Titre: Title:	Trip mode detection from massive smartphone data
Auteurs: Authors:	Alexis Viallard, Jean-Simon Bourdeau, Catherine Morency, Martin Trépanier, Edwin Vargas, & Hicham Benzamane
Date:	2024
Type:	Article de revue / Article
Référence: Citation:	Viallard, A., Bourdeau, J.-S., Morency, C., Trépanier, M., Vargas, E., & Benzamane, H. (2024). Trip mode detection from massive smartphone data. Transportation research procedia, 76, 37-47. https://doi.org/10.1016/j.trpro.2023.12.036

Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 76 (2024) 37–47

Transportation Research Procedia

www.elsevier.com/locate/procedia

12th International Conference on Transport Survey Methods

Trip mode detection from massive smartphone data

Alexis Viallard[a,b], Jean-Simon Bourdeau[a,b], Catherine Morency[a,b], Martin Trépanier[a,b] *, Edwin Vargas[c], Hicham Benzamane[c]

[a]*Polytechnique Montréal, P.O. Box 6079, station Centre-Ville, Montréal, Canada, H3C3A7*
[b]*Interuniversitary Research Centre on Entreprise Networks, Logistics and Transportation (CIRRELT)*
[c]*Pelmorex Corportation*

Abstract

Nowadays, some smartphone applications require the location of users to be able to provide circumstantial information. However, this data may not be fluid and continuously recorded in a way that can be easily analysed for transport planning purposes. This paper proposes a methodology to reconstruct trips and detect modes from a weather smartphone app data, combined with a validation survey. These results can be useful to create origin-destination matrices and other analyses based on trip data. Our study shows that the Artificial Neural Network (ANN), combined with a proposed data processing framework, provides the best travel mode detection.

Keywords: Smartphone data, travel behaviour, transport planning, GPS data

1. Introduction

In the recent years, with the advent of massive data from tracking devices, it is now possible to recreate trip and path information useful for transport planning purposes. However, this data, in addition to lacking sociodemographic attributes and representativeness, it is often incomplete, if not erroneous, and it brings a challenge for identifying the activity locations, and especially the travel modes. Moreover, many types of tracking devices may be used, each one having its one pros and cons. On the one hand, cellphone location data (related to cellphone towers or zones) may not give precise geographical locations and suffer from poor sampling rates. On the other hand, smartphone apps that are dedicated to travel surveying are quite precise and data-rich but are not widely used and may suffer from

* Corresponding author. Tel.: +1-514-343-7240.
E-mail address: mtrepanier@polymtl.ca

10.1016/j.trpro.2023.12.036

representativeness deficit. At last, commercial smartphone apps will collect much more data, but may also have representativeness issues.

In this paper, we propose a method to process data from a commercial smartphone app that is used by about 1.9 million devices in the province of Quebec (8 million inhabitants). The popular Meteomedia app is used to provide real-time meteorological conditions at the location of the smartphone; this permit to gather geolocation data from users at a frequent pace. Of course, all user information is completely anonymous and privacy measures are strictly enforced.

Dealing with such data quantity (600 million points per day for all the app users around the World) arise computational challenges related to calculation time and resources required to process data. The most refined data processing and mode detection techniques, such as those based on advanced map matching and path calculation, are not applicable here due to calculation time and cost constraints.

Hence, in this paper we propose a method based on feature preprocessing and machine learning techniques to detect activity location of users (so, also trips) and their travel mode. The paper is organised as follows: after a brief literature review, the data source, the framework and many processing methods will be described. Then, the result section will provide a comparison to find the most suitable method for this specific type of data.

2. Literature Review

Literature frequently refers to two main methodologies to detect transport modes. The point-based methods usually infer a transport mode on each recorded point thanks to computed features (Prelipcean et al., 2015; Shah et al., 2014; Yu et al., 2014). Segment-based methods propose to segment trips from raw location data, compute features and finally classify modes (van Dijk, 2018; Xiao et al., 2015). For these works, authors are usually using a dedicated smartphone app or a portable GPS device to collect data, providing a great accuracy and a good frequency of records, but with a low battery autonomy in return (Reddy et al., 2010). Few methods are using passive data from a smartphone app not intended to collect transport-oriented data.

Table 1. The use of different methods for mode classification in the literature.

Authors	Artificial Neural Network	Bayesian Models	Decision Trees (incl. Random Forest)	Fuzzy Logic Rules	Support Vector Machines	Comment
Bantis & Haworth (2017)	X	X	X		X	Uses GIS features to reinforce
Biljecki et al. (2013)	X					Relies on expert systems
Feng et al. (2016)	X	X	X		X	53,258 data points / use dwell times for trip ends
Li et al. (2022)	X					End-of-end framework using CNN
Li et al. (2021)	X		X			Detection of 5 modes
Namdarpour et al. (2021)	X				X	Use Genetic Programming
Rasmussen et al. (2011)				X		100 persons diary / assumes short walking distance between modes
Stenneth et al. (2011)	X	X	X			6 persons for 3 weeks / use closeness to buses
Zheng et al. (2008)		X	X		**X**	45 persons tracked over 6 months

After data has been preprocessed (point-based or segment-based), three main categories are found in the literature to achieve mode classification: statistical methods, rule-based methods, and machine learning methods (Sadeghian et al. 2021). The main idea is to train a model with a sample of data that is labelled with the “correct” modes, and then let the model estimate the mode for the remaining sample. Table 1 summarises the use of these methods by some authors. The reader may also read the prolific literature review provided by Sadeghian et al. (2021) for further information.

3. Methodology

In this section, we will first present the data sources. Then, the general framework is explained, followed by the explanation of the successive methods used to estimate the ends and the mode of the trips.

3.1. Data Sources

This paper deals with rather noisy data, collected indirectly by a location-based smartphone application that, we recall, is not dedicated to transport studies. The sequence of locations provided looks noticeably different from one phone to each other, depending on the platform and the model of smartphones used. In this case study, data is generated by the phone operating system when it is required by the weather app that is running in it. Collecting information this way sometimes makes the record frequency totally unstable, especially if the phone is in low battery mode, or if privacy parameters were enforced on it. The locations provided by the phones are gathered from three different ways: 1) the phone GPS feature (in most cases), 2) GSM positioning (if GPS data is not available, like in the Montreal subway, which is completely underground), and 3) WIFI for phones that are indoor. Moreover, like for any urban location device, its precision relates on the terrain conditions (presence of urban canyons, underground use in the subway system, etc.). Figure 1 provides examples of data collected. In the first case, the location data is frequent and thus this creates a very neat trace that can be analysed thoroughly. In some cases (b), a neat trace may be followed by irregular points, possibly due to phone interruptions. In normal cases (c), the frequency is not so high, but a regular pattern appears.

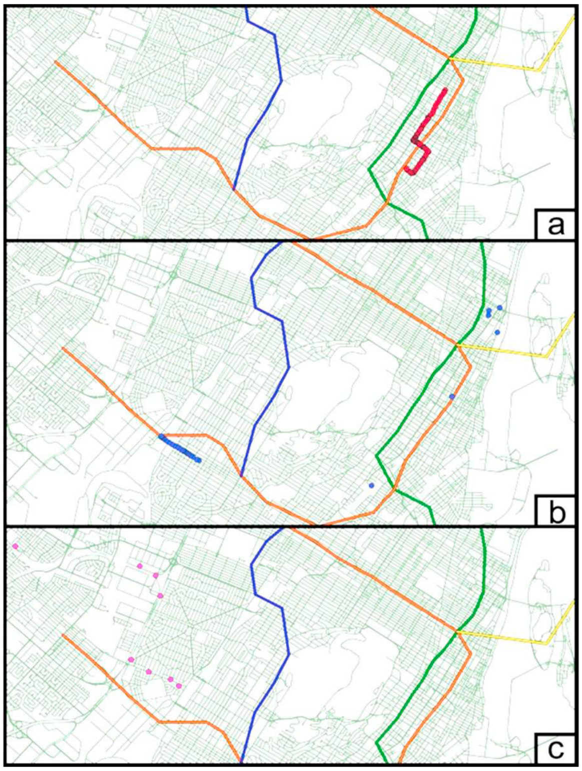

Fig. 1. Sequence of location points: (a) best case; (b) worst case; (c) normal case.

In this study, we aim to use the smartphone app data, covering several days of operation and different weather conditions to test the model (collected in 2019). A hidden feature of a beta version of the weather app was used by employees to label 155 trips used for validation. Modes are added via an interface that displays a list of trips made during the day. But first, the main training of the models necessitates the use of labelled data. This came from a second data source: 8,343 trip observations from the MTL-Trajet smart phone app. MTL-Trajet is an app that is used by volunteers to report their trip in the Montreal area (MTL-Trajet 2022). The city of Montreal provides the traces as open data. The traces cover trips made by inhabitants of Montreal, so most points are located in the Montreal Island and in the surrounding. The traces are very neat; so, as explained later, a degrading process has been used to mimic the weather smartphone app.

3.2. Framework

To tackle this kind of data, we present a framework that uses a segment-based method able to process a massive amount of data with low frequency of records. Figure 2 illustrates the procedure. Denoising and filtering (A) is the first step to be done when dealing with such data quantities. Notably, we remove incomplete traces and outliers that do not fit in the point sequences. This framework is comparable to the one found in the literature, except for the "transform" step (E) that was needed in this case as explained later.

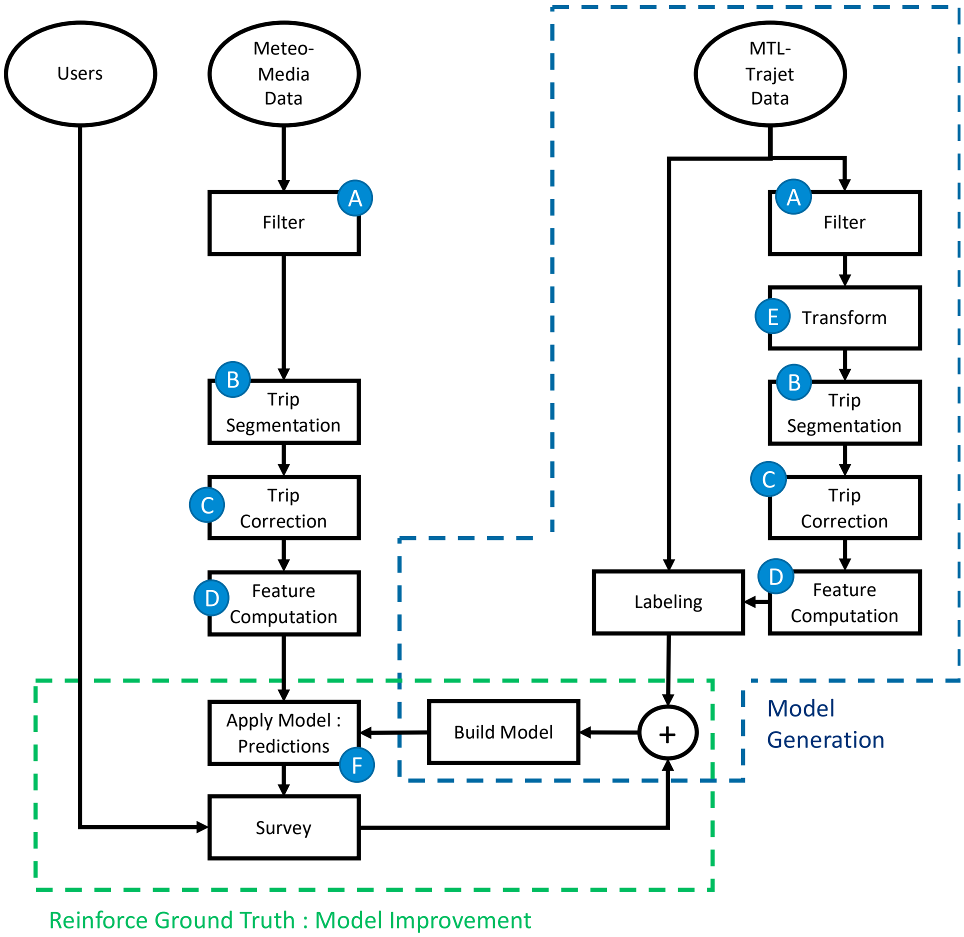

Fig. 2. Overall framework.

3.3. Trip Segmentation

The next step is the trip segmentation method (B). It is based on a proof by contradiction: moving points can be selected because they are not being static. Thus, stationary point detection is made in three steps (Figure 3): first, a density-based clustering algorithm (DBSCAN) is applied on the spatiotemporal data of each unique user id (projected latitude, projected longitude, time), generating heaps of points which are composed of most of the stationary points. This requires a minimum number of 3 points. This clustering algorithm presents the advantage of being fast and not noise sensitive. Since location data is not continuously collected, a second step computes the gap times between successive points and force them to stay stationary in high-value cases. Of course, the trip correction algorithm removes points in congestion (accumulated) from stationary points. Finally, all successive non-stationary points are identified as trip segments.

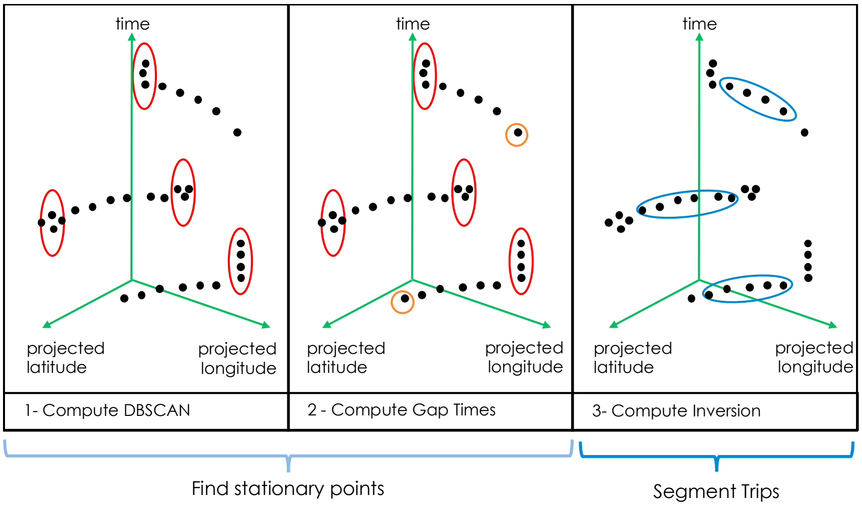

Fig. 3. Trip segmentation process.

3.4. Trip Correction

The next step of the framework is the trip correction, which is added to enhance trip segmentation (C, see figure 4). The first part is to identify the start and end points for each trip segment. For example, if two successive points are within 100 m and less than 20 minutes apart. The next step is to merge points altogether if there is a short time-lapse (i.e., less than 2 minutes) and a short variation in the movement (less than 50 m). Finally, an additional step is needed to split the walking part at the beginning or the end of a trip (for example, if a car is used). This is done using the speed attributes of the points. This method has been developed from empirical evidence from the data we have. The values may be different for another dataset.

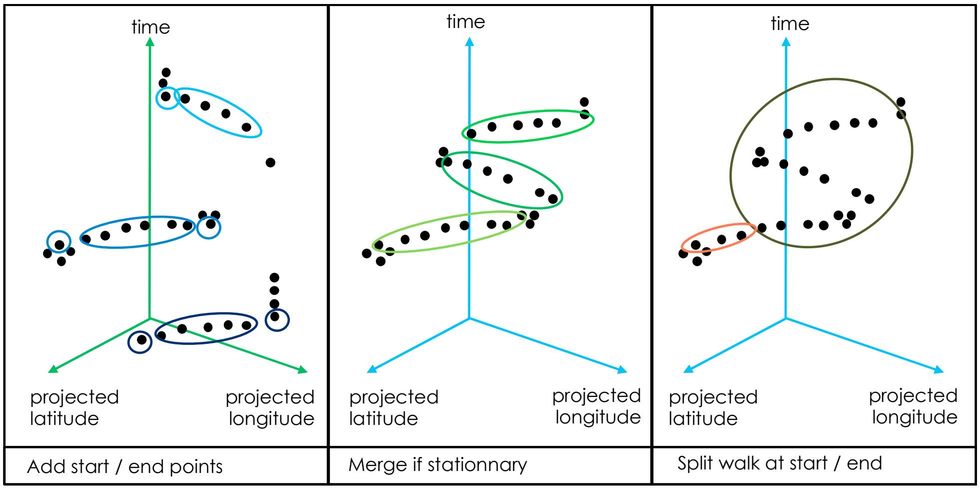

Fig. 4. Trip correction process.

3.5. Feature Computation

Feature computation (D) is the step of defining statistics for each segment. Usually, literature refers to basic statistics including the duration of each trip, the mean, the maximum and the standard deviation of the speed (Yang et al., 2015). According to different data collecting devices, some authors were using accelerometer information (Feng & Timmermans, 2013) or heading changes (Zhang et al., 2015). The introduction of open-source GIS information generally improves classification accuracy (Biljecki et al., 2013). Indeed, knowing that users are on the highway will prevent the classifier to predict walking or riding a bike.

In our case, due to the large quantity of data and the need for real-time processing of the dataset by our research partner, and to reduce computing costs, we do not apply a map-matching procedure to the whole road network, but some important transport features, such as subway and train stations, bike lanes, bus routes and main highways only, are matched to the trips and added as attributes. With this, we obtain the features in Table 2.

Table 2. Trip features used in this framework.

Feature	**Description**
Accuracy	Mean of accuracy provided by the phone app; usually low in the metro network, meaning a more precise location (m).
Speed	Standard deviation, average, 95th percentile of speed (km/h) calculated with two successive point (Euclidean).
Time	Time difference between two points, time of day (s).
Access to mode	Mean distance to bike lane, subway, bus routes (m).
Bus use	Ratio of points on the same bus route.
Location on stations	Tells if points are on bikesharing station, metro station, main highway.
Weather	Weather conditions (rain/snow, air temperature) (mm, ºC).

3.6. Prediction model

Classification algorithms must learn from ground truth. Instead of using data from several volunteers as some papers do (Stenneth et al., 2011), we use the learning open data of trips made in Montreal from the MTL-Trajet app (MTL Trajet, 2017). Here comes a relevant issue: MTL-Trajet data looks slightly different from the weather app data.

Indeed, this framework has been built from applying a survey after a trip segmentation to know the used modes. Even if the main information already looks the same, implicit characteristics as accuracy and frequency are always better. The stake is to transform (E) MTL-Trajet data to look like MM data in an implicit way and then apply the same methodology to get trip features. In this case, we have randomly removed points from the MTL-Trajet data to obtain the same average amount of points per trip than the weather-app dataset.

We use an artificial neural network (ANN) as a classifier. ANN is renowned to be an efficient mode detection (Bantis & Haworth, 2017; Feng & Timmermans, 2016) and can learn more complex data structures. The model is built from the MTL-Trajet labelled trips (F). As an improvement, the weather app has been modified to include a mini-travel survey that is used to continuously reinforce the model. At this stage, this feature is at beta stage and has been used only by employees of the research partner. This way, in a long-term, this information would replace the MTL-Trajet ground truth and decrease the possible bias caused by the data transformation process.

4. Results

In this section, we first present the experimental setup that was developed, then show an example of results, and compare some methods applied to the case study.

4.1. Experimental setup

The framework has been implemented in Python and applied to MTL-trajet data (8,343 trips for learning) and the weather app data. For the latter, a total of 155 trips were labelled with the help of the trip mode labelling feature of the weather app. The results of the various models developed (many mixes of features and type of prediction method were tested) are presented in two Microsoft Excel dashboards that gather results from the Python environment.

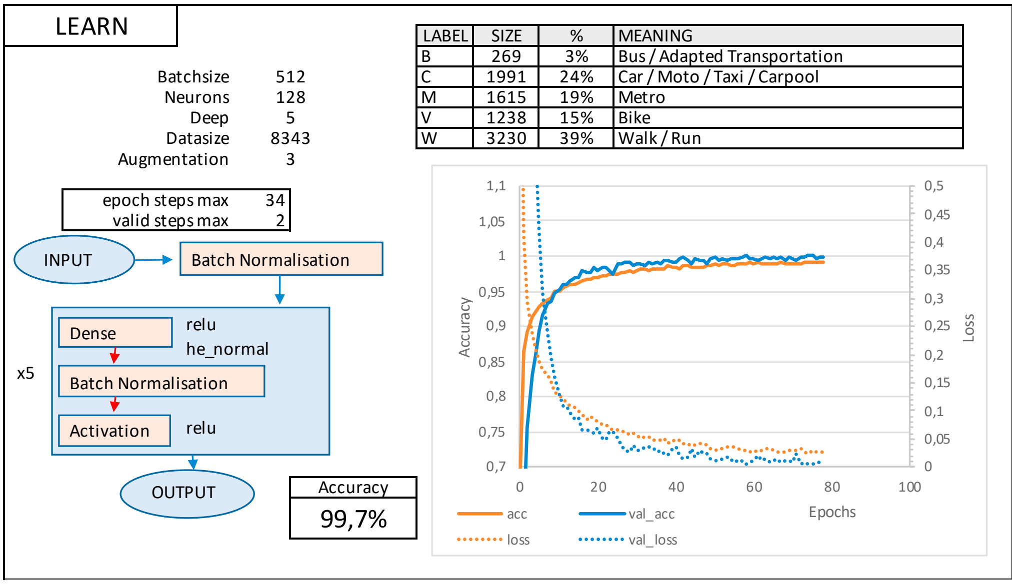

Fig. 5. Experimental dashboard for the learning phase.

The “learn” dashboard (figure 5) presents the results of the application of the artificial neural network to the MTL-trajet dataset. The figure shows the set of hyperparameters that were chosen, following many tests of combinations, validated with a minimal loss function value. The distribution of trips through modes is shown at upper right. At the left are the properties of the ANN. The chart shows the evolution of the ANN learning attributes. The “predict” dashboard (figure 6) shows the confusion matrix of the modes predicted by ANN versus the labels of the weather app dataset. This illustrates which modes are better predicted and helps to fine-tune the model, especially for the trip detection and correction steps where arbitrary decisions are made to identify trips. In this case, the "walk" mode has a higher detection success. Note that the success rate applies to the mode detection only, it is not related to the trip segmentation process.

PREDICT

Accuracy
78%

LABEL	SIZE	%
B	5	3%
C	37	24%
M	30	19%
V	23	15%
W	60	39%
Total	155	

		PREDICTIONS				
		B	C	M	V	W
TRUE	B	3				2
	C	2	22	1	12	
	M	1	1	24	2	2
	V		6		16	1
	W	1		3		56

True True	60%	59%	80%	70%	93%

Fig. 6. Experimental dashboard for the predicting phase (B=bus, C=car, M=subway, V=bike, W=walk).

4.2. An Example of Feature Used for Mode Detection: Mean Speed

The features available for each trip and each trip segment (part of a trip) can vary a lot from an observation to another. Thus, there is a need to use a model that is flexible enough to "learn" from the different combinations of features. That is why machine learning or deep learning models (such as ANN) are used. Figure 7 is presented here to exemplify this, applied to the mean speed of trip segments. In the figure, we see that there are distinctive profiles for walk (slower speed) and car (with higher speed). However, the speed distributions associated to subway, bike and bus overlap totally or in part, so the model needs additional features to choose the mode.

4.3. Comparison of Prediction Models

We tested four prediction models: Artificial Neural Network (ANN), Bayesian networks, Random Forest and Support Vector Machine (SVM). Every model has been calibrated to provide the best result possible. The LEARN column shows the results for the learning process applied to the MTL-trajet data, and the PREDICT column relates to the application of the models to the weather app data. As an additional information, we also provide results from previous authors who applied these models to their dataset (however, the models do not have the same characteristics). As mentioned in the literature, it appears that the best type of model to use is strongly depended on the nature of data and its features. Table 3 presents our results of the application of the four models. In our case, the ANN performs the better, with a prediction success of 82.5%. We consider that it is a good result, given the "poor" quality of this weather app dataset, not designed for transport studies.

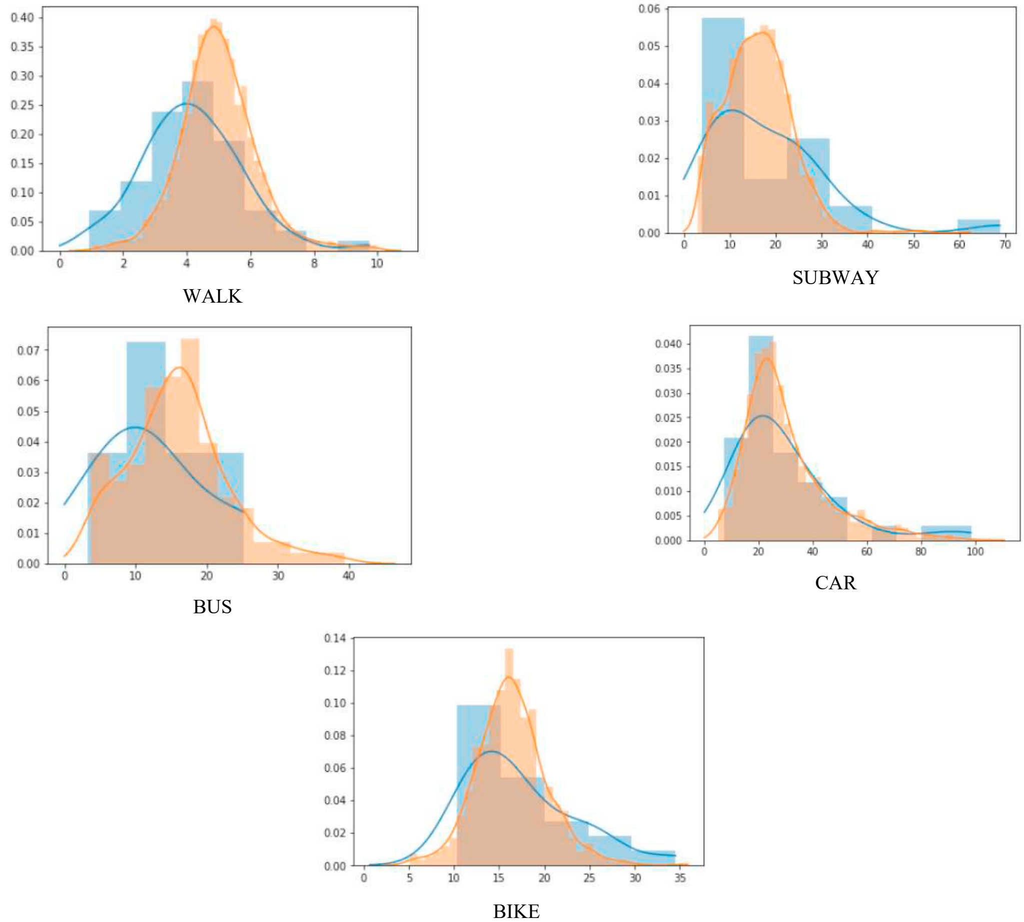

Fig. 7. Mean speed distribution for labelled modes (orange = MTL-trajet data, blue = weather app data) (X axis = mean speed (km/h); Y axis = proportion of trip segments).

Table 3. Comparison of models in the literature and with our dataset.

Model Type	Stenneth et al. 2011	Feng et al. 2016	Bantis et al. 2017	Proposed method MTL-Trajet trips (LEARN)	Proposed method Weather app trips (PREDICT)
Neural Network	83.3 %	97 %	82 %	**96.0 %**	**82.5 %**
Bayesian Network	92.5 %	**99 %**	**90 %**	80.7 %	69.7 %
Random Forest	**93.7 %**	98 %	82 %	92.9 %	79.4 %
SVM	N/A	94 %	85 %	74.4 %	60.6 %

5. Conclusion

In this section, we will recall the main contributions of this paper, then discuss about some limitations of the approach and state some research perspectives.

In this paper, we proposed a general framework to detect the trips and the travel mode for a dataset provided by a smartphone weather app. The characteristics of the dataset made it impossible to use with methods found in the literature. In addition, the large number of records to be processed forced us to choose a method that involved less map matching. In our case, the artificial neural network (ANN) was found to be the most effective in detecting travel mode.

Some limitations can be identified regarding this work. First, the validation dataset is quite small as this point because the travel mode validation function has not been implemented in the weather app yet. The "transform" (step E) process is also a limitation because of the randomness of the point removal process used. More relevant statistical measures could be used for the comparison. Next, the hyperparameters of the model were found with a simple heuristic, a more elaborate process could be put in place to find the best combination. Another limitation is the inability to provide a success rate for the trip segmentation process, it only applies to mode detection. Privacy concerns amongst the users of the app may hamper this development. Another limitation is the actual choice of features in the model and the way these features are calculated. As stated, the computational power needed to calculate them forced us to make some simplification choices, but his may evolve with the availability of more computer clouding power. Last, but not least, are the privacy measures that are now put in place by phone operation systems developers (Apple iOS in particular): a user can now blur and block the geolocation features of his/her phone. For a weather app, the exact location is not needed, so this will degrade much further the data available for transport studies.

However, there is still much to do with this type of data. The dataset could be used for the estimation of intercity and international trip data (the app is used around the world). Another challenge is to be able to measure the representativeness of this data; this is further needed to use this data to estimate overall volumes on mode usage, and origin-destination matrices.

Acknowledgements

The authors wish to acknowledge the financial support of the Natural Science and Engineering Research Council of Canada. They also thank Pelmorex that provided the data. The contribution of the City of Montreal, through the use of the MTL-trajet data, is also highlighted.

References

Bantis, T., & Haworth, J. (2017). Who you are is how you travel: A framework for transportation mode detection using individual and environmental characteristics. Transportation Research Part C: Emerging Technologies, 80, 286–309.

Biljecki, F., Ledoux, H., & Oosterom, P. van. (2013). Transportation mode-based segmentation and classification of movement trajectories. International Journal of Geographical Information Science, 27(2), 385–407.

Feng, T., & Timmermans, H. J. P. (2013). Transportation mode recognition using GPS and accelerometer data. Transportation Research Part C: Emerging Technologies, 37, 118–130.

Feng, T., & Timmermans, H. J. P. (2016). Comparison of advanced imputation algorithms for detection of transportation mode and activity episode using GPS data. Transportation Planning and Technology, 39(2), 180–194.

MTL-Trajet (2022), Service des infrastructures, de la voirie et du transport - Direction des transports. Trips MTL Trajet. from https://montreal.ca/programmes/etude-mtl-trajet (visited on Feb 2022).

Li, Z., Xiong, G., Wei Z., Lv, Y., Anwar, N., Wang, F.-Y. (2022). A Semisupervised End-to-End Framework for Transportation Mode Detection by Using GPS-Enabled Sensing Devices. IEEE Internet Of Things Journal, 9(10), 7842-7852.

Li, J., Pei, X., Wang, X., Yao, D., Zhang, Y., Yue, Y. (2021). Transportation Mode Identification with GPS Trajectory Data and GIS Information. Tsinghua Science And Technology, 26(4), 403-416.

Namdarpour, F., Mesbah, M., Gandomi, A.H., Assemi, B. (2021). Using genetic programming on GPS trajectories for travel mode detection. IET Intelligent Transport Systems, 16, 99-113.

Prelipcean, A. C., Gidofalvi, G., & Susilo, Y. O. (2015). Comparative framework for activity-travel diary collection systems. 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 251–258.

Rasmussen, T.K., Bláfoss Ingvardson, J., Halldórsdóttir, K., Nielsen, O.A. (2015). 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, Volume 54, pp. 301-313.

Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., & Srivastava, M. (2010). Using Mobile Phones to Determine Transportation Modes. ACM Trans. Sensor Netw.

Sadeghian, P., Håkansson, J., Zhao, X. (2021). Review and evaluation of methods in transport mode detection based on GPS tracking data, Journal of Traffic and Transportation Engineering (English Edition), 8(4), 467-482.

Shah, R. C., Wan, C., Lu, H., & Nachman, L. (2014). Classifying the Mode of Transportation on Mobile Phones Using GIS Information. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 225–229.

Stenneth, L., Wolfson, O., Yu, P. S., & Xu, B. (2011). Transportation Mode Detection Using Mobile Phones and GIS Information. Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 54–63.

van Dijk, J. (2018). Identifying activity-travel points from GPS-data with multiple moving windows. Computers, Environment and Urban Systems, 70, 84–101.

Xiao, G., Juan, Z., & Zhang, C. (2015). Travel mode detection based on GPS track data and Bayesian networks. Computers, Environment and Urban Systems, 54, 14–22.

Yang, F., Yao, Z., & Jin, P. J. (2015). Gps and acceleration data in multimode trip data recognition based on wavelet transform modulus maximum algorithm. Transportation Research Record: Journal of the Transportation Research Board, (2526), 90–98.

Yu, M.-C., Yu, T., Wang, S.-C., Lin, C.-J., & Chang, E. Y. (2014). Big Data Small Footprint: The Design of a Low-power Classifier for Detecting Transportation Modes. Proc. VLDB Endow., 7(13), 1429–1440.

Zhang, L., Liu, L., Bao, S., Qiang, M., & Zou, X. (2015). Transportation Mode Detection Based on Permutation Entropy and Extreme Learning Machine. Mathematical Problems in Engineering.

Zheng, Y., Liu, L. Wang, L. & Xie, X. (2008). Learning transportation mode from raw gps data for geographic applications on the web. In Proceedings of the 17th international conference on World Wide Web (WWW '08). Association for Computing Machinery, New York, NY, USA, pp. 247–256.