A DEEP LEARNING AND DATA FUSION APPROACH TOWARDS AN OFF-LINE STOP DETECTION METHOD FOR SMARTPHONE-BASED TRAVEL SURVEYS

A PREPRINT

Valentino Servizi
Department of Management Engineering
Technical University of Denmark
Kgs. Lyngby (DK)
valse@dtu.dk

Francisco C. Pereira
Department of Management Engineering
Technical University of Denmark
Kgs. Lyngby (DK)

Otto A. Nielsen
Department of Management Engineering
Technical University of Denmark
Kgs. Lyngby (DK)

February 28, 2019

ABSTRACT

Different machine learning methods such as transport mode detection or purpose inference are based on GPS trajectories collected from smartphone devices. These rely on stop detection and trajectory segmentation techniques at their very root. One of the main difficulties in approaching stop detection as classification problem is defining what a stop is. Let us assume that we are going to catch a metro covering by bicycle the distance between home and the metro stop. In this simple scenario we count 2 stops and observe "bike->walk->metro" as the mode chain if we use "walking mode" as segmentation criterion. We count only 1 stop and observe "bike->metro" as the mode chain if we define a set or rules based on a spatial range and a dwell time centered on the metro station position. In this work we focus on the former criterion, potentially suitable for detecting the most challenging stops. With the purpose of orienting the next steps of this research, we describe: (i) the architecture deployed to collect and label the trajectories, (ii) the data preparation process (iii) and the results of a preliminary experiment based on Deep Learning and data fusion.

Keywords stop detection · mining smartphone data · data fusion · deep learning

1 Introduction

One of the main difficulties in approaching stop detection as classification problem is defining what a stop is.

The literature is rich of studies about trajectories segmentation and stop detection, where dwell time and range are two of the main variables used to define what a stop is [1]. We reviewed a long list of effective heuristics about what are the best dwell time and range to classify the portions of trajectories where the user stops [2].

Another approach identifies the walking mode as indicator for a stop. Every time we stop we walk either before, after or both [2]. For example, when we go picking up the car from the parking or when we switch train. Even when we catch a bus right outside the metro station, between 2 different modes of transportation we always walk.

In smartphone-based travel surveys the data collected to perform stop classification includes GPS trajectories, some times fused with other sensors' measures such as acceleration, heading, gyroscope and possibly Geographic Information Systems (GIS).

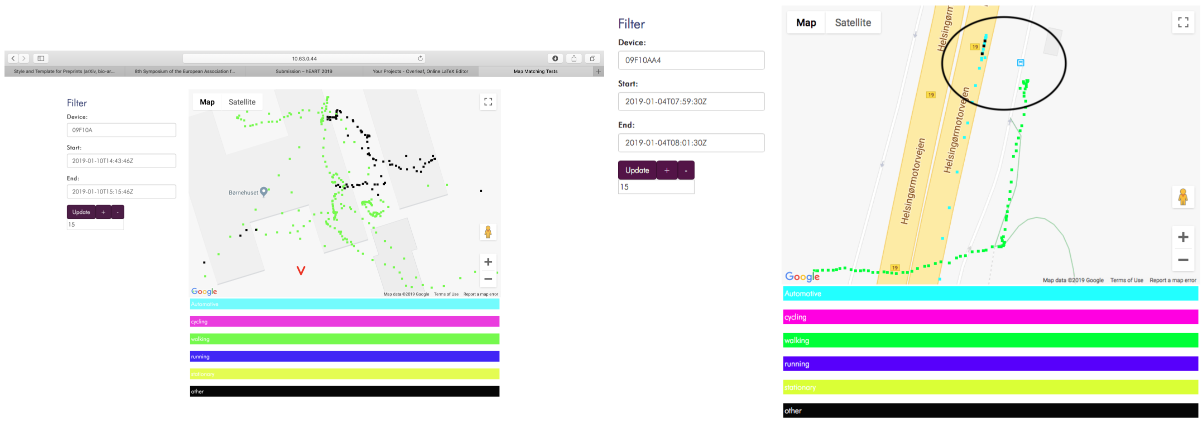

Figure 1: Stop visualization, long (left) versus short (right)

Among the most common methods applied to trajectories for stop classification, rule-based methods seem to perform at the top [3]. About machine learning methods for stop detection, we find various clustering approaches. Popular methods such as Random Forests or Deep Neural Network seem to offer the opportunity of further research, perhaps because of the lac of properly labelled data and high quality ground truth.

There are complex transportation mode chains where some of the stops can be learned and classified only on very few observations (see Fig.1). While there are other stops where the amount of observation is very large (see Fig.1). The former category of stops is very challenging to detect due to the small amount of points observable. Rule based algorithms normally ignore by design such extreme cases, which are not rare in daily trips.

2 Related Work

In this application field we can organise the relevant machine learning methods along 2 main dimensions: (i) location agnostic versus location specific (ii) and user agnostic versus user specific. Each method has advantages and disadvantages.

For example, methods relying on user/ location agnostic features can be trained on any geographic area with users belonging to the local population and then deployed on a different area where another population is present.

On the other hand, even though user/location specific data are difficult to obtain they seem to enable more accurate predictions. This claim is hard to prove due to the heterogeneity of the results available in the relevant literature. Although these results are hardly comparable, within each relevant study we find evidence about the positive contribution of user/location specific classifiers [4, 5, 6].

The approaches and methods about stop-detection and transition-detection described below have been applied for segmenting GPS trajectories with the purpose of e.g. mode-detection and/or activity inference. In [5] the authors focus on Future Mobility Sensing (FMS). They highlight the impact of stop-detection performance on the quality of the ground truth collected from users which validate their trips on the mobile device. The method presented consists in six steps: (i) trajectory cleansing based on the accuracy provided by the AGPS; (ii) rule based stop detection candidates, where stops are points within 50 meters range and 1 minute time window; (iii) check stop-candidates against user frequent stop location; (iv) rule based merging of the resulting stops applying various range/time thresholds; (v) detect still mode by applying a learned classifier based on acceleration measures; (vi) remove extra stops after mode detection algorithm.

In [7] the authors apply a rule based algorithm to detect activity points based on the on-off status of the GPS (because the GPS units employed in the study), speed/time threshold and range/time threshold. Transition-points are identified by applying a threshold on computed acceleration and speed as well as on the time, based on the assumption that travellers walk to change mode.

In [8] the authors define Stay-Points as a geographical area where travellers stay within a range for a certain time. Then, based on these two rules, they apply an *"affinity propagation clustering method"* [8]. Stay-Points belong to a different definition than the transition points used to identify where the travellers change mode in a complex travel mode chain and are identified on a different set of rules based on speed as well as on the assumption that the noisy data typically detected in correspondence of transition points is temporary, while the change in speed are permanent.

In [9], in order to perform mode classification on the trajectories available in the Geolife[1] data-set, the authors apply fixed-size segments of 200 points for both seen and unseen trajectories (where 200 is the median of GPS points in all trips composing the data-set). Then they concatenate together consecutive segments with the same label. They discard segments with less than 10 GPS points. Finally, the trajectories are processed with a Savitzky-Golay filter for smoothing purpose.

In [10], the authors tests four segmentation methods: distance-, time-, bearing- and window- based. They highlight that while the last three are statistically equivalent, the first leads to varying sample size within each segment due to the different speeds in complex transport mode-chains. Stop detection is not mentioned explicitly. The work aims directly to transportation mode detection. Thus, transition points might be identified where there is a discontinuity in the mode-chain detected on these segments.

In [11, 3, 12] we find an extensive list of methods for trajectory segmentation which are presented from different perspectives.

For example, [11] perspective highlights the point- and segment- based methods and the difficulty of comparing the performances between them. Therefore they introduce a penalty system by looking at where these methods make mistakes.

In [3] the authors review a large amount of travel-surveys worldwide and when it comes about stop-detection and trip-segmentation they report that rule-based stop detection techniques relying on range, time, speed or acceleration thresholds are the most common. The authors also highlight the challenges about signal loss and signal noise in the detection of short stops.

In [12] the authors argues *"that the presence of nearby points in Euclidean space may be indicative of an activity, while the absence of nearby points may be indicative of travel"*[12]. Therefore, referring to [2], in order to acquire a local density of points they suggest to deploy a moving window that preserves the relationship with 30 preceding and 30 succeeding points within a 15-m range.

2.1 Problem definition

In the off-line classification of GPS trajectory points and segments according to stop versus motion state of the user carrying a smartphone dedicated to collect the data, signal loss and signal noise [3] are among the main challenges. The classification methods we reviewed often lead to over/under-segmentation [8] of the trajectories and are not designed for detecting short stops. Can deep learning improve detecting erogenous short stops by learning the thresholds, therefore reducing drastically any specification of heuristics?

3 Methodology

In the following paragraphs we describe how data has been collected and processed in order to prepare the data-set.

3.1 Data Collection Front-end

In this phase we focused only on iOS and we implemented a logger App which has been distributed via Test Flight[2]. The application relies on the following services:

1. Core Location [3]. It provides longitude, latitude, altitude, speed, course etc., including the uncertainty measures for location and altitude.
2. Core Motion Accelerometer[4]. It provides the raw acceleration measures on three axes.
3. Core Motion Gyroscope [5]. It provides the measures of the mobile device rotations around the three axes.

[1]Geolife data-set by Microsoft. Rretreived from web 01/01/2019.
[2]Test Flight is the official Apple platform for beta testing. Retreived from web 01/01/2019
[3]Apple Developers Core Location Manual - retrieved from web 01/01/2019
[4]Apple Developers Core Motion Accelerometer Manual - retrieved from web 01/01/2019
[5]Apple Developers Core Motion Gyroscope Manual - retrieved from web 01/01/2019

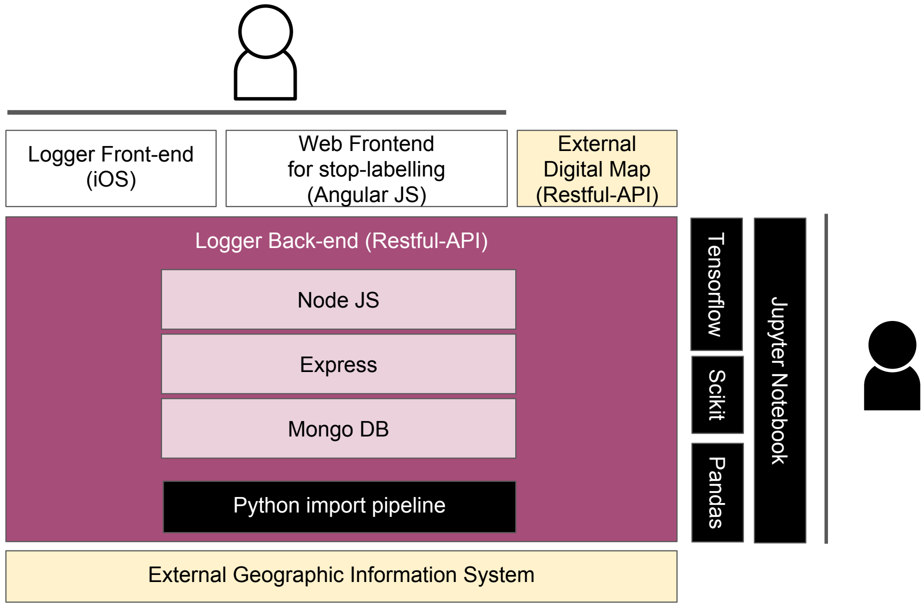

Figure 2: Technical Architecture for transport data collection and processing

4. Core Motion Activity [6]. It provides a classification of the activity performed by whom it carries the mobile device, using six classes: stationary, walking, running, automotive, cycling, unknown.

While e.g. raw acceleration and rotation measurements can be accessed and the update frequency can be set within the app source-code, the location information requires further attention. In particular, due to the very high battery consumption of the Global Positioning System (GPS) necessary to access the location, the operating system provides such a location via the Assisted Global Positioning System (AGPS). At the best of our knowledge, it is not possible hardware-level control within the source-code e.g. on the location update frequency. Furthermore, iOS will override the location updates by shutting down the app i.e. when this is not active while in background, which is the main status of our logger app e.g. when the user is at home or at work. In order to overcome the above issue, the consequence of which is the interruption of the data collection, we focused on some of the tools that iOS provides for battery optimization. (i) We set the second best available location resolution option, which provides a trade-off balance between position accuracy and battery consumption. (ii) we configured the background activity of the app in order to shut down automatically when the telephone is stationary for more than 1 hour. (iii) While shutting down we switch data collection update control choosing the only option that enables the app to restart automatically the data collection when a "relevant" change of position is detected. (iv) Since the option described in (iii) has a lower level of location resolution, we customized the corresponding class in order to allow switching back to the update best resolution as soon as the app has re-started the data collection. (v) In order to further increase the battery efficiency and avoid loss of data in case of missing link between front- and back-end of the logger, we implemented a local database on the mobile device and a rule based data transfer routine.

3.2 Data Collection Back-end

The back-end of the platform built for this research is RESTful API based on MEAN stack which stands for Mongo DB, Express, Angular, NodeJS. Its purpose is to collect the data and make them available for the labelling-operation via web-based interface (see Sec. 3.3) and for the machine learning research (see Fig. 2). The main motivations behind our technological choice are the following. (i) The stack is very fast to build and possibly to integrate with machine learning algorithms; (ii) it is very flexible offering standard access to any client, e.g. mobile-, web- or python-based; (iii) because of the native geo-query features [7] built within Mongo DB, it is extremely powerful when it comes to the interaction with geo-based information.

[6] Apple Developers Core Motion Activity Manual - retrieved from web 01/01/2019
[7] Mongo geo-spatial Query Operators - retrieved from web 01/01/2019

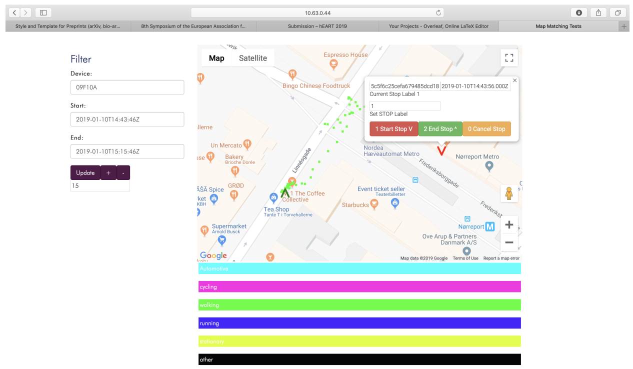

Figure 3: Labelling trajectories, start and end of the stops

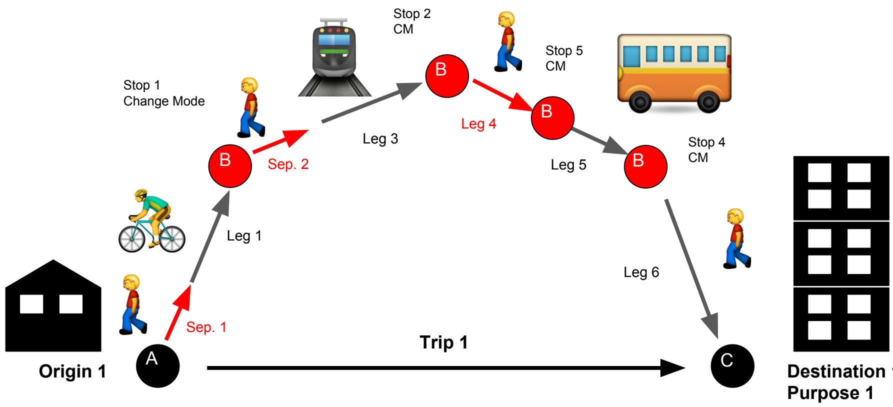

Figure 4: Stop Definition

3.3 Stop labelling and web front-end

The collection of the ground truth which is accomplished through a web Angular-based front-end is one of the most critical steps of the methodology. The interaction-design side of this component, considered crucial in large scale deployment of any smartphone-based travel survey, in this preliminary experiment has been reduced to the minimum. The user is required to mark 2 points for each stop, the first corresponding to the start the second corresponding to the end of the stop (see Fig. 3. The instructions to mark the stops are described in Fig. 4. The Figure provides also the definition of stop taken into consideration in this work. A, B and C are the stops we want to label and each of the stops is either followed or preceded by a walking mode. Therefore, walking mode is considered the separation mark between trip-legs and in this way it determines the location of the stop (see Fig. 4, Sep.1 and Sep.2). However, often walking mode is not just the separation gauge but the transport mode of the whole trip-leg (see Fig. 4, Leg 4 and Leg 6). Often the stop is very short (see Fig. 1). Therefore, only few points can describe these cases. The consequence is that labelling, training and classifying short stops before performing any mode detection can be very challenging.

3.4 Geographic Information System

We leveraged the Mongo DB native geo-query features described in Sec. 3.2 by importing all the relevant data available from Open Street Map directly on Mongo DB. Thanks to the architecture described in Fig.2, the data preparation steps such as the "location specific" features extraction can be performed directly from the Jupyter Notebook which we choose to implement and train the stop classification model.

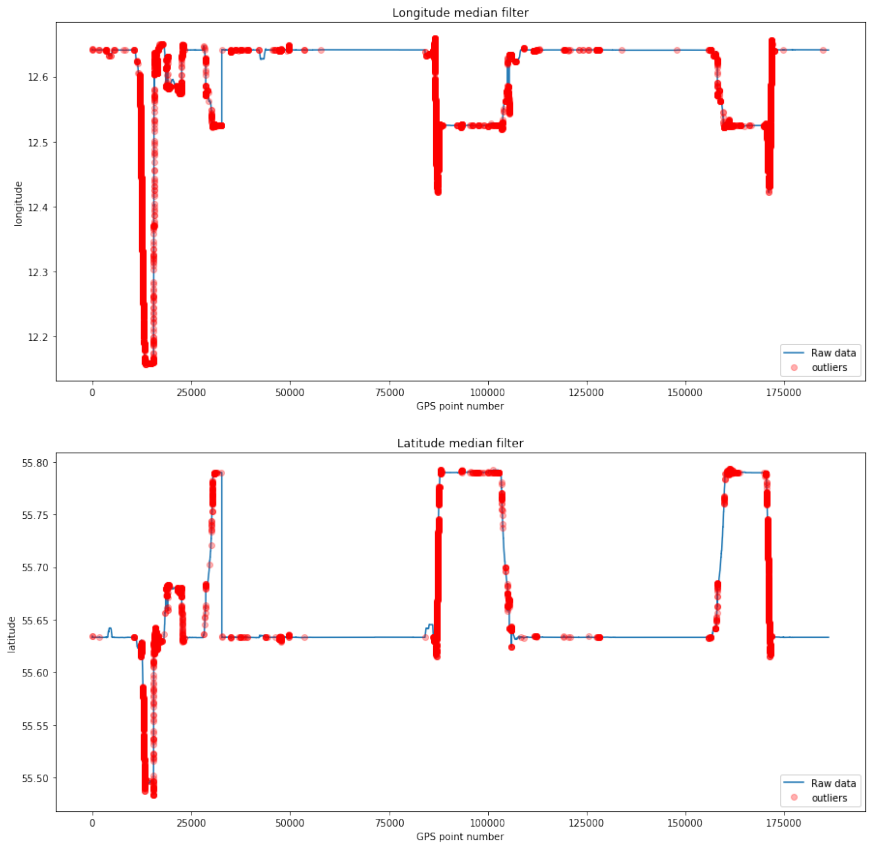

Figure 5: Median filter on longitude and latitude

3.5 Data pre-processing

The data preparation consists in following main steps. (i) Assigning labels according to the two classes stop and motion. The trajectories have been classified as stop-class between the stop markers provided as ground truth by the user (see Sec. 3.3) and as motion-class in any other point. (ii) Trajectories cleansing. AGPS trajectories collected by our logger where often affected by noise and gaps. The main critical areas are overlapping with the trip-legs underground, where the transportation mode was the Metro. However, since Apple does not provide any documentation about how their location service really works (at the best of our knowledge), we can't say whether these faulty segments are caused by well known GPS limitations such as canyon effect [13] or from the error propagation of alternative location estimation techniques deployed either to deal with GPS signal gaps or with battery savings [14, 15, 16, 17, 18, 19]. In order to clean the trajectories we've looked into the main filters such as Naïve Bayesian Classifiers, Kernel Density Methods or Kalaman Filter [20]. Since at the moment they seemed too complex given the stage of this work, we opted for a simple median filter applied on longitude and latitude as described in [9] (see Fig. 5). The application of the filter removed about 2% of the points from the data-set.

4 Deep Learning on Location Specific Features

This section concerns the data-set prepared for the classification experiment, the Deep Learning model implementation and the experiment set-up.

Data-set Description The data-set used for the following experiment is small and it is composed by 6 days trajectories of 2 users for a total of about 185000 GPS points collected in the greater Copenhagen area. For each point we computed the distance from each of the following categories of points of interest (POI) present within the range of 15m: subway stations, railway stations, parking lots, bus stops, traffic lights, supermarkets, fuel stations, charging stations, schools and kindergartens, for a total of 286 relevant features. For each point we collect 3 axes acceleration and and gyroscope measurements. We also apply the heuristics from [2] including the distances of 30 preceding points and 30 succeeding

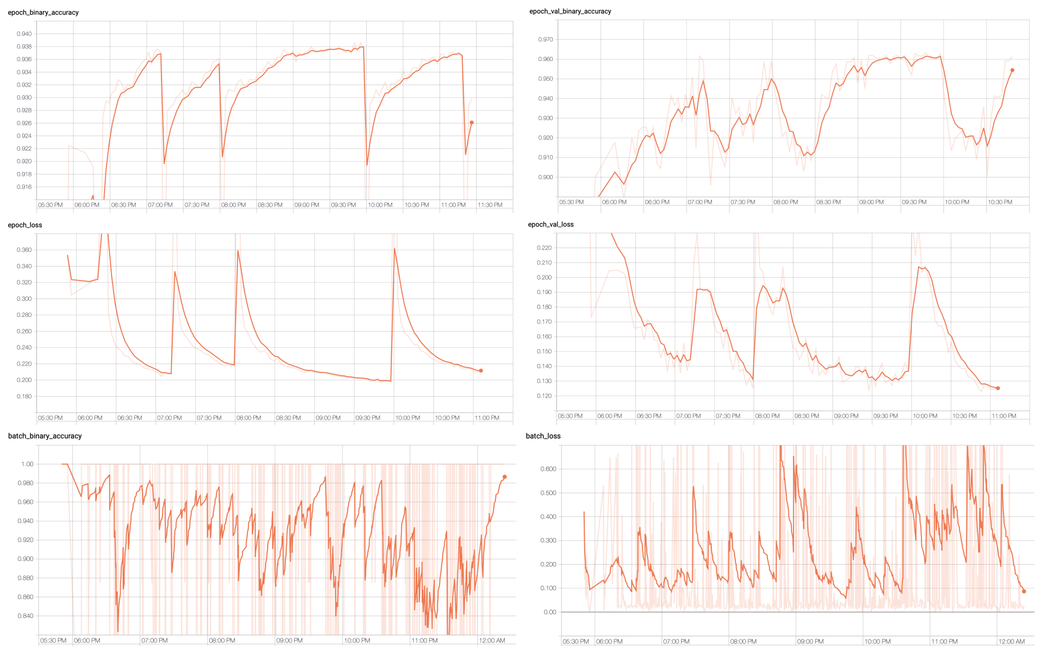

Figure 6: Binary accuracy and loss of training and validation sets

points. Lastly, we provide a temporal context by applying a 4 bins discretisation of the Time of Day (see Fig.7). 85% of the points are labelled as stop, while the remaining points are labelled as motion.

Deep Neural Network Description The network structure in Fig.7 takes inspiration from [9] regarding the convolutional layers connected with the features describing the distances. The layers connected with accelerations and rotations take inspiration from the design of the Recurrent Neural Network described in [10]. Finally, the time discretisations are connected directly to the Multy Layer Perceptron (MLP). Since these three categories of features have different properties they must be processed separately in the bottom layers. Afterwards the output can be processed in the fully connected top layers towards the Softmax layer which provides the final classification. In the present work we focused only in the Convolutional Neural Network highlighted in Fig.7.

Experiment implementation and setup The experiment consists in a 5-stratified-fold cross validation implemented with StratifiedKFold [8]. The data-set is randomized twice, when applying the 5 split and when each split is processed by the model. The model is implemented with Keras[9] Tensorflow. ReLU is the activation function. The batch size is 8 points and the Adam optimizer learning step is 0.0001. With higher values for the learning step we experienced extreme oscillations in the validation set, the size of which is 20% of the training set. In order to compare the results, we train and test a random forest with the same cross-validation scheme and Sklearn[10].

5 Conclusion and future directions

In this preliminary study we analyse the existing work related to stop detection in the field of smartphone-based travel surveys. We find that results are hardly comparable due e.g. to lac of standardisation and heterogeneity of data-sets. The most prominent approaches are rule-based. Such methods are not designed to capture short stops and they could be complemented by more advanced ones. The research gaps on stop detection seem to be at least two: (i) lac of data-sets properly labelled and high quality ground truth, (ii) missing investigation on machine learning methods such as Deep Learning. In order to address the problem defined in Sec. 2.1, we design and implement a dedicated architecture. Then we carry out real data collection and labelling, focusing on stop detection purpose. We process the collected data in order to remove random errors and noise as well as to extract a set of location-specific features. We take into consideration also agnostic features, e.g. acceleration and rotation (see Fig.7). In this work we experimented only with the former

[8] Python Sklearn StratifiedKFold API - retrieved from web 01/01/2019

[9] Keras Tensorflow API - retrieved from web 01/01/2019

[10] Sklearn Random Forest Classifier - retrieved from web 01/01/2019

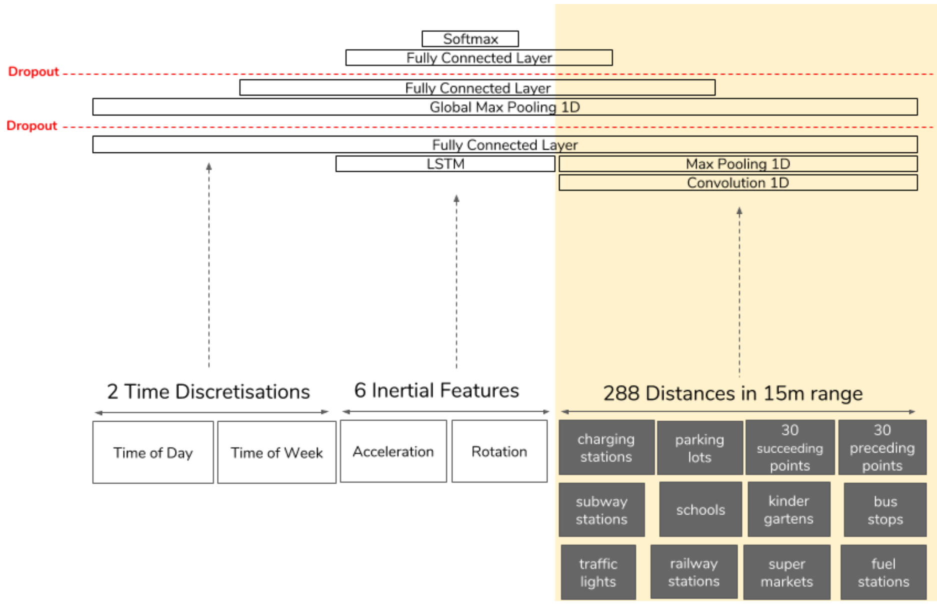

Figure 7: Data and Deep Neural Network Structure

Table 1: Confusion Matrix Convolutional Neural Network

Class	Precision	Recall	F1-Score	Support Points
Motion	0.80	0.75	0.77	23472
Stop	0.96	0.97	0.97	159756
avg / total	0.94	0.94	0.94	183228

location specific group.The classification accuracy measured as average F1 over two classes, namely stop and motion, is 94% (see Fig. 6 and Tab. 1). The Random forest performs slightly better than the Convolutional Neural Network at this stage. Despite the size of the data-set, which does not help to achieve strong conclusions, we could acquaint ourselves with important future directions functional to the scale-up of the study, such as: (i) considering transition stops and activity stops as two classes, (ii) including embeddings in the full model described in Fig.7, (iii) testing a broader range of values for the heuristic applied to measure POIs distances, because the range of 15 meters found in [2] seems not adequate in consideration of the GPS accuracy.

References

[1] Glenn Cich, Luk Knapen, Tom Bellemans, Davy Janssens, and Geert Wets. Threshold settings for TRIP/STOP detection in GPS traces. *Journal of Ambient Intelligence and Humanized Computing*, 7(3):395–413, 2016.

[2] Nadine Schuessler and Kay W. Axhausen. Processing raw data from global positioning systems without additional information. *Transportation Research Record*, 2105(1):28–36, 2009.

[3] Li Shen and Peter R. Stopher. Review of GPS Travel Survey and GPS Data-Processing Methods, 2014.

[4] Youngsung Kim, Ajinkya Ghorpade, Fang Zhao, Francisco C Pereira, P Christopher Zegras, and Moshe Ben-akiva. Activity Recognition for a Smartphone and Web- Based Human Mobility Sensing System. *IEEE Intelligent Systems*, 33(August):5–23, 2018.

[5] Fang Zhao, Ajinkya Ghorpade, Francisco Câmara Pereira, Christopher Zegras, and Moshe Ben-Akiva. Stop detection in smartphone-based travel surveys. *Transportation Research Procedia*, 11(2010):218–226, 2015.

[6] Ivana Semanjski, Sidharta Gautama, Rein Ahas, and Frank Witlox. Spatial context mining approach for transport mode recognition from mobile sensed big data. *Computers, Environment and Urban Systems*, 66:38–52, 2017.

[7] Thomas Kjær Rasmussen, Jesper Bláfoss Ingvardson, Katrín Halldórsdóttir, and Otto Anker Nielsen. Improved methods to deduct trip legs and mode from travel surveys using wearable GPS devices: A case study from the Greater Copenhagen area. *Computers, Environment and Urban Systems*, 54:301–313, 2015.

[8] Qiuhui Zhu, Min Zhu, Mingzhao Li, Min Fu, Zhibiao Huang, Qihong Gan, and Zhenghao Zhou. Identifying transportation modes from raw GPS data. In *Communications in Computer and Information Science*, 2016.

[9] Sina Dabiri and Kevin Heaslip. Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C: Emerging Technologies*, 86(November 2017):360–371, 2018.

[10] Xiang Jiang, Erico N de Souza, Ahmad Pesaranghader, Baifan Hu, Daniel L. Silver, and Stan Matwin. TrajectoryNet: An Embedded GPS Trajectory Representation for Point-based Classification Using Recurrent Neural Networks. In *Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering*, pages 192–200, 2017.

[11] Adrian C. Prelipcean, Gyözö Gidofalvi, and Yusak O. Susilo. Measures of transport mode segmentation of trajectories. *International Journal of Geographical Information Science*, 30(9):1763–1784, 2016.

[12] Justin van Dijk. Identifying activity-travel points from GPS-data with multiple moving windows. *Computers, Environment and Urban Systems*, 70(September 2017):84–101, 2018.

[13] Stratis Kanarachos, Stavros Richard G. Christopoulos, and Alexander Chroneos. Smartphones as an integrated platform for monitoring driver behaviour: The role of sensor fusion and connectivity. *Transportation Research Part C: Emerging Technologies*, (March):0–1, 2018.

[14] Theodoros Anagnostopoulos, Denzil Ferreira, Alexander Samodelkin, Muzamil Ahmed, and Vassilis Kostakos. Cyclist-aware traffic lights through distributed smartphone sensing. *Pervasive and Mobile Computing*, 31:22–36, 2016.

[15] Ulrich Bareth and Axel Küpper. Energy-efficient position tracking in proactive location-based services for smartphone environments. *Proceedings - International Computer Software and Applications Conference*, pages 516–521, 2011.

[16] Thomas Olutoyin Oshin, Stefan Poslad, and Athen Ma. Improving the energy-efficiency of GPS based location sensing smartphone applications. *Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012*, pages 1698–1705, 2012.

[17] Anhua Lin, Jianzhong Zhang, Kai Lu, and Wen Zhang. An efficient outdoor localization method for smartphones. *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, pages 1–8, 2014.

[18] Zhenyu Liu, Mengfei Wu, Konglin Zhu, and Lin Zhang. SenSafe: A smartphone-based traffic safety framework by sensing vehicle and pedestrian behaviors. *Mobile Information Systems*, 2016, 2016.

[19] Zhenyun Zhuang, Kyu-Han Kim, and Jatinder Pal Singh. Improving energy efficiency of location sensing on smartphones. *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, page 315, 2010.

[20] Xue Yang, Luliang Tang, Xia Zhang, and Qingquan Li. A data cleaning method for big trace data using movement consistency. *Sensors (Switzerland)*, 18(3), 2018.