38] by reaching a variational lower bound, denoted by Equation (5).

$$\log p_{\theta}(\mathbf{Y}|\mathbf{X}) \geq -\mathbf{K}\mathbf{L}(q_{\phi}(\mathbf{z}|\mathbf{X},\mathbf{Y})||p_{\theta}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{X},\mathbf{Y})}[\log p_{\theta}(\mathbf{Y}|(\mathbf{X},\mathbf{z}))],$$
(5)

where $p_{\theta}(\mathbf{z})$ is the prior that can be made independent from the input **X** [40] and is drawn from $\mathcal{N} \sim (0, I)$. For the complete derivation of the lower bound, we recommend readers to have a look at [38,39].

The form of Equation (5) is interpreted as an auto-encoder, as the first term on the right side "encodes" both the input and the output into the latent variables and the second term "decodes" the output from the input and the latent variables. The decoder is also called the generative model. Note that compared to a traditional auto-encoder, here the CVAE model predicts the output **Y**, rather than reconstructs the input **X**.

The Kullback–Leilber divergence **KL**(.) between the approximated posterior and prior distributions (both Gaussian) can be solved analytically. The reconstruction loss $\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{X},\mathbf{Y})}(,)$ can be approximated by the Monte Carlo sampling [38]. The non-linear mapping functions of both θ and ϕ are parameterized by neural networks. In order to enable the gradient in the sampling process, a re-parameterization trick [42] is applied for back propagation, where

$$\mathbf{z}^{(l)} = g_{\boldsymbol{\phi}}(\mathbf{X}, \mathbf{Y}, \boldsymbol{\epsilon}^{(l)}) = \boldsymbol{\mu}^{(l)} + \boldsymbol{\sigma}^{(l)} \odot \boldsymbol{\epsilon}^{(l)}, \text{ and } \boldsymbol{\epsilon}^{(l)} = \mathcal{N} \sim (0, 1).$$
(6)

Then the loss of the CVAE model is optimized via stochastic gradient descent. Equation (5) denotes the optimization process.

$$\mathcal{L}_{\text{CVAE}}(\mathbf{X}, \mathbf{Y}; \theta, \phi) = -\mathbf{K}\mathbf{L}(q_{\phi}(\mathbf{z}|\mathbf{X}, \mathbf{Y})||p_{\theta}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L}\log p_{\theta}(\mathbf{Y}|(\mathbf{X}, \mathbf{z}^{(l)})).$$
(7)

2.2.3. Framework Pipeline and Input Features

In this sub-section, we introduce the overall pipeline of the CVAE model for the sequence-to-sequence classification task using GPS tracks with a sliding window and weighting function, demonstrated in Figure 3.

The CVAE model has two different information flows in training and inference/prediction, respectively. In the training process, both the GPS signals and the corresponding arm regulator are available. First, the label of the arm regulator is duplicated to align with the signal time steps. Then a sliding window is applied to exhaust the GPS signal and arm regulator sequences in parallel. After that both sequences are concatenated as a complete input for training a variational encoder for the latent variables. In the end, the decoder is trained by using the GPS track information and the latent variables for predicting the signal-wise arm regulators, which is later summarized by the weighting function for achieving the track-wise prediction. In the inference process, only the GPS signals are available. In order to predict the arm regulator, the GPS signals are concatenated with the latent variables directly sampled from the Gaussian distribution. We used Long Short-Term Memories (LSTMs) [43] for both the encoder and the decoder.

The GPS signals are extracted from the relative *x* and *y* positions of the UTM (Universal Transverse Mercator) coordinates in relation to the center of the given junction. First we used a predefined distance to select the relevant GPS tracks. Only the tracks within the threshold are of the interest. Because large distance may cause a track to traverse multiple different junctions. In addition, considering that the signals after the junction are not as important as the ones before the junction, when the vehicle is leaving, we set another threshold (maximum one window size) for the GPS signals after the junction center. Figure 4 exemplifies the GPS tracks extracted from some junctions regulated by priority-signs, traffic-lights and uncontrolled, respectively.

Second, after the extraction, we enriched the GPS signals by calculating the distance *d*, the *x*and *y*-offset denoted as Δx and Δy , the speed *v* in relation to the junction center. Note that because the raw GPS signals are not evenly distributed over time, we also added the time interval of two consecutive GPS signals Δt as an input feature. The enriched GPS feature vector is denoted as $\mathbf{X}^{(T)} = \{x, y, d, \Delta x, \Delta y, v, \Delta t\}_{t=1}^{T}$.



Figure 4. GPS tracks extracted from the given junction for traffic regulators such as priority-signs, traffic-lights and uncontrolled. (a) Priority-sign junction (PS). (b) Traffic-light junction (TL). (c) Uncontrolled junction (UC).

2.2.4. Experimental Settings

We ran the CVAE model multiple times using different parameters and set the values based on the best performance. The most important parameters as defined after this process are listed as follows:

- For the GPS tracks extraction, the distance threshold for selecting the relevant GPS tracks regarding the given junction is set to 65 m, the sliding window size is set to 8 and and the stride to 2;
- For the data partitioning, the GPS tracks are randomly split into 70:30 for training and test, respectively;
- For the neural networks of the CVAE model, the dimension of the latent variables **z** is set to 2, the dimension for the LSTM hidden state for both the encoder and decoder is set to 128;
- For the training hyper-parameters, the batch size is set to 256, the number of training epochs to 500 and an early stop with 50-epoch patience. The learning rate is set to 1e 3 using the Adam optimizer [44] with a decay rate of 1e 8;
- We use the average weighting function for summarizing the signal-wise prediction to the track-wise prediction.

More details of the settings and the code of the CVAE model can be found at the repository (https://github.com/haohao11/Traffic_Control_Recognition).

2.2.5. Comparison Model

The proposed CVAE model was compared with the performance of a random forest model [26] using the same dataset (1328 in total regulators) as in [26], enlarged by additional 1609 regulators from the same city/road-network (in total 2937 regulators). Different from the sequential features mentioned above, the random forest model uses two types of features summarized from the GPS tracks: physical and statistical features. The physical features are, for example, the number, percentage, duration and distance of the standstill phases; duration and distance of the last standstill phase relative to the given junctions, mean and maximum speed of each GPS track. The statistical features are the statistics, such as *minimum, maximum, mean* and *variance* of all the aforementioned physical features. Different strategies were leveraged to boost the performance of the random forest model, such as, random oversampling and bagging or AdaBoost [26].

3. Results

In this section, we present the empirical results for the CVAE model and the random forest model with different boosting strategies. The performance was measured by accuracy for classifying the three majority traffic regulators, i.e., priority-signs, traffic-lights and uncontrolled on the test dataset. Table 3 shows the evaluation results for the random forest model and the CVAE model.

Classifier	Classifier Strategy		Dataset Size ²	Accuracy of Test
Random Forest	basic	complete	1328	0.83
Random Forest	basic	no turning tracks	937	0.85
Random Forest	oversampling	no turning tracks	937	0.85
Random Forest	oversampling & Bagging	no turning tracks	937	0.80
Random Forest	oversampling & AdaBoost	no turning tracks	937	0.88
CVAE model	majority voting scheme	complete	2937	0.90

Table 3. Evaluation results for the random forest model with different boosting strategies and the CVAE model.

¹ There are two types of tracks in the complete dataset: no turning tracks and turning tracks. ² is the original number of tracks in the dataset.

The basic random forest classifier achieved an accuracy of 0.83 for the test GPS tracks, including both non-turning and turning tracks. The performance was improved by removing the turning tracks that were more difficult to classify. However, the removal reduces the size of the dataset. Random oversampling was leveraged to increase the samples of the minority class and an increase in the performance was achieved by that way. The best performance (0.88 accuracy) was accomplished by the classifier using oversampling and the AdaBoost strategy.

The CVAE model was trained to classify all the GPS tracks, including both non-turning and turning tracks for the complete dataset. The CVAE model first predicts the traffic regulators for each single GPS signal and then summarizes the weighted signal-wise predictions to the track-wise prediction for each GPS track (see Section 2.2.1). A majority vote of the classified results for the tracks traversed along the given junction arm is the final classification. Overall, the CVAE model outperformed the random forest model using all the tracks (0.90 vs. 0.83), and as well the random forest model using only no-turning tracks with oversampling and AdaBoost strategy (0.90 vs. 0.88).

Figure 5 shows the confusion matrices for the best random forest classier with oversampling and AdaBoost (Figure 5a) and the CVAE model (Figure 5b) for traffic regulator classification. The confusion matrices demonstrate that compared to the random forest classifier the CVAE model achieved slightly better (0.87 vs. 0.86) and superior (0.91 vs. 0.88) precision for predicting uncontrolled and priority-sign regulators, respectively. It outperformed the random forest classifier by a large margin (0.90 vs. 0.80) for predicting traffic-light regulators.



Figure 5. The confusion matrices (in percentage) for the best random forest classier and the CVAE model for junction arm rule prediction. (**a**) Confusion matrix for the random forest classier with oversampling and AdaBoost. (**b**) Confusion matrix for the CVAE model for junction arm rule prediction.

From the above results, in comparison with the random forest model, the CVAE model generated better accuracy for junction arm rule predictions. Additionally, the CVAE model was generalized to both non-turning and turning tracks and validated on a larger dataset. Moreover, the CVAE

model did not use any advanced boosting strategies. The empirical results confirm that the proposed framework, a sequence-to-sequence classifier with a sliding window and an average weighting function, is suitable for dealing with both linear and non-linear GPS tracks of varying sequence length for traffic regulator classification.

4. Discussion

In this section, we first analyze the performance of the sequence-to-sequence CVAE model in terms of signal-wise prediction and the importance of the features used for the classification task, then we discuss the applicability of the model based on GPS signals for real-world traffic regulator detection.

The detailed results for the signal-wise predictions are listed in Table 4. Overall, the accuracy at signal-level for a three-class classification task is 0.73. It indicates that predicting traffic regulators for each GPS signal is very challenging, because the motion of the vehicles changes over time, e.g., decelerating, stopping and accelerating when they approach junctions. However, the accuracy is much higher than a random guess, which sheds light on the overall junction arm rule classification. The accumulated signal-wise predictions by the average weighting function and the track-wise predictions by the majority voting scheme lead to a very accurate junction arm rule classification (0.90 accuracy, see Table 3). It proves that the weighting function and the voting scheme can tolerate low-level errors (wrong classification of GPS signals), and most importantly the accuracy of individual information can be enhanced by accumulated crowd-sourcing information (multiple GPS signals and tracks). With a close look at the results for each class, the detailed precision and F-measure indicate that predicting the regulators for priority-signs and uncontrolled at the signal-wise level is more difficulty than predicting the regulator for traffic-lights. First, the respective sample size (support) of the priority-sign and the uncontrolled classes is smaller than the sample size of the traffic-light class. Note that we did not use any boosting strategy to increase the samples in the minority class. The priority-sign and uncontrolled classes might not be as well trained as the other class. Second, the regulators of priority-signs and uncontrolled are similar in the regard of enforcement; drivers need to practice their courtesy, which is highly individual dependent. On the other hand, the traffic-light regulator is clearly defined and has a stronger impact than the other two regulators on the drivers' behavior, which makes the classification more accurate.

Item	Precision	Recall	F-Measure	Support
Priority sign	0.60	0.75	0.67	5027
Traffic light	0.84	0.74	0.78	8150
Uncontrolled	0.75	0.71	0.73	5328
Weighted avg.	0.75	0.73	0.74	18,505
Accuracy	0.73			

Table 4. The results of detecting traffic regulators for the signal-wise classification by the CVAE model.

The CVAE model was trained using different feature combinations, in order to analyze how these features contribute to the classification performance. Namely, the features identified in Section 2.2.3 are the relative *x* and *y* coordinates and the distance *d* to the junction center, the *x*- and *y*-offset denoted as Δx and Δy between two consecutive GPS signals with the time interval Δt , and the speed *v*. Table 5 lists the detailed results. From the table we can see that the CVAE model (A) using *x* and *y* coordinates had a very limited performance compared to the other models using different or more features. This is because the coordinates are not evenly distributed over time and no accurate dynamic information (e.g., speed) is provided to indicate how vehicles approach junctions. The model (B) had a slightly better performance by only adding the distance feature *d*. But due to the same reason, its performance was rather limited. On the other hand, the model (C) using the offsets Δx and Δy achieved significantly better performance measured by all evaluation metrics compared to the previous two models. This is because the offset feature indicates how fast a vehicle crossing the junction changes

its position between two consecutive GPS signals. When the coordinates, distance and offset features were leveraged, the model (D) achieved a further improved performance. Interestingly, adding the speed feature v did not provide a positive contribution, i.e., model (E) vs. model (D). But the time feature Δt contributed to a slightly improved performance, i.e., model (F) vs. model (D). When all the aforementioned features were used, the model (G) achieved the best results measured by all the evaluation metrics.

CVAE	E Feature Combination.							Performance			
Model	x	у	d	Δx	Δy	v	Δt	Accuracy	Precision	Recall	F-Measure
(A)				-	-	-	-	0.69	0.80	0.69	0.67
(B)				-	-	-	-	0.73	0.79	0.73	0.70
(C)	-	-	-			-	-	0.84	0.85	0.84	0.84
(D)						-	-	0.86	0.86	0.86	0.86
(E)							-	0.85	0.85	0.85	0.85
(F)						-		0.86	0.87	0.86	0.86
(G)		\checkmark				\checkmark		0.90	0.90	0.90	0.90

Table 5. Evaluation results for the CVAE model using different feature combinations.

It becomes clear from the above analyses that the sequence-to-sequence CVAE model detects traffic regulators by learning the motion information (speed-profiles) from the vehicles driving through junctions. The motion information captured by GPS tracks can be easily acquired by a mobile phone application, which is relatively cheaper than acquiring images that require, e.g., larger storage and communication bandwidth. In addition, GPS signals are used as sequences. There is no extra computational overhead to pre-process the GPS signals for feature extraction locally on the mobile phone or remotely on the server side [5]. These advantages make the model light-weight and applicable for real-world traffic regulator detection tasks. Most importantly, as we analyzed above, the model provides a solution to tolerate the errors in the VGI—crowdsourced GPS tracks—generated by commuters. One single GPS signal may not correctly represent the traffic regulator. But a sequence of GPS signals and the aggregated GPS tracks via the majority voting scheme represent a highly accurate detection for the traffic regulator.

5. Conclusions and Future Work

In this paper, we propose a conditional generative framework for traffic control recognition using crowdsourced GPS tracks data. First, we discuss the advantages of using light-weight GPS data compared to image-based data. Second, we explain how our proposed novel framework differs from previously suggested methods that normally use statistical features summarized from GPS tracks. We propose to use the fine-grained GPS signals as sequences and train a sequence-to-sequence classifier based on the Conditional Variational Auto-Encoder (CVAE). A sliding window mechanism was applied to process sequences of varying length and an average weighting function for summarizing the signal-level prediction to the track-wise prediction. The proposed CVAE model outperformed a random forest model and achieved 0.90 accuracy tested on the mobile phone GPS data collected in the German city Hannover for both no-turning and turning junctions. The sequence-to-sequence method with the average weighting function and the majority voting scheme provides a solution to tolerate the errors generated by individual users. The usage of GPS signals as sequences makes our model easily applicable for real-world traffic regulator detection tasks.

In the future, different strategies will be investigated to further increase the prediction accuracy, for instance, using augmentation techniques to increase the number of GPS tracks and interpolation techniques to smooth the GPS signals. In addition, road images extracted from maps or satellite imagery can be fed to the CVAE framework for more accurately solving the task of traffic regulator detection. We will extend our model not only for detecting the most common traffic regulator classes, but also for more generalized classes.

Author Contributions: All authors contributed to conceptualization, methodology, writing—review and editing and project administration. Hao Cheng contributed to software, validation, formal analysis, data curation, visualization and writing—original draft preparation. Stefania Zourlidou contributed to data resources and formatting, investigation, visualization and writing—original draft preparation. Monika Sester contributed to supervision and funding acquisition. All authors have read and agreed to the published version of the manuscript.

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

Conflicts of Interest: 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.

Abbreviations

The following abbreviations are used in this manuscript:

- CVAE Conditional Variational Auto-Encoder
- MCS Mobile Crowd Sensing
- OSM OpenStreetMap
- PR Priority-Signs
- RW Right-of-Way rule
- SS Stop-Signs
- TL Traffic-Lights
- UC Uncontrolled
- YS Yield-Signs

Appendix A





(b)

1064



(a)





T-junction Y-junction Four-way Five-way Circular Total

(d)



1000

800

600

400

200

0

Number of Junctions



Figure A1. Statistics that describe the dataset used for testing the proposed methodology. (a) Distribution of dataset's trajectories according to their length (Km). (b) Distribution of dataset's trajectories according to trip duration (minutes). (c) Distribution of dataset's regulators according to their type (junction arms having at least one crossings). (d) Distribution of dataset's intersections according to their shape type. (e) Distribution of dataset's trajectories according to the regulator type of the junctions they cross. (f) Distribution of dataset's intersections according to the number of trajectories they cross each of them.

References

- Nam, T.; Pardo, T.A. Conceptualizing smart city with dimensions of technology, people, and institutions. In Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, College Park, MD, USA, 12–15 June 2011; pp. 282–291.
- 2. Antrop, M. Landscape change and the urbanization process in Europe. *Landsc. Urban Plan.* **2004**, *67*, 9–26. [CrossRef]
- 3. Link, H.; Dodgson, J.S.; Maibach, M.; Herry, M. *The Costs of Road Infrastructure and Congestion in Europe*; Springer: Berlin/Heidelberg, Germany, 2012.
- Dornbush, S.; Joshi, A. StreetSmart traffic: Discovering and disseminating automobile congestion using VANET's. In Proceedings of the 2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring, Dublin, Ireland, 22–25 April 2007; pp. 11–15.
- 5. Hu, S.; Su, L.; Liu, H.; Wang, H.; Abdelzaher, T.F. SmartRoad: Smartphone-Based Crowd Sensing for Traffic Regulator Detection and Identification. *ACM Trans. Sens. Netw.* **2015**, *11*, 1–27, [CrossRef]
- Goodchild, M. Citizens as sensors: The world of volunteered geography. *GeoJournal* 2007, 69, 211–221, [CrossRef]
- 7. OpenStreetMap Contributors. Planet Dump. 2019. Available online: https://planet.osm.org; https://www.openstreetmap.org (accessed on 11 May 2015).
- 8. Estellés-Arolas, E.; de Guevara, F.G.L. Towards an integrated crowdsourcing definition. *J. Inf. Sci.* **2012**, *38*, 189–200, [CrossRef]
- 9. Boubiche, D.E.; Imran, M.; Maqsood, A.; Shoaib, M. Mobile crowd sensing—Taxonomy, applications, challenges, and solutions. *Comput. Hum. Behav.* **2019**, *101*, 352–370. [CrossRef]
- 10. Zhang, C.; Xiang, L.; Li, S.; Wang, D. An Intersection-First Approach for Road Network Generation from Crowd-Sourced Vehicle Trajectories. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 473, [CrossRef]
- 11. Zhang, Y.; Liu, J.; Qian, X.; Qiu, A.; Zhang, F. An Automatic Road Network Construction Method Using Massive GPS Trajectory Data. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 400, [CrossRef]
- 12. Zhang, L.; Sester, M. Incremental Data Acquisition from Gps-traces. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences—ISPRS Archives 38*; International Society for Photogrammetry and Remote Sensing: London, UK, 2010.

- Nie, Y.; Xu, K.; Chen, H.; Peng, L. Crowd-parking: A New Idea of Parking Guidance Based on Crowdsourcing of Parking Location Information from Automobiles. In Proceedings of the IECON 2019—45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal, 14–17 September 2019; Volume 1, pp. 2779–2784, [CrossRef]
- Dimri, A.; Singh, H.; Aggarwal, N.; Raman, B.; Ramakrishnan, K.K.; Bansal, D. BaroSense: Using Barometer for Road Traffic Congestion Detection and Path Estimation with Crowdsourcing. *ACM Trans. Sens. Netw.* 2019, *16*, [CrossRef]
- 15. Mozas-Calvache, A.T. Analysis of behaviour of vehicles using VGI data. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 1–20, [CrossRef]
- Ganti, R.K.; Pham, N.; Ahmadi, H.; Nangia, S.; Abdelzaher, T.F. GreenGPS: A Participatory Sensing Fuel-efficient Maps Application. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys '10, San Francisco, CA, USA, 15–18 June 2010; ACM: New York, NY, USA, 2010; pp. 151–164, [CrossRef]
- 17. Holderness, T.; Turpin, E. From Social Media to GeoSocial Intelligence: Crowdsourcing Civic Co-management for Flood Response in Jakarta, Indonesia. In *Social Media for Government Services*; Nepal, S., Paris, C., Georgakopoulos, D., Eds.; Springer: Cham, Switzerland, 2015; pp. 115–133, [CrossRef]
- Moreira, R.; Degrossi, L.; De Albuquerque, J. An experimental evaluation of a crowdsourcing-based approach for flood risk management. In Proceedings of the 12th Workshop on Experimental Software Engineering (ESELAW), Lima, Peru, 22–24 April 2015.
- Feng, Y.; Brenner, C.; Sester, M. Flood severity mapping from Volunteered Geographic Information by interpreting water level from images containing people: A case study of Hurricane Harvey. *ISPRS J. Photogramm. Remote Sens.* 2020, 169, 301–319. [CrossRef]
- 20. Burghardt, D.; Nejdl, W.; Schiewe, J.; Sester, M. Volunteered Geographic Information: Interpretation, Visualisation and Social Computing (VGIscience). *Proc. Int. Cartogr. Assoc.* **2018**, *1*, 1–5. [CrossRef]
- 21. Zourlidou, S.; Sester, M. Traffic Regulator Detection and Identification from Crowdsourced Data—A Systematic Literature Review. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 491, [CrossRef]
- 22. Tian, Y.; Gelernter, J.; Wang, X.; Li, J.; Yu, Y. Traffic Sign Detection Using a Multi-Scale Recurrent Attention Network. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 4466–4475, [CrossRef]
- 23. Protschky, V.; Ruhhammer, C.; Feit, S. Learning Traffic Light Parameters with Floating Car Data. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, 15–18 September 2015; pp. 2438–2443, [CrossRef]
- 24. Liu, J.; Shen, H.; Narman, H.S.; Chung, W.; Lin, Z. A Survey of Mobile Crowdsensing Techniques: A Critical Component for The Internet of Things. *ACM Trans. Cyber-Phys. Syst.* **2018**, *2*, [CrossRef]
- 25. Pribe, C.A.; Rogers, S.O. Learning To Associate Observed Driver Behavior with Traffic Controls. *Transp. Res. Rec. J. Transp. Res. Board* **1999**, 1679, 95–100, [CrossRef]
- Golze, J.; Zourlidou, S.; Sester, M. Traffic Regulator Detection Using GPS Trajectories. KN J. Cartogr. Geogr. Inf. 2020, [CrossRef]
- 27. Qiu, H.; Chen, J.; Jain, S.; Jiang, Y.; McCartney, M.; Kar, G.; Bai, F.; Grimm, D.K.; Gruteser, M.; Govindan, R. Towards Robust Vehicular Context Sensing. *IEEE Trans. Veh. Technol.* **2018**, *67*, 1909–1922, [CrossRef]
- 28. Aly, H.; Basalamah, A.; Youssef, M. Automatic Rich Map Semantics Identification Through Smartphone-Based Crowd-Sensing. *IEEE Trans. Mob. Comput.* **2017**, *16*, 2712–2725, [CrossRef]
- 29. Carisi, R.; Giordano, E.; Pau, G.; Gerla, M. Enhancing in vehicle digital maps via GPS crowdsourcing. In Proceedings of the 2011 Eighth International Conference on Wireless On-Demand Network Systems and Services, Bardonecchia, Italy, 26–28 January 2011; pp. 27–34, [CrossRef]
- 30. Kuntzsch, C.; Zourlidou, S.; Feuerhake, U. Learning the Traffic Regulation Context of Intersections from Speed Profile Data. In Proceedings of the Accepted Short Papers from the GIScience 2016 Workshop on Analysis of Movement Data (AMD'16), Montreal, QC, Canada, 27 September 2016.
- Zourlidou, S.; Fischer, C.; Sester, M. Classification of street junctions according to traffic regulators. In Proceedings of the Accepted Short Papers and Posters from the 22nd AGILE Conference on Geo-Information Science, Cyprus University of Technology, Limassol, Cyprus, 17–20 June 2019.
- Meneroux, Y.; Guilcher, A.; Saint Pierre, G.; Hamed, M.; Mustiere, S.; Orfila, O. Traffic signal detection from in-vehicle GPS speed profiles using functional data analysis and machine learning. *Int. J. Data Sci. Anal.* 2020, 10, 101–119, [CrossRef]

- Saremi, F.; Abdelzaher, T.F. Combining Map-Based Inference and Crowd-Sensing for Detecting Traffic Regulators. In Proceedings of the 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems, Dallas, TX, USA, 19–22 October 2015; pp. 145–153.
- 34. Wang, C.; Hao, P.; Wu, G.; Qi, X.; Lyu, T.; Barth, M. Intersection and Stop Bar Position Extraction from Crowdsourced GPS Trajectories. In Proceedings of the Transportation Research Board 96th Annual Meeting, Washington, DC, USA, 8–12 January 2017.
- 35. Méneroux, Y.; Kanasugi, H.; Pierre, G.S.; Guilcher, A.L.; Mustière, S.; Shibasaki, R.; Kato, Y. Detection and Localization of Traffic Signals with GPS Floating Car Data and Random Forest. In Proceedings of the 10th International Conference on Geographic Information Science (GIScience 2018), Leibniz International Proceedings in Informatics (LIPIcs, Melbourne, Australia, 28–31 August 2018; Winter, S., Griffin, A., Sester, M., Eds.; Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik: Dagstuhl, Germany, 2018; Volume 114, pp. 11:1–11:15.
- 36. Munoz-Organero, M.; Ruiz-Blaquez, R.; Sánchez-Fernández, L. Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving. *Comput. Environ. Urban Syst.* **2018**, *68*, 1–8. [CrossRef]
- 37. Cheng, H.; Liu, H.; Hirayama, T.; Shinmura, F.; Akai, N.; Murase, H. Automatic Interaction Detection Between Vehicles and Vulnerable Road Users During Turning at an Intersection. In Proceedings of the 31st IEEE Intelligent Vehicles Symposium, Las Vegas, NV, USA, 19 October–13 November 2020.
- Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings, Banff, AB, Canada, 14–16 April 2014.
- Sohn, K.; Lee, H.; Yan, X. Learning structured output representation using deep conditional generative models. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 3483–3491.
- 40. Kingma, D.P.; Mohamed, S.; Rezende, D.J.; Welling, M. Semi-supervised learning with deep generative models. In Proceedings of the NIPS, Montreal, QC, Canada, 8–13 December 2014; pp. 3581–3589.
- 41. Cheng, H.; Liao, W.; Yang, M.; Rosenhahn, B.; Sester, M. MCENET: Multi-Context Encoder Network for Homogeneous Agent Trajectory Prediction in Mixed Traffic. In Proceedings of the 23st International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020.
- 42. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In Proceedings of the 31st International Conference on International Conference on Machine Learning, Bejing, China, 22–24 June 2014; Volune 32, pp. 1278–1286.
- 43. Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780. [CrossRef]
- 44. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the ICLR, San Diego, CA, USA, 7–9 May 2015.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).