

# On Improving Operational Planning and Control in Public Transportation Networks using Streaming Data: A Machine Learning Approach

Luis Moreira-Matias<sup>1,2</sup>,  
João Mendes-Moreira<sup>2,3</sup>, João Gama<sup>2,4</sup>, and Michel Ferreira<sup>1,5</sup>

<sup>1</sup> Instituto de Telecomunicações, 4200-465 Porto, Portugal

<sup>2</sup> LIAAD-INESC TEC, 4200-465 Porto, Portugal

<sup>3</sup> DEI-FEUP, U. Porto, 4200-465 Porto, Portugal

<sup>4</sup> Faculdade de Economia, U. Porto 4200-465 Porto, Portugal

<sup>5</sup> DCC-FCUP, U. Porto, 4169-007 Porto, Portugal

{luis.m.matias, joao.mendes.moreira, jgama}@inescporto.pt,  
michel@dcc.fc.up.pt

**Abstract.** Nowadays, transportation vehicles are equipped with intelligent sensors. Together, they form collaborative networks that broadcast real-time data about mobility patterns in urban areas. Online intelligent transportation systems for taxi dispatching, time-saving route finding or automatic vehicle location are already exploring such information in the taxi/buses transport industries. In this PhD spotlight paper, the authors present two ML applications focused on improving the operation of Public Transportation (PT) systems: 1) Bus Bunching (BB) Online Detection and 2) Taxi-Passenger Demand Prediction. By doing so, we intend to give a brief overview of the type of approaches applicable to these type of problems. Our frameworks are straightforward. By employing online learning frameworks we are able to use both historical and real-time data to update the inference models. The results are promising.

**Keywords:** GPS data, Public Transportation, Bus Bunching, Taxi-Passenger Demand, Probabilistic Reasoning, Online Learning.

## 1 Introduction

The increasing number of running road vehicles worldwide is enlarging the complexity of transportation networks - especially of its design and operations. Therefore, it is becoming more difficult to maintain the reliability of this means of transportation, thereby decreasing passenger satisfaction. On the other hand, rising fuel price are increasing its operational costs.

**GPS (Global Positioning System)** devices are already in place in many of these networks. There are also many **Intelligent Transportation Systems (ITS)** that already successfully explore this kind of data, such as Intelligent Routing [1], Bus Travel Time prediction [2] or efficient taxi dispatching [3].

Despite the intrinsic online characteristics regarding the aforementioned problems, many of the techniques employed are batch learners - which are not prepared to detect the concept drift often introduced by unexpected events, which emerge in the system, such as traffic jams or a massive demand.

In fact, GPS data is mainly an unbounded stream of data. This kind of data is produced continuously at a high speed from multiple locations and time granularities, while its distribution may change over time (e.g. a continuous but speedy vehicular flow on a highway is heavily decreased by a traffic jam). However, state-of-art Machine Learning (ML) algorithms have strong assumptions - such as stationary data distributions and the existence of finite training sets - that make their application to this kind of data inefficient or even useless [4]. On the other hand, techniques that **learn from data streams** have seen great development over the last decade and there has also been an increase in applications used in many sensor networks - such as our own - and which have already proved to be efficient in dealing with these characteristics.

In this PhD thesis, we intend to explore such techniques related to GPS data streams broadcasted by the vehicular networks comprised in Public Transportation (PT) networks, namely, 1) from buses and 2) from taxis. Our goal is to produce ITS applications that will increase the profitability of companies, by providing important information that otherwise would not be possible to mine. The above mentioned networks have common characteristics and synergies that should be explored together. They can be enumerated as follows:

1. Both provide a **continuous stream of data** about the network's behavior, based not only on **vehicle location**, but also on other status variables, such as the number of passengers travelling within or of a mechanical nature.
2. Both enclose **vehicular networks**, and their operations rely on the a) **dependences** and/or b) **correlations** between vehicle behaviors. Some examples of these could be a) the delay propagation effect of a highly frequented bus route, introduced by a vehicle that is failing to fulfill its schedule, or b) the expected distance of a taxi service, departing from a location of interest, given that the last  $N$  vehicles, which departed from such a spot, experienced a cruising distance greater (or smaller) than a given time threshold.
3. The **passenger demand** in PT services, provided by such networks, is highly dependent on the **regularities of human behavior**, such as the sleep period at night, or the difference between travel origins and destinations on weekdays/weekends.
4. The **planning** of these networks is highly dependent on **seasonal events exhibited during the year**, as in periods of school holidays or over the Christmas season. Important planning stages of these networks (e.g.: the location of taxi stands in a given urban area, or the planning of bus schedules) are relevant examples of the dependency in place.
5. The **real-time control** of both is highly sensitive to **anomalous demand events** that may unbalance the expected relationship between service offer and demand - and thereby provoke **unexpected disruptions** in such services. Examples of this issue could be overcrowded buses caused by large

scale events (e.g. sporting events, concerts, etc.), which may cause a temporary absence of taxi offers in some locations, due to an exponential increase in passenger demand.

The aforementioned characteristics represent similarities that are reflected in the data provided, namely, 1) by exhibiting the same periodicity of existing regularities (daily, weekly) and 2) common passenger origin/destination matrices; 3) by revealing the existence of anomalous demand events (thereby allowing their detection in both time and space) or 4) even common data distributions of the time series of passenger count. Consequently, such streaming data provides opportunities to improve both the operational planning and control of networks, by exploring methods to learn and therefore identifying these patterns.

In this spotlight paper, we present two ML frameworks for solving two real-world problems: 1) Bus Bunching (BB) Online Detection and the 2) Taxi- Passenger Demand Prediction. By doing so, we intend to give a brief overview of the type of approaches applicable to these type of problems.

## 2 Case Study

Our case study took place in the city of Porto in Portugal. Two data streams of events from two PT companies operating in Porto were used to evaluate our approaches. This city is the center of a medium-sized urban area, consisting of 1.3 million inhabitants.

To test our BB online detection framework, we used data collected from STCP, the Public Transport Operator in Porto. It describes trips of three distinct lines (A, B, C) during 2010. Each line has two routes - one for each way A1, A2, B1, B2, C1, C2. Line A is a common urban line between *Viso* (an important neighbourhood in Porto) and *Sá da Bandeira*, a downtown bus hub. Line B is also an urban line, although it is an arterial one. It traverses the main interest points in the city by connecting two important street markets: *Bolhão* - located in the downtown area - and *Mercado da Foz*, in the most luxurious neighbourhood in the city. Line C connects the city's downtown area to the farthest large-scale neighborhood in the region (*Maia*).

Concerning the second application case, we focused on the event data stream from a taxi company (which contains 441 running vehicles) operating in Porto, Portugal, between August 2011 and April 2012. This dataset contains information about more than one million fared trips.

## 3 Bus Bunching Online Detection

It is known that some schedule instability exists, especially in highly frequent routes (10 minutes or less). In these kinds of routes the **headway** (time separation between vehicle arrivals or departures) regularity is more important than the fulfilment of the arrival time at bus stops. In fact, a small delay in a bus'

arrival increases the number of passengers at the next stop. This number increases the dwell time (time period where the bus is stopped at a bus stop). On the other hand, the next bus will have fewer passengers and shorter dwell times without delays. This will continue as a snow ball effect and, at a further point of that route, the two buses will meet at a bus stop, forming a platoon. This phenomenon is denominated as **Bus Bunching(BB)**.

### 3.1 Related Work

One of the first works to address the BB phenomenon was presented by Powell and Sheffi [5]. After this paper, many other works followed the stability concept (i.e. if we guarantee a stable headway, BB events will never emerge) by constantly introducing corrective actions into the system. Some examples are the work in [6], where each bus is an agent that negotiates with the other buses about which bus is holding up time at each stop or in [7], where the negotiation is centered around the cruising speed.

The employment of historical data to address this problem is very recent. In [8], a model to determine the optimal holding time at each stop based on real-time location is presented. Delgado et al. [9] also suggested preventing passengers from boarding by establishing maximum holding times to maintain the headway stable. The efficiency of these types of frameworks is usually demonstrated through simulations, assuming 1) stochastic demand or 2) using historical data. Despite their usefulness, all these works do not account for the use of both historical and real-time data simultaneously. Moreover, they have low interpretability because their outputs do not provide any insight on what the best corrective action is. The predictive method presented in the next section is able to deal with the network's stochasticity, regardless of which corrective action we opt to take. Finally, it is important to highlight that the majority of the work described in the literature tries to maintain a stable headway at the cost of some schedule uncertainty (introduced by the constant corrective actions), despite the existing risk of forming a bus platoon at a further stop.

### 3.2 Methodology

The most important variable in regards to BB events is the headway (i.e.  $h$ ). Theoretically, the headway between two consecutive trips should be constant. However, due to stochastic events that arise during bus trips, the headway suffers some variability. BB does not only occur when a bus platoon is formed, but occurs as soon as the headway becomes unstable. The headway between two consecutive buses is defined as unstable whenever it is strictly necessary to apply a corrective action in order to recover the headway value to acceptable levels. Such a threshold is usually defined in function of the frequency  $f = h^{-1}$  (time between the departure of two consecutive buses) [10].

Let the arrival time be defined as  $T_{k,i+1} = T_{k,i} + dw_{k,i} + CTT_{k,i,i+1}$  where  $dw_{k,i}$  denotes the dwell time at the stop  $i$  and  $CTT_{k,i,i+1}$  stands for the Cruise Travel Time between those two consecutive stops. Therefore, it is possible to

anticipate the occurrence of BB events if we are able to predict the value of  $dw_{k,i} + CTT_{k,i,i+1}$ , which is often denominated by **Link Travel Time** (LTT) [11]. Let the LTT Prediction be defined as an offline **regression** problem where the target variable is the cruising time between two consecutive bus stops. Such predictions are computed on a daily basis (the forecasting horizon) using the  $\theta$  most recent days (the learning period) to train our model. Consequently, we obtain a set of predictions for all the  $t$  trips of the day denoted as  $\mathbb{P} = \bigcup_{i=1}^t P_i = \{P_{1,1}, P_{1,2}, \dots, P_{1,s}, \dots, P_{t,s}\}$ .  $\mathbb{P}$  is then incrementally refined in two steps: 1) trip-based and 2) stop-based. Both steps are based on the Perceptron's Delta Rule [12] by reusing each prediction's **residuals** to improve the further ones.

Let  $e$  denote the last trip completed before the current trip starts (i.e.  $c$ ). The trip-based refinement consists of comparing the predictions of  $e$   $P_e = \{P_{e,1}, P_{e,2}, \dots, P_{e,s}\}$  with the real times  $T_e$  to update  $P_c$ . Firstly, we compute the residuals as  $R_e = T_e - P_e$  and then its average value as  $\nu_e = \sum_{i=1}^s \frac{T_{e,i} - P_{e,i}}{s}$ . Secondly, an user-defined parameter  $0 < \alpha \ll 1$  is employed to set a threshold  $th$  able to identify trips where the error is larger than expected. Consequently,  $th = \alpha * f_e$ . Three other variables are then defined:  $\vartheta_p = 0, \vartheta_n = 0$  and  $\beta' = \beta$ . The first two are counters that are incremented whenever the prediction error is heading to the same way (positive/negative) in consecutive trips (e.g. if  $\mu_e > th$   $\vartheta_p$  is incremented; otherwise,  $\vartheta_p = 0$ ). The beta value  $\beta'$  stands for the residual's percentage to be added to  $P_c$  (its initial value is user-defined). It is initialized with another user-defined parameter  $0 < \beta \ll 1$  and updated according to a user-defined learning rate  $0 < \kappa \leq 1$ . Consequently, if  $\vartheta_p$  or  $\vartheta_n$  are incremented, the  $P_c$  and  $\beta'$  are updated as  $P'_c = P_c \pm (\beta' \times P_c)$  and  $\beta' = \beta' + \vartheta * (1 + \kappa) * \beta$ , respectively. If both  $\vartheta$  stay the same,  $\beta'$  resumes its original value as  $\beta' = \beta$ . The error tests are always performed over the regression results  $P_c$  and not over the updated arrays  $P'_c$ . These updates are performed incrementally for every trip.

Given the updated predictions of two consecutive trips ( $P'_c, P'_{c+1}$ ), it is possible to obtain the predicted headways  $E_c = P'_{c+1} - P'_c$  (i.e. an offline prediction). The second refinement uses the headway residuals  $HR_c = H_c - E_c$  to update  $E_c$  stop-by-stop. Incrementally, we can obtain online headway predictions as  $E'_{c,i} = H_{c,i-1} + E_{c,i} - E_{c,i-1}, \forall i \in \{2, s\}$ . The problem resides in updating the headway online prediction for the next stop  $E'_{c,i}$  given the value of  $HR_{c,i-1}$ . Let  $\gamma' = \gamma$  be the residual's percentage to add to the prediction where its initial value for each trip ( $0 < \gamma \ll 1$ ) is an user-defined parameter.  $E'_{c,i}$  can be updated as  $E''_{c,i} = E'_{c,i} + (HR_{c,i-1} * \gamma')$ . Finally,  $\gamma'$  is also updated by comparing the residuals of  $E_c$  and  $E'_c$  ( $HR_c$  and  $HR'_c$ , respectively). If  $|HR_c| > |HR'_c|$ , then  $\gamma' = \gamma' * (1 - \gamma)$ . Otherwise,  $\gamma' = \gamma' * (1 + \gamma)$ . The progression of  $\gamma'$  is bounded by an user-defined domain  $[\gamma_{min}, \gamma_{max}]$ . The value of  $E''_{c,i}$  is also used to update the offline predictions for further stops as  $E'_{c,j} = E''_{c,j-1} + E_{c,j} - E_{c,j-1} \wedge j = i + 1$  and  $E'_{c,j} = E'_{c,j-1} + E_{c,j} - E_{c,j-1}, \forall j \in [i + 2, s]$ .

### 3.3 Experimental Results

In the offline regression problem, a state-of-art algorithm was employed: Random Forest (RF). We did so by following previous work about this topic, which used

data from the same source [13]. The experiments were conducted using the R Software [14]. A sensitivity analysis was conducted on the regression parameters. The best parameter setting was `mtry=3` and `ntrees=750`. The learning period used was  $\theta = 7$  days by employing our domain-knowledge. The error threshold to trigger the inter-trip update rule was set to  $\alpha = 0.05$  while the initial value for the residual’s percentage to be employed is  $\beta = 0.01$ . The learning rate *kappa* was set to 0.3. The initial residual’s percentage employed on the stop-based update rule is  $\gamma = 0.1$  while its domain is  $\gamma \in [0.005, 0.3]$ . Finally, the  $\rho$  was set to 360 seconds. All these parameters were set by employing an apriori cross-validation test on some close range of values.

It is possible to divide the evaluation of our framework into two distinct contexts: (i) the Mean Absolute Error (MAE) and (ii) the BB detection accuracy. With the first one, we employed a prequential evaluation [15] by evaluating *just* the prediction made for the LTT performed for the next bus stop. We did so by using the MAE on (1) the offline regression output and then on the (2) inter-trip and (3) intra-trip refinement. In the BB detection context, Accuracy, Precision and Recall were used as evaluation metrics. A weighted Accuracy was also employed, by weighing the trips where a BB event emerged ten times more than the remaining ones. The Average Number of Stops Ahead is also displayed to show which our forecasting horizon is. The results are presented in Table 1. More than only just identifying a problematic link or stop, the BB online detection framework also identifies the vehicle pair where a corrective action must be made. In the current dataset, it was able to detect BB events thirteen stops ahead (on average), which gives more than enough room to perform any of the four possible corrective actions. Despite its achievements, this framework also presents some limitations, namely, with the regression task and with the employed parameters. The regression task was tested using only one algorithm. Even though considering that it presented good results with similar data [13], we do not know if there is another that could perform better, by using a similar computational effort. On the other hand, both the prediction refinements and

Table 1: Experimental results. The times are in seconds. The ALL column contains the average for the first two spans and the sum for the last one.

	A1	A2	B1	B2	C1	C2	ALL
<b>MAE offline regression</b>	1356.96	643.99	1475.22	1871.01	473.61	2776.57	1432.88
<b>MAE inter-trip update</b>	148.85	92.91	124.99	148.85	40.65	123.77	113.34
<b>MAE incremental update</b>	13.21	26.35	22.67	13.21	31.79	27.47	22.45
<b>Accuracy</b>	97.99%	96.34%	97.08%	97.83%	96.63%	93.83%	96.62%
<b>Weighted Accuracy</b>	93.97%	93.57%	94.57%	95.52%	95.73%	91.51%	94.14%
<b>Precision</b>	65.88%	40.85%	41.53%	45.70%	69.44%	51.67%	52.51%
<b>Recall</b>	81.81%	83.18%	83.07%	83.24%	94.48%	87.95%	85.62%
<b>Avg. Nr. of Stops Ahead</b>	11.85	14.78	13.88	15.01	12.96	14.52	13.83
<b>Correct BB Predictions</b>	558	460	363	303	1811	1497	4992
<b>Real BB Events</b>	682	553	437	364	1917	1702	5655

the event detection framework rely on a large set of parameters. To get a fair parameter setting can be a hard task – especially if the user has no expertise with the case study approached.

## 4 Taxi-Passenger Demand Prediction

Taxi driver mobility intelligence is an important factor to maximize both profit and reliability within every possible scenario. Knowledge about where services will actually emerge can be an advantage for the driver – especially when there is no economic viability to adopt random cruising strategies to find passengers. The GPS historical data is one of the main data sources for this topic because it can reveal underlying running mobility patterns. Multiple works in the literature have already successfully explored this type of data with various applications, such as modelling the spatiotemporal structure of taxi services [16], building passenger-finding strategies [17] or even predicting taxi location through a passenger-perspective [18]. Despite their useful insights, most techniques reported are tested using offline test-beds, discarding some of the main advantages of this type of signal. In other words, they do not provide any live information about the location of a passenger or the best route to pick-up a passenger at the current date/time (i.e. real-time performance).

### 4.1 Methodology

Let  $S = \{s_1, s_2, \dots, s_N\}$  be the set of  $N$  taxi stands of interest and  $D = \{d_1, d_2, \dots, d_j\}$  be a set of  $j$  possible passenger destinations. Consider  $X_k = \{X_{k,0}, X_{k,1}, \dots, X_{k,t}\}$  to be a discrete time series (aggregation period of  $P$ -minutes) for the number of demanded services at a taxi stand  $k$ . Our goal is to build a model which determines the set of service counts  $X_{k,t+1}$  for the instant  $t + 1$  per each taxi stand  $k \in \{1, \dots, N\}$ . To do so, we propose three distinct short-term prediction models: 1) a Time Varying Poisson Model which handles the long term memory; 2) a Weighted Time Varying Poisson Model, which sets weights to each data point according with its date - where the most recent points weigh more than the older ones and the weights are calculated through an Exponential Smoothing model; 3) an ARIMA model which handles the short-term memory through its high reactivity to bursty changes to the process in place. The output prediction is an Ensemble of the outputs produced by the aforementioned methods. We employed a Sliding Window ensemble that computes and weighs the average of those outputs. The weights are inverse to their error in the *most recent* samples.

This model is deeply described in section III in [19]. Moreover, an extended version of this framework was also presented in [20], where both the Poisson and the ARIMA models were extended to be calculated **incrementally**.

### 4.2 Experimental Results

The model was programmed using R language [14]. The prediction periodicity was set to 5 minutes. Both the ARIMA model and the weight set were firstly

set (and updated every 24h) through learning the underlying model (i.e. auto-correlation and partial autocorrelation analysis) running on the historical time series curve of each stand, during the last two weeks. To do so, we used an automatic time series function in the `[forecast]` R package - `auto-arima` and the `arima` function from the built-in R package `[stats]`. The Time Varying Poisson averaged models (both weighted and non-weighted) were also updated every 24 hours. A sensibility analysis carried out with data previous to the one used on these experiments determined the optimal values for the parameters  $\alpha$ ,  $\beta$  and  $H$  as 0.4, 0.01 and 4 (i.e. a sliding window of 20 minutes), respectively.

The error measured for each model is presented in Table 2. The results are firstly presented per shift and then globally. The overall performance is good: the maximum value of the error using the ensemble was **25.90%** during the evening shift. The sliding window ensemble is always the best model in every shift and case study considered. The models just present slight discrepancies within the defined shifts. Our model took - on average - 37.92 seconds to build the next prediction about the spatiotemporal distribution the demand by all stands.

### 4.3 Related Work

Little research regarding the demand prediction problem exists. Kaltenbrunner *et al.* [21] detected the geographic and temporal mobility patterns over data acquired from a bicycle network running in Barcelona. The authors' goal was to forecast the number of bicycles at a station to improve their deployment. Yuan *et. al* presented in [22] a complete work containing methods about a) how to divide the urban area into pick-up zones using spatial clustering; b) how a passenger can find a taxi; and c) which trajectory is the best to pick-up the next passenger. Although its results are promising, it is focused on improving the trajectory of a single driver, disregarding the position of the remaining drivers.

The most similar work to our own is presented by Li *et. al* [16]. The authors present a recommendation system to improve the drivers' mobility intelligence. To do so, data from a taxi network running in Hangzhou, China was used. Firstly, they calculated the city hotspots - urban areas where pick-ups occur more frequently. Secondly, they used ARIMA to forecast the number of pick-ups at these hotspots over periods of 60 minutes. Thirdly, they presented an

Table 2: Error Measured on the Models using *sMAPE*.

Model	Periods			
	00h-08h	08h-16h	16h-00h	24h
Poisson Mean	27.67%	24.29%	25.27%	25.32%
W. Poisson Mean	27.27%	24.62%	25.66%	25.28%
ARIMA	28.47%	24.80%	25.60%	26.21%
<b>Ensemble</b>	<b>24.86%</b>	<b>23.14%</b>	<b>24.07%</b>	<b>23.77%</b>



improved ARIMA depending both on time and day type. Despite their good results, it just uses the most immediate historical data, discarding the mid and long-term memory of the system; moreover, the proposed approach assumes fixed periods of 60 minutes (i.e. there is just one prediction per hour). Our framework is *incremental*. By doing so, it is able to produce predictions **on-demand**, which is a true advantage facing the real-time characteristics of this type of decisions.

## 5 Final Remarks

GPS devices are now widespread. Taking full advantage of this rich source of spatiotemporal data to support daily human activities comprises a relevant challenge for the Data Mining community. In this PhD spotlight paper, the authors presented two ML applications focused on improving PT operations. The results are promising. However, there are still many issues to be solved over a mid-term period. The demand prediction must be used to perform commendations in the most profitable urban areas to go to pick-up the drivers' next service. The BB detection framework still requires a large-scale sensitivity analysis of its parameter set. How to select the most adequate corrective action for each situation will also be addressed in our future work. Using both data sources simultaneously on one of these problems can also provide a step forward in this research area. However, further research must be employed to know how beneficial such *shared* knowledge may be.

## Acknowledgments

The authors would like to thank Geolink, Lda. and STCP for the data supplied. This work was supported by the VTL: "Virtual Traffic Lights" (PTDC/EIA-CCO/118114/2010), by MAESTRA (ICT-2013-612944), by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and also by the Portuguese Funds through the FCT (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-037281.

## References

1. Zhu, W., Boriboonsomsin, K., Barth, M.: Defining a freeway mobility index for roadway navigation. *Journal of Intelligent Transportation Systems* **14**(1) (2010) 37–50
2. Chien, S., Ding, Y., Wei, C.: Dynamic bus arrival time prediction with artificial neural networks. *Journal of Transportation Engineering* **128**(5) (2002) 429–438
3. Lee, J., Park, G., Kim, H., Yang, Y., Kim, P., Kim, S. In: *A Telematics Service System Based on the Linux Cluster*. Volume 4490 of LNCS. Springer Berlin / Heidelberg (2007) 660–667
4. Gama, J.: *Knowledge discovery from data streams*. Chapman and Hall/CRC (2010)

5. Powell, W., Sheffi, Y.: A probabilistic model of bus route performance. *Transportation Science* **17**(4) (1983) 376–404
6. Gershenson, C., Pineda, L.: Why does public transport not arrive on time? the pervasiveness of equal headway instability. *PloS one* **4**(10) (2009) 72–92
7. Daganzo, C., Pilachowski, J.: Reducing bunching with bus-to-bus cooperation. *Transportation Research Part B: Methodological* **45**(1) (2011) 267–277
8. Daganzo, C.: A headway-based approach to eliminate bus bunching. *Transportation Research Part B* **43**(10) (2009) 913–921
9. Delgado, F., Muñoz, J.C., Giesen, R., Cipriano, A.: Real-time control of buses in a transit corridor based on vehicle holding and boarding limits. *Transportation Research Record: Journal of the Transportation Research Board* **2090**(1) (2009) 59–67
10. Moreira-Matias, L., Ferreira, C., Gama, J., Mendes-Moreira, J., de Sousa, J.: Bus bunching detection by mining sequences of headway deviations. In: *Advances in Data Mining. Applications and Theoretical Aspects*. Volume 7377 of LNCS., Springer. (2012) 77–91
11. Chen, G., Yang, X., An, J., Zhang, D.: Bus-arrival-time prediction models: Link-based and section-based. *Journal of Transportation Engineering* **138**(1) (2011) 60–66
12. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* **65**(6) (1958) 386
13. Mendes-Moreira, J., Jorge, A., de Sousa, J., Soares, C.: Comparing state-of-the-art regression methods for long term travel time prediction. *Intelligent Data Analysis* **16**(3) (2012) 427–449
14. R Core Team: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. (2012)
15. Dawid, A.: Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society. Series A (General)* **147** (1984) 278–292
16. Li, X., Pan, G., Wu, Z., Qi, G., Li, S., Zhang, D., Zhang, W., Wang, Z.: Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science in China* **6**(1) (2012) 111–121
17. Li, B., Zhang, D., Sun, L., Chen, C., Li, S., Qi, G., Yang, Q.: Hunting or waiting? discovering passenger-finding strategies from a large-scale real-world taxi dataset. In: *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. (March 2011) 63–68
18. Yuan, J., Zheng, Y., Xie, X., Sun, G.: Driving with knowledge from the physical world. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM* (2011) 316–324
19. Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L.: Predicting taxi-passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems* **14**(3) (2013) 1393–1402
20. Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L.: On predicting the taxi-passenger demand: A real-time approach. In: *Progress in Artificial Intelligence*. Volume 8154 of LNCS. Springer (2013) 54–65
21. Kaltenbrunner, A., Meza, R., Grivolla, J., Codina, J., Banchs, R.: Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system. *Pervasive and Mobile Computing* **6**(4) (2010) 455–466
22. Yuan, J., Zheng, Y., Zhang, L., Xie, X., Sun, G.: Where to find my next passenger? In: *13th ACM International Conference on Ubiquitous Computing (UbiComp)*. (2011)