A Preliminary Investigation of Machine Learning Approaches for Mobility Monitoring from Smartphone Data.

Claudio Gallicchio [1][0000-0002-6692-2564] Alessio Micheli [1 [0000-0001-5764-5238]], Massimiliano Petri [2 [0000-0003-2402-2055]], Antonio Pratelli [2]

[1] Department of Computer Science, University of Pisa, Pisa, Italy
gallicch@di.unipi.it, micheli@di.unipi.it
[2] Department of Industrial and Civil Engineering, University of Pisa, Pisa, Italy
m.petri@ing.unipi.it, a.pratelli@ing.unipi.it

Abstract. In this work we investigate the use of machine learning models for the management and monitoring of sustainable mobility, with particular reference to the transport mode recognition. The specific aim is to automatize the detection of the user's means of transport among those considered in the data collected with an App installed on the users smartphones, i.e. bicycle, bus, train, car, motorbike, pedestrian locomotion. Preliminary results show the potentiality of the analysis for the introduction of reliable advanced, machine learning based, monitoring systems for sustainable mobility.

Keywords: Sustainable Mobility, Machine Learning, Transport Mode Recognition.

1 Introduction

The results of the GOOD_GO platform testing application made in Leghorn Municipality show how the union of its disincentive system for bike theft with the sustainable mobility rewarding system are able of attracting citizens to the use of the APP, providing an important flywheel to encourage sustainable mobility and for bottom-up and low-cost monitoring of daily movements and the impacts of mobility actions implemented by city administrations.

Results also indicated some critical points of the platform, in particular the following elements:

- many users forget to start the movement monitoring and to indicate the mode of transport used inside the GOOD_GO App for smartphone;
- the stolen bicycles detection system is excessively expensive and requires more automation;
- despite the advantage of the very low cost, RFid passive tags are affected by the metal noise of nearby bicycles, significantly disturbing radio frequency messages.

In this paper we describe a way to address the first critical point by investigating the use of Artificial Intelligence, and in particular of Machine Learning, approaches to the

management and monitoring of sustainable mobility. The aim is to automatize the recognition of the user's means of transport among those considered in the data collected with the current application (GOOD_GO smartphone App), namely: bicycle (bike), bus, train, car, motorbike, and pedestrian locomotion (foot). In order to develop an automatic detection system, a series of activities have been pursued, in relation to: data acquisition and pre-processing for mobility purposes, formulation of the computational task in the Machine Learning context, selection of input features and of learning models, analysis of the results.

The rest of this paper is structured as follows. In Section 2 we introduce the innovative features of the GOOD_GO system and its links to rewards and anti-theft systems. In Section 3 we describe the adopted methodologies for estimation of the transport mode from smartphone gathered streams of data, focusing on the required pre-processing steps and on the phases of data and learning models selection. The results of our experimental analysis are given in Section 4. Finally, in Section 5 we delineate conclusions and future perspectives of our work.

2 The GOOD_GO System

Briefly, here, the whole GOOD_GO sustainable mobility platform and its related SaveMyBike system is presented, remanding readers to other papers where its framework is described in more detail [1, 2, 3, 4]. Moreover, we present some data relative to the prototypical test already done involving about one thousand inhabitants of Leghorn Municipality in the end of 2018.

The Good_Go Platform is a ‘space of services’ for sustainable mobility users linked to ITS sensors and an ICT social platform capable of:

- monitoring bicycle trips and all the other transport modes by using an APP for smartphone;
- creating secure areas for private bike parking;
- finding stolen bicycles;
- rewarding people who perform sustainable trips in the city;
- organizing sustainable mobility competition at different scale level (whole city, institutional system like hospital, university or single company/school).

The platform, by means of the previous presented services, tries to develop features able to attract the interest of citizen in the use of the App and so able to build a significative population sample (at low-cost respect to other data acquisition method like the use of ITS) with the relative trips data. The open source nature of the platform follows the same criteria of low-cost monitoring system, with the possibility to use the code and the App without any license and, then, enabling the mobility monitoring also for little-medium municipalities where financial resources for expensive ITS system are not available.

The testing application done for 4 months in the end of 2018 has showed a great appeal for citizen because in only one week we reached the maximum number of participants (for the prototypical application) of one thousands subscribers to the

GOOD_GO app. In the application more than 1.500 trips were collected along with all information regarding transport modes, emissions, cost and health indices (see **Fig. 1**).

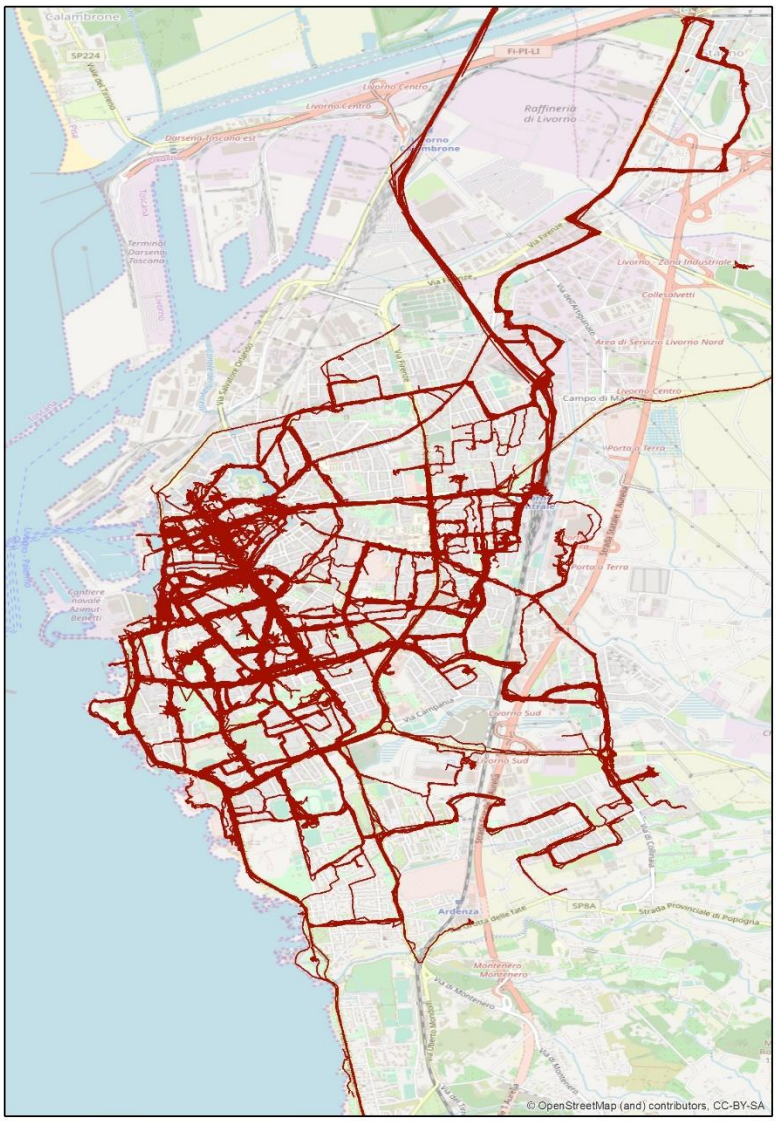

Fig. 1. Monitored trips in the Leghorn testing case.

The App, at the moment, has a section where users need to indicate manually the transport mode (see **Fig. 2**). In this way, the manual insertion of the transport modes with the tracking by a non-automatic start and end becomes an element of weakness of the system as it introduces possible errors due to following facts:

- the user can forget to start or stop the tracking;
- the user can forget to change transport mode in intermodal trips;
- the user can insert wrong transport mode trying to collect a greater number of points for the reward system (even if there are, however, empirical rules useful to identify these erroneous data entries).

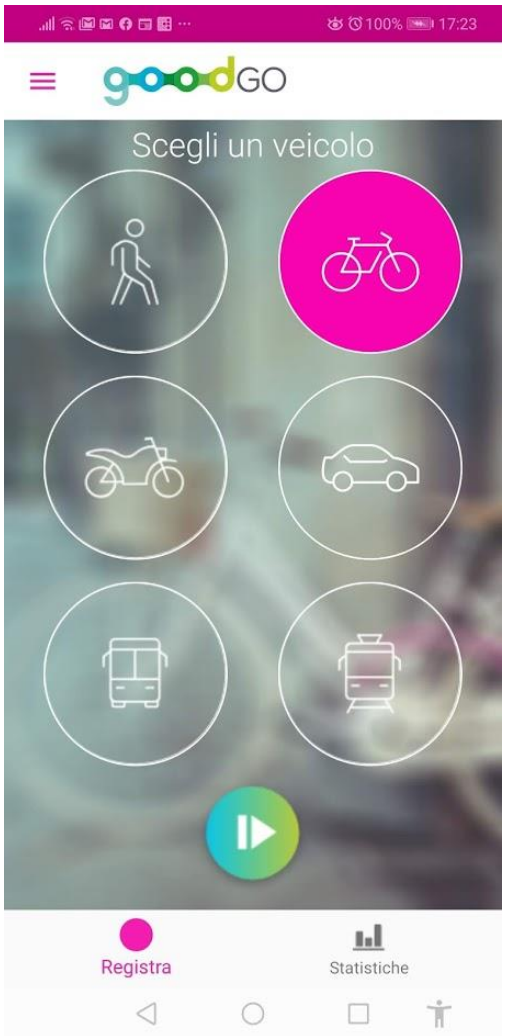

Fig. 2. The form of the smartphone App where to start and stop tracking, selecting the transport mode used.

3 Methods

In this section, we describe the adopted methodology for data processing and setup of the learning models. Specifically, we describe the aspects of data pre-processing and learning task definition in Section 3.1, while in Section 3.2 we focus on feature and learning models selection.

3.1 Data acquisition, pre-processing and formulation of the computational learning task

A careful analysis of the data collected from the users registered to the system through the App developed in the project was necessary. The data were critical in terms of uniformity of sampling, number of available samples for the different means of transport, missing data, significance of the available features (attributes), as well as noise of the samples. While the last two characteristics are common to applications that require the use of Machine Learning methods, and motivate its use, the former have required pre-processing operations that include filtering and imputation (replacement of values).

The input variables include 3D accelerometers, pressure, proximity, speed, longitude, latitude, roll, pitch, bearing, and lumen. In particular, the pre-processing operations applied to each input variable were the following: (a) uniform resampling at constant 1 second resolution, i.e., 1 Hz; (b) filling of NaN (not available) values using padding with the last valid observation; (c) filtering using moving average over periods of 1 minute; and (d) uniform resampling at constant 1 minute resolution. Finally, to create the inputs to be provided to the Machine Learning models, for each input sequence we extracted 3 features for each input variable, namely *average* (avg), *standard deviation* (std), and *maximum value* (max). The feature extraction in the considered form has allowed the analysis through a set of Machine Learning approaches for vector data (see Section 3.2), in the light of a first evaluation of the involved challenges. The learning problem is configured as a multi-class classification task with 6 classes, one for each transport mode, starting from the streams (traces) of sensors data extracted from the smartphone App (and preprocessed as described above).

Overall, the extracted dataset includes 2636 samples, 33 input variables, plus 1 target class variable that encodes the transport medium.

A significant and critical aspect in the present dataset is the strong unbalance of the classes present, i.e. the sequences recorded for each type of transport mode. As shown in **Fig. 3**, the vast majority of the data pertains to the bike transport mode ($\approx 82\%$), followed by foot ($\approx 8\%$) and bus ($\approx 7\%$), while train and car modes of transportation both represents $\approx 1\%$ of the available data. Only 12 samples for motorbike transportation are available (less than 1% of the data). To counteract the effects of this imbalance in the available data, resampling policies (oversampling) have been considered. Also in consideration of these aspects, the assessment of the learning models' performance was conducted by using both multi-class accuracy (on the 6 classes), and macro F1 score, as follows:

$$accuracy = \frac{\sum_{i=1,\dots,6} tp_i}{\sum_{i=1,\dots,6} N_i},$$

$$F1 = 2\frac{precision_{av}\, recall_{av}}{precision_{av} + recall_{av}},$$

where tp_i indicates the number of true positives for the i-th class, N_i is the total number of samples pertaining to the i-th class, $precision_{av}$ and $recall_{av}$ respectively denote the precision and the recall measures, macro-averaged among the classes.

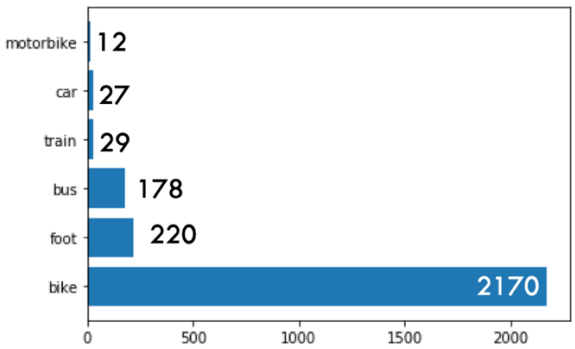

Fig. 3. Available samples for each transport mode.

3.2 Selection of models and features

In consideration of the peculiarity of the available data, and to favor the simplicity of the system, the considered approach has been based on the use of features extracted on whole sequences (recorded temporal traces), as described above in Section 3.1. Data was divided into a training (or development) set and an external test set (unseen during model calibration phase), according to a stratified 80%-20% split.

We explored different version of the learning tasks, originated from different feature selection policies on the available data. Specifically, we analyzed the following three configurations: (a) *full features*, in which all the input features (processed as described in Section 3.1) were considered; (b) *selected features*, comprising the 10 features that were found to be maximally correlated (in absolute value) with the target variable (i.e., max, avg and std of device bearing, max, avg and std of speed, max lumen, std pressure, avg X-accelerometer, and std Z-accelerometer); (c) *ad-hoc features*, where a minimal set of features were selected based on an a-priori presumable significance (i.e., max speed and std on the 3D accelerometers).

In our preliminary experimental analysis, we considered a set of classification models comprising different methodologies, including feed-forward Neural Networks, instantiated as Multi-layer Perceptrons (MLPs) with 1 or 2 hidden layers [5], Random Forest (RF) [6], and K-Nearest Neighbors (K-NN) [7]. All of these learning models

were evaluated on all the data configuration described above, i.e. full features, selected features, and ad-hoc features. The software library (scikit-learn) is publicly available [8].
The hyper-parameters of each learning model were optimized (individually for each model) on a nested level of stratified 5-fold cross validation on the training (development) split, using grid search. In particular, for MLP with 1 hidden layer, i.e. MLP-1, we explored values of the hidden layer's size (number of units) in $\{10, 50, 100, 500\}$. For MLPs with 2 hidden layers, i.e. MLP-2, we explored cases the 2 hidden layers had the same size, varying in $\{10, 50, 100, 500\}$. For RF, we explored configurations with a number of estimators (decision trees) in $\{10, 20, 50, 100, 200, 500\}$. For K-NN, we explored the size K of the neighborhood in $\{3, 5, 10, 50, 100\}$. All other hyper-parameters were set to the default values, using the scikit-learn library [8].

4 Results

In this section, we describe the results achieved by our experimental analysis. In consideration of the fact that the available dataset is heavily imbalanced, the macro F1 score was used at phase of model selection, while for test assessment we used the accuracy, on order to have a score that is closer to human understanding.

The achieved results are reported in **Table 1**, which shows the validation and test performance achieved by MLP-1, MLP-2, RF and K-NN on the three dataset configurations considered (i.e., full features, selected features, ad-hoc features).

Within the limits of the preliminary investigation targeted in this work, the results appear to be very good, with the best models having F1 values and accuracy greater than 0.9 on both validation and external test data. Overall, the best result is achieved by RF in the case of full features configuration, reaching 0.904 of F1 score on validation, 1.00 accuracy on training, and 0.919 accuracy on test. We can also observe that, in general, the higher performance is obtained in the full features configuration, with a gentle reduction in correspondence of the cases of selected features and ad-hoc features configurations. This indicates that the quality of the estimation does not degrade dramatically when a smaller set of input sources is available to the system.

The performance of the best overall model (RF with full features) is further analyzed in **Fig. 4**, which shows the corresponding confusion matrix on the test set. It is evident to see that the model achieves high accuracy especially on the 3 classes that are sufficiently sampled, i.e. foot, bike and bus. The performance is lower in correspondence of the under-sampled classes, i.e. motorbike, car and train which have 1% of the data compared to the bike class alone, and on which therefore also the test estimation is much less statistically significant.

Table 1. Results achieved on all the dataset configurations by the considered learning models. Best results are highlighted in bold font.

Model	Features	Validation F1-score	Test accuracy
MLP-1	*full*	0.600	0.748
MLP-1	*selected*	0.426	0.608
MLP-1	*ad-hoc*	0.455	0.689
MLP-2	*full*	0.688	0.828
MLP-2	*selected*	0.549	0.773
MLP-2	*ad-hoc*	0.578	0.710
RF	*full*	**0.904**	**0.919**
RF	*selected*	0.902	0.903
RF	*ad-hoc*	0.858	0.814
K-NN	*full*	0.542	0.813
K-NN	*selected*	0.545	0.777
K-NN	*ad-hoc*	0.521	0.754

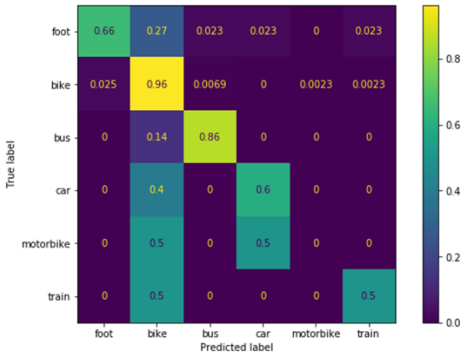

Fig. 4. Confusion matrix for RF, computed on the test set.

Having covered with significantly high values the accuracy for the three main transportation modes where sampling was sufficient for calibration and testing (i.e. bike, bus and foot), the system has shown its potentiality and its flexibility in this context. In addition, confusion matrices of other learning models, such as those based on neural networks, showed a better behavior than RF in some cases confined to specific classes. For example, in **Fig. 5** we show the confusion matrix for MLP-1 with full features, from which we can see a gain, in comparison to RF in **Fig. 4**, on the transportation modes of foot, bus and motorbike. This consideration puts forward further enhancing potentialities in relation to investigations of the interplay between different learning models.

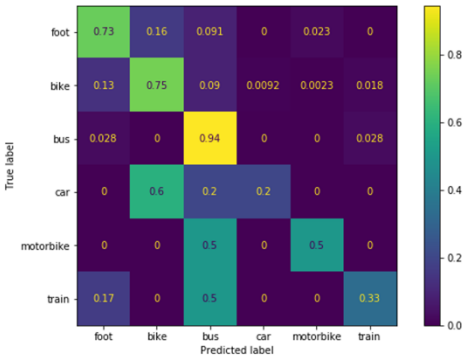

Fig. 5. Confusion matrix for MLP-1, computed on the test set.

5 Conclusions and Perspectives

In this paper we have presented a preliminary experimental analysis of the application of Machine Learning methodologies to the problem of estimating human transportation mode from smartphone sensors. The achieved results were significant. In view of the peculiarities of the data, the overall external test accuracy over 90% represents a positive aspect. The performance scales with the number of samples in the classes, independently of the learning models used. This indicates as a possible line of broadening of the study, the continuation of a data collection, with a focus on the classes that were less sampled so far (i.e., non-cycling vehicles).

Finally, the preliminary research presented in this paper opens the way to further studies also from a Machine Learning perspective. In this regard, an interesting direction consists in conducting a more in-depth cross-validation of the learning models. Another relevant line would be to extend the analysis to learning models for time series, e.g. Recurrent Neural Networks [9] (or hybrid neural architectures), enabling to naturally taking into account the temporal nature of the mobility data involved in the predictions. The final aim is that of contributing to the creation of an advanced and reliable human monitoring system for sustainable mobility. Moreover, monitored data, in the future, will be geoprocessed with data coming from other sources (health or metereological-data, land use data [10], data coming from ITS located in the city [11] or data regarding urban and building/activities field) so to extract further important knowledge elements useful for decision support system.

References

1. Petri, M., Frosolini, M., Lupi, M. and Pratelli, A., 2016, June. ITS to change behaviour: a focus about bike mobility monitoring and incentive—The SaveMyBike system. In 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC) (pp. 1-6). IEEE.
2. Pratelli, A., Petri, M., Farina, A. and Lupi, M., 2017, September. Improving bicycle mobility in urban areas through ITS technologies: the SaveMyBike project. In Scientific And Technical Conference Transport Systems Theory And Practice (pp. 219-227). Springer, Cham.
3. Petri, M. and Pratelli, A., 2019, July. SaveMyBike–A Complete Platform to Promote Sustainable Mobility. In International Conference on Computational Science and Its Applications (pp. 177-190). Springer, Cham.
4. Pratelli, A., Petri, M., Farina, A., and Souleyrette, R.R., 2020. Improving Sustainable Mobility througModal Rewarding: the GOOD_GO Smart Platform, in Proceeding of the 6th InternationaConference on Mechanical and Transportation Engineering (ICMTE 2020) – In publication.
5. Haykin, S., 2009. Neural networks and learning machines, 3rd edn, Prentice Hall.
6. Breiman, L., 2001. Random forests. Machine learning, 45(1), pp.5-32.
7. Cover, T. and Hart, P., 1967. Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), pp.21-27.
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), pp.2825-2830.
9. Kolen, J.F. and Kremer, S.C. eds., 2001. A field guide to dynamical recurrent networks. John Wiley & Sons.
10. Petri, M., Pratelli, A., Barè, G. and Piccini, L., 2019, July. A Land Use and Transport Interaction Model for the Greater Florence Metropolitan Area. In International Conference on Computational Science and Its Applications (pp. 231-246). Springer, Cham.
11. Pratelli, A., Petri, M., Ierpi, M. and Di Matteo, M., 2018, June. Integration of Bluetooth, vehicle count data and trasport model results by means of Datamining techniques. In 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe) (pp. 1-6). IEEE.