sciendo

Transport and Telecommunication, 2022, volume 23, no. 4, 273–283
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2022-0022

SMARTPHONE-BASED RECOGNITION OF ACCESS TRIP PHASE TO PUBLIC TRANSPORT STOPS VIA MACHINE LEARNING MODELS

Seyed Hassan Hosseini[1], Guido Gentile[2]

[1] Università di Roma La Sapienza
Via Eudossiana 18 - 00184 Roma, Italy
seyedhassan.hosseini@uniroma1.it
[2] Università di Roma La Sapienza
Via Eudossiana 18 - 00184 Roma, Italy
guido.gentile@uniroma1.it

The usage of mobile phones is nowadays reaching full penetration rate in most countries. Smartphones are a valuable source for urban planners to understand and investigate passengers' behavior and recognize travel patterns more precisely. Different investigations tried to automatically extract transit mode from sensors embedded in the phones such as GPS, accelerometer, and gyroscope. This allows to reduce the resources used in travel diary surveys, which are time-consuming and costly. However, figuring out which mode of transportation individuals use is still challenging. The main limitations include GPS, and mobile sensor data collection, and data labeling errors. First, this paper aims at solving a transport mode classification problem including (still, walking, car, bus, and metro) and then as a first investigation, presents a new algorithm to compute waiting time and access time to public transport stops based on a random forest model. Several public transport trips with different users were saved in Rome to test our access trip phase recognition algorithm. We also used Convolutional Neural Network as a deep learning algorithm to automatically extract features from one sensor (linear accelerometer), obtaining a model that performs well in predicting five modes of transport with the highest accuracy of 0.81%.

Keywords: transport mode detection, machine learning, trip phase recognition, urban trips on public transport

1. Introduction

In urban studies, understanding daily travel patterns and transport modes can help transportation planners to build a more efficient transportation system. Data collection is a primary step. There are multiple techniques to collect trip information, such as questionnaires, interviews (Tennøy *et al.,* 2022) and GPS-based surveys (Axhausen *et al.,* 2003). However, participants may report inaccurate data (Chia *et al.,* 2016).

Researchers should consider smartphone batteries when using mobile sensors for data collection, so sensor selection and sampling frequency are among the main factors for battery usage. The greater sampling frequency leads to higher energy usage (Khan *et al.,* 2016) .In this paper, a two-step algorithm tries to recognize access trip phase at public transport stops via machine learning. The first stage determines transit mode of users in an urban context, and the second step includes access trip phase recognition.

In the following, Section two includes a review of similar studies. Section three presents two different datasets, data preprocessing (cleaning and windowing), feature extraction, and training and testing stages. Evaluation metrices and a two-step access trip phase recognition algorithm, are described in section four. Section five concludes the paper.

2. Theoretical Background

This section discusses various studies applying machine and deep learning models in transport mode recognition.

2.1. Transport Mode detection

This section discusses various studies applying machine and deep learning models in transport mode recognition.

2.1.1. Machine Learning Models

Machine learning techniques usually follow two key steps to predict mode of transportation. Frist, creation hand-crafted features and select the best set of features. Second, train and test models with machine learning algorithms. Accelerometer sensor data was used in one of the earliest motion categorization studies (Devaul and Dunn, 2001) to distinguish between sitting, walking, biking, and riding with a multi-component Gaussian mixture model. Apart from motion sensors in mobile phones, GSM data in one of the first studies on recognizing commuter modes applied and they reported that GSM signals are inadequate for distinguishing modes of transportation (Muller, 2006).

In the following years, advances in GPS had a positive influence on not only monitoring individual trajectories but also identifying their mode of travel. However, poor reception of location data and high-power consumption make real-world application with some difficulties. As a GPS based research (Zheng *et al.*, 2008) classification model arrived to 72.8 percent accuracy to detect four different modes (cycling, driving, bus, and walking).

Only a few public transport mode detection datasets are currently available, and one of the public datasets (Carpineti *et al.*, 2018) tried to distinguish between five various modes, with highest accuracy of 96 percent. Additionally, in another research using just GPS (Quintella *et al.*, 2016) data for six-months, decision tree had the best performance among other classifiers. In another paper (Manzoni *et al.*, 2010) accelerometer sensor used with 25Hz frequency and windowing into 10-24 second timesteps with 50% overlap (increasing overlap can increase the output accuracy) and resulting in an accuracy of 82.14 percent for eight different modes.

In this investigation (Wang *et al.*, 2010) accelerometer saved for six modes of transport with an 8-second time window without overlapping between windows. Decision three gained the best results. Use an approach similar to previous research, they (Nham *et al.*, 2008) arrived to 93.88 percent accuracy for classifying four different modes at a frequency of 50Hz and five-second windows with a 50% overlap. Adding more sensors can improve the performance of the machine learning models. Investigation (Lorintiu and Vassilev, 2016) tried to recognize seven modes of transit by using three key sensors (GPS, Accelerometer and Magnetometer). Results show that classification performance reaching up to 96 percent when GPS data is included and 94 percent with an accelerometer and magnetometer.

In another investigation, the authors (Efthymiou *et al.*, 2019) addressed a classification problem including accelerometer, gyroscope, orientation, and GNSS data, and random forest and gradient boosting were the main machine learning techniques. The biggest problem with this research is that just a few transport modes were considered. Random forest model arrived at the best results.

One of the comprehensive experiments (Ferreira *et al.*, 2020) tried to recognize different modes including walking, cycling, driving, bus, and train. Frequency to save Accelerometer and GPS data are 1 Hz and 10 seconds respectively. The results show that walking is the most easily identified form of transportation, and the most difficult part belongs to detection car and bus. They reported that random forest has better performance than decision tree.

2.1.2. Deep Learning Models

Most investigations in transport mode detection area have recommended mode detection models based on hand-crafted features directly from raw data, but time consuming and human errors are main drawbacks of this approach.

In this study (Dabiri and Heaslip, 2018) GPS data after a preprocessing step used to predict mode of transport. Convolutional Neural Network as a deep learning model arrived at the highest accuracy of the test data (84.8 percent). They reported that CNNs outperform classic machine learning algorithms. In another paper (Liang and Wang, 2017) using just an accelerometer applied a Convolutional Neural Network to build a deep learning model (CNN). In this study, data was recorded at a frequency of 50Hz (every 20 milliseconds), and seven modes of transportation, distinguished with an accuracy of 94.48 percent, which was the highest among similar studies. To minimize the impact of rotation of the phones, magnitude of three axis was computed.

Recurrent Neural Networks as one of the popular time series prediction algorithms in deep learning models was developed (Jiang *et al.*, 2017) and achieved classification accuracy over 98 and 97 percent when detecting four and seven modes of transportation. In another study (Asci and Guvensan, 2019) using HTC dataset, a recurrent neural network (RNN) applied with new input set data to distinguish 10 different modes of transportation. They employed three different types of sensors (accelerometer, magnetometer and, gyroscope) with a 96.82 percent accuracy. Similar to previous study, a deep Bi-LSTM

(bidirectional long short-term memory) neural network architecture (Zhao *et al.*, 2019)tried to identify transportation modes. Accelerometer and gyroscope sensors played the main role to arrive 92.8 percent of accuracy to classify six types of transport. Final model trained width sliding window of 2.56 seconds with 50 percent overlap.

To compare machine and deep learning performance, in paper (Priscoli *et al.*, 2020) data was recorded for seven modes of transportation. The sensor data was collected by 18 volunteers at a sample rate of 50 Hz and total of 140 hours of data. Random Forest had the best performance among machine learning models, with an accuracy of 81.4 percent, while DeepCNN arrived at an accuracy of 98.6 percent.

2.2. Trip Phase Recognition (public transport access walking and waiting time)

There are several methods to calculate walking time to public transportation stops and waiting time inside bus and metro stations. The commonly used techniques include self-reported walking (El-Geneidy *et al.*, 2014; He *et al.*, 2018) by users about their actual walking distance or time from home or work office to public transport stations.

Traditional travel studies are unable to capture the real walking or biking distance to reach public transportation stops. Access time is often determined by self-reported interviews. To reduce the role of humans in data recording part of investigations, GPS sensors are used to calculate the walking distance from home to public transportation stations. In study (Tennøy *et al.*, 2022) the purpose is to calculate the duration and distance of walking journeys to public transportation stops in four Norwegian cities. Participants responded to questions regarding the mode of transportation they used from home and workplace side of the public transport trip, as well as the travel time and distance of trips. Participants at the public transport stops reported the distance and duration of their walking journeys to public transportation stations. Distances were also calculated from self-reported duration using an average walking velocity of 80 meters per minute to provide a deeper understanding of the variances between self-reported distance and computed distance.

In another study (Zuo *et al.*, 2018) data from GPS-based Household Journey Survey (HTS) was used to calculate the travel distance and duration to reach public transportation stations. They used GPS data to calculate walking and cycling distances to public transportation stations. In one of a few studies (Voss *et al.*, 2015) where GPS and Accelerometer were selected to extract trip characteristics, a total of 100 trips were taken by 42 participants. They conclude that combining GPS and accelerometers improves our knowledge of travel behaviors in terms of physical activity.

In paper (Nygaard, 2016) the waiting time of 1145 passengers at 24 unique bus stops were recorded. They also recorded the precise time each commuter arrived at the bus stop and the exact time each bus departed. In this study, to compute waiting time they defined that the waiting time for each traveler is calculated by subtracting their arrival time to bus stop from the departure time of bus from that station.

To the best of our knowledge, in all aforementioned studies, waiting time and access time to public transport stops rely on self-reported data of user or some computational methods, while this study is the first investigation that tries to compute these indicators via machine learning models.

3. Methodology

In this part, we discuss about 1) Bologna and Sapienza datasets 2) Preprocessing and feature extraction procedures 3) Classification algorithms (machine and deep learning 4) Access trip phase recognition algorithm.

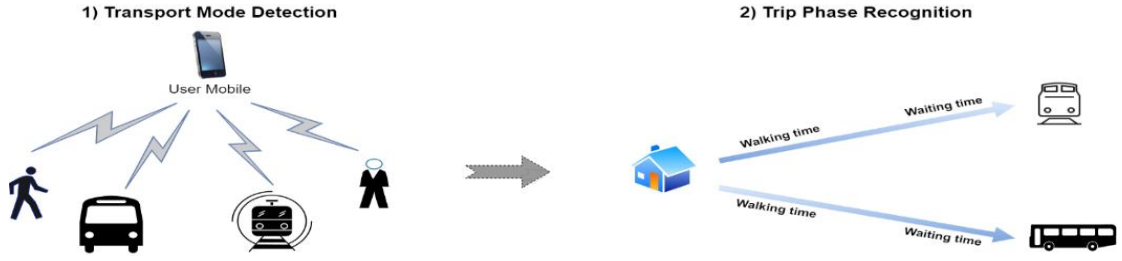

Figure 1. Two-step access trip phase recognition algorithm

3.1. Bologna and Sapienza Dataset

Sensor data was recorded in bologna dataset (Carpineti *et al.*, 2018) into five transport modes including walking, vehicle, stationary, train, and bus with 20 Hz frequency .Totally 31 hours of data to train classification models. For the Sapienza dataset, three users saved five hours of data in Rome, recording linear accelerometer and magnetometer with 1 Hz frequency and GPS points saved each 10 second which is similar to this study (Ferreira *et al.*, 2020) including walking, standing, metro and bus. Mobile phones were not placed in a specific position when recording data. Furthermore, the main reason to save magnetometer is that the results of final model can improve with this sensor when try to detect metro segments.

3.2. Mobile Sensors, Preprocessing, Windowing and Feature Extraction

Mobile phone position can change during recording data, and to solve this problem, most of the research (Reddy *et al.*, 2010; Wang *et al.*, 2010) suggests computing the magnitude of three axes (x, y, z) for each record of data.

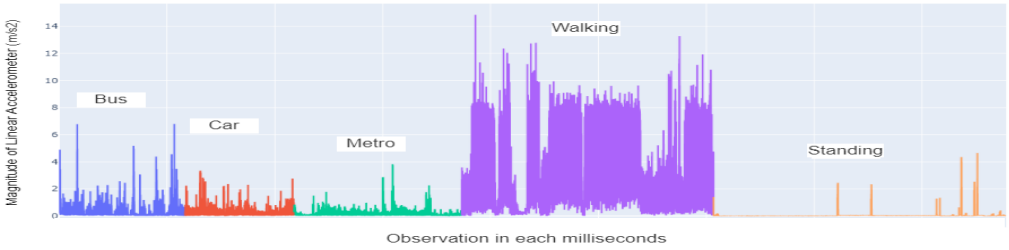

Figure 2. Bologna dataset after filtering

There are two major steps for the data preprocessing including filtering and windowing. The purpose is to reduce noise caused by the user's abrupt movements. The median filter (Erkan *et al.*, 2018) as a non-linear noise filtering process of eliminating noise from images and signals was used to have more smooth data. After filtering, data divided into five second time window. Therefore, mean, standard deviation, minimum and maximum of each window as a training feature were computed.

3.3. Machine Learning Classification

Support vector machines (SVM), K-nearest neighbor, and Random Forests algorithms, as well-known and often used in current literature were selected for the training, testing, and prediction phase. Random Forests (Ho, 1995) are ensemble methods that create a group of "weak learners" to form a "strong learner", able to solve a complex problem. In random forests, the weak learners are decision trees (Breiman *et al.*, 1984). Moreover, support vector machines (Andrew, 2001) are supervised learning models and, due to their ability to deal with nonlinear classification problems are widely used classification algorithms. Finally, K-nearest neighbors (KNN) stores all available samples and classifies new samples based on a similarity measure (e.g., distance functions) (Mucherino *et al.*, 2009).

3.4. Convolutional Neural Network Architecture

3.4.1. Input Layer

The input data contains magnitude of linear acceleration for five seconds (fixed time window). A matrix with 100 height (20 records in each second multiply by five seconds) and a width of one (one sampling value of linear accelerometer). In our CNN model, the input shape for five second time window is 100x1.

3.4.2. Convolutional Layer

Convolutional layer defines a filter (also known as a feature detector) with a height of three (also called kernel size). With the defined kernel size and considering the length of the input matrix, each filter will contain weights. The result from the first CNN will be fed into the second CNN layer. We will again define different filters to be trained on this level. Convolution filters with equal size (1×3×C) selected for all convolutional layers and C indicates the number of channels and in our case is one (one dimensional convolutional network).

3.4.3. Activation layer

In our model, activation function is Rectified Linear Units (ReLU) which substitutes all negative values in feature map with a value of zero. With respect to other activation functions such as tanh(x) and the sigmoid function, the CNN with ReLU has a substantially quicker learning rate (Lin and Shen, 2018).

3.4.4. Batch Normalization

Batch normalization layer can be used to normalize inputs before or after activation layers (Ioffe and Szegedy, 2015) and it can greatly accelerate the training process of a neural network and, in certain situations, enhance model performance via a minor regularization impact.

3.4.5. Pooling Layer

The goal of pooling layer is to reduced computation, and control overfitting by reducing the dimensionality of each feature map (Scherer *et al.,* 2010). Global Average Pooling is a function that computes the average output of each feature map in the preceding layer. There are no trainable parameters in this layer.

3.4.6. Dropout

Dropout is the most feasible and extensively utilized method for dealing with the overfitting problem in CNNs. Dropout is the process of randomly selecting certain weights to zero in order to boost the network resiliency (Srivastava *et al.,* 2014).

3.4.7. Fully Connected Layer with SoftMax Activation

Several Fully Connected (FC) layers can be used in CNN architecture. In our model, we used 200 and 100 neurons to link the two final layers to the prior layer. Finally, the data are transferred into a 5×1 output vector with full connection where the SoftMax activation function is used to build a probability distribution across the transportation labels.

3.4.8. CNN Configurations

To find the best, different configurations has been tested and results reported to find the optimal one for our purpose. We begin with fewer filters and layers, then add more layers and filters inside each layer (See Table 1).

3.4.9. Training Process

To compute the error at final layer (prediction step), we selected categorical cross-entropy as the loss function. In the back propagation phase, Adam optimizer used to update model parameters with 5 iterations to train our model, and over each iteration our learning rate is divided by 10. We used a batch size of 64 and an initial learning rate of 0.01. The accuracy on test data is calculated after each epoch of training.

Table 1. Multiple CNN configurations per column

	A	**B**	**C**	**D**	**E**
Input Layer	The shape of input layer is (1×100×1) linear accelerometer segments where height and width are 100 and 1 respectively				
Convolutional	32	64	128	256	256
Batch Normalization	yes	yes	yes	yes	yes
Convolutional	64	128	256	512	512
Batch Normalization	yes	yes	yes	yes	yes
Convolutional	64	64	128	256	256
Batch Normalization	yes	yes	yes	yes	yes
Global Average Polling	yes	yes	yes	yes	yes
FC	No	200	200	100	200
Dropout	No	0.2	0.2	0.2	No
FC	No	100	100	50	100
Dropout	No	0.2	0.2	0.2	No
FC	5	5	5	5	5
Accuracy	79.00	79.50	80.30	80.45	81.00

3.5. Trip Phase Recognition

Trip phase algorithm is the last part of this study, it consists of three major phases:

1) Training a random forest model and prediction transit modes
2) Post-preprocessing on transport mode detection results
3) Counting the number of windows in each phase, multiply by window size to define the time spent on each phase of the trip.

We trained a random forest model with Sapienza dataset, using linear Accelerometer and Magnetometer (1 Hz frequency), and extract features (mean and standard deviation) of each one-minute segments. Data was labeled with (walking, standing, and in-vehicle). It means that we labeled bus and metro as in-vehicle, the main goal of access trip phase algorithm is to categorize public transport base trips into walking, waiting time, and in vehicle.

After preprocessing (filtering and windowing), all segments are fed into a pretrained machine learning model, and the output is transport mode detection results. The prediction algorithm in some cases predicts the transport mode incorrectly since it is not perfect. To improve the final mode, understanding the logic of trip phases can reduce the errors of machine learning prediction.

As part of the post-processing, an algorithm is defined to filter the final forecast. A window with length three (3 minutes) start searching and find predictions that are not similar to one before and after their index, i.e., a trajectory is predicted as walking, walking, standing, walking, walking.

Post-preprocessing algorithm change standing in this trajectory to its previous window that here is walking. In another case when there are two standing, before and after walking, the walking can convert to standing. In these cases, a sudden movement can result saving linear accelerometer as walking. Final predictions will not change where the prediction results are walking, walking, standing, in vehicle.

To recognize two main parts of access trip to public transport stations, we define walking and waiting time as following:

(1) Walking time to public transit stop is the time from origin (home, or office) before waiting time at public transport stop.
(2) Waiting time is the time after finishing walking time to public transport stop and before boarding a bus or metro.

The last part of the algorithm works as following:

(1) The start time of in-vehicle prediction will be fined. The condition is when algorithm detects three in-vehicle prediction segments respectively.
(2) It will save all windows from start time a trip before starting in-vehicle prediction. The number of all walking prediction multiply with window length (one minutes) is defined as access time to public transport stops.
(3) The number of all standing prediction after walking and before boarding vehicle multiply by window length is waiting time at public transport station.

The main reason to not saving GPS points is that individuals are concerned about their privacy about saving GPS data. The final goal is to deploy our approach in a real-world situation where saving mobile phone battery and privacy is essential for consumers.

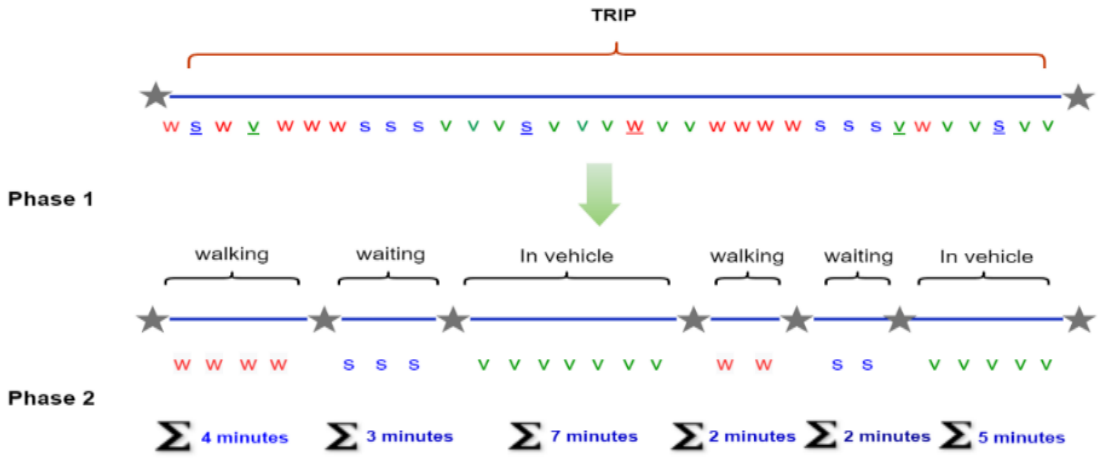

Figure 3. Post-Processing algorithm for the access trip phase recognition

4. Results and Discussion

We will discuss the results of machine and deep learning, as well as testing access trip phase recognition algorithm with new provided trajectories.

4.1. Machine Learning

First, we evaluate the performance of our machine learning models. To compare our results with investigation (Carpineti *et al.*, 2018), that prepared bologna dataset, the same time window (5 second) and features (mean, standard deviation, max and min) were used similar to bologna investigation in both our machine and deep learning part. Dataset divided with 70 percent of data for training models and 30 percent for the test part.

Each ML algorithm includes unique parameters (hyper-parameters) that are relevant to internally operation and to choose the best hyper-parameter combination, we tested many combinations and selected number of estimators equal to 160 and bootstrap fixed as true. For the machine learning model evaluation, similar to (Carpineti *et al.*, 2018), the accuracy on the test data is reported while deep learning models are evaluated with accuracy on the test data and confusion matrix, more details about the metrics can be found (Sokolova and Lapalme, 2009).

Table 2. Results obtained from random forest algorithm selecting different sensors

Sensor	Accuracy	
	Bologna results	Our proposed random forest model
Accelerometer	0.74	0.78
Linear Accelerometer	0.74	0.77
Gyroscope	0.68	0.74
Orientation	0.61	0.63

We applied the same machine learning algorithm (random forest) to compare our findings more effectively with Bologna results. Accelerometer sensor and linear accelerometer both reported improvements of 4 and 3 percent, respectively, while orientation accuracy improved up to 2 percent (See Table 2).

4.2. Deep Learning

CNN architecture has been developed and implemented in Keras, a Python deep learning open-source library. Results of CNN model on the test data using just linear accelerometer arrived at 81.48%. Precision and recall indicators are the performance indicators used for deep learning results.

Table 3. Performance of different algorithms on linear accelerometer data

Model	Accuracy
CNN	0.81
random forest	0.77
k nearest neighbor	0.73
support vector machine	0.62

Table 4. Performance of CNN model using only Linear Acceleration, 5 seconds time window

Transport Mode	Precision	Recall
Bus	0.62	0.33
Car	0.78	0.95
Standing	0.97	0.98
Train	0.65	0.81
Walking	0.94	0.91
Overall	0.79	0.81

Table 4 shows that deep learning layers can extract meaningful features from non-motorized modes (walking and standing) and predict segments with more than 0.94 percent accuracy. Train and bus had the lowest accuracy, and our model confused so we can safely claim that more features and sensors should apply to improve the accuracy of our model like magnetometer that is more sensitive in train stations.

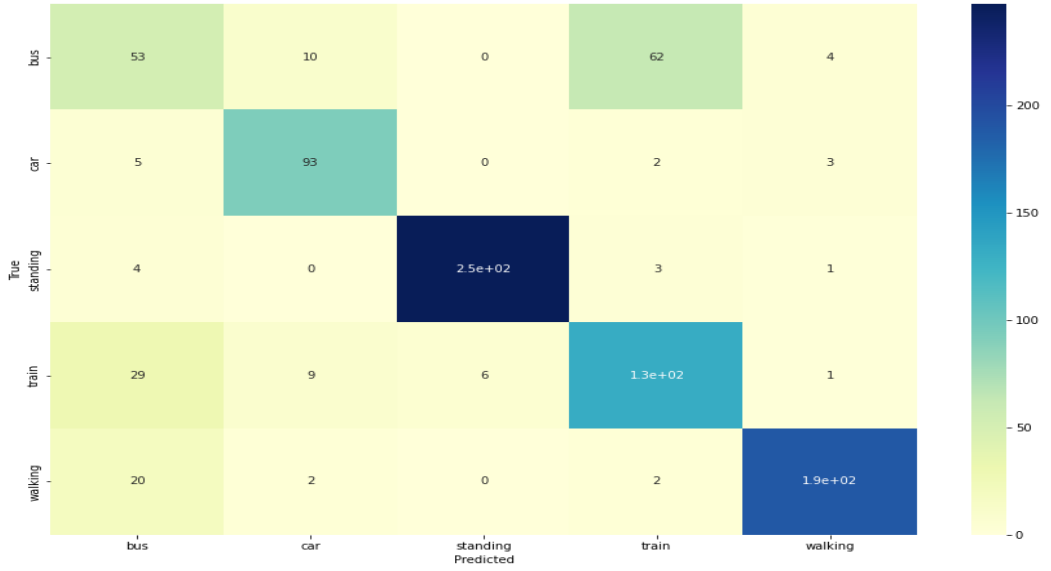

Figure 4. Confusion matrix for CNN results, 5 second time window, linear acceleration

The result of CNN model reported in a confusion matrix, the model predicts 247 standing segments and 190 walking segments correctly while 20 segments of walking predicted as in a bus, the reason can be some movements in a bus or errors of the model. Challenging part similar to machine learning model belongs to detection motorized transit modes.

4.3. Trip Phase Results

The purpose of this section is to identify the main components of any public transportation base journey, including walking time to PT stations and waiting time before boarding a bus or metro via pretrained machine learning model. We asked from participants to record public transport-based trips in Rome (using bus or metro in their trips and not using car) and fed trajectories into our algorithms to evaluate its effectiveness.

Table 5. Prediction Walking time and Waiting time at bus and metro station

Trajectory Number	Self-report by participants	Predicted walking time	Predicted waiting time
1	walking-standing-metro	6 min	2 min
2	walking-standing-bus	5 min	0 min
3	walking -standing-metro-walking-standing-bus	12 min	3 min
4	walking-metro-walking-bus	3 min	0 min
5	walking-standing-bus	5 min	8 min
6	walking-standing-metro	4 min	5 min

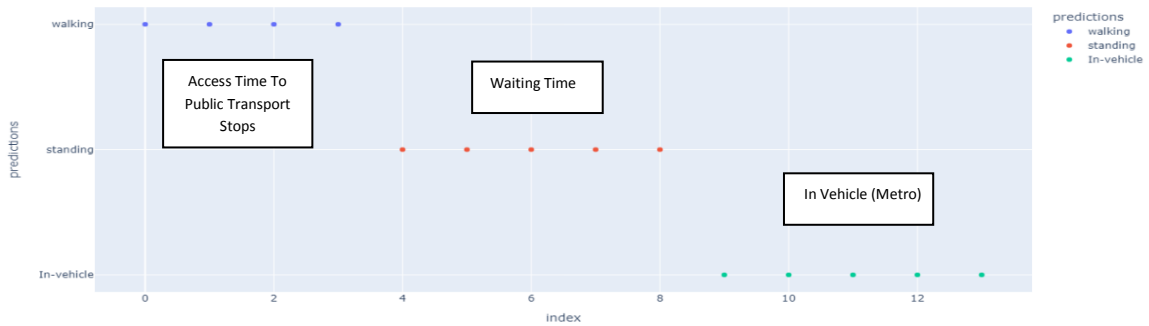

Figure 5. Results from prediction Random Forest model

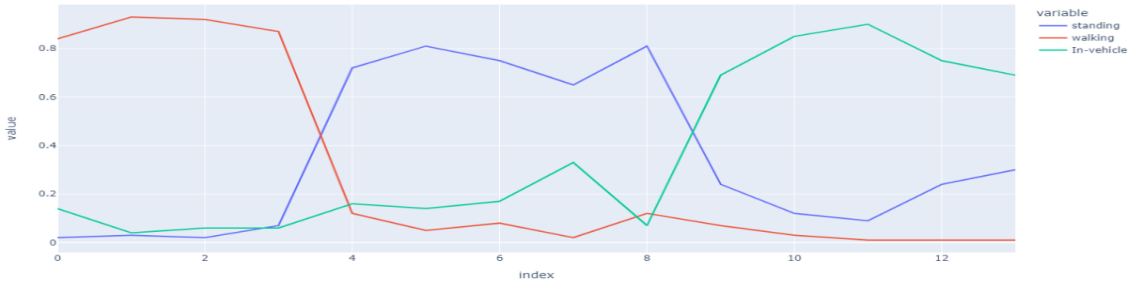

Figure. 6. Predicted probability of each one minutes

Table 5, Figure 5, and Figure 6 shows algorithm results tested on some new trajectories saved by participants in Rome. Participants did not provide details about their journey stages. They described their activities as walking, waiting, and using metro or bus. The algorithm provides additional information about trajectories and reporting trip phases, as well as the time spent on each of them. There are some algorithm errors, i.e., in trip number 2, there is no predicted waiting time and participant claims they waited for a bus. From the algorithms performance when waiting time is less than 1 minute it considers waiting as in-vehicle when the mode of transport is changing. In total, walking prediction from random forest model was more than 95 percent, it means that most of the computed walking time were predicted correctly.

5. Conclusions

The first aim of this paper is to recognize transit modes using bologna dataset that contains data from five different modes. Our results show that, given the same set of features and time windows similar to bologna investigation, our suggested machine learning models outperform their results. Moreover, we applied a CNN deep learning model using just Linear Accelerometer. However, they did not use deep learning models. The results first have shown the effectiveness of the suggested deep learning model, which achieved approximately 81% classification accuracy and find that random forest model among machine learning models had the best performance for the classification task. Second, a random forest machine learning model trained with Sapienza dataset for access trip phase recognition algorithm to recognize trip phases including access time and waiting time at train and bus stations. As a first investigation to recognize access trip phases to PT stops with machine learning models, instead of using self-reported phases by users or applying computational GPS based methods, our algorithm predicts duration of walking and waiting to public transportation stations with high precision.

References

1. Andrew, Alex M. (2001) An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. *Kybernetes*.
2. Asci, Guven, and Guvensan, M. Amac. (2019) A Novel Input Set for LSTM-Based Transport Mode Detection. In: *2019 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2019*, 107–12. DOI: 10.1109/PERCOMW.2019.8730799.
3. Axhausen, Kay W., Schönfelder, Stefan, Wolf, Jean, Oliveira, Marcelo and Samaga, Ute. (2003) 80 Weeks of GPS-Traces: Approaches to Enriching the Trip Information. *Arbeitsberichte Verkehrs- Und Raumplanung*, 178.
4. Breiman, L., Friedman, J., Olshen, R. and C. Stone. (1984) *Classification and Regression Trees–Crc Press*. Boca Raton, Florida.
5. Carpineti, C., Lomonaco, V., Bedogni, L., di Felice, M. and Bononi, L. (2018) Custom Dual Transportation Mode Detection by Smartphone Devices Exploiting Sensor Diversity. In: *2018 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2018*, 367–72. DOI: 10.1109/PERCOMW.2018.8480119.
6. Chia, Jason, Lee, Jinwoo and Kamruzzaman, M. D. (2016) Walking to Public Transit: Exploring Variations by Socioeconomic Status, 10(9), 805–14. DOI: 10.1080/15568318.2016.1156792.

7. Dabiri, Sina, and Heaslip, Kevin. (2018) Inferring Transportation Modes from GPS Trajectories Using a Convolutional Neural Network. *Transportation Research Part C: Emerging Technologies*, 86, 360–371. DOI: 10.1016/J.TRC.2017.11.021.
8. Devaul, Richard W., and Dunn, Steve. (2001) *Real-Time Motion Classification for Wearable Computing Applications*.
9. Efthymiou, Alexandros, Barmpounakis, Emmanouil N., Efthymiou, Dimitrios and Vlahogianni, Eleni I. (2019) Transportation Mode Detection from Low-Power Smartphone Sensors Using Tree-Based Ensembles. *Journal of Big Data Analytics in Transportation 2019*, 1(1), 57–69. DOI: 10.1007/S42421-019-00004-W.
10. El-Geneidy, Ahmed, Grimsrud, Michael, Wasfi, Rania, Tétreault, Paul and Surprenant-Legault, Julien. (2014) New Evidence on Walking Distances to Transit Stops: Identifying Redundancies and Gaps Using Variable Service Areas. *Transportation*, 41(1), 193–210. DOI: 10.1007/S11116-013-9508-Z/FIGURES/4.
11. Erkan, Uğur, Gökrem, Levent and Enginoğlu, Serdar. (2018) Different Applied Median Filter in Salt and Pepper Noise. *Computers & Electrical Engineering*, 70, 789–798. DOI: 10.1016/J.COMPELECENG.2018.01.019.
12. Ferreira, Paulo, Zavgorodnii, Constantin and Veiga, Luís. (2020) EdgeTrans - Edge Transport Mode Detection. *Pervasive and Mobile Computing*, 69, 101268. DOI: 10.1016/J.PMCJ.2020.101268.
13. He, Jinliao, Zhang, Ruozhu, Huang, Xianjin and Xi, Guangliang. (2018) Walking Access Distance of Metro Passengers and Relationship with Demographic Characteristics: A Case Study of Nanjing Metro. *Chinese Geographical Science 2018*, 28(4), 612–623. DOI: 10.1007/S11769-018-0970-6.
14. Ho, Tin Kam. (1995) Random Decision Forests. In: *Proceedings of 3rd international conference on document analysis and recognition*, 1, 278–282.
15. Ioffe, Sergey, and Szegedy, Christian. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: *International conference on machine learning*, 448–56. PMLR.
16. Jiang, Xiang, de Souza, Erico N., Pesaranghader, Ahmad, Hu, Baifan, Silver, Daniel L. and Matwin, Stan. (2017) TrajectoryNet: An Embedded GPS Trajectory Representation for Point-Based Classification Using Recurrent Neural Networks'. *Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering, CASCON 2017*, 9, 192–200. DOI: 10.48550/arxiv.1705.02636.
17. Khan, Inayat, Khusro, Shah, Ali, Shaukat and Ahmad, Jamil. (2016) Sensors Are Power Hungry: An Investigation of Smartphone Sensors Impact on Battery Power from Lifelogging Perspective'. *Bahria University Journal of Information & Communication Technologies (BUJICT)*, 9(2).
18. Liang, Xiaoyuan, and Wang, Guiling. (2017) A Convolutional Neural Network for Transportation Mode Detection Based on Smartphone Platform. *Proceedings – 14th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2017*, 338–342. DOI: 10.1109/MASS.2017.81.
19. Lin, Guifang, and Shen, Wei. (2018) Research on Convolutional Neural Network Based on Improved Relu Piecewise Activation Function. *Procedia Computer Science*, 131, 977–984. DOI: 10.1016/J.PROCS.2018.04.239.
20. Lorintiu, Oana, and Vassilev, Andrea. (2016) Transportation Mode Recognition Based on Smartphone Embedded Sensors for Carbon Footprint Estimation. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 0, 1976–1981. DOI: 10.1109/ITSC.2016.7795875.
21. Manzoni, Vincenzo, Maniloff, Diego, Kloeckl, Kristian and Ratti, Carlo. (2010) Transportation Mode Identification and Real-Time CO2 Emission Estimation Using Smartphones. *SENSEable City Lab, Massachusetts Institute of Technology, Nd*.
22. Mucherino, Antonio, Papajorgji, Petraq J. and Pardalos, Panos M. (2009) K-Nearest Neighbor Classification. *Data mining in agriculture*, 83–106. Springer.
23. Muller, Ian Anderson Henk. (2006) Practical Activity Recognition Using GSM Data. In: *Proceedings of the 5th International Semantic Web Conference (ISWC). Athens*, 1. Citeseer.
24. Nham, Ben, Siangliulue, Kanya and Yeung, Serena. (2008) Predicting Mode of Transport from Iphone Accelerometer Data. *Machine Learning Final Projects, Stanford University*.
25. Nygaard, Magnus Frestad. (2016) *Waiting Time Strategy for Public Transport Passengers*. 61.
26. Priscoli, Francesco Delli, Giuseppi, Alessandro and Lisi, Federico. (2020) Automatic Transportation Mode Recognition on Smartphone Data Based on Deep Neural Networks. *Sensors 2020*, 20(24), 7228. DOI: 10.3390/S20247228.
27. Quintella, Carlos Alvaro De M. S., Andrade, Leila C. V. and Campos, Carlos Alberto v. (2016) Detecting the Transportation Mode for Context-Aware Systems Using Smartphones. *IEEE*

Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2261–2266. DOI: 10.1109/ITSC.2016.7795921.

28. Reddy, Sasank, Mun, Min, Burke, Jeff, Estrin, Deborah, Hansen, Mark and Srivastava, Mani. (2010) Using Mobile Phones to Determine Transportation Modes. *ACM Transactions on Sensor Networks (TOSN)*, 6(2). DOI: 10.1145/1689239.1689243.
29. Scherer, Dominik, Müller, Andreas and Behnke, Sven. (2010) Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In: *International conference on artificial neural networks*, 92–101. Springer.
30. Sokolova, Marina, and Lapalme, Guy. (2009) A Systematic Analysis of Performance Measures for Classification Tasks. *Information Processing & Management*, 45(4), 427–37.
31. Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya and Salakhutdinov, Ruslan. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.
32. Tennøy, Aud, Knapskog, Marianne and Wolday, Fitwi. (2022) Walking Distances to Public Transport in Smaller and Larger Norwegian Cities. *Transportation Research Part D: Transport and Environment*, 103, 103169. DOI: 10.1016/J.TRD.2022.103169.
33. Voss, Christine, Winters, Meghan, Frazer, Amanda and McKay, Heather. (2015) School-Travel by Public Transit: Rethinking Active Transportation. *Preventive Medicine Reports*, 2, 65–70. DOI: 10.1016/J.PMEDR.2015.01.004.
34. Wang, Shuangquan, Chen, Canfeng and Ma, Jian. (2010) Accelerometer Based Transportation Mode Recognition on Mobile Phones. *APWCS 2010 - 2010 Asia-Pacific Conference on Wearable Computing Systems*, 44–46. DOI: 10.1109/APWCS.2010.18.
35. Zhao, Hong, Hou, Chunning, Alrobassy, Hala and Zeng, Xiangyan. (2019) Recognition of Transportation State by Smartphone Sensors Using Deep Bi-LSTM Neural Network. *Journal of Computer Networks and Communications*. doi: 10.1155/2019/4967261.
36. Zheng, Yu, Li, Quannan, Chen, Yukun, Xie, Xing, and Wei Ying Ma. (2008) Understanding Mobility Based on GPS Data. *UbiComp 2008 - Proceedings of the 10th International Conference on Ubiquitous Computing*, 312–321. DOI: 10.1145/1409635.1409677.
37. Zuo, Ting, Wei, Heng and Rohne, Andrew. (2018) Determining Transit Service Coverage by Non-Motorized Accessibility to Transit: Case Study of Applying GPS Data in Cincinnati Metropolitan Area. *Journal of Transport Geography*, 67, 1–11. DOI: 10.1016/J.JTRANGEO.2018.01.002.